Semantic Digital Twins in the Industrial Internet of Things

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

von
Sebastian Richard Bader
aus
Regensburg, Deutschland

Bonn, August 2021

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn.

1. Gutachter:	Prof. Dr. Jens Lehmann
2. Gutachterin:	Prof. Dr. Maria Maleshkova
Tag der Promotion:	01.04.2022
Erscheinungsjahr:	2022

Abstract

The seamless connection of control devices, physical assets, and IT systems through internet-based networks is one of the current megatrends. Especially the manufacturing industry experiences disruptive changes in how products are designed, documented, and provided to customers. The appropriate digital representation of products becomes a must-have for future competitiveness, and the standardized integration with arbitrary third-party applications is no longer an additional feature but strictly demanded by customers. The complete product lifecycle needs to be represented in comprehensive and self-descriptive manners and integrated into scalable, decentralized communication networks from the engineering phases through production, delivery, operation, maintenance, and disposal.

While the general potential is without question, the recent activities in industrial companies and the research community have brought specific challenges to the surface. Consistent and industry-wide data modeling, standardized and reliable data exchange mechanisms, and distributed and scalable communication architectures have not yet been achieved. Limited attempts with small numbers of partners and controlled domains have implemented working prototypes, but an industry-wide, comprehensive approach is still not achieved. Inconsistent representations from the involved domains, vendor-specific technologies and protocol breaks, and missing standards and conflicting conventions comprise the most critical challenges.

Combining formal data modeling paradigms of semantic technologies with reliable and well-known internet technologies forms the foundation of this work. The concept of Digital Twins is extended and specified to consistent and self-declarative entities imposing the atomic building blocks for the integration of components, facilities, and applications across networks, companies, and domains. Based on proven technologies, widely accepted conventions, and new extensions, new patterns are outlined to overcome six identified key challenges: the unpredictable requirements of IIoT settings, their Brownfield nature, the communication between heterogeneous devices, the non-transparent design decisions of the device interfaces, the dynamic changes in the networks, and the need to keep the sovereignty of the involved data.

This thesis contributes to the state of the art by examining the communication between heterogeneous assets of any kind and analyzing how the recommended patterns can be transferred into the so-called Industrial Internet of Things. In particular, it presents (1) how the assets themselves need to be modeled to become suitable, broadly usable Digital Twins, (2) which interactions they need to provide and how they have to be consumed by legacy and future services, and (3) how the flexible ecosystems of tomorrow's manufacturing processes look like and how the current standardization landscape supports and defines it.

The first regarded area focuses on the consistent and formalized representation of Digital Twins. The vast amount of proposed patterns in the literature did not reach a consensus but even instead increased the overall complexity through incompatible models and conflicting technology stacks. This thesis analyzes the shortcomings of existing models and develops a comprehensive approach for the Digital Twin entity itself and the surrounding Cloud ecosystem in close collaboration with relevant standardization organizations. This procedure is necessary to ensure the required wide adoption and thereby to overcome the mentioned hurdles.

The scope of the second contribution area is the *connection and communication* of thereby expressed Digital Twins. The current IIoT landscape involves a wide variety of protocols with differing operational semantics and serializations. The extraction of atomic interactions and their homogeneous interpretation throughout the protocol layer allows the explicit construction of complex workflows and data exchange patterns.

The combination of the formalized representation and their interaction patterns enables the realization of *flexible, federated architectures*. The third research area examines their characteristics and requirements, and supplies a comprehensive analysis of existing reference models and technical standards. The formal presentation of their relations and creating interactive examination tools on top enables thorough surveys of standard practices and gaps in the specifications.

The insights of the presented work have been incorporated in several research publications at reputed conferences, journals and have already been integrated into upcoming industry standards. They have been discussed with the research community and used in various standardization activities to perpetuate the outlined approaches. Namely, the frameworks and technical specifications of the International Data Spaces (IDS), the Plattform Industrie 4.0, and the upcoming Gaia-X Cloud Ecosystems present the core communities in which the contributions of this thesis are incorporated and which enabled the direct evaluation of the results in active environments.

Keywords: industrial internet of things, digital twin, semantic web, industry 4.0, linked data

Acknowledgments

A Ph.D. thesis is never the result of individual work but the combined contribution and support of many. However, naming all is not possible, so I need to apologize first to everyone who is missing in the following. I have met so many brilliant, open-minded people in the last years who were surprisingly motivated to discuss ideas over and over again until something that was made up and fuzzy in my head finally made it to a contribution worth being published.

I want to thank in particular my doctorate supervisor, Prof. Dr. Jens Lehmann, who invited me into his research group and created an environment in which amazing people can come together and push boundaries every day. Prof. Dr. Stefan Wrobel for the continuous support and excellent opportunities at Fraunhofer IAIS where I could bring my ideas from abstract concepts into real-world applications.

Dr. Maria Maleshkova always guided me in the right direction, for her patience and continuous support to bring me back on track whenever I lost orientation. She always believed in my potential and ideas and was more convinced than me that this thesis will finally make it.

Prof. Dr. York Sure-Vetter and Prof. Dr. Rudi Studer gave me the chance to start my research career in Karlsruhe. You have pointed me a way about which I didn't know anything and helped an ignorant graduate to learn what research really means. I also want to thank all my co-authors and colleagues who had the bad luck working with me for all the constructive discussions, for broadening my view with their ideas, and for their effort to meet all the deadlines through countless night shifts.

My wife Fang-Ju Chou for her support throughout this journey. Without her, this thesis would have never been finished. You teach me every day about the essential things in life. And Leon, who proves every day that there is no ground-breaking research and no project deadline that is more important than the smile of a little child.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Scenario: Understanding the IIoT . . . 2
  - 1.2.1 The IIoT from a Business Viewpoint . . . 3
  - 1.2.2 The IIoT from an Asset Viewpoint . . . 5
- 1.3 Targeted Challenges . . . 6
- 1.4 Research Questions . . . 8
- 1.5 Contributions and Publications . . . 9
- 1.6 Research Methodology . . . 12
- 1.7 Relevance and Impact . . . 14
- 1.8 Thesis Structure . . . 15

2 Preliminaries **17**
- 2.1 Semantic Web . . . 17
  - 2.1.1 The Resource Description Framework . . . 18
  - 2.1.2 Ontologies and Formal Knowledge Representation . . . 20
  - 2.1.3 Linked Data and distributed RDF . . . 21
  - 2.1.4 Linked Data Platform . . . 21
- 2.2 Industrial Internet of Things . . . 22
  - 2.2.1 The International Data Spaces . . . 24
  - 2.2.2 IIoT Standardization Initiatives . . . 25
- 2.3 Digital Interactions and Data Exchange . . . 27
  - 2.3.1 Web Services . . . 28
  - 2.3.2 Representational State Transfer . . . 30
  - 2.3.3 Combining Web Services . . . 31
- 2.4 Digital Twins . . . 31
- 2.5 Usage Policies . . . 33

3 Related Work **35**
- 3.1 Semantic Digital Twins . . . 35
  - 3.1.1 Semantic Input Modelling . . . 37
  - 3.1.2 Semantic Description Languages . . . 39
- 3.2 Semantic Web and IIoT . . . 41
- 3.3 Unpredictable Requirements . . . 43
- 3.4 Digital Twins in Brownfield Scenario . . . 44
- 3.5 Heterogeneous Communication in IIoT Networks . . . 45
- 3.6 Digital Workflows in Dynamic Environments . . . 47
- 3.7 Data Sovereignty . . . 48

3.8 Distribute Architectures for Digital Twins in IIoT . . . 51
3.8.1 Visualization of interlinked Entities in the Semantic Web . . . 51
3.8.2 Technical Standards for the IIoT . . . 52
3.8.3 IIoT Landscape . . . 53
3.9 Reference Architectures for IIoT . . . 53
3.10 Summary . . . 55

4 The Semantic Digital Twin 57
4.1 From Assets to Digital Twins . . . 58
4.1.1 The IIoT from an Integration Viewpoint . . . 59
4.1.2 IIoT Assets in Distributed Architectures . . . 60
4.1.3 Self-Descriptive APIs . . . 61
4.1.4 Self-governed IIoT Services . . . 63
4.1.5 Summary: IIoT Assets and Digital Twins . . . 63
4.2 A Data Model for the Digital Twin: The AAS and Virtual Representations . . . 64
4.2.1 Introduction . . . 64
4.2.2 Methodology . . . 65
4.2.3 Asset Administration Shell as the Digital Twin . . . 66
4.2.4 The SAAS Data Model . . . 69
4.2.5 Mapping to RDF: The Digital Twin from a Data Viewpoint . . . 70
4.2.6 Reasoning . . . 72
4.2.7 Schema Validation . . . 72
4.2.8 Use Cases . . . 73
4.2.9 Experimental Evaluation . . . 73
4.2.10 Summary: A Data Model for the Semantic Digital Twin . . . 75
4.3 A Data Model for IIoT Ecosystems: The IDS Information Model . . . 76
4.3.1 Introduction . . . 76
4.3.2 Governance and Context of the IDS Information Model . . . 77
4.3.3 Methodology . . . 81
4.3.4 Evaluation . . . 82
4.3.5 Summary: A Data Model for IIoT Ecosystems . . . 83
4.4 A Usage Control Language for Data Sovereignty . . . 84
4.4.1 Introduction to a Usage Policy for IIoT Digital Twins . . . 85
4.4.2 The IIoT from a Data Sovereignty Viewpoint I . . . 85
4.4.3 Policies and Machine-readable Restrictions . . . 86
4.4.4 Dimensions of Usage Control Aspects . . . 90
4.4.5 Operators for Usage Control Rules . . . 92
4.4.6 Summary: Sovereignty of Digital Twins . . . 92
4.5 Summary . . . 94

5 Interaction Patterns for the IIoT 97
5.1 IIoT Requirements Model . . . 98
5.1.1 Functional Requirements . . . 99
5.1.2 Non-functional Requirements . . . 101
5.1.3 Protocol Characteristics . . . 102
5.1.4 Data Representation . . . 103
5.1.5 Summary: An Overview of IIoT Requirements . . . 104

5.2 State-based Integration Through Resource-orientation . . . 105
5.2.1 The IIoT from an Interaction Viewpoint . . . 106
5.2.2 The Virtual Representation of the Digital Twin . . . 107
5.2.3 Summary: A scalable Integration Pattern . . . 111
5.3 Brownfield Integration at the Edge of IIoT . . . 112
5.3.1 Iterative Brownfield Deployment . . . 112
5.3.2 IIoT Framework for Digital Twins at the Edge . . . 114
5.3.3 Implementation . . . 116
5.3.4 Deployment . . . 116
5.3.5 Service generator . . . 117
5.3.6 Evaluation . . . 117
5.3.7 Summary: Brownfield-Integration at the Edge . . . 118
5.4 RESTful APIs with Linked Data: The SOLIOT Approach . . . 118
5.4.1 Guiding Principles . . . 119
5.4.2 The IIoT from a Data Sovereignty Viewpoint II . . . 120
5.4.3 Mapping and Interaction Model . . . 122
Functional Characteristics . . . 122
Non-Functional Characteristics . . . 124
MQTT and CoAP Protocol Bindings . . . 125
Data Representation . . . 130
Usage Control Model . . . 133
5.4.4 Architecture and Prototypical Implementation . . . 134
5.4.5 Interactions and Protocols . . . 135
5.4.6 Use Case Scenario . . . 136
5.4.7 Evaluation . . . 137
5.4.8 Summary: An IIoT Framework for Semantic Digital Twins . . . 140
5.5 Summary . . . 141

6 Distributed Architectures and IIoT Reference Frameworks 143
6.1 Industrial Standards and Norms for the IIoT . . . 144
6.1.1 Design and Technical Quality . . . 145
6.1.2 Knowledge Graph for the IIoT . . . 148
6.1.3 Availability of I40KG . . . 150
6.1.4 Reusability of the Graph Content . . . 151
6.1.5 Technical Evaluation . . . 152
6.1.6 Summary: The Landscape of Industrial Standards . . . 152
6.2 Generalized Architecture Model for IIoT Digital Twins . . . 152
6.2.1 Business Aspects of IIoT Digital Twins . . . 156
6.2.2 Usage Aspects of IIoT Digital Twins . . . 156
6.2.3 Functional Aspects . . . 157
6.2.4 Data Formats and Semantics . . . 157
6.2.5 Implementation Aspects . . . 158
6.2.6 Transport and Internet Layer . . . 159
6.2.7 Security and Trustworthiness . . . 160
6.2.8 Governance and Compliance . . . 162
6.2.9 Summary: A Layered Architecture for the IIoT . . . 163

6.3 Visual Analytics of Standards and Frameworks . . . 163
6.3.1 Interactive Presentation and Information Selection . . . 165
6.3.2 The IIoT Landscape from a Standardization Viewpoint . . . 166
6.3.3 Considered Concerns . . . 169
6.3.4 Selecting the Reference Architectures . . . 170
Reference Architecture Alignment Process . . . 172
The IDS as a Use Case Model . . . 173
6.3.5 Limitations . . . 174
6.3.6 Findings and Best Practices . . . 176
6.3.7 Lessons Learned and Outlook . . . 179
6.3.8 Summary: A new Way to Analyse the IIoT . . . 179
6.4 Summary . . . 180

7 Conclusion **181**
7.1 Standardized Data Models for IIoT Ecosystems and Digital Twins . . . 182
7.2 Modular Communication Patterns for IIoT Digital Twins . . . 183
7.3 Standardization and Reference Architectures for the IIoT . . . 184
7.4 The Impact of Semantic Digital Twins for the Industrial Internet of Things . . . 185

8 Future Work **187**
8.1 Ownership and Control of digital Information . . . 188
8.2 Human and Machine Interaction . . . 188
8.3 Uncertainty and incomplete Information . . . 188
8.4 Semantifying Industrial Catalogues . . . 189
8.5 Real-time Control and Orchestration of Production Facilities . . . 190
8.6 Upcoming Megatrends . . . 190

Bibliography **193**

A Publications **219**

B Scenario: Viewpoints of the IIoT **223**
B.1 Business Viewpoint . . . 223
B.2 Asset Viewpoint . . . 224
B.3 Sovereignty Viewpoint . . . 225
B.4 Standardization Viewpoint . . . 226
B.5 Interaction Viewpoint . . . 227

List of Figures **229**

List of Tables **233**

Glossary **235**

CHAPTER 1

Introduction

The Internet of Things (IoT) comprises the recent trend of equipping nearly any kind of physical object with network access and thereby enabling connecting them to a local intranet or the global internet. Similar to the great success of the Web at connecting humans, the IoT envisions a world where nearly all facilities, assets, and devices become connected through the internet. Assets of any kind shall be enabled to send, retrieve, and process digital messages. While the original vision for an 'Internet **for** Things' by Sarma, Brock, and Ashton [1] focused on RFID tags, through which "[...] electronic devices are networked and every object, whether it is physical or electronic, is electronically tagged with information pertinent to that object" ([1] p. 4), this interpretation has changed slightly. In the meantime, the focus has shifted towards embedded devices, for instance defined in the joint EC/EPoSS expert workshop on RFID/Internet-of-Things [2]:

Definition 1.0.1 *Internet of Things "The semantic origin of the expression is composed by two words and concepts: 'Internet' and 'Thing', where "Internet" can be defined as 'The world-wide network of interconnected computer networks, based on a standard communication protocol, the Internet suite (TCP/IP)', while 'Thing' is 'an object not precisely identifiable'. Therefore, semantically, 'Internet of Things' means 'a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols.'"*

Gartner estimated that in 2017 already 8.4 billion devices were connected to the global internet and predicted that this trend would continue to more than 20 billion connected objects in 2020 [3]. Having nearly all relevant real-world objects also digitally accessible will not only allow more efficient and faster coordination between them, and at the same time, create completely new flexibility for establishing and controlling complex workflows but also increasing the transparency to understand and analyze activities in real-time.

1.1 Motivation

The syntactic and semantic integration of information and services, however, is the decisive factor for any use case. While the Web has been proven to be a suitable answer to scalability challenges, the addition of formal vocabularies and shared meaning – known as semantic technologies or the Semantic Web – has developed answers to describe and automatically understand the *meaning*. In addition, the combination of the rich experiences and high maturity standards of *machine-to-machine communication* and control processes developed by the industrial automation community provides the foundation to create completely new ways of manufacturing assets with the support through digital technologies.

Especially in the context of smart factories, notable efforts are conducted to bring the diverse and heterogeneous components from the shop floor to an integrated digital layer. Generally aspired goals are, for instance, lower conversion times, data-driven maintenance, or automated machine configurations in order to reduce downtimes and to increase the overall efficiency.

Nevertheless, this requires nearly full information about the current and the past state of all relevant machines, their respective components and parts, and the involved materials and products. Whilst many critical components and parts are not observable with economically reasonable investment (due to missing accessibility, not available network access, or high update costs), continuous and complete monitoring of any involved system, sub-system, or sub-sub-system cannot be achieved.

These outlined opportunities and challenges are commonly framed by the so-called *Industrial Internet of Things* (IIoT) [4] or Industry 4.0 [5]. The shared vision of locally and globally interwoven digital and physical production processes across facilities, organizations, and domains has drawn significant attention in research and industry. However, as the huge efforts of recent years have shown, the key challenges are still unresolved. Information from a wide variety of sources needs to be processed and managed in distributed, dynamic networks where actors do not necessarily know each other beforehand. Nevertheless, it is widely accepted that future business models in the industrial domain heavily rely on collaboration, the combined co-creation of value and intelligent data usage. The thereby created contrast of still necessary developments and at the same time arising business needs call for innovative approaches and new concepts.

Definition 1.1.1 ***Industrial Internet of Things****[4] "[The] Industrial Internet of Things (IIoT) describes systems that connects and integrates industrial control systems with enterprise systems, business processes, and analytics."*

As stated, the underlying interoperability and integration challenges remain unsolved. The gained insights of the Semantic Web, on the one hand, and the automation expertise of the manufacturing domain, on the other hand, are therefore promising fields to drive the IIoT developments. A core assumption in all the related approaches, however, is the vision that the practical business cases will emerge as soon as the IIoT vision has been successfully realized. Slightly different from traditional investments, the finally applicable – disruptive – business models might even not be known yet. This challenge has been explained by the initial works on the IoT by Sarma, Brock, and Ashton [1]. They argue that successful architectures and systems must be open to continuous improvements and adaptions, as new technologies and thereby imposed requirements are hardly known when these devices are initially deployed. Nevertheless, as the great success of, for instance, the silicon valley companies shows, similar technology breakthroughs need to be executed first, while the business cases may automatically arise after. Whether this assumption will hold, however, is not in the scope of this work.

Assuming, however, that the observable enthusiasm has a valid basis, and the expected opportunities can be exploited in ways that can not yet be predicted, the situation imposes the challenge that the decisive requirements are currently not known either. A promising approach therefore must be as flexible and adaptive as possible while at the same time be able to solve the already well-observed outlined connectivity and information exchange challenges.

1.2 Scenario: Understanding the IIoT

The examinations and approaches presented in this thesis target several interwoven aspects of the IIoT. To increase the readability and support the understanding, the provided contributions are explained by using one representative scenario and examining it from different perspectives. The described use case

motivates the conducted research agenda by introducing the key roles and participants, their related problems, as well as it outlines the limitations of the current state of the art.

As mentioned, the scenario is discussed from different angles. These are called *Viewpoints* in the following, in accordance with ISO 42010. Each viewpoint focuses on a different group of stakeholders and their respective motivations and requirements, enabling the switch of perspectives and a more comprehensive picture of the use case. The Business viewpoint of the example is explained first, aiming to describe the general situation using BPMN [6], followed by a presentation how the physical objects (Asset Viewpoint) are related. Additional perspectives are introduced alongside the contributions. For instance, Section 4.2.5 presents how the assets are represented as Digital Twins, and Section 5.2.1 extends the previous explanations with communication flows and corresponding UML diagrams. Annex B contains all viewpoints, including the complete diagrams.

In the following, the terminology of a Digital Twin is used to refer to the digital representation of a physical asset. Typical characteristics of a Digital Twin involve visual representations, simulation models, machine-readable descriptions and defined interfaces, virtual representation of physical attributes, bidirectional relations between the physical and the virtual entity, and more. Close or even identically used are terms such as Cyber-physical Systems, IoT Devices, Avatars, Virtual Representations, Asset Administration Shell etc. While many definitions of Digital Twins exist as terms used, for instance [7–9], in this thesis mainly the one by Glaessgen and Stargel [10] is followed: "The Digital Twin is an integrated multiphysics, multiscale, probabilistic simulation of an [as-built vehicle or] system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding [flying] twin". Tao et al. extend this understanding further that a "Digital Twin[, it] consists of three parts: physical product, virtual product, and connected data that tie the physical and virtual product [...]" [7]. These aspects are all combined in the definition provided in the Encyclopedia of Production Engineering by Stark and Damerau [11] that is used in this thesis:

Definition 1.2.1 ***Digital Twin*** *"A digital twin is a digital representation of an active unique product (real device, object, machine, service, or intangible asset) or unique product-service system (a system consisting of a product and a related service) that comprises its selected characteristics, properties, conditions, and behaviors by means of models, information, and data within a single or even across multiple lifecycle phases."*

Stark and Damerau relate the digital representation with a 'product' or 'product-service system', already indicating thereby that this aspect of the Twin is nearly as hard to define as the Digital Twin itself. One possible distinction is that the represented entity, further called the *Asset*, must contain a certain amount of value that the representation effort towards a Digital Twin is justified[12, 13]:

Definition 1.2.2 ***Asset*** *The Asset represented by the Digital Twin is a tangible or intangible entity that has a significant value in a certain usage scenario.*

These definitions are the foundation for the following description of a typical IIoT use case. It contains the key aspects of the IIoT implementations, starting with an examination from a business viewpoint and is then continuously extended with additional perspectives.

1.2.1 The IIoT from a Business Viewpoint

The regarded situation involves three manufacturers M_1, M_2, and M_3 that form a simple supply chain to design and produce a robot gripper arm (cf. Fig. 1.1). M_1 is responsible for the electric motor, and M_2 produces the required joint. M_3 combines these parts and finalizes the robot. It is assumed that no additional parts are necessary or can be neglected in the following.

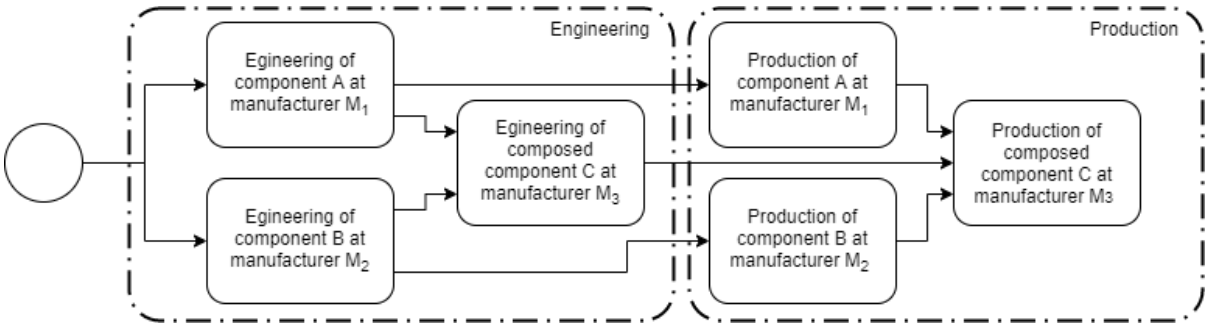


Figure 1.1: Engineering and production phases of the robot gripper arm and its components.

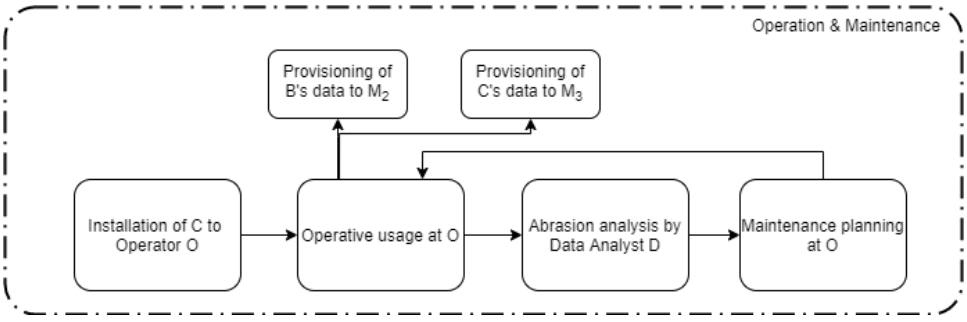


Figure 1.2: Integration of the robot arm and predictive maintenance based on data analytics.

The third manufacturer needs digital representations of the components from M_1 and M_2 already at the design time of the robot. These representations must include the geometry and explanations on the used materials and their structural limits but also documentation forms and safety certifications. M_1 and M_2 provide all this information in the form of Digital Twins through the IIoT. M_3 then integrates the endpoints into his engineering environment and can thereby create blueprints, production plans but also technical specifications for the whole robot. He combines the Digital Twins received from M_1 and M_2 to a new Digital Twin and makes it accessible to his potential customers.

At this point of the product lifecycle, no physical Asset has been created yet. All information exchange is focused on its *type* – in contrast to the individual *instances* that are being produced. The type combines the general attributes, for example, the dimensions and documentation manuals, that are true for the whole series. The instances represent individual Assets with an exact production date and an individual usage history. Both kinds are represented as Digital Twins and are exposed in the same ways.

After receiving the physical motor and the joint, M_3 produces the robot arm and delivers it to his customer, the Operator O. O installs it into his running production line and integrates it into his control systems (cf. Fig. 1.2). He further engages the data analyst D to implement a predictive maintenance strategy to prevent unforeseen failures and reduce the downtime due to unnecessary inspections. Furthermore, both M_2 and M_3 have signed contracts to contiguously access the operation states of the robot. They use this data to improve their designs but also to assist O in case of breakdowns.

After a certain amount of time, O decides to add an additional sensor unit to the arm (cf. Fig. 1.3). This modification happens independently of the original manufacturer M_3 and is therefore not foreseen in the communication interfaces of the original design and also not represented in the provided Digital Twin. O can however easily reconfigure his Digital Twin and also configure that this new information source is not exposed to M_2 and M_3 but only to the data analyst. Solely relying on the Digital Twin as the single data access mechanism relieves all involved parties from further reconfiguring their communication

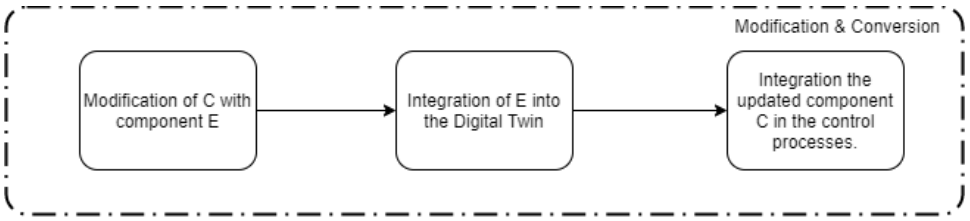


Figure 1.3: The new sensor module is integrated into the physical Asset as well as into the Digital Twin.

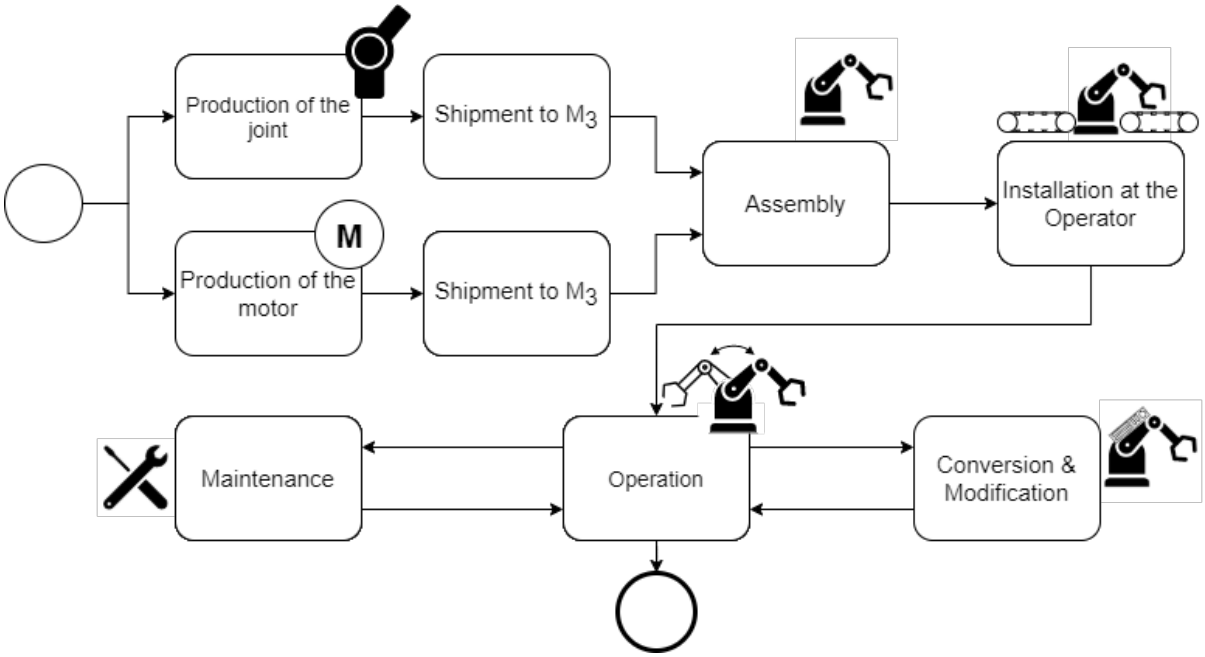


Figure 1.4: The physical Assets are assembled, shipped, installed and reconfigured involving different stakeholders and owners.

infrastructure.

1.2.2 The IIoT from an Asset Viewpoint

The physical Assets undergo a slightly different process. While the Business Viewpoint focuses on the relations between the involved companies, this section takes the perspective of the components and their interactions between each other and their surrounding. The joint and motor are assembled to a new Asset, the robot gripper arm, shipped to different locations, and integrated with other machines into a running facility (cf. Fig. 1.4). Most prominently, the engineering phase is entirely out of scope for this viewpoint.

After the successful installation at O, the robot arm executes its tasks and thereby experiences, for instance, abrasion and breakdowns as the parts are being exposed to mechanical stress. As a consequence, maintenance and repair activities need to be executed at the Asset – something that is impossible solely in the virtual world.

Additional activities mainly executed in the Asset Viewpoint are, for instance, the upgrade or modification of the robot arm with additional sensors or control units. The digital world is obviously affected by such steps, as the amount of available and presentable data is increased but dependent on the capabilities of the installed units. The Digital Twin must therefore reflect the activities and modifications of the

physical Asset and provide the respective update mechanisms.

At this point, the limits of the scenario are reached. Later phases like the reselling of the robot arm or the deconstruction and disposal are explicitly not regarded. Nevertheless, it might be a reasonable assumption that the distinct requirements of these phases are somewhat limited, and therefore, the focus on the mentioned lifecycle phases is justified.

1.3 Targeted Challenges

Despite many and intense activities in the various IIoT domains, the overall vision of a flexible and interoperable industrial internet has not yet been achieved. The existing challenges are still not overcome, and the current approaches are either not yet mature enough to reach large-scale deployment or do not sufficiently address the different challenges. The overall complexity of the topic is more and more regarded as the critical issue to apply IIoT approaches in productive environments [14]. In this context, we have identified the following six key hurdles CH1-CH6 on the way to an applicable IIoT.

CH1: Unpredictable Requirements
The conditions and demands of future settings are hard to predict, making it nearly impossible to implement the necessary capabilities into currently operating systems that have been designed according to the requirements at their time. The disruptive nature of the digitization, however, has completely reshaped their environment and the now appearing requirements. A common solution is to design systems and devices as flexible as possible, to simplify future adaptations and integration steps.

In the scenario of the robot gripper arm, this flexibility is necessary to enable the integration of D's predictive maintenance algorithm added independently of the component manufacturers and the extension with the sensor module E. The usage of this module has not been intended by M_3 and independently executed by O. Nevertheless, the Asset and its Digital Twin must support the upgrade both on the physical and virtual level.

CH2: Brownfield Settings
Obviously, CH1 is not limited to today's design and integration efforts but has also been the case for the designed systems of the past. As a consequence, the currently installed equipment is only partly suitable for the discussed IIoT requirements. Furthermore, the high investment costs prevent the complete renewal with modern facilities. Such environments are broadly called Brownfield environments [15], in contrast to the relatively rare settings where the design can start from scratch, a Greenfield.

In the IIoT scenario, the robot gripper arm is already enabled with his Digital Twin. The surrounding facilities – conveyors, upstream and downstream components – originate from previous investment decisions and lack the rich IIoT-driven interaction features of the new Asset. Still, their complete replacement is not possible due to their high acquisition costs. The new robot, therefore, has to be adjusted towards the given legacy systems.

CH3: Heterogeneous Communication Patterns
While sub-domains and specialized communities have developed an integrated set of information and communication standards of their own, for instance, shop floor data exchange relying on OPC UA, an overall integration standard does not exist. Different protocols (HTTP, MQTT, CoAP etc.), data formats (binary, JSON, XML, Automation ML etc.) and interaction sequences (flooding, message queues, request/response, publish/subscribe) make reliable integration solutions hardly possible.

For instance, the operator O locates the robot arm in a production line with several different components, for instance, with upstream and downstream units that are connected with the robot through conveyor

belts. While the robot itself has an OPC UA interface, the upstream unit pushes sensor signals through MQTT to the central control server and the downstream one follows the CoAP protocol with alarm messages in a vendor-specific JSON format.

CH4: Non-transparent Design Decisions and Assumptions
The behavior and implications of interactions are only transparent for the directly involved developer and designer of a component. Later maintenance or integration steps however are usually executed by other parties, requiring an immense amount of documentation and formal explanation. The resulting systems require enormous efforts to reconfigure systems and devices for new application environments.

This challenge typically appears when the Assets are used outside of the original ecosystem or designed IT platform. The Data Analyst D in the given scenario executes his abrasion analysis algorithm independently of M_3. That means that D must understand the data provisioning behavior of the robot, the meaning of the data objects, the accuracy of the available measurements, their quantities and units, and much more. M_3 on the other hand, was not aware of the required input parameters of D at the engineering time of the robot. The resulting integration gap can be narrowed through intensive documentation materials or directly applied training for the analysts. Any of these options however is very time-intensive and therefore significantly increases the required investment costs.

CH5: Dynamic Environments
The update cycles in traditional industries become shorter and shorter. Still, the development and shipment times of software and digital services – with update cycles of months instead of years – do hardly fit to the significantly longer design and operation lifecycles of manufacturing machines, calculated in years and decades. Currently developed systems must therefore reflect the dynamics in their environment and must be able to rapidly adjust to newly introduced or updated components [16].

In the scenario of the robot arm, the operator O decides at some point in time that he needs a third-party sensor module mounted on the arm. Such upgrading processes are typical for long-running facilities, for instance, to enable new operation modes or to enable new usage operations and thereby increase the overall service lifetime.

CH6: Data Sovereignty
Digital business models will only succeed if companies are willing to collaborate and exchange information. Business-critical data needs to be shared with previously unknown organizations, and business relations are created and terminated at a much higher frequency. The current customer-supplier-relationships do not reflect this fact, including the question of how the necessary trust and control of the critical information can be ensured [17]. Data-sovereign ecosystems, therefore, need to be developed [18] to reduce the uncertainty and risk from all involved participants and to enable on the fly interactions across organizations. Established trust levels are a prerequisite for any scalable, data-driven business model where autonomous Assets communicate with each other.

One good use case for the seamless data integration over the complete Asset lifecycle through the IIoT is the provisioning of operational data to the engineering departments of the Asset vendor. By knowing the actual usage scenarios, more effective designs can be developed and thereby increase the value of his customers. However, O faces the risk to reveal his competitive advantage by providing access to the robots sensor streams and configuration parameters, as domain experts can very likely guess product features, production plans or even customer requirements based on such information. O therefore must stay in control of all the created data while at the same time needs to be able to manage the access and usage of the shared data in a fine-grained manner. One example might be the sharing of sensor streams with the trustworthy manufacturers M_2 and M_3 as well as the Data Analyst D, while prohibiting all of them to distribute further this data and any dependent resources with the not trusted M_1.

The presented challenges present a – undoubtedly incomplete – list of critical obstacles that currently prevent the broad adoption of the industrial internet. While in specific use cases, additional reasons might be the decisive factors, it is assumed that the identified six challenges appear in the majority of IIoT scenarios and therefore need to be answered before any further steps are taken into account.

1.4 Research Questions

The first-class citizen in IIoT use cases is generally the Asset, either as a production facility, one of its components and devices, or in the form of the used material, intermediate and final products. The Assets bind a significant ratio of the used capital and all other production factors, for instance, logistics, energy or IT, must support the value adding process centered around the Assets. Any approach to solve the six declared challenges must therefore originate from the Asset and locate the proposed solutions relative to it. Different to digitization efforts in other domains, the Asset is the indispensable entity that needs to be considered, and any virtual extensions must increase the efficiency and effectiveness of its usage.

Therefore this thesis investigates the following research gap:

How can the combination of Assets and their digital representations solve the key challenges of the Industrial Internet of Things?

In order to solve this question, it must be examined how such Assets actually appear in the desired environment, how they can interact with each other and in parallel also with legacy systems, shop and office floor applications, and how they can be combined to overall architectures ensuring the desired flexibility for the future business models.

Several different perspectives are possible to address this. One possible direction would be the view of the IIoT as a global network and standardized infrastructure in which Assets are registered. On a local scale, the various devices in a car are manually integrated into the CAN bus, and the restrictions and requirements of the bus determine the communication interfaces. In a production line, however, the network requirements are usually determined by the Assets, not the Assets selected depending on their network compliance.

Another possible perspective could put the data integration platforms at the center. Such Cloud-based environments, usually provided by one of the big hyperscalers, consume the Asset's raw data stream and then execute all further processing inside this platform. The specifications of the hyperscaler for data formats, interfaces, and offered services ensure compliance with each other and thereby create a uniform environment. Necessary liftings or mappings of data and operations are applied at the entry point of the platform, relieving the downstream applications from further integration steps.

None of these views however reflect the nature of the industrial manufacturing domain, where the ownership of the Asset and therefore the authority on its data is bound together but must also move between organizations according to the changing ownership situations. The data management in Cloud platforms makes the Asset subject to the hyperscaler's procedures and forces the Asset owner into a critical dependency.

As stated in the use case scenario, the concept of the Digital Twin regards the virtual representation of the Asset as a combined entity. While this view reflects the reality of the manufacturing domain, each of the six challenges needs to be solved for every single Asset. This thesis intends to provide generic patterns and applicable methods on the presentation of the Asset as a Digital Twin, the interactions between such Digital Twins among each other and with the surrounding applications, how they need to be designed to achieve their soundness for future usages and how the already available standardization

and referencing activities can be used to reach compliance.
We explore the research topic therefore by addressing the following three questions:

RQ1: Enabling Integration – How should IIoT Assets be represented in order to enable their seamless integration in IIoT settings? (cf. Ch. 4)

RQ2: Enabling Interaction – How do IIoT Assets and their Digital Twins interact with each other in dynamic and unpredictable environments? (cf. Ch. 5)

RQ3: Standards & Architectures – How can standardization activities ensure the compliance of independent IIoT integration efforts? (cf. Ch. 6)

The answer to **RQ1** demands a formalized, machine-readable, and extensible data model for the Asset itself but also for the ecosystem it is placed in. This data model acts as the common denominator between system and company borders and defines the virtual representation of the Asset and its characteristics and abilities.

RQ2 requires a general understanding of the IIoT domain and a detailed model of the impacting aspects. Such a conceptional framework reveals blank spots in existing models and thereby significantly increases the chances to meet future requirements. This foundation enables a scalable and unambiguous paradigm for data exchange and remote operation calls. It is necessary to align this communication model closely with the internet, more precisely with the Web, as this is the main entry point for any stakeholder. It is further necessary to clearly define the responsibilities and abilities of the involved parties to enable the on-the-fly coupling at runtime.

RQ3 is divided into an examination of the technical standardization landscape and analysis together with the use case-specific exploitations of the industry-wide recognized reference frameworks. While the digital communities are mainly organized by conventions, discussion states, and best practices, industrial manufacturing requires more formal processes and long-lasting specifications. The combination of both worlds need to be organized and interrelated for a complete picture, and the inherent complexity of the current landscape needs to be encapsulated through adaptive access methods.

1.5 Contributions and Publications

The identified research gaps have been addressed by the contributions CO1-CO6 (cf. Fig. 1.5) and described in peer-reviewed publications in highly regarded conference proceedings or journals (cf. Annex A). In the following, the main publications are grouped by the contribution they address. All the conference and workshop papers, as well as the journal articles, have undergone an independent peer-review process and have been evaluated by experts of the respective research domains. The data model of the Plattform Industrie 4.0 has the status of an industry specification and went through an elaborate review process executed by the associated companies.

RQ1 is answered by presenting interoperable information models that follow the design principles and conventions of the Semantic Web. They serve as extendable structures to encode the identities and attributes of Assets and formulate machine-readable statements to exchange information between heterogeneous systems.

CO1: Data Model for Digital Twins
The digital representation model for Assets presented in this thesis originates from the Digital Twin model of the Plattform Industrie 4.0, which was influenced and extended to semantic technologies. The alignment with the standardization processes ensures the adaption and applicability of the model

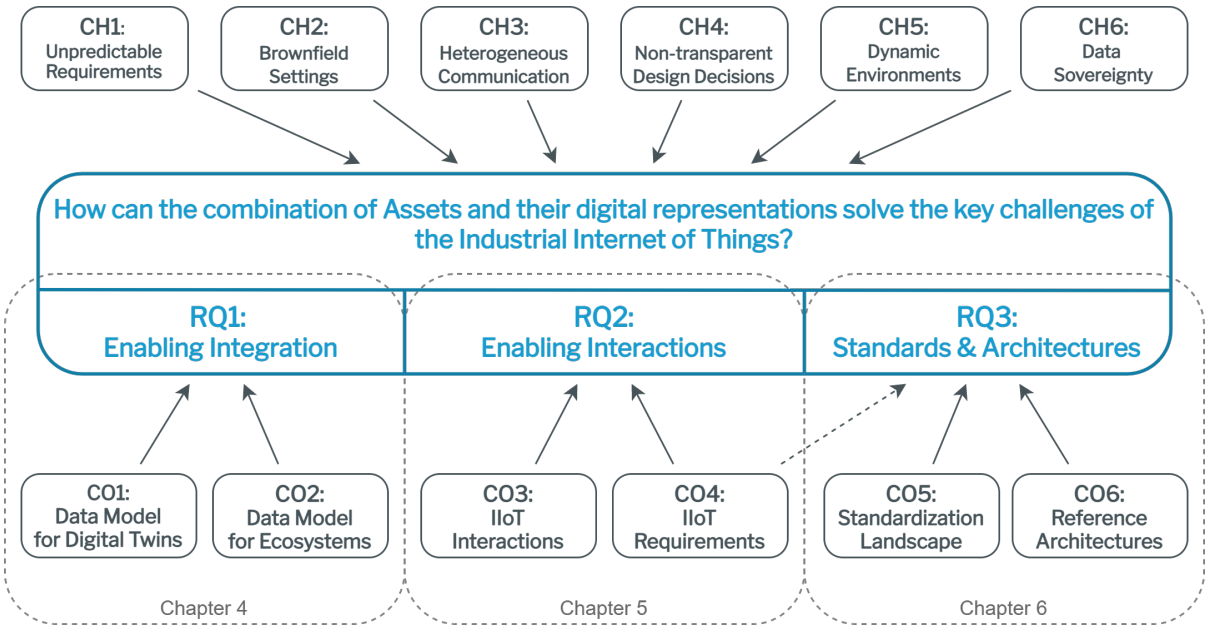


Figure 1.5: Overview of the regarded challenges, contributions, and research questions of this thesis.

while incorporating the technical characteristics of web resources. The resulting model is continuously enhanced and updated through an active community.

CO1 contains a vocabulary to express the Digital Twins of the robot from M_3 and its components from M_1 and M_2 throughout their lifecycles. It enables the quick integration into the facilities at O and the consumption through D. All involved applications know what to expect and are further able to extend the Digital Twins with domain- and use case-specific attributes.

The results of CO1 have been presented at the SEMANTiCS conference (SEMANTiCS2019 [19]) and as parts of the data model specification of the Plattform Industrie 4.0 [12].

CO2: Data Model for IIoT Ecosystems

The environment of IIoT Digital Twins is expressed in the second contribution. The definition of infrastructure components, their roles, and responsibilities in combination with a defined semantics on access and usage restrictions for digital objects paves the way for seamless IIoT networks. In combination with CO1, a concrete, standardized presentation of the digital entity itself as well as the surrounding ecosystem is delivered. Both together provide the answer to RQ1 and define the descriptive layer of IIoT Digital Twins.

The developed data model for IIoT environments is based on the works inside the International Data Spaces and strongly related to its specifications. The created language describes the hosting systems of all involved participants and trustworthy claims on their abilities to protect supplied data. It gives, for instance, the data analyst D the expressiveness to state the technical enforcement capabilities of his IT services and the operator O the capabilities to formulate protection rules bound to his Digital Twin.

The detailed aspects of CO2 have been published at the International Semantic Web Conference (ISWC2020 [20]) and in further peer-reviewed workshops [21] [22].

RQ2 requires the detailed examination of requirements in IIoT networks and, building on the thereby gained insights, defining a consistent operation semantics and interaction model for Digital Twins. The thereby specified interaction layer builds on the descriptive definitions from CO1 and CO2 and extends

them with homogeneous communication interfaces.

CO3: IIoT Interactions
The operation semantics and meaning of operations are defined in the third contribution. By incorporating and extending the standards of Linked Data, RESTful interfaces, and read/write operations on web resources, atomic IIoT patterns have been formulated. The clear separation of roles and their responsibilities simplifies the interactions and enables plug & control between heterogeneous Assets.

The proposed interaction paradigm gives O and D the mechanisms to integrate the robot without developing elaborated mediators and wrapping modules but relying on standardized APIs both for consuming data and sending commands. The resulting interaction layer is entirely compliant with the current web architecture and can thereby overcome the separation of the OT and IT worlds.

The contents of CO3 have been provided as a part of an article for the Future Internet Journal Special Issue on "Internet of Things (IoT) Applications for Industry 4.0" [23], at the IESS 2018 [24] and in peer-reviewed workshops [25, 26].

CO4: IIoT Requirements
The future-proven design and development of IIoT systems demands a consistent and comprehensive overview of the relevant features and requirements. CO4 comprises a detailed analysis of the functional and non-functional aspects, data and protocol-related considerations, and their implications on the digital sovereignty of the involved participants.

The fourth contribution defines and locates the determining categories in a layered architecture model to provide useful guidance towards comprehensive IIoT solutions, containing the required capabilities for future modifications and retrofitting of Assets. This contribution is the foundation for CO3 but also lays the basis for the generalized framework that is reflected in the examination of RQ3 (cf. Fig. 1.5). CO4 has been published as a part of an article for the Future Internet Journal Special Issue on "Internet of Things (IoT) Applications for Industry 4.0" [23] together with parts of CO3 and at the 9th Conference Professional Knowledge Management [27] .

The requirement model is, for instance, relevant for the manufacturers M_1, M_2, and M_3 to ship their Assets with adaptable interfaces and descriptions. It also gives O the first indication for relevant topics he needs to regard when creating its local IIoT network, and what to address during the updating activities. CO4 furthermore serves as a conceptional basis for the following contributions and thereby defines the analysis of the standardization landscape and the examination of IIoT frameworks.

RQ3 examines the previously described (CO1 & CO2) and interacting Digital Twins (CO3 & CO4) from an environmental perspective. They need to comply with detailed technical standards and form loosely coupled and adaptable architectures.

CO5: Landscape of Standards and Norms
The fifth contribution analyses the current state of the standardization activities in the IIoT. A semantically annotated knowledge graph is created according to the Linked Data and FAIR [28] principles and populated with the standards of the relevant standardization bodies. The thereby shaped landscape of technical standards is visualized through a configurable web service, a new analysis tool to discover and examine the content of IIoT standards and their interdependencies.

The knowledge graph and visualization service on top of it enable the target-oriented and straightforward discovery of relevant specifications. This enables, for instance, engineers of M_1, M_2, and $M3$ to achieve high-quality Assets and the network architects of O and D to create scalable and reliable communication networks.

The knowledge graph for IIoT standards has been discussed at the ESWC2020 conference [29], extending work that has been presented at the SEMANTiCS2017 conference [30], and the visualization

approach to analyze its content at the 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA2019) [31].

CO6: Reference Architecture Analysis
While CO5 focuses on standardization documents and their content, the sixth contribution analyses IIoT reference frameworks and architecture models. The former is dominated by long-established standardization bodies, whereas the latter is driven by relatively new research and industry consortia. This contribution combines an in-depth review of the current state with the knowledge graph and visualization service of CO5. The resulting new possibilities to examine this dynamic and unstructured domain enables the identification of areas with broad consensus but also of critical blind spots in the specifications.

The content of CO6 enables, for instance, *O* to select a suitable architecture pattern for his organization. He can rely on the specified best practices and guidelines and discover and combine them using the curated metadata from the graph. The visualization tools also help the engineers of M_3 to integrate the delivered components and design his final product according to the essential standards and best practices.

The details of CO6 have been outlined in an article of the Future Internet Journal [32], as parts of a paper published at Extended Semantic Web Conference (ESWC2017) [33] as well as at the 6th International Workshop on Services and Applications over Linked APIs and Data [34].

Furthermore, the content of the six contributions influences a number of industry specifications, for instance, the Reference Architecture Model of the IDS [35], and white papers [36]. In addition, a series of chapters in the Springer books "Smart Services" ([37], [38], [39], and [40]) as well as in the "Handbuch Industrie 4.0" [41] have been published to explain the gained insights also to the general audience.

The dissemination of the gained insights of this thesis is also ensured through the collaboration and active contributions in several standardization consortia. The IDS ecosystem has been influenced to a large extent through the developed approaches for IIoT interactions (primarily through a protocol binding based on the proposal of Chapter 5) as well as the data model with extensive data sovereignty aspects (Chapter 4). Similarly, the standard data model for the Asset Administration Shell [12], its interactions and security model reflect the recommendations gained through this research. In addition, the 2019 founded Gaia-X initiative presents an additional dissemination channel to transfer the research results into real-world applications. This is ensured through the tight collaboration in the official Gaia working groups and the contribution to the specifications, mainly the Gaia-X Technical Architecture Document [42] and the Federated Catalogue [43].

1.6 Research Methodology

As pointed out by Demeyer [44], research of computer science is at the intersection of mathematics with his hypothesis/proof approach and the more empirical and measurement-driven engineering domain. A plain mathematical-inspired research approach formulates hypotheses about the state of the world, which can be either verified or contradicted using formal deductions and proof theory. For instance, the developments in complexity theory are prominent representatives of this pattern. Empirical research, on the other hand, is based on observing and measuring the state of a defined system. Reproducible experiments, as for instance common in biology or medicine using statistical evaluations, are another proven method to gain new knowledge.

However, the target environment of this research work is also highly influenced by sociological aspects as communities of developers, system architects, or business deciders and their desires and actions affect the acceptance of proposed approaches. In general, crucial success metrics like community penetration or interoperability potential cannot be precisely measured. Estimating such criteria may be based on the judgment of experts and the opinion of the affected community.

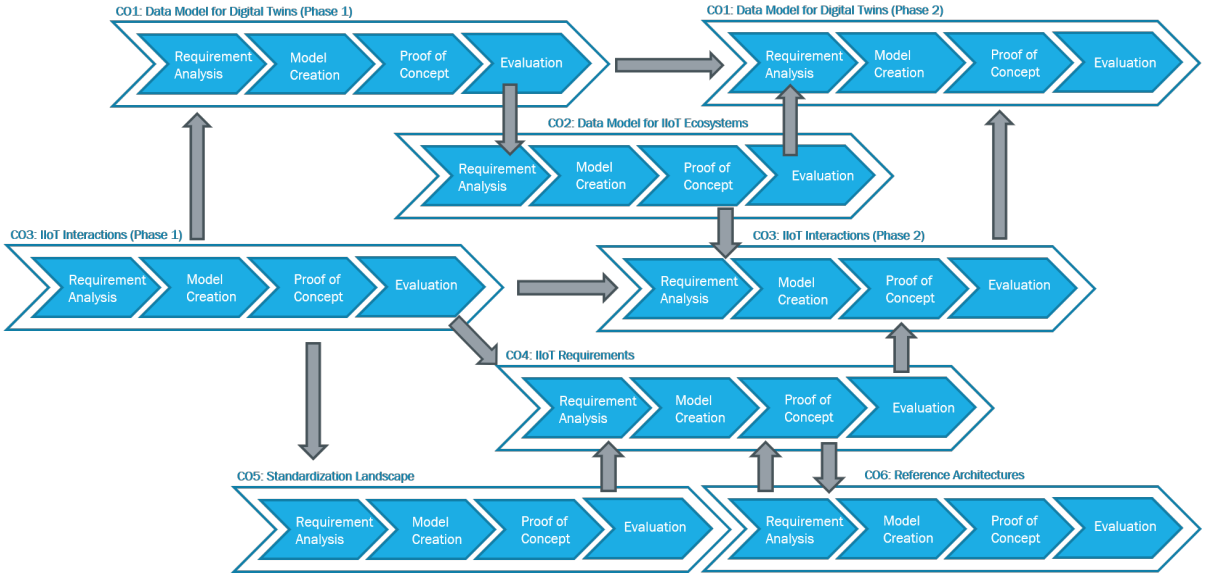


Figure 1.6: The research process has been partitioned into the depicted modules that impact and influence each other.

In order to target the identified research gap of this work, a mixture of the named methods and approaches must be selected. The previously defined main research question is examined by analyzing the three sub-problems. The single aspects of each sub-problem are then, in case they are objectively measurable (e.g., scalability, performance overhead, response times, etc.), empirically evaluated, and the results put into context. Wherever non-measurable aspects are affected, the evaluation is conducted through the judgment of domain experts and the acceptance in the respective community. Besides, the adaption of the contributions in standardization efforts is used as a quality reference.

The general research target is split into smaller research questions. Every one is addressed by several contributions and research activities that have been individually evaluated in peer-reviewed publication processes. Each single contribution is therefore quality-assured by independent experts and examined on its methodical soundness and scientific relevance.

Based on the decent view of the current state of discussion, a hybrid approach is applied to evaluate the different contributions. Empirical evaluation methods are combined with qualitative discussions and use case prototypes with proof of concepts. Each contribution is evaluated (Sections 4.2.9, 4.3.4, 5.3.6, 5.4.7, and 6.1.5) and their implications and limitations explained. The complete evaluation data is publicly available in open repositories, pointed out in the respective sections.

As explained, this combined approach is necessary due to the nature of the research field where technological factors, as well as social factors, are critical. Therefore, the adoption and implementation of the gained insights into practical solutions are as important as the soundness and correctness of the underlying theories.

Fig. 1.6 contains the selected research process. The contributions have been achieved in an iterative manner where the gained insights of each activity impacted the other ones. This is especially visible in the two phases for CO1 and CO3 that incorporate the results of the other contributions after a first, preliminary stage has been achieved. Grouped also by their impact on the research questions, the contributions CO1 and CO2 form a logical group as well as CO4, CO5, and CO6. Both groups are related by the two iterations of CO3, which combines all activities to a consistent output.

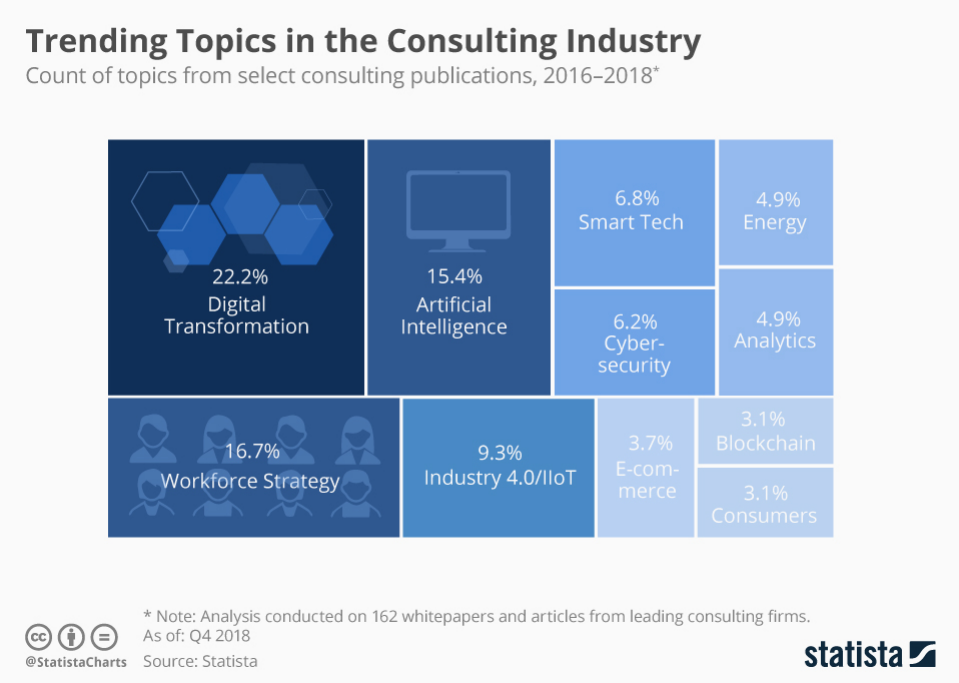


Figure 1.7: The IIoT is ranked as the fourth most relevant trending topic for consultants in 2018 [45].

1.7 Relevance and Impact

Research groups, companies, and governmental agencies around the world try to find answers to these questions. For instance, several international consortia and associations have been founded to drive the future IIoT. The German “Plattform Industrie 4.0” and the “International Data Spaces Association”, the American “Industrial Internet Consortia”, and the Chinese “Made in China 2025” are only a few examples. At the core, all these initiatives follow a similar approach: They specify a generic asset model, define standardized interfaces and data exchange workflows and prepare reference architectures. Still, the products and applications implementing these specifications are only very few. The lacking adoption can be seen as an indicator that additional research is still required and that further solutions are needed to realise the IIoT potential.

Driving this development is therefore crucial, especially to overcome the typical chicken-egg problem: Without significant investments, suppliers will not equip their devices with the necessary technologies and capabilities, and without the working proves, the investments will still be rather low. While indeed a trend in the right direction cannot be overseen anymore, the amount of actually implemented IIoT-ready facilities is still nowhere close to its promised impact.

A study conducted by the business association bitkom revealed that in 2017 and 2018 companies located in Germany ranked *Industry 4.0* and the *Internet of Things* directly behind IT Security and Cloud Computing as the most relevant IT Trends [46][47], even before Big Data, Blockchains and Mobile Applications. Similarly, as illustrated in Fig. 1.7, IIoT applications and the highly related Digital Transformation receive significant attention from consultants. A further study by 451 Research calculated that between 2017 and 2019, more than 860 million U.S. dollars had been invested worldwide in merging and acquisitions of IIoT-related companies [48], nearly twice the invested amount in the second biggest IoT sector. The overall market for IIoT solutions is estimated to reach an overall revenue of more than 77 billion U.S. dollars in 2020 and grow to 110 billion by 2025 [49], an average growth per year of around

7.4%[1].

The business potential in Germany confirms this impression. Continuously growing investments in IIoT applications reached 2.6 billion euro (roughly 3 billion dollars) in 2020 [50]. Most relevant, however, is to enable faster adjustments to change requirements and market changes significantly before reducing production costs [51]. That means that IIoT is regarded as a tool to improve the manufacturing capabilities rather than a cost reduction approach. In total, more than half of the German companies view themselves as well-prepared or even leaders in the domain [52]. The high awareness of the topic is further underlined by the perception that IIoT will affect nearly every manufacturing company. Only 1% of asked managers in 2020 regarded it as not important for their company, compared to still 9% in 2018. In general, one can observe a development from a general awareness of IIoT topics around 2015 to first projects mature enough to impact operative processes [53]. Nevertheless, still less than 10% of German companies claim that they are extensively using IIoT in their factories.

This shows the already existing impact of the IIoT vision on German and worldwide businesses and illustrates the vast potential for the future. This work is intended to support the observed trend by targeting the stated challenges and thereby help to drive the IIoT forward.

1.8 Thesis Structure

This thesis outlines first the methodological background of the presented research and the preliminaries of the regarded domain. The most relevant developments and publications build the foundation for the examination of the three research questions.

An overview of the existing work in the literature and an intense discussion of discussion state of the research community on the overall topic is further addressed in Chapter 3 after introducing the underlying technology stack as the preliminaries of this thesis (Chapter 2). Building on this foundation, the Chapters 4, 5, and 6 explain the contributions in detail and answer the research questions.

The respective answers to RQ1, RQ2, and RQ3 are then briefly repeated in Chapter 7 and combined to the solution for the overall research agenda. The delivered insights are discussed and their limitations examined. Possible extensions to cope with these limitations as well as additionally necessary contributions for the IIoT are explained in Chapter 8 to inspire further research activities.

The eight chapters of this thesis are therefore organized in the following manner, according to the relations and dependencies illustrated also in Fig. 1.5:

- **Chapter 1 - Introduction:** gives an overview of the thesis, motivates the topic, and explains the modules of this thesis and their connection points. It also contains the definitions of the core concepts, an overview of the scientific publications, and how they fit into the overall structure of this research work.
- **Chapter 2 - Preliminaries:** presents and explains the underlying technologies and developments and how they create the foundation for this work.
- **Chapter 3 - Related Work:** presents the state of the art and relevant research work, which is put into the context of this thesis and its contributions.
- **Chapter 4 - The Semantic Digital Twin:** explains the conducted work on the integrated data models for the IIoT and how the Assets need to be represented accordingly.

[1] Forecasts did not include the effects of the Corona pandemic and its implication on the worldwide economy.

- **Chapter 5: Interaction Patterns for the IIoT:** proposes interaction patterns and explains the contributions for flexible communication patterns based on the representation model of Assets.
- **Chapter 6 - Standardization Landscape and Reference Architectures:** combines the contributions from Chapter 4 and Chapter 5 to flexible IIoT architectures and proposes solutions to integrate norms and technical standards are into them.
- **Chapter 7 - Conclusion:** summarizes the executed work, answers the research questions, and explains the impact and the limitations of the presented contributions.
- **Chapter 8 - Future Work:** outlines potential next steps and exposed research gaps, which need to be addressed by further research activities.

CHAPTER 2

Preliminaries

This section introduces the prerequisites and foundations of this thesis. The most relevant developments can be grouped into the foundations of semantic technologies (Section 2.1), the developments around Web Services and their orchestration (Section 2.3.1), the Industrial Internet of Things (Section 2.2), and Digital Twins (Section 2.4). As all of these areas, their focus and understanding, undergoes a constant development, the meaning and shared understanding of the community changes over time. In order to illustrate these processes, and to present the current state of the art, each section outlines a brief timeline of the core developments.

2.1 Semantic Web

The term Semantic Web combines various technologies and developments to describe the meaning of data, information, and relations between resources in formal, machine-readable ways (cf. Fig. 2.1). Mainly relying on the Resource Description Framework (RDF), the Semantic Web contains a consistent set of standards and conventions to model self-describing information and to process them in further processes. The meaning of the data objects, its semantics, is encoded with the data itself. This self-descriptiveness of digital information enables the interpretation of data objects independent of additional definitions or external catalogs.

The information in the Semantic Web is regarded in the form of directed graphs. Data resources, nodes in such graphs, are interlinked with other resources through relations. The thereby created graphs can also be distributed across different data sets but also across IT systems. The combination of the RDF data model with the technologies from the Web, mainly hypermedia links and HTTP interactions, create a distributed and interrelated ecosystem of data resources on top of internet technologies.

The so-called Semantic Web Stack outlines the building blocks of this ecosystem. Mainly maintained by the World Wide Web Consortium (W3C), the Semantic Web Stack serves as a reference architecture for systems working in the Semantic Web. As such, the origins of the Semantic Web can be located in data modeling or communities and in contributions from formal logic domains, resulting in ontologies and formal expression languages like the Web Ontology Language (OWL) on top of RDF. The resulting standards, mostly W3C Recommendations, define the syntax of data resources, ways to express their semantic meaning, and the interaction or operation semantics with them.

While originally intended to represent machine-readable knowledge in the Web, the latest developments have created new requirements. The impression of a research-focused technology with less practical relevance is more and more faced with works that, for instance, add schema restrictions and validations (cf. SHACL), lightweight vocabularies that are easier to understand and use [54], and REST interactions

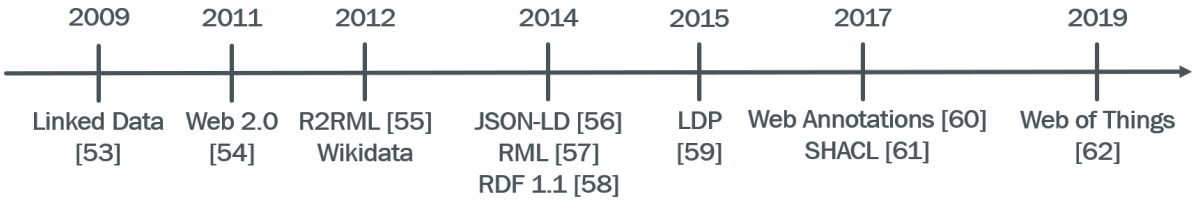


Figure 2.1: Timeline of selected Semantic Web developments from 2009 on.

Listing 2.1: RDF example of the robot arm in the NTriple syntax.

```
<http://manufacturer3.com/aas/robot/>
   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
   <https://admin-shell.io/aas/2/1/AssetAdministrationShell> .
<http://manufacturer3.com/aas/robot/>
   <http://www.w3.org/2000/01/rdf-schema#label>
   "Robot Gripper Arm"^^<http://www.w3.org/2001/XMLSchema#string> .
<http://manufacturer3.com/aas/robot/>
   <https://admin-shell.io/aas/2/1/asset>
   <http://manufacturer3.com/asset/gripper/755003377> .
```

Listing 2.2: RDF example in Turtle, equivalent representation as in Listing 2.1

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix aas: <https://admin-shell.io/aas/2/1/> .

<http://manufacturer3.com/aas/robot/>
  a aas:AssetAdminsitrationShell ;
  rdfs:label "Robot Gripper Arm"^^xsd:string ;
  aas:asset <http://manufacturer3.com/asset/gripper/755003377> .
```

(cf. Section 2.1.4). All these activities have the vision to create an interoperable layer for data and system integration, and to overcome the heterogeneity of systems, data silos, formats, and operations.

2.1.1 The Resource Description Framework

RDF [59] is a set of W3C Recommendations to model resources in the form of triples (subject, predicate, object). These triples form the basic building blocks to enable directed graphs of data resources. Each part of an RDF triple can either be a resource in the form of a globally unique IRI, a so-called Blank Node or, in case of the object position, also a sequence of characters referred to as value Literals (cf. Listing 2.1 and 2.2 and Fig. 2.2).

RDF representations therefore seamlessly support the connection of distributed data across (Web) documents by utilizing typed links. RDF also allows several serialization formats, for instance XML as RDF/XML or JSON as JSON-LD, and therefore serves as a flexible data format for both schema descriptions and instance data.

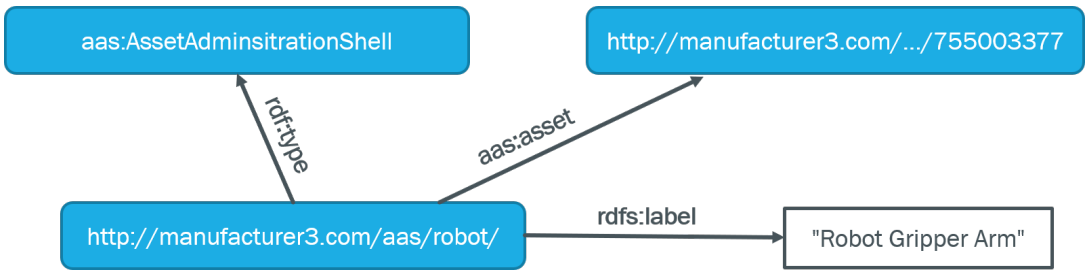


Figure 2.2: Visual representation of the example from Listing 2.1

In order to describe the nature of data objects, we use the following terms.

A Resource is anything that can have an identifier. This can be tangible objects, for instance machines or raw materials, but also intangible concepts like software, ideas, or attributes. In RDF, resources are the atomic building blocks. *rdfs:Resource* is the class of resources, therefore containing "everything". For instance, all entities in the example are included in *rdfs:Resource*.

Classes are groups of resources that share the same characteristics. The RDF class *rdfs:Class* is the class of classes. In the example listings, *aas:AssetAdministrationShell* is such a class.

Classes are realized by Instances, which are expressed by the *rdf:type* attribute (cf. Fig. 2.2).

Attributes, referred to as Properties in the following, are explicitly stated characteristics of classes and instances modeled in RDF. They appear as edges in the RDF graph and connect the nodes with each other. *rdf:type*, *rdfs:label*, and *aas:asset* all are Properties.

As expressed by the semiotic triangle [63], a Resource itself cannot be viewed, processed or exchanged as it is only an abstract idea of an Asset (cf. Fig. 2.3). With our example, while the reader reads the printed text about the robot gripper arm, an idea of this concept appears in the reader's mind. It's identifier, an IRI (*http://manufacturer3.com/aas/robot/*), refers to this concept. It is noteworthy to state that a single Resource can have more than one identifier, for instance the IRI "http://manufacturer3.com/aas/robot/" and the textual label "Robot Gripper Arm". Obviously, humans can manage such variations without problems. Machines on the other hand need explicit knowledge about the interrelations, an often underestimated obstacle for information processing systems.

As illustrated in Fig. 2.3, the actual Asset is also different to the already mentioned concepts. While the Resource is an abstract idea, this is - in general - not the case for the Asset itself[1]. Following the example, the physical robot gripper arm is obviously different from its identifier - printed in this text - and the idea about it - located in the mind of the reader. This is especially relevant for the later definition of the Digital Twin, where the explicit relation between the Asset and its digital representation are depend on this understanding.

Still, neither of the previously introduced concepts meet the requirements to exchange digital information about a Resource. In order to do so, an information carrier is required. This *Document*, called Representation in the following, contains descriptions about the Resource. For example, the Representation of the robot gripper arm makes statements on its type, name, and related asset. It is easy to see that the number of Representations is not limited, and usually depends on the amount of consuming parties and their use case. Furthermore, each representation must be encoded into a Serialization. Serializations might be - taking this thesis as an example - as a PDF, or a printout on paper, but also as NTriples (Fig. 2.1), Turtle (Fig. 2.2), or as a graphical notation (Fig. 2.2). The information content does not depend

[1] Remark from the authors: Even though often disregarded, the inequality of Resources - located in the mind of humans - and (physical) Assets - in general existing outside of such - is a very healthy fact and therefore relevant even beyond philosophical discussions. Experiments intended to prove the opposite did not work out well for the involved parties.

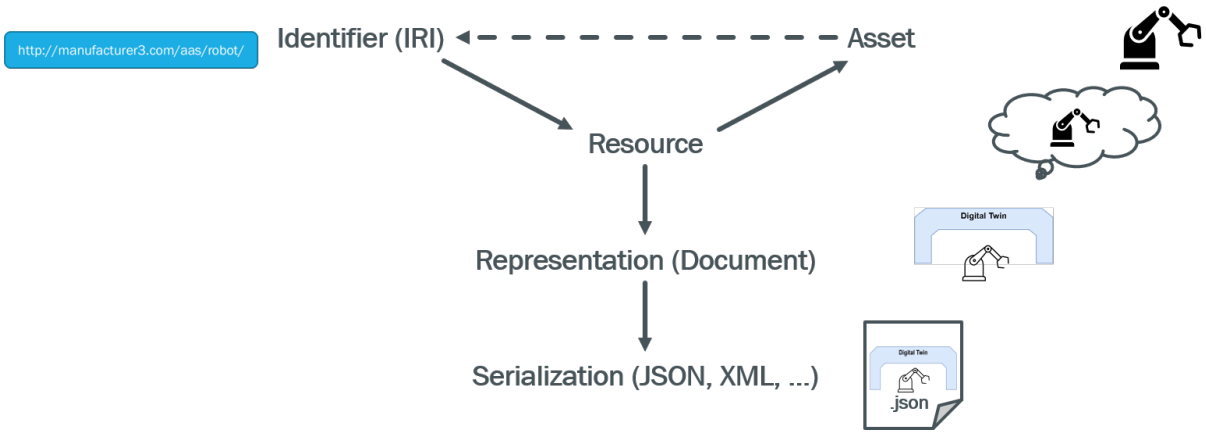


Figure 2.3: Information representation (inspired by Ogden and Richards [63], own illustration)

on the Serialization but solely on the Representation, and can be translated without information loss according to the needs of the involved parties.

This distinction is crucial to further define the interaction semantics and operations on digital resources. A significant amount of misunderstandings in the research and industry community results from differing implicit - but usually not made explicit - views whether the Resource, its Representation, or the Asset itself is targeted.

2.1.2 Ontologies and Formal Knowledge Representation

The RDF core and its extension, RDFS, define a core set of terms. In addition, first entailment rules give instructions how to derive new statements given a set of triples. For instance, the *rdf:type* semantic allows the conclusion, that *aas:AssetAdministrationShell* is a *rdfs:Class*. Such rules, or axioms, are either expressed by the definition of the used vocabularies but can also be expressed within the regarded triples.

Sets of triples, as for instance shown in the example of the robot gripper arm, are called Graphs. Graph partitions only containing describing classes and properties are usually called TBox, whereas information about instances form the ABox. Both together build a knowledge base, for which often the term ontology is used. Following the widely accepted definition from Gruber, “An ontology is an explicit specification of a [shared] conceptualization.” [64] As such, an ontology can be expressed using the RDF syntax. However, in order to have further expressiveness, usually a combination with the Web Ontology Language (OWL) is used to describe a domain of interest.

The proposed vocabularies within the Rdf and RDFS recommendations only build the frame of a usable RDF graph. To describe the content, and to understand the characteristics of resources, several commonly-known vocabularies have been proven their usefulness[2]. The Dublin Core Metadata Vocabulary [65] (dc and dcterms) and the Simple Knowledge Organization System Namespace (SKOS) [66] are examples for reappearing namespaces. The Data Catalog Vocabulary (DCAT) [67] is a related W3C Recommendation making use of well-established vocabularies to describe the distribution of (static) data sets. Nevertheless, DCAT does not include relations to originating organisations nor allows for the description of data-related service APIs.

In addition to a shared and clearly defined data format the APIs have to define the restrictions on their

[2]An indication for this might be the ranked list by Cyganiak: http://richard.cyganiak.de/blog/2011/02/top-100-most-popular-rdf-namespace-prefixes/ (accessed 19.10.2020)

input data. The Shapes Constraint Language (SHACL) [61] defines how constraints on RDF data objects can be formulated and which form validation results have. SHACL introduces schema-like mechanisms into the open world of RDF. As the statements itself are encoded in RDF, the language allows a Web API provider to demand certain information in a required format and publish this constraints in the same way as the rest of its RDF description.

Mappings of relational or otherwise formatted data to RDF are possible with the RDB to RDF Mapping Language R2RML [56] or the broader applicable RDF Mapping Language RML [58], which also enables mappings from JSON, XML or CSV to RDF. The desired transformations are also formulated in RDF by defining the output graph structure by so-called Maps and URI templates. While R2RML strictly relies on tables, and uses column names as resource and attribute identifiers of row-based data objects, RML also transforms JSON and XML data by identifying objects according to their keys. Even though some tools have been introduced in order to support the creation of mappings for both approaches, the possibility to collaboratively work on mappings was not part of the design requirements and is still missing.

2.1.3 Linked Data and distributed RDF

The Linked Data principles [68] are a set of four conventions to publish data on the Web, which make use of the combination of RDF and HTTP: To allow for data retrieval of resources, the first two principles suggest the usage HTTP URIs as Identifiers. The third principle recommends to use standard-conforming data representations such as RDF. To allow for the discovery of new information, the fourth principle advocates to include links in the representation document to point to further resources. Following these conventions results in interoperable and easy to parse Web documents, which can be regarded as partitions of a huge interconnected graph spanned on top of the infrastructure of the Web. Each commonly used Web client is thereby capable of following links, traversing the edges of this graph, and to discover new information on the fly.

The potential of Linked Data arises from the combination of the capabilities of the formal data modeling with the existing, scalable and distributed Web systems. The combination of best practices from ontologies with proven conventions how to publish and interact with resources on the Web, the complete World Wide Web can be regarded as the hosting system of semantic data. The Linked Open Data Cloud [3] illustrate the success of this approach and the feasibility to combine the two technology stacks. Staab, Lehmann and Verborgh give an overview of the design principles and discuss further developments especially regarding reliability and trustworthiness of the data [69].

2.1.4 Linked Data Platform

While the Linked Data principles are about data publishing, the interaction with web resources that are described in RDF was not in their scope. The W3C's Linked Data Platform (LDP) [60] specification closes the gap between the HTTP and the RDF specification for the RESTful interaction with web resources. The recommendation defines Linked Data Platform Resources and Linked Data Platform Containers. A LDP Resource guarantees a minimal set of common read operations and specifies how servers publishing such resources need to react to HTTP requests targeting the resource. LDP containers are collection LDP Resources with additional functionalities for creating new resources.

[3] https://lod-cloud.net/

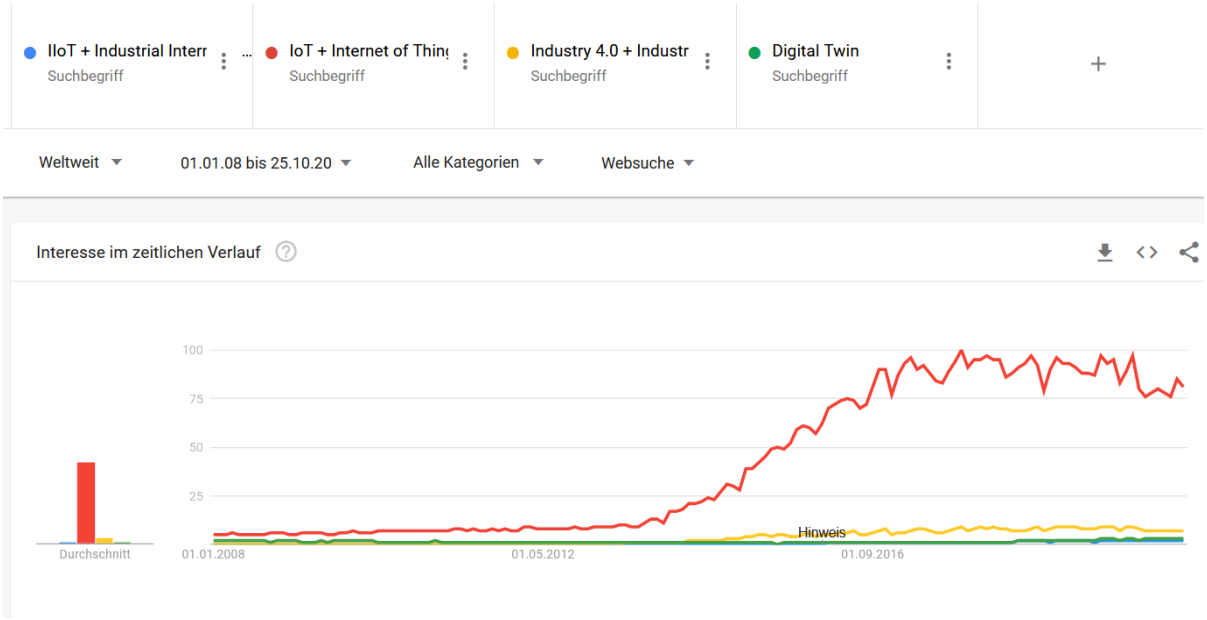


Figure 2.4: Popularity of terms as measured by Google Trends[4].

2.2 Industrial Internet of Things

In the following, the definitions as collected by Boyes et al. [70] are used. The terms 'Industrial Internet' (as defined by the IIC [4]), 'Industry 4.0' (as defined by Herman, Pentek and Otto [71]), 'Industrial Internet of Things' (as defined by Boyes et al. [70]) or even 'Cyber-physical Systems' (as for instance defined by Baheti and Gill [72]) all have different variations (cf. Fig. 2.9). In order to increase the readability, only the terms 'Industrial Internet of Things' and the acronym 'IIoT' as defined in Def. 1.1.1 are used as unifying terms.

This section outlines the core aspects of the IIoT and its differences to the traditional internet and the Web (cf. Fig. 2.5). Main differences are, for instance, the restricted computing resources and power supply of the devices, the high heterogeneity in terms of interfaces, interaction patterns, and capabilities, the strong interrelation between the virtual and the physical world and the still missing conventions on interaction and integration patterns. Figure 2.4 indicates the popularity and interest over time. One can note that the main interest has started around 2014.

The terminology of the IIoT combines *Internet of Things* (IoT) with industrial automation networks (cf. Fig. 2.5). Even though the specific origin of the term itself is hard to determine, one of the earliest usages is certainly the work of Sarma, Brock and Ashton around the year 2000 [1]. Nevertheless, the underlying concepts have already been mentioned before at several occasions.[5] While Ashton used the term 'Internet **for** Things', the term slightly changed and - as IoT - became a major trend topic in the following years.

In their original white paper, Sarma, Brock and Ashton strongly connect the IoT with physical tags inspired by the RFID [73] technology. At the time, the development of embedded devices driven by internet-enabled micro-controller was lacking the dissemination of the recent years ("However, TCP/IP

[4]Query request: https://trends.google.de/trends/explore?date=2008-01-01%202020-10-25&q=IIoT%20%2B%20Industrial%20Internet%20of%20Things,IoT%20%2B%20Internet%20of%20Things,Industry%204.0%20%2B%20Industrie%204.0,Digital%20Twin (accessed on 25.10.2020)

[5]for instance in a Forbes interview https://www.forbes.com/global/2002/0318/092.html#57711b5f3c3e

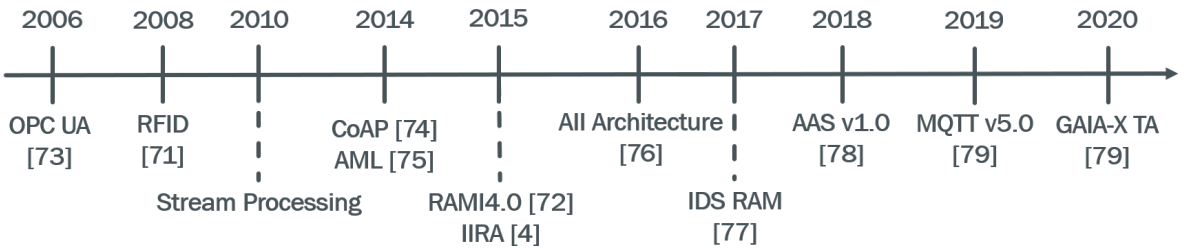


Figure 2.5: Timeline of selected IIoT developments from 2006 on.

controllers are expensive today." [1] p. 8), explaining the different direction of their understanding of the things in IoT networks. Apart of this, their proposal is quite similar to the current state of discussion.

Sarma, Brock and Ashton formulate a number of requirements to realize their vision of the IoT:

1. "A single, open architecture system for networking physical objects is more valuable than multiple, smaller scale, alternatives"
2. "The open architecture system must be platform independent and highly interoperable"
3. "The open architecture system must require a minimum of performance from the tag technology embedded in the objects"
4. "The open architecture system must be flexible and adaptable to change and improvement long after it has been installed"

The requirements 1 and 2 have been incorporated by all major IoT associations and standardization groups (Industrial Internet Consortium [4], Plattform Industrie 4.0 [74], Web of Things [62] etc.). Requirement 3 obviously is motivated by the strong relation towards RFID, a technology which did not appear as relevant as envisioned. Requirement 4 however is opening a field which is usually disregarded by the following research and standardization activities.

One of the core characteristics of (industrial) settings is their long operation time comparing to typical IT systems. While the later have update cycles of months (mobile applications) to years (operating systems or office tools), industrial facilities are in place for decades. The corresponding high investment costs enforce long-time operations. The challenge therein is that at the design time of such facilities and devices, the future use cases and requirements are hardly known. Suitable solutions therefore must be as adjustable and ready for upgrades as possible, even though that creates additional cost at the first place.

Current activities do not sufficiently regard this challenge. As today's greenfield environments quickly become tomorrow's legacy systems, the adaptability to new applications is one of the most crucial criteria deciding about the success or failure of a system.

The embedding of network and Internet-enabled devices into nearly any production-related asset, requiring digital communication between devices from different manufacturers, is definitely not a new development. Machine-to-machine protocols intend to solve this challenge, with the OPC UA stack [75, 81] as the most widely adopted solution. OPC UA defines the interactions, data structures and information models for entities on the shop floor, also often referred to as Operational Technology (OT). AutomationML is further often used to describe the specific logic in OPC UA-driven production workflows as e.g., described by Volz et al. [82].

Especially in the automation domain, communication and control networks relying on OPC UA provide vendor-independent machine-to-machine architectures. The OPC UA protocol stack specifies the data

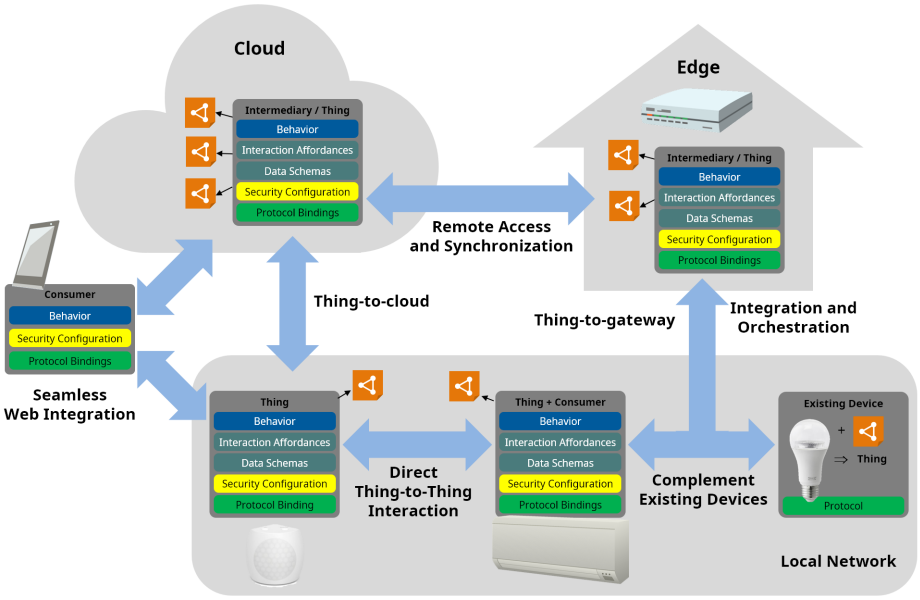


Figure 2.6: Reference Model as proposed by the WoT Community Group [62].

representation, remote interactions with resources and functionalities and its own information model. OPC UA-driven architectures have a strong focus on the operational technology domain, for instance intended for shop floor devices. Malakuti et al. [83] present a layered architecture for Digital Twins based on OPC UA. The interactions are enabled through either OPC UA sockets or REST APIs. Generally, the interactions close to the asset or devices are executed through OPC UA binary communication and the ones between the Digital Twin and higher-level applications rely on REST over HTTP.

The internet and in particular the World Wide Web offers already a well understood and widely accepted infrastructure to exchange data. Well-established Web technologies like URIs, HTTP and hyperlinks have proven to allow easy and reliable communication mechanisms in a decentralized manner. Cloud services and on demand solutions offer fast and flexible deployment of applications, a strict requirement for a smart factory. The Semantic Web further adds meaning to data objects and reduces the integration effort. However, the so called Web of Things does neither specify the interaction patterns of the regraded things nor does it model the intended relationship with the physical world.

The Web of Things (WoT) is an initiative of the W3C to model and outline common aspects of physical assets and represent them in the Web. An elaborate vocabulary and an interaction model demonstrate how independent entities can be described, operated, and orchestrated using the current practices and conventions of Web connections. The regarded requirements are technology-oriented and provide system architects and developers with implementable guidelines based on the currently used technologies.

2.2.1 The International Data Spaces

The International Data Spaces initiative (IDS; formerly “Industrial Data Space”) is building an ecosystem to facilitate data exchange in a secure, trusted, and semantically interoperable way. It aims at providing a basis for smart services and cross-company business processes, while at the same time guaranteeing data owners’ sovereignty over their content. The IDS Information Model is an RDFS/OWL ontology

defining the fundamental concepts for describing actors in a data space, their interactions, the resources exchanged by them, and data usage restrictions. After introducing the conceptual model and design of the ontology, we explain its implementation on top of standard ontologies as well as the process for its continuous evolution and quality assurance involving a community of industry and research organisations. We demonstrate tools that support generation, validation, and usage of instances of the ontology with the focus on data control and protection in a federated ecosystem.

Seamless collaboration and information exchange are the foundations of digital business models. Huge internet-based platforms have emerged, connecting people around the world and exchanging information in unprecedented speed. While end-users got used to such convenient communication and data exchange in their private interactions, they expect similar characteristics in their professional environment. However, data exchange in business-to-business relations faces a significant amount of still unresolved challenges. Data, and any form of digital information, is a crucial aspect of the intellectual property and thus competitive advantage of companies. However, in data-driven business models, most value is created through cooperation between several players, making data sharing a necessity. Both aspects lead to a typical dilemma of digital strategies – sharing valuable data involves the risk of losing the company's competitive advantage, whereas not participating prevents innovative business models and undermines upcoming revenue opportunities.

The International Data Spaces initiative[6] targets the requirements mentioned above by promoting a standard for virtual data spaces with reliable data exchange among business partners. To achieve the goal of sovereign data exchange, aspects of data management, semantic data integration, and security have to be addressed.

2.2.2 IIoT Standardization Initiatives

Several consortia have been formed in the context of Industry 4.0. One of the most prominent is the Industrial Internet Consortium (IIC), which developed the so-called Industrial Internet Reference Architecture (IIRA) [4]. Based on the structure of ISO 42010, the IIRA categorizes the outlined content in four viewpoints for business-, usage-, functional-, and implementation-related topics. Additional relevant documents in-depth examine methods of connectivity [84] and security [85]. The main scope of the IIC publications is in providing an overview of reasonable patterns and methods for the Industry 4.0 domain and providing a framework to reach a shared understanding. Therefore, the specifications are less restrictive than other guidelines.

The Reference Architecture Model for Industry 4.0 (RAMI4.0, cf. Fig. 2.7), developed by the Plattform Industrie 4.0, promotes a three-dimensional, layered model with additional dimensions regarding IEC 62890 lifecycle phases and asset hierarchies according to IEC 62264. The first-class citizen in terms of RAMI4.0 are asset of any kind, for instance production plants, machines, components, materials, but also software or services. The information carrier is the Asset Administration Shell (AAS) [12], which serves as a digital model of the asset itself. The strong focus on the AAS concept underlines the manufacturing oriented view of the model. Led by the necessity to seamlessly integrate assets, RAMI4.0 specifies Industry 4.0 concepts close to standards and norms to detail technical characteristics. Together with the American IIC, the Plattform Industrie 4.0 can be regarded as the most influential initiatives for IIoT topics. According to a bitkom survey, the majority (27%) of asked companies regard the U.S. as the leading country for IIoT directly followed by Germany (22%) [86].

The FIWARE Foundation promotes an open-source software stack to accomplish interoperability also beyond IoT use cases. The Next Generation Service Interface (NGSI) is a standardized Web API for

[6] https://internationaldataspaces.org

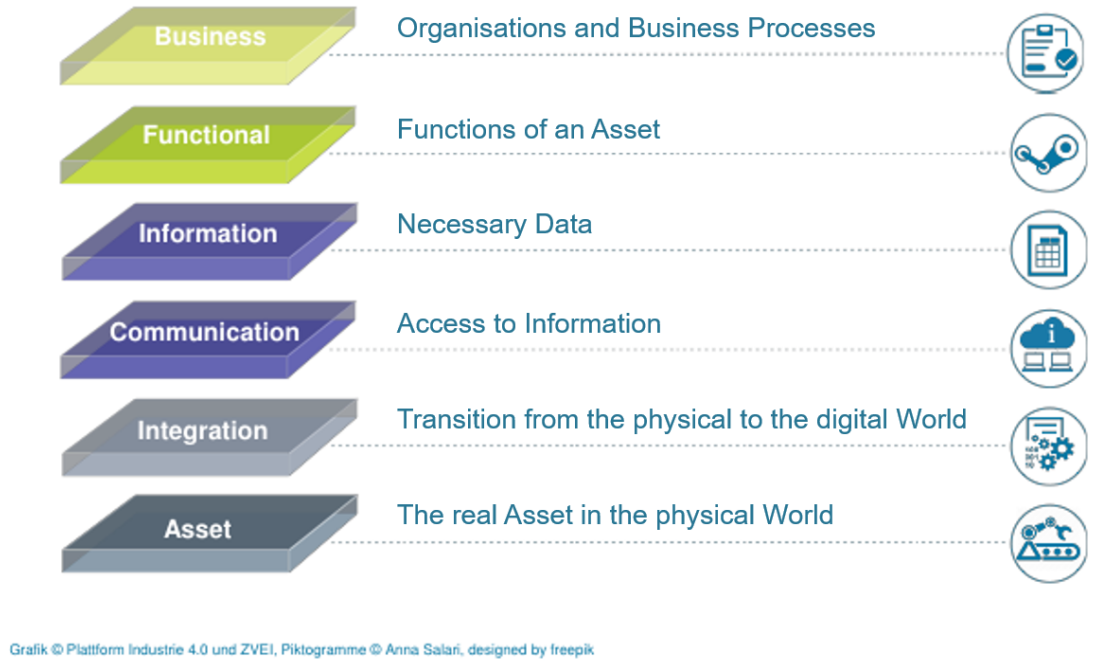


Figure 2.7: Layer structure of the Reference Architecture Model for the Industry 4.0

the IoT restricted to RESTful interactions. Any IoT protocol can be connected by suitable agents or wrappers, providing data towards the Orion Context Broker as the intermediary component for data and command transformation and translation. The currently specified NGSI-LD [87] provides a semantically annotated JSON syntax for context modeling and guidelines to interact with the respective resources.

The FIWARE reference architecture provides documentation for developers and system architects on cloud computing and how Big Data possibly enhances IIoT architectures on the higher network levels instead of regarding physical assets where concepts from for example, RAMI4.0 or IIRA are more detailed. In addition to HTTP serving as the suggested protocol with specified bindings FIWARE defines protocol-agnostic methods and context representations [87].

One part of the Chinese government's program 'Made in China 2025' is the Alliance of Industrial Internet (AII).While the prominent announcements about 'Made in China 2025' have drawn significant attention in the international media, the low interest in terms of web searches as shown in Table 6.4 is remarkable. One reason certainly is the, for instance in comparison with IIC activities, low visibility of the Alliance of Industrial Internet in English-speaking events and publication channels, even though the main reference architecture [77] illustrates the general scope also for the international audience.

Mainly driven by the focus on data-driven interactions at the sensors and actuator level, production data analytics level and the exchange of enterprise data, the AAI reference framework outlines the various necessary communication channels between the next generation of information technology (IT) systems for the office floor and operational technology (OT) systems for the production facilities.

The Industrial Internet Consortium aims to introduce common standards for the Industrial Internet of Things. As part of their reference architecture 2.8 the consortium discusses the trend to "base their control decisions on the simulation model rather than a control engineer's equation" [4]. Especially the introduced Functional Viewpoint contains the modeling of things but still only targets objects directly

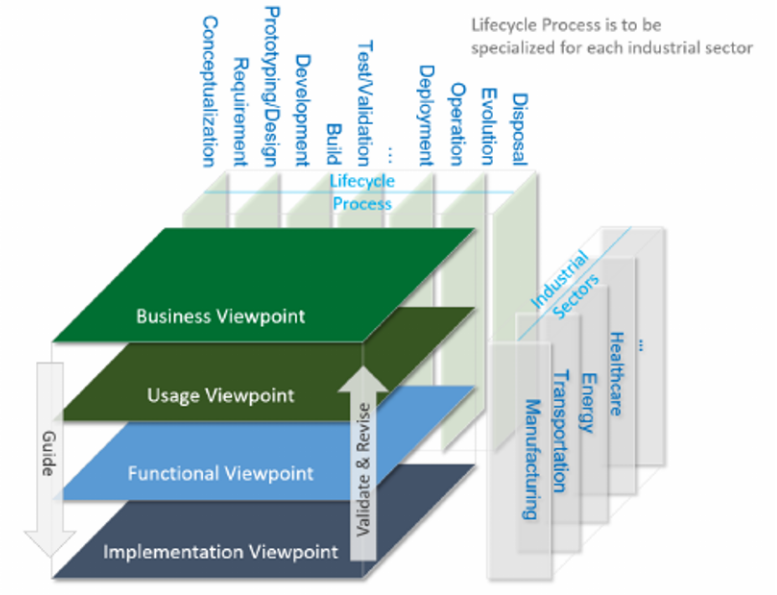


Figure 2.8: Industrial Internet Communication Framework [84]

equipped with sensors or actuators. In addition, context information is briefly discussed but limited to the semantically described relationships of the previously mentioned entities.

More from a Cloud perspective are the activities centered around the GAIA-X initiative [42]. It intends to create the specifications and standardize the interfaces and services that are necessary to enable the seamless consumption of services between different Cloud providers and the simple transport of data resources. The target is a federated network of independent clouds that can be used as a single, consistent data ecosystem driven by shared principles. The strong focus of the GAIA-X specifications towards data sovereignty and the assurance of data control presents it as an alternative to the currently dominated Cloud platforms controlled by single companies. As of 2021, GAIA-X is however still in the early phase of its specifications, with first prototypes being developed and evaluated.

2.3 Digital Interactions and Data Exchange

The distribution of IT systems and the partitioning of monolithic architectures into loosely-coupled services reduced the maintaining efforts and increased the flexibility of modern IT landscapes (cf. Fig. 2.10). Service-oriented Architectures (SOA) have become the dominant design pattern, connecting independent Services through internet protocols. Further developments like Cloud Computing have further paved the way to distributed applications.

In general two massaging styles are widely used for data providers and consumers in the internet. A push communication is started by the provider of information. An example are publish/subscribe systems where a data producer sends his content to anyone who has previously subscribed to his topic. Changes of a resource of interest can be communicated by periodically sending the current state of a resource or alternatively by publishing the occurrence of events which represent a significant change of the observed resource.

[7] Query request: https://trends.google.de/trends/explore?date=2008-01-01%202020-10-27&q=%2Fm%2F0n5rs,%2Fm%2F03nsxd,%2Fm%2F077dn,%2Fg%2F11cn3w0w9t,%2Fm%2F06kl5 (accessed on 27.10.2020)

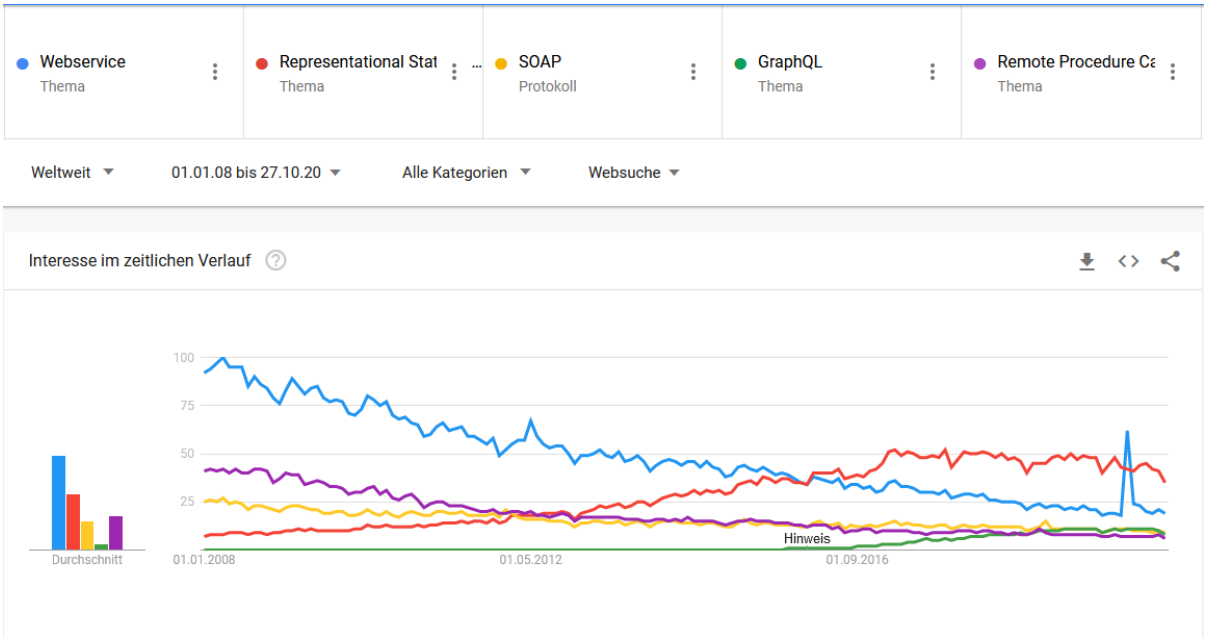


Figure 2.9: Popularity of interaction paradigms as measured by Google Trends[7].

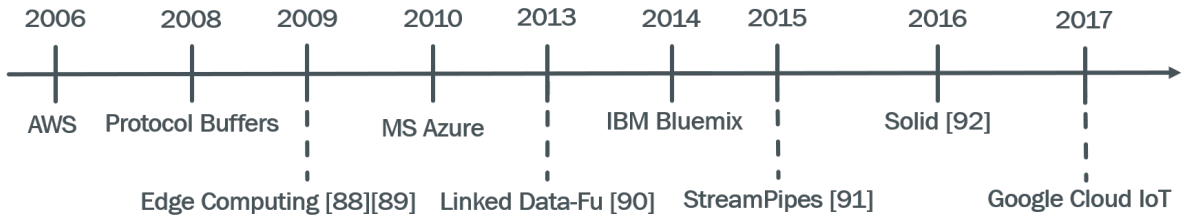


Figure 2.10: Timeline of selected communication technology developments from 2006 on.

In a pull interaction the consumer initiates the communication by sending a request to a specified provider. The provider then responses with a message containing the requested data, therefore this pattern is also referred as request/response. The REST principles (cf. Section 2.3.2) specify how the data handling shall be implemented. The common interaction with Web browsers and Web pages follows the request/response model.

2.3.1 Web Services

Web Services, and in particular the thereby framed technology stack of SOAP [93] for the protocol stack and its description language WSDL [94], composed an early approach to standardize remote service calls. The aim was to distribute functionalities across servers, and provide rich operation interfaces with standardized descriptions. The core challenge however for the success of Web Services is the complexity in the call semantics, an issue appearing when SOAP-based Web Services need to be integrated by foreign parties without direct contact to the original developers.

The core challenge of selecting suitable communication patterns for distributed systems is the agreement on implicit assumptions between the provider and consumer of remote interfaces. In situations where the same developer designed both the server and the client API, both sides can easily communicate

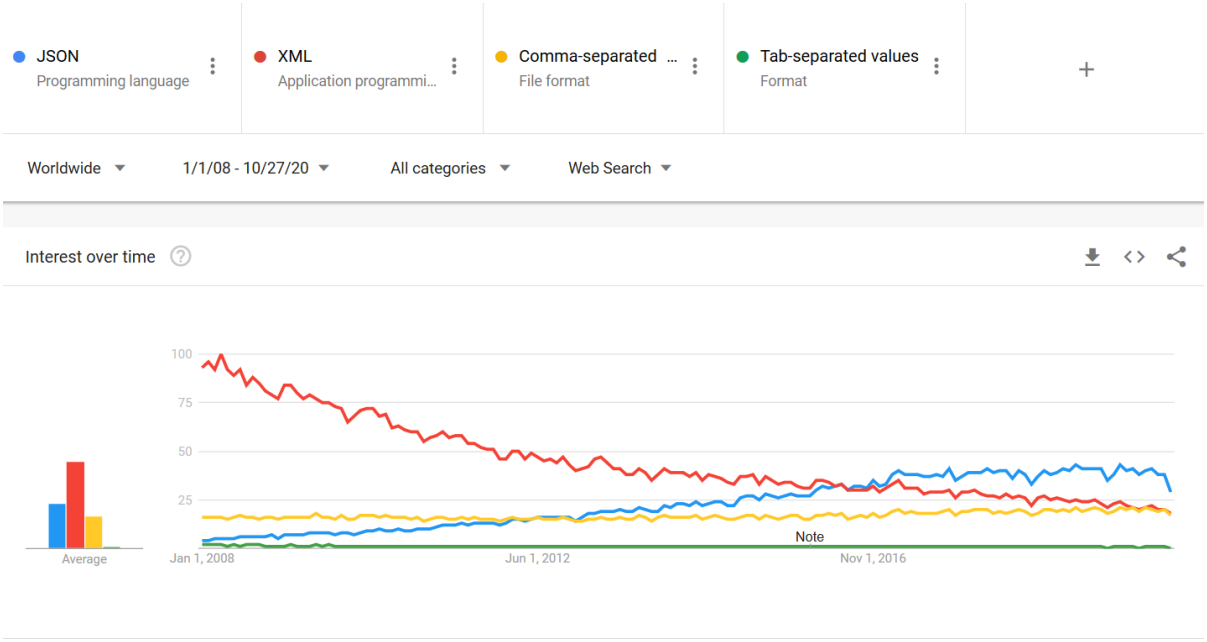


Figure 2.11: Popularity of structured data formats as measured by Google Trends[8].

and understand the meaning, the content, and the consequences of each interaction. However, this is usually not the case, leading to two potential strategies. The server API creator can describe the provided interface in detail, and provide this document to the client developer. Textual descriptions in natural language but also structured formats like WSDL, OWL-S, or WSMO serve this purpose. The obvious advantage of this approach is the high implementation freedom. Still, it creates a significant effort to properly describe the API. This effort usually is even higher on the client developer side, as the description very likely is incomplete, and can only contain issues and topics the API designer expected. As one of the core aspects of the Industrial Internet of Things lies in the re-usage of existing systems in originally unintended ways, this approach quickly reaches its boundaries.

On the other hand, the API designer can rely on defined practices and standardized rules. That way, the client can - independently of any documentation - interact with the provided functionalities. The therefore required high degree of regulation is however unrealistic as not one system API is exactly the same as another. Consequently, a decent trade-off between both approaches need to be found, which establish a common understanding on the operation semantics, the content and format of transferred data, and the implications of interactions.

Operation semantics contains the understanding on the meaning of a specific request. In our example, most developers would intuitively expect that “GET /aas” will return a description of the /aas resource. Nevertheless, protocols and standards can differ, therefore an explicit model of the meaning of operation calls is a necessity. The data content is independent of the operation semantic, and requires further definition. At the moment, a strong trend towards JSON as the data format of choice can be observed (cf. Fig. 2.11). Still, the meaning of resources and attributes is unclear and needs to be described.

In addition, both server and client need to share the same meaning on the consequences of an interaction. For instance, if the server offers an API to start or stop a process, maybe even put involved humans into a potentially hazardous situation, the client needs to be aware of this. Misunderstandings of this kind are

[8]Query request: https://trends.google.com/trends/explore?date=2008-01-01%202020-10-27&q=%2Fm%2F05cntt,%2Fm%2F08745,%2Fm%2F02hznc,%2Fm%2F0ds3b6k (accessed on 12.04.2021)

critical, especially if control systems are involved.

2.3.2 Representational State Transfer

Only making these assumptions explicit enables interoperable systems. The **RE**presentational **S**tate **T**ransfer (REST) paradigm for interactions in the Web combines a set of conventions and best practices into characteristics and guidelines for APIs. Originally formulated by Fielding [95], RESTful APIs are resource-oriented and follow a state-less communication model. As such, the effect of a single interaction is independent of previous interactions, simplifying the operation model for both the server and the client. Furthermore, the transitions of the resources themselves are regarded as changes of resource' states. The *state-less* nature of REST interactions is only related to the communication itself but not restricting the characteristics of the target resource. Its state is encoded in the representation document of the resource, which are exchanged between the server and client.

On the web, Uniform Resource Identifiers (URIs) [96] serve as the identifiers for resources. In our examples, we use URI templates as specified in [97]. A resource has a state that may change over time. To access and transfer the state of resources between components, we need a way to represent this state. Those resource representations need to include references to resources to allow for linking and discovering previously unknown resources in decentralised networked environments such as the web or the Web of Things.

The deep integration of Web principles into RESTful APIs is further underlined by the HATEOAS principle (Hypermedia As The Engine Of Application State), requiring hypermedia links to point the client to further resources. In addition, the REST principle give clear semantics on the possible HTTP operations by stating the meaning of the HTTP verbs (GET, PUT, POST, DELETE, PATCH, HEAD, OPTIONS, CONNECT), and explaining which are safe and which are idempotent. *Safe* interactions do not change the state of the world, for instance GET/read, while usually the state before a PUT/update operation is different to the one after. The *idempotence* characteristic states whether or not the multiple execution of an operation has a different result than a single one. For instance, deleting a single resource once has the same effect of executing it more often, making it an idempotent operation.

REST APIs therefore simplify the need to document an API behavior, as both the API provider and the client API developer follow the same principles. Obviously, still a decent documentation or explanation of the API is necessary. As a drawback, a REST API returns the complete representation of the resource, which is in general more data than required and therefore creating an unnecessary overhead. Furthermore, the client needs to interact in several steps instead of directly asking for the desired information.

But in order to allow human and machines to integrate Web APIs they have to understand not only the technical restrictions as data formats or input conditions but require a deep understanding of the used vocabulary. The OpenAPI Initiative[9] provides a small set of standardized field names to describe meta data of Web APIs. Though it is non RDF it applies semantic meaning and thereby helps structuring descriptions. Without some exceptions the definitions are very open in the form of supplied information and therefore not understandable for automated agents. Nevertheless, the clear structuring and the appealing displaying by the strongly connected SWAGGER tool[10] explain its wide usage. Hydra [98] on the other hand defines a strict semantic vocabulary for RDF which is also understandable by consuming programs. By supplying descriptions on available HTTP operations and their characteristics HYDRA supports the automated interpretation and configuration of HTTP connections. RESTdesc [99] in comparison does not describe the technical characteristics but the functionality of Web Services in the

[9] http://www.openapis.org/
[10] http://swagger.io/

form of rules. These rules explain the interaction of the service in the form of input data as a condition and an operation with resulting data in the head of the rule.

2.3.3 Combining Web Services

As already stated several exchange patterns for data are possible. If the consumer of data is not aware of update frequencies or always requires the current status of objects, a pushing of information from a data producer towards the consumer might be efficient. In this case, the producers trigger HTTP requests including the information to a set of consumers and therefore are the active component. The main problem here for the producer is to know which consumer is interested in his data and how to reach it. Common protocols like MQTT [80] solve this problem with message brokers or queues where interested consumers register themselves. Incoming data is distributed regarding topics or other filter criteria.

The StreamPipes approach [90] requires semantically annotated services and connects them through user input in a web application. The semantic information on input and output data is used to propose connectable services and to configure the data exchange automatically. Thereby, the user is liberated from a deep understanding of technical details and can focus on the overall process.

While the StreamPipes approach supports the management of pushing-based services, it is not capable of integrating pulling-dependent services. Especially in the Web, most resources (e.g. web sites) are provided on servers waiting for incoming requests by clients (e.g. browsers). Pushing data is not reasonable for them as it is usually unknown which of the potentially unlimited number of web clients is interested in their content. Therefore, the client needs to trigger the interaction. Linked Data-Fu [100] is an integration engine for these types of linked data components. Its declarative programming patterns directly targets a state-based view of the world as native to the web. The included reasoning capability for retrieved data enriches incoming RDF data regarding their semantic meaning and thereby supports the integration and interpretation of formerly unknown data objects.

2.4 Digital Twins

Digital Twins are comprehensive, independently operating digital representations of physical Assets, provisioned and manipulated through standardized interaction patterns, which thereby merge the tangible and virtual world. Real-world developments are reflected in digital models and vice versa. The concept of Digital Twins combine these facets to integrated entities, specifying the description, appearance and behavior of real-world entities in virtual models.

Originally introduced as a simulation-driven virtual representation of space and aircraft vehicles, the focus of Digital Twins has shifted to data interoperability concerns and to serve as a foundation for exchanging comprehensive data models of nearly any kind of physical assets (cf. Fig. 2.12). As this idea has gained a lot of attention, uncountable approaches, models and variation appeared in the recent years. Worth mentioning are however the proposals of the W3C working group Web of Things [101], the Digital Twin as described by the Industrial Internet Consortium [9], and the Asset Administration Shell specified by the Plattform Industrie 4.0 [12]. All three approaches are promoted by large communities and therefore comprise a sufficiently wide-spread consent.

Similar to the Industrial Internet Consortium the Plattform Industrie 4.0 proposes the ‘Asset Administration Shell’ containing an identifier, a referent, the referent definition and a set of characteristics [102]. It contains basic descriptions (header) of its provided data objects and functions in the body. An ‘Asset Administration Shell’ can be seen as a specification of the virtual entity concept from cyber-physical systems but does also only regard connected objects. Even though an ‘Asset Administration Shell’ can

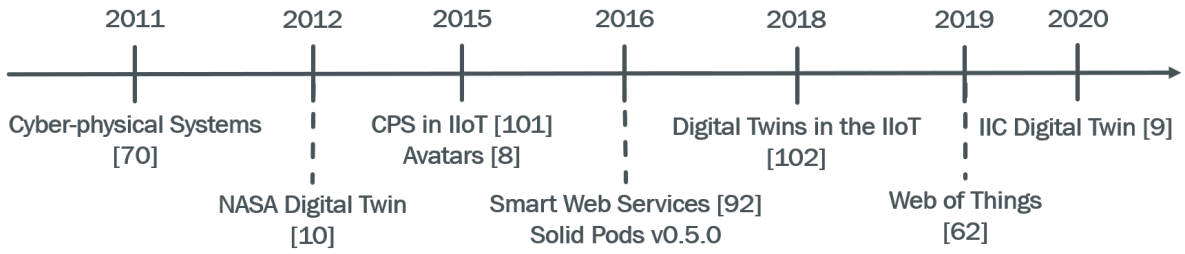


Figure 2.12: Timeline of selected concepts of Digital Twins from 2011 on.

also represent software or a Web service the authors do not discuss any patterns how the general concept also allows the introduction of unconnected things to the digital shop floor.

The Asset Administration Shell (AAS) is the promoted solution as defined by the Plattform Industrie 4.0 and published in DIN SPEC 91345 in accordance with RAMI4.0. The AAS concept contains a data model, protocol bindings and lifecycle specifications together with a security concept according to Attribute-based Access Control principles.

IoT data is currently mainly exchanged in either JSON or XML. These commonly used data formats ease the serialization and parsing by providing specifications for the syntactic structure of the data objects. Additional information on the meaning of keys/values is usually specified in customized data models and schemata. The latest specification of the AAS also follows this convention [12]. The AAS is the declared Digital Twin of the German Plattform Industrie 4.0 and encompasses the interpretation of the digital representation of any production-related asset. As such, materials and products, devices and machines but also software and digital services have a respective digital version.

While the predefined structure and the usage of specific keys reduce the heterogeneity inherent in the data exchange processes of current industrial scenarios, all real-world scenarios still require a thorough understanding of the specific terms and values. Therefore they are dependent on extensive manual work and understanding of the extended AAS model, followed by a time consuming data mapping. A semantic formalization of entities and data objects has several advantages in this context. The mature Semantic Web technology stack around RDF enables clear references to classes, properties and instances in the form of URIs, beyond the scope of single AAS objects but also across applications, domains, and organizations. The defined meaning of the used entities further allows its combination with predefined logical axioms, which allow the automatic derivation of new knowledge.

The security declarations of Asset Administration Shells follow the Attribute-based Access Control (ABAC) scheme. While being already included in the specification of the AAS data model, the maturity and expressiveness of the ABAC clauses have not yet reached the required level for implementations. For instance, the security-related part of the AAS model does not sufficiently reflect the particular requirements of sovereign data interactions, in particular the lack to specify data usage after leaving the original IT environment, which imposes a significant shortcoming for cross-organizational IIoT use cases.

The IIC Digital Twin concept is a functional analysis and requirement description specifying the virtual behavior. The IIC defines a Digital Twin as a formal digital representation of some asset, process or system that captures attributes and behaviors of that entity suitable for communication, storage, interpretation or processing within a certain context [9]. It is important to note that a Digital Twin does not only provide access to the lifecycle information of its asset. It also models the asset behavior through different types of asset models (such as physics-based, data-driven, etc.) and it offers value-added services (e.g. anomaly prediction) for industrial applications or other Digital Twins. The IIC focuses on

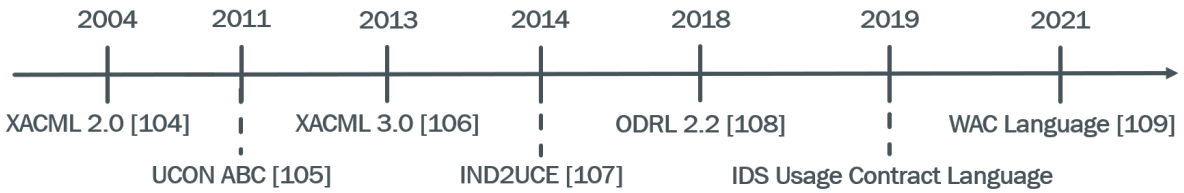


Figure 2.13: Timeline of selected developments for Usage and Access Control policies from 2004 on.

the aspects of interoperable systems and architectures but also specifies a brief vocabulary [103], intended to enable discussions between experts but not to serve as a formal information model for a machine to machine interactions. Consequently, the ideas and suggestions of the IIC Digital Twin model require further extensions and combinations with other frameworks before being applicable in IIoT scenarios.

2.5 Usage Policies

The proposed solutions of this thesis rely on the concepts of Data Sovereignty and *Data Usage Policies* through Access and Usage Control languages (cf. Fig. 2.13). These topics are briefly summarised in the following.

In current business interactions, permissions and obligations are declared through textual contracts. Wherever a formal, legally binding agreement needs to be documented, lawyers are required to express the intended meaning in written form. This is not suitable for the data-driven IIoT, as interactions are formed on the fly, dynamically and evolve and dissolve quickly.

In addition, the relation between the involved organisations needs to be reflected by the technical implementations. Currently, arrangements between business partners rely on responsible human actors, which are trusted by the opposite party to treat their assets according to the previously concluded agreements. In case of violations, the textual contracts and the legislative system serve as escalation and disciplining stages. The interaction speed and data volume of the regarded scenarios, however, limit the monitoring capability of the involved humans, therefore restricting their options to control the data processing processes. The solution is to demand the implementation of data restrictions directly as part of the respective systems and, as far as this thesis is concerned, into the Digital Twins.

The situation is slightly different for person-related data, especially in member countries of the EU. The General Data Protection Regulation (GDPR) defines certain rights and obligations for digital information related to humans. Still, the relevant data objects of the IIoT do usually refer to machine to machine communication and, therefore, are not in the scope of GDPR laws. As no other legislative option exists to enforce control on digital information, respective agreements have to be established on a case-by-case basis.

In the following, the terminology of *Contracts* is used when talking about legally binding agreements, which usually appear in written, natural language form. In contrast to these human-readable contracts, *Policies*, are understood as formally modelled descriptions:

Definition 2.5.1 *A **Data Usage Policy** is machine-readable representation of an agreement between two legal entities, exactly one assigner and one assignee, defining the permissions on a defined data artefact. Policies are serialised in a common data format (e.g. XML or JSON) and use shared vocabularies to unambiguously outline their intentions.*

In general, more than two parties can participate in the scope of a policy. This is intentionally left open for future work, together with other extensions that would result in further increases of the complexity

of a policy statement. It is also important to understand that Data Usage Policies, as defined here, are currently not legally binding. A binding agreement still requires textual contracts. Policies however omit the (intended) blurriness of legal clauses and need to be objectively decidable. Contracts on the other hand must be interpreted based on their context and the prevailing legal understanding of the contained clauses. The challenge is therefore to translate the established patterns of textual contracts into the digital world, and to create an equally recognised level of trust.

CHAPTER 3

Related Work

This chapter outlines the current state of the art on Digital Twins in the industrial internet and the previous work from the community regarding the six challenges. The relevant and most influential works are presented, their context and contributions analyzed and their relation to the content of this thesis evaluated. The literature analysis is organized in four main parts. The explained works of this chapter target the previously stated challenges and present the state of the research regarding the research agenda of this thesis. In addition, each contribution contains a focused literature analysis regarding the respective developments. As such, Section 3.1 targets the literature on the semantic modeling, Section 3.5 presents approaches to create data exchange under unknown requirements, and Section 3.8 and 3.9 proposals for flexible IIoT architectures.

3.1 Semantic Digital Twins

The usage of concepts is not consistent in the literature. Even though the mentioned publications use varying terms, they are used in this thesis according to the following understanding. A Asset, as introduced in Definition 1.2.2, can be either a physical object, a software program, a piece of information or even a human or any other kind of identifiable resource. With the IIoT, more and more Assets are represented in the virtual world. In order to establish a digital information exchange, a virtual Resource is created to identify and supply its attributes and provide control interfaces. Different terms are used in the literature for such virtual resources, for instance Digital Counterpart, Avatar, or Virtual Representation. Within this work, this digital resource is mentioned as the Digital Twin, even if the referenced publications write about “IoT devices” [110], “digital shadow” [111] or “virtual representation” [25].

Glaessgen and Stargel formulate requirements on so called digital twins regarding NASA and US airforce vehicles [10]. Their focus is to simulate any perceived incident to the physical vehicle at the virtual one in order to get a higher accuracy predicting the current state of the vehicle. Tao et al. focus on the product lifecycle (design, manufacturing, service) [7]. They identify a research gap in the field of Product Lifecycle Management in the form of a disconnection between the physical object and the virtual information available during the several lifecycle stages. Their digital twin concept follows the definition of Glaessgen and Stargel and focuses on information delivered in the form of virtual objects but do not consider any manipulations of those. Therefore, their digital twin concept mainly serves as a virtual model and information container and rather the Asset itself as any (virtual) interaction pattern is missing.

Even though the scope of Digital Twins therefore is large, significant standardization activities have been started to create coherent definitions of how a Digital Twin can be defined, which functionalities it needs to provide and which requirements have to be faced. The German Plattform Industrie 4.0

(PI4.0) and the American Industrial Internet Consortium (IIC) have published first specifications for the manufacturing industry.

The internet and in particular the World Wide Web offers already a well understood and widely accepted infrastructure to exchange data. Well-established Web technologies like URIs, HTTP and hyperlinks have proven to allow easy and reliable communication mechanisms in a decentralized manner. Cloud services and on demand solutions offer fast and flexible deployment of applications, a strict requirement for a smart factory. The Semantic Web further add meaning to data objects and further reduce the integration effort. However, the so called Web of Things does neither specify the interaction patterns of the regraded Assets nor does it model the intended relationship with the physical world.

Perera et al. write about the Avatar concept as an interoperability concept between objects "using standard protocols and technologies defined by the W3C for the Web, coupled with a distributed service-oriented mediation infrastructure" [110]. Thus, avatars can serve as a representation for both physical and software Assets but do not contain the formalism to simulate unconnected objects. The terminology of 'avatars' however is for instance also used by Hribernik et al. in the sense of a digital shadow with autonomous decision making capabilities [111]. This is only one example that the definitions are not used consistently and highly depend on the publication time, the research community and sometimes even the research group.

Agents as e.g. mentioned by Hendler [112] on the other hand are in general not restricted to physical Assets or resources but perform higher level actions on behalf of a user. Consequently, agents in that sense are not meant to constantly represent the status of an IIoT Asset and are not applicable for the described use case.

'Physical entities' and 'virtual entities' are basic concepts in the domain of cyber-physical systems. Lee, Bagheri and Kao introduce a general five layered architecture for cyber-physical systems. Their cyber layer includes 'cyber twins' which capture and preprocess the captured data for higher level applications [113]. The thereby created virtual modeling of the factory and its entities are the consistent data suppliers for the factory management. Unfortunately, they do not consider the enormous necessary effort to fully represent all machines of a production line and require to have all entities at every step and do not discuss how an environment with only partly digitized machines can benefit from their vision.

The intention behind the concept of Digital Twins have been examined by many works, among them Negri, Fumagalli, and Macchi [114], Jocaby and Usländer [115], and Souza et al. [116]. The overview provided by Negri, Fumagalli, and Macchi observes an evolution towards virtual factories, while still the information appearing at the different lifecycle phases of the Assets are not yet sufficiently connected [114]. In general, they note a development from an Asset'S health oriented view, over attempts to digitally mirror the life of the Assets, to an entity intended to support decision-making processes. The presented works focus on the physical aspects of Digital Twins, rather than regarding their potential also as self-descriptive building blocks for system integration.

Souza et al. interpret the current consensus in a more communication oriented manner [116]. The Digital Twin in their architecture encapsulates an IIoT gateway between the Asset and the Digital Twin representation, which redirects all connections through a consistent and standardized way. Their understanding however misses the need to not only align operations and protocols but also integrate, and previously understand, the meaning of the supplied attributes. This aspect is also included in the overview from Jacoby and Usländer [115]. They follow the Meta Object Specification from the Object Manangement Group [117] to describe the relations of the different kinds of necessary data and information models. A Digital Twin in their interpretation combines standardized communication and identification capabilities on the interaction level but also rich semantic descriptions and definitions to explain its contained resources. This interpretation is already close to the one proposed in this thesis, nevertheless the authors miss to propose concrete specifications and standard patterns for each identified

aspect.

The discussion about the Digital Twin in the IIoT is not possible without relating it to the different manifestations they can have. Perera et al. classify IIoT Digital Twins in six categories according to two dimensions - their *computing capability* (resourceful/constrained/resource-less) and their *location* (fixed/mobile)) [110]. This requires that any regarded Asset can at least be digitally identified in the physical world - for instance through RFID tags or bar codes - or even actively identify themselves, as for instance TCP/IP capable devices.

Sjarov et al. examine different categories based on the usage purpose [118]. They distinguish *type*, *instance*, and *business-related* aspects. *Types* frame all attributes shared across all *instances* of a certain product type, for instance the physical shape or manufacturer, while the latter are specific to one distinct device or component, like operation time or abrasion state. They further distinguish the attributes of IIoT Digital Twins into descriptive, exploratory features and activity-oriented features. The first are limited to present the current state and enable transparency throughout the network, while the second aim to induce specific decisions, for instance, to autonomously decide on optimal maintenance windows. This classification scheme makes it obvious that the operating, actually connected IIoT devices can only be instance representations. Type-related attributes always need to be merged from further sources, like engineering systems. It is therefore important to regard the different kind of the target resource, and reflect this in the representation as a Digital Twin.

3.1.1 Semantic Input Modelling

The benefits of an integrated data management approach were already discussed in 1994 and that it can lead to a significant competitive advantage [119]. By comparing more than 100 service providers, the authors point out how standardization and supporting systems can improve the overall efficiency. Some of the suggested improvements, like e.g. wireless communication, have become widely used techniques in the meantime. But the overall problem of supplying the necessary information for the right task at the right time is still not solved.

Yamauchi, Whalen and Bobrow claim that informal information sources like short notes and activity reports are crucial in case a non-trivial problem is faced [120]. According to their observations, people first try to find an explaining story based on own experience and advice by colleagues. If this procedure fails, they continue by searching the controlled documentation base. Consequently, both sources are necessary but at different stages of the process. Schweitzer and Aurich propose a continuous improvement process for organization where both controlled and uncontrolled documents are shared in the maintenance network [121]. This approach allows especially to grant customers but also suppliers with read/write access to the knowledge base. Although outlining the economic necessity, the authors do not propose any solution in order to effectively connect the various systems and to guarantee a common understanding of data through the whole network.

Modularizing and classifying information artifacts with semantic annotations provides two major advantages, namely an improved retrieval by revealing the power of semantic queries and a better interoperability across different systems. Uren et al. [122] give an overview of manual and automated tools for semantic annotations. They state that a manual annotation process is too labour-intensive and therefore must be automated. They outline the three major strategies of rule- or pattern-based systems, supervised, and unsupervised machine learning approaches. They claim that the required skills to configure the automated annotation systems and the amount of effort to create training data is not always justified with a suitable annotation quality.

Well-established Web technologies such as URIs, HTTP and hyperlinks have proven to allow easy and reliable communication mechanisms in a decentralized manner. Cloud services and on-demand Web

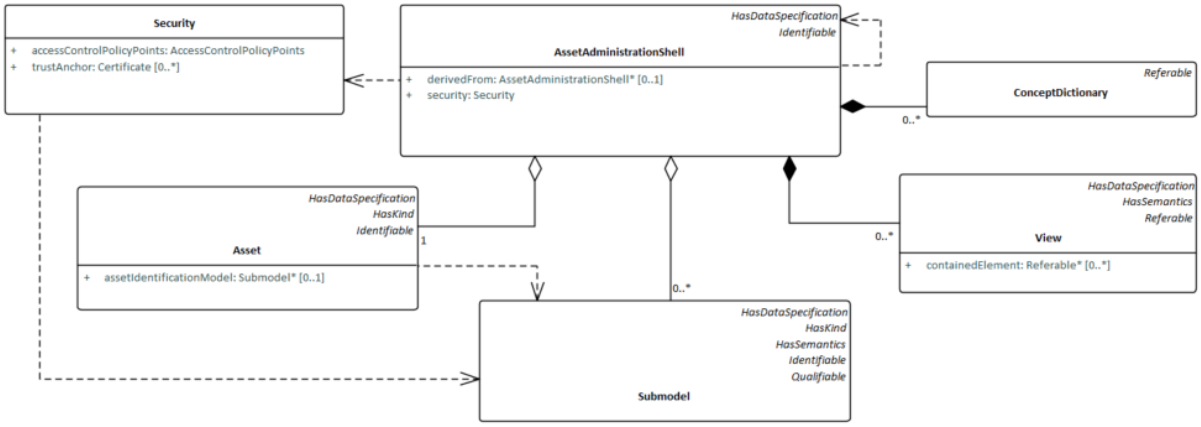


Figure 3.1: Core classes of the Asset Administration Shell Data Model according to [12] (page 44).

solutions offer fast and flexible deployment of applications, a necessary requirement for a smart factory. Maleshkova, Pedrinaci and Domingue analyzed the state of deployed Web APIs and RESTful services [123, 124]. They conclude that a significant number of APIs lacks sufficient descriptions and at the same time miss necessary information on both the input and output data sets.

While there is already a nearly uncountable variety of proposals how the Digital Twin concept needs to be integrated into the IIoT, the Asset Administration Shell (AAS) can be regarded as the one with the most active community, sufficient support through dedicated associations and the required acceptance of the industry (cf. Fig. 3.1). This section outlines the relevant works to enhance the AAS to the identified challenges of this thesis.

Barnstedt et al. define the data model of the AAS [12], the form of identifiers, access rights and roles, as well as XML and JSON serializations and a preliminary transport format. The textual documentation of the model is enhanced with XML and JSON schemata, derived from the leading UML model. It defines a basic set of keys and properties, and outlines extension points for custom vocabularies and terminologies. Additional parts of the specification will further standardize the APIs and interaction functions of the AAS, and how operations can be provided and described for the Industrie 4.0.

Grangel-González provide a first RDF data model for the AAS and the respective technical standards as published by ISO, IECC, and DIN [125]. They further extended the work in [126] with a formalized model of the Reference Architecture for Industrie 4.0 (RAMI4.0) and entities for units of measurements and provenance, and show a prototypical mapping using R2RML examples. However, the mapping itself was not generally applicable to other Asset Administration Shells as a common data model was not specified at this time. The work therefore shows the principle applicability of semantic technologies to the AAS but can not serve as a usable reference for actual implementations.

Tantik and Anderl [127] present an analysis how recommendations of the W3C fit to the guidelines of the Plattform Industrie 4.0. They outline various suggestions how standardized Web technologies can be integrated into Asset Administration Shells. The authors present best practices and integration methods through a sample implementation scenario but do not discuss the implications on the data model itself.

Katie et al. [128] show by integrating the machine-to-machine communication protocol OPC-UA for servers and clients how semantic descriptions, in particular SAWSDL annotations, bridge the gap between the heterogeneous devices of the shop floor. The use of uniquely identified semantic descriptions supports the automatic orchestration of decoupled Cyber-physical Systems. However, only the specific input and output requirements of the OPC-UA methods are described. Neither the data objects nor the OPC-UA general information model is reflected.

This gap is targeted by the "OPC UA for Asset Administration Shell Companion Specification"[1]. A Companion Specification is an official release of the OPC UA Foundation to define the interactions and data model in OPC UA-conform networks. It explains the data syntax and the mapping to OPC UA nodes, which can be hosted by OPC UA servers and requested by OPC UA clients. The *meaning* itself however is only encoded in the form of named labels for the used entities. Further expression of formal relations or references to non-OPC UA-based resources are not intended.

Dietrich et al. examine the semantic characteristics of the Asset Administration Shell in [129]. They outline the identification of attributes and properties through cross-industry standards, mainly IEC 61360 and ECLASS. The IEC 61360 series specifies the *Common Data Dictionary*, an IEC-maintained catalog of product classifications. Its entries are organized as a taxonomy with International Registration Data Identifiers (IRDI) to unambiguously encode terms. IEC CDD terms all share the ISO-reserved International Code Designator "0112".

Similar to the IEC CDD, the ECLASS catalog propose a taxonomy of IRDI-identified terms in a hierarchical manner but under the International Code Designator "0173". Both ECLASS and IEC CDD provide a set of human-readable fields, for instance the preferred name and the version and revision of the term. The catalogs thereby enable the exchange of product details with clear and world wide unique identifiers, as well as online services for look ups. However, neither ECLASS or the IEC CDD have the expressiveness to state formal relations beyond the underlying taxonomy, or supply dereferencable identificators. The corresponding Web portals target human users, and can hardly be used directly by machines.

Dietrich et al. further discuss mappings of the AAS data model to AutomationML and OPC-UA in general. However, they do not regard any relations to the Web but focus solely on machine-to-machine communication on the shop floor. Furthermore, Dietrich et al. do not show how to integrate the Asset Administration Shell with the technology stack of the Semantic Web, which in the end again leads to the distinction between the OT area and the IT world. This shortcoming however is one of the most critical ones to overcome for a real integration into a seamless IIoT.

There was no sufficiently complete RDF representation of the officially released data model of the Asset Administration Shell. This is however necessary to build the required bridge between the latest approaches of data provisioning models in the manufacturing domain and the rich and mature data integration and formalization capabilities of the Semantic Web. As such, an RDF data model has the potential to ease the information exchange but also provides the capabilities to introduce logical reasoning to the AAS as the leading Digital Twin concept of the IIoT.

3.1.2 Semantic Description Languages

Semantic descriptions of Web components can be formulated in various languages and ontologies. Currently most important are the Web Service Ontology Language (WSMO) [130], OWL-S [131] and Linked USDL [132]. RESTdesc [99] utilizes a N3 Syntax to specify input and output parameters and how they are connected. Similarly, Dimou et al. [133] combine access information with data mappings in the RDF Mapping language (RML). Verborgh et al. provide a survey [134] on machine-interpretable Web API descriptions. The types of descriptions can be organized in the categories behavioral, functional and non-functional. The technical details to operate the component are part of the behavioral sections whereas functional statements include basic information on the component's purpose. Non-functional information contain additional details on e.g. prices, provenance or provided Quality of Service. However, only a small fraction of available Web components is explicitly equipped with such descriptions. Karma [135]

[1] The Companion Specification is under review as of December 2020. It is expected to be released at https://opcfoundation.org/about/opc-technologies/opc-ua/ua-companion-specifications/

supports the annotation process by proposing a syntactic and semantic model based on samples of API requests.

Also, central registries for Web APIs like RapidAPI[2] or ProgrammableWeb mostly do not provide semantic information, making an automated discovery a hard task. In the approach of Sande et al. [136] for Linked Data sets the data server recognizes other components by dereferencing its existing RDF data or utilizing the Referer Header of incoming HTTP requests and therefore gains knowledge about other data sources.

A huge amount of semantic description languages for interfaces and federated systems have been proposed in the past. The SOAP technology stack and its service description language WSDL has been extended with the WSMO and WSMO-Light ontologies [137]. OWL-S is a similar OWL-based ontology for semantic descriptions of services. Furthermore, description languages for REST APIs have recently gained popularity, most prominently OpenAPI[3]. In addition, API Blueprint[4] and RAML[5], but also the RDF-based Hydra [98], illustrate the great need to unambiguously describe remote interfaces in both human and machine-readable manners.

The IDS Information Model's definition of an *Interface*, e.g., of a Resource, is technology-agnostic, comparable to Web Service Interfaces in WSDL 2.0[6] and the concept of Service Profiles in OWL-S ontology[7]. There are two alternative approaches in the Information Model to operationalize it. Still, the focus is on the functionality of the endpoints itself, disregarding the challenges proposed through data protection and trust requirements.

In addition to plain description languages, several ecosystems have been designed to seamlessly exchange data. bIoTope[8] aims at enabling interoperability between vertical IoT silos via a standardised open API. Data integration is supposed to be based on vocabularies to describe the different data sources. FIWARE[9] provides data through a RESTful API with RDF semantics called NGSI-LD[10]. Besides the claim to reduce JSON payload costs and a full REST adoption, it offers a more powerful query language, especially for geospatial queries. The NGSI-LD interfaces contain an own presentation of the graph structure of semantic data, however, they have in common the "Context" concern of data, e.g., in a spatio-temporal sense. Nevertheless, these ecosystems do not sufficiently express the conditions and restrictions imposed through digital information exchange.

Web agents and representations of avatars have been suggested by Mrissa et al. [8] as a resource-oriented extension of CBS with Web capabilities. The one-to-one relation of an Avatar and its cyber-physical object are intended to lift CBS to a homogeneous, Web-driven integration layer. Their interfaces appear through REST APIs and make use of semantic descriptions, both of their capabilities and the offered and consumed data. The main focus of the Avatar approach is the interoperability challenge of IIoT scenarios, with less detailed specifications on data protection and security. Virtual Representations [25] go into a similar direction, with read/write capabilities also on functions and services.

Several attempts have been made to cover Web technologies in industrial settings. One of them is the W3C Web of Things Working Group proposing an Architecture and Vocabulary [62] for the intersection of the Semantic Web and IIoT. One of them, the W3C recommendation for a Web of Things, focuses on

[2] https://www.rapidapi.com/
[3] https://swagger.io/docs/specification/about/
[4] https://apiblueprint.org
[5] https://raml.org
[6] https://www.w3.org/TR/wsdl20/#Interface
[7] https://www.w3.org/Submission/OWL-S/#4
[8] http://www.biotope-project.eu
[9] https://www.fiware.org
[10] https://fiware-datamodels.readthedocs.io/en/latest/ngsi-ld_howto/

the promotion of Web technologies with elaborated semantic meaning. The Asset is closely integrated with conventions and specifications used in the Web. In particular, the discovery of capabilities and resources at runtime through previously uninformed clients is at the heart of the recommendation. That enables a lose coupling of server and clients, thereby allowing a real decentralized and scalable network. Through regarding an IIoT network as an open Web environment, the proven security mechanisms from Web applications (encryption, authorization, identities) are introduced into operational communication. The WoT recommendation also proposes the according Thing Description ontology (TD) as the core vocabulary to denote the WoT attributes and properties.

First mappings from the Web of Things community towards IIoT protocols have already been created [11].

The CoAP and MQTT bindings in form of RDF vocabularies are however still on the level of examples, and do not represent the full characteristics of these protocols. For instance, the interaction and discovery mechanisms are still missing.

Other vocabularies for IIoT applications and devices are for instance the IoT-Lite ontology [138] or FIESTA-IoT [139]. Their focus on formal descriptions and automated inferencing support integrates so-called top-level ontologies and thereby eases the relation cross domains. Well-known and commonly used vocabularies such as RDF, DCTERMS or QUDT are aligned, mapped and extended with IIoT-specific concepts. The Lov4IoT catalog [140] collects a broad range of such ontologies and formally described IIoT vocabularies.

However, the inherent overhead of the linked-data approaches due to their strict usage of HTTP messages has led to mappings into less expressive but resource saving protocols. An LDP to CoAP translation [141] outlines patterns and solutions to bridge the gap between restricted devices and rich interactions on the application layers. The additional introduction of Edge or Fog layers through dedicated gateways [142, 143] can further lift the resource-restricted data sources to the higher-level communication layers.

As shown by the presented works, the semantic technologies still have a huge potential to solve the identified issues of Digital Twins when combined to a comprehensive approach. Current proposals fall short, for instance, to adapt to the specific requirements of the IIoT or only regard a certain aspect of the domain. The following section in particular examines the current state and proposals how the expressiveness of semantic technologies can be implemented in the IIoT.

3.2 Semantic Web and IIoT

A broad overview of the usage of semantic technologies and in particular ontologies in the IIoT is provided by Kumar et al. [144]. The authors collect and in depth analyze the state of semantic models in the manufacturing domain and extract a number of best practices. The presentation of shared knowledge, removing of ambiguities, and the reasoning capabilities are a few examples. They name 29 domain ontologies that shall be merged through four high-level approaches towards a consistent environment. Kumar et al. unfortunately do not provide the implementation of this idea but stop at the conceptual strategy.

The review on IIoT trends form a semantic perspective is also conducted by Whitmore, Agarwal, and Xu identify several unresolved requirements [145]. In particular, they state that improvements need to appear for the identification, sensing, networking, and processing capabilities of Assets. Whitmore, Agarwal, and Xu state that the target of the Semantic Web, to empower machines to recognize and process digital content, is very similar to these challenges. The authors collect a significant number of IIoT publications and group them into several categories according to their technology focus or proposed

[11] https://github.com/w3c/wot-binding-templates/blob/master/ontology/ (accessed on 10 June 2020)

architecture pattern. Based on their extensive review, they come to the conclusion that the established functions of consuming the Web, the browsers and search engines, need to be introduced into IIoT applications as well. Still, Whitmore, Agarwal, and Xu do not further develop their proposition how such tools should look like.

Nevertheless, as Dohr et al. [146] and Jara et al. [147], these tools alone will not be sufficient but an ubiquitous computing approach and context-awareness are key requirements for any value-adding intelligence in the domain, which therefore must be one of the development goals of the IIoT. Integration steps through semantic models still require further extensions, for instance as the current consensus is limited to the used data formats (XML and JSON-LD) and that ontologies in general are needed to establish the metadata and meaning necessary to support interoperability ([148–151]). The more advanced integration steps that automatically benefit from the formalized content however still need more alignment steps.

One prominent application scenario of this approach, virtual integration layers through semantic data modeling on top of enterprise data environments, have received significant attention in the recent time [152] [153] [154]. The information flow is usually three-fold: (1) the original data from the various databases are consumed using (2) predefined mappings and then thereby (3) lifted into the virtual knowledge graph layer. The primary query is then either translated into different query languages to bring the queries to the target systems, or the data is physically lifted into the target scheme and stored redundantly [155]. A different approach is presented by Lehmann, Sejdiu and Jabeen with the SANSA framework [156]. SANSA leverages state-of-the-art technologies to maintain the distributed graph, for instance with Apache Spark. On top, sophisticated data analytic and AI modules are applied. The intended loose-coupling, however, is thereby hardly accomplished, as the mappings and the configuration of the middle integration layer are strictly necessary. Changes in the underlying data scheme automatically require according - manual - updates of the mapping rules. Consequently, the domain or technology expert maintaining these rules becomes the new bottleneck.

The internet and in particular the World Wide Web offers already a well understood and widely accepted infrastructure to exchange data. Well-established Web technologies like URIs, HTTP and hyperlinks have proven to allow easy and reliable communication mechanisms in a decentralized manner. Cloud services and on demand solutions offer fast and flexible deployment of applications, a strict requirement for a smart factory.

The Semantic Web further adds meaning to data objects and further reduces the integration effort. Pfisterer et al. promote the term *Semantic Web of Things* (SWoT) to express the combination of machine-readable data with Semantic Web capabilities and the global Linked Open Data Cloud [157]. The integration of Assets is intended on the fly and without any registry, simply by client look-ups and well-known paths (cf. RFC 5785), and integrating representations of the Assets current state. Even though the focus of the SWoT project is the description and discovery of Assets, an important building block for the challenges identified in this thesis, the missing support through a commonly-accepted standardization authority makes it hard to implement the ideas and recommendations directly in applications.

As a consequence, a further institutionalization through the so called Web of Things has been triggered. It does neither specify the interaction patterns of the regraded Assets nor does it model the intended relationship with the physical world. The Web of Things (WoT) is an initiative of the W3C to model and outline common aspects of physical assets and represent them in the Web. An elaborate vocabulary and an interaction model demonstrate how independent entities can be described, operated, and orchestrated using the current practices and conventions of Web connections. The regarded requirements are technology-oriented and provide system architects and developers with implementable guidelines based on the currently used technologies.

To further enhance interoperability of services, a shared understanding on how to exchange data and

commands through the web and a clear semantic meaning of those is still not sufficient. In order to realize the vision of flexible integration of formerly unconnected services, a certain level of smartness needs to be introduced. In this context, services or applications are 'smart' if they are capable to adapt their functionality depending on observed or communicated situations. They therefore include a certain degree of decision logic, in the form of e.g. decision trees, rule-based inferring, models based on machine learning or others. This allows a Smart Service to deliver different results depending on its perceived situation. One very simple example is a weather service which regards the location of the service requester or a knowledge management system filtering available documents dependent on predictions of his demands.

In particular in the context of the Web, Smart Components [158] path the way to flexible and context-dependent integration. A Smart Component not only exchanges semantically defined data (through its regular Web API) but also can adjust their program code through a so called meta API, using exactly the same interaction mechanisms enabled by the Linked Data-Fu engine. In that sense, Smart Components have the capability to provide data and also consume other Web resources by sending requests. They form 'Prosumers' (producer and consumer of data) and thereby overcome the strict separation of servers (as resource providers) and clients (as consumers).

Smart Web Services[92] utilize the formerly outlined methods and techniques in order to cope with new situations. By consuming and producing RDF data they can exchange information without relying on a strict data scheme, similar to Smart Components. Smart Web Services in that sense combine the technical characteristics of Smart Components and the usage of the Semantic Web Stack with independent decision logic. Still, distributed applications of Smart Web Services have to be composed and orchestrated by central controller in the form of e.g. the StreamPipes coordinator.

Smart Web Services [92] approach this challenge coming from a Cloud-oriented perspective. The ability to perceive, model, and react dynamically on different context environments allows this model to behave autonomously and reevaluate their own AI-powered decision model. While the flexibility of Smart Web Services may be a key to today's and tomorrow's heterogeneity challenges, the still missing understanding on consequences of such automated decision processes, in particular regarding safety and liability, still prevents their actual implementation.

The concept of Smart Web Services however is also a good example how the different communities merge their ideas, even though the used terminology might be still different. For instance, the definition for cyber-physical systems from Sahlab, Jazdi and Weyrich is very similar to the core aspects of Smart Web Services from Maleshkova et al. [92], and both concepts are corresponding to the understanding of Digital Twins.

3.3 Unpredictable Requirements

The focus change related to the Digital Twin concept during the last decade vividly illustrates that business-critical developments are hardly predictable for the designers of IIoT systems and the engineers of devices. As stated, the high investment costs and long deployment times however require the adaption to new trends and developments. As Sahlab, Jazdi and Weyrich argue, a comprehensive understanding of *context* is required [159]. Analyzing the shortcomings of current CPS applications, they come to the conclusion that only formalizations using ontologys and thereby created Knowledge Graphs can cover this.

According to Sahlab, Jazdi and Weyrich, the formal reasoning capabilities of ontologies need to be extended with uncertainty inferencing, for instance using Fuzzy Logic or probabilistic models. It is an open question, whether the newly proposed extensions to the RDF data model, like for instance

RDF* [160], fulfill the thereby created requirements. Property Graphs on the other hand may provide the necessary flexibility but still lack the maturity and standardization level of traditional RDF data models. Combining these technologies however does still not solve the challenge *what context really is* or maybe even more important, *what is excluded*. Novotny and Bauer analyze the literature for context-aware systems, extracting that "context may be anything that could be used to describe the situation of an entity." ([161] page 2) While this high-level definition might very well be the common denominator of the current research discussion, it is certainly not contributing to actual implementations. For this reason, Novotny and Bauer examine that the categories *time*, *location*, *device*-related aspects, *communication*, *network*, and *infrastructure* are - in this sequence - the most often proposed categories for context information.

Li et al. suggest an approach for decentralized IIoT platforms by using a metadata registry to dynamically discover context information, called a *distribution broker* [162]. They follow the assumption that the metadata about provided information is less likely to change than the data itself (for instance: "The PIP provides a stream of observations" vs. the actual data stream). They propose the usage of the Semantic Web stack (mainly JSON-LD and SPARQL) to overcome the integration hurdles and to integrate previously unknown context information. The outlined patterns of Li et al. still need to be transferred into a production-ready standardization framework, like for instance the IDS or the Plattform Industrie 4.0, to reach the necessary maturity level to allow the implementation on a broad scale.

3.4 Digital Twins in Brownfield Scenario

Caesar et al. describe a Digital Twin, more precise a *digital shadow* model, for physical attributes of existing Assets [163]. For instance, they define the location, the occupied space and the material characteristics. Relying on VDI 3682, they examine how spatial and temporal descriptions allow the planning of processes and production steps. The proposed data model however lacks the necessary interoperability features and adoption of a formally described standard, and does not explain the interaction and communication semantics.

All relevant players in a connected, data-driven manufacturing are continuously exchanging information on the state of regarded products, production units and materials over the whole supply chain and product lifecycle [164]. IIoT devices and cyber-physical systems form global networks and provide more flexibility to the production processes. International organizations like the Industrial Internet Consortium or the Platform Industrie 4.0 drive the development of standards to enable the seamless integration of machines, software applications and products. The target is to reach a secure but at the same time flexible integration of any kind of production related unit based on the internet. Main advantages are the reduction of applied protocols, formats and interaction patterns to simplify the digital information exchange and to support a plug-and-play like deployment. This will not only allow faster adjustments to existing production processes but also to apply analysis driven by existing information and not hampered by previously designed interfaces, data silos or interaction patterns. Yet, the current specifications are still high level proposals how a connected production shall be implemented. Commonly agreed technology stacks and transaction formats are still missing, therefore a seamless connection is yet not possible.

Smart Manufacturing [165] comprises efforts to establish a reference architecture with nodes representing physical components in the manufacturing facility to ease the integration and create a generic platform. The promoted modularized approaches model virtual resources similar to their physical counterpart in order to enable rapid deployments and portability. But even though they outline a reference architecture, the targeted integration aspect is still unclear and directly implementable specifications are missing. Hedengren and Eaton [166] further discuss time based mathematical simulation and optimization on highly dynamic measurements. They discuss various types of update frequencies and how to derive

predictions. All of the discussed models require decent preprocessed and, most of all, accordingly synchronized input data. Especially in brownfield scenarios, such a state is a major accomplishment and not a prerequisite.

Data Lakes, as e.g. discussed in [167] or [168], are one concept to make data from heterogeneous sources and in different formats accessible. Established technologies like Apache Hadoop provide solutions for NoSQL clusters and enable queries also on dynamic data without a fixed schema. The Data Lake concept is only partly scalable in terms of the underlying cluster but also forms a single point of failure and potentially another data silo with tight coupling which will hamper the data usage in future cases. The not required data format enables the simple data storage but makes an effective data integration without previous knowledge on each data object a challenging task.

3.5 Heterogeneous Communication in IIoT Networks

Countless efforts have been made to control the variety of communication patterns, protocols, and data exchange procedures. To structure the landscape of approaches, one might can order them whether they regard the topic from a business or process-oriented view (top-down) or from the protocol and interaction level (bottom-up).

Hohpe and Woolf for instance present a set of 65 Enterprise Integration Patterns, each describing a building block in a system architecture [16]. They define abstract functionalities, like *Message Queues* or *Remote Procedure Calls*, and explaining their necessary behaviour and characteristics. This includes synchronous or asynchronous behaviour or typical side effects. The Enterprise Integration Patterns however do not specify how the interactions in terms of protocol calls need to be implemented, nor how the involved Resources are described.

The similar shortcoming appears in the *Digital Twin Structure Model* from Lechler et al. [169]. Their conceptional approach to combine a physical with cyber-focused layers extends the basic RAMI4.0 with a special scope on the lifecycle dimension. The noteworthy aspect of the Digital Twin Structure Model however is the consideration of different integration timings. As their focus is the bi-directional interaction between the Asset and the virtual representation, *real-time* and *near real-time* patterns are described. These obviously crucial aspects to properly interpret provided data correctly, many other Digital Twin models tend to ignore it. Still, even Lechler et al. fall short in their description of the consequences for the applied communication patterns, especially as they summarize all further possibilities in just one additional category.

A more bottom-up approach is illustrated by Pautasso, Ivanchikj, and Schreier [170]. Their patterns combine HTTP requests, more precisely REST calls, to so called *conversations*. Each conversation follows a defined workflow to reach a certain effect. Fig. 3.2 for instance illustrates the steps to create a new remote resource and add its content. The conversation extends the basic REST specifications by further distributing responsibilities between the involved parties. For the pattern in Fig. 3.2, the server may delete created but empty resources at some point in time. If both client and server know that they follow this conversation pattern, the client can expect this behaviour and act accordingly. If it however solely acts on the basic REST patterns, the client would be surprised by the disappearance, potentially leading to an unsafe application state.

In particular in the context of the Web, Smart Components [158] path the way to flexible and context-dependent integration. A Smart Component not only exchanges semantically defined data (through its regular Web API) but also can adjust their program code through a so called meta API, using exactly the same interaction mechanisms enabled by the Linked Data-Fu engine [171]. Käfer further extends this approach to complex, decentralized workflows [172]. Similar to BPMN models, the state transitions are

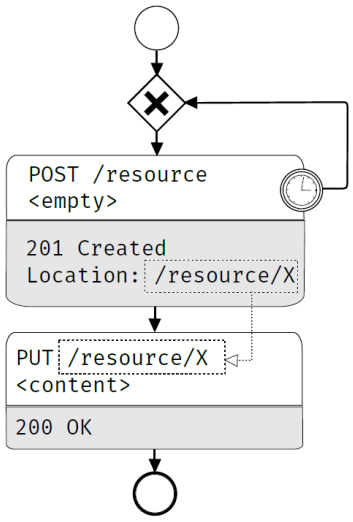


Figure 3.2: Graphical notation of a conversation between a client and a server to create exactly one resource (from Pautasso, Ivanchikj, and Schreier [170])

enabled through basic Web requests and can easily be federated among applications. Thereby, an agent is able to execute its operations following a state machine model while interacting with potentially complex services.

Xu and Li give an extensive overview on IIoT trends and driving factors [173]. Even though the authors use the terminology of Cyber-physical Systems, their focus is on the IIoT developments as discussed in this thesis. Their review explains the history of the disruptions faced with each industrial revolution and how those affect the current disruptive developments of the IIoT. They analyze that the current IT landscape is not ready for the desired intra-organizational and inter-organizational integration. In particular, Xu and Li argue that the necessary *scalability* is not addressed by current IIoT applications, as the expected massive amount of real-time data can not be processed effectively. This is caused by the lack of the one standardized platform of choice, which hinders the communication between the various systems in the shop floor and the office floor (CH3). They therefore call for an “unified information infrastructure”(page 2953 [173]), which is based on the same principles as the outside information networks.

In particular, the authors formulate seven research gaps. At a basis, the *integration* (1) of IIoT systems (cf. CH3), the communication between the heterogeneous devices and their dynamic behaviour need to be solved. On top of that, the *verification* (2) of IIoT systems is not yet sufficient. Xu and Li further state that the *Blockchain* (3) technology is crucial for the manufacturing domain. Unfortunately, the explanation on potential IIoT use cases for Blockchain technologies is missing, as well as the linkage of the technology with the technical challenges of IIoT networks.

The next gap they raise is the integration of *artificial intelligence* (4) into smart and autonomous devices. According to the authors, this has also the potential to increase the *resilience* (5) of the overall network. Resilience in that regard is the capability of a system - or a network composed of such systems - to cope with disruptions. Xu and Li conclude with mentioning potential implications on future *enterprise systems* (6) and on the effects of the IIoT on the *society* (7) in general.

3.6 Digital Workflows in Dynamic Environments

Service-oriented Architectures (SOA) promote the encapsulation of functionalities in independent units. The design-time decisions on how to structure the different services heavily affects the later composition and orchestration challenges when the different services are combined to value-adding applications. The Arrowhead Framework [174] describes a possible approach how to establish the necessary infrastructure at runtime, composed of a *service registry*, an *orchestrator*, and an *authorization* component. Using these basic building blocks and relying on the specifications of IEC 61499, Romanato et al. show how a safe and unsafe interactions can be implemented on the fly [175]. While this work targets the operational semantics of the interactions in conjunction with the data format, the *meaning* of the regarded data is not further specified.

Edge computing [176], edge analytics or fog computing [177] proposes to conduct substantial part of computing tasks close to the physical Asset at the *edge* of the network. The main advantages are less network traffic and response times as data processing takes place close to the data sources. Even though simulating resources close to their physical location can be seen as some kind of edge computing the paradigm itself lacks standardized methods on how to implement and how to interact with the resources of interest.

Grieves and Vickers examine the behaviour of complex systems and how Digital Twins can improve their organization [178]. They distinguish the predicted and the unpredicted behavior, and in particular the unpredicted undesirable behavior as a subcategory of the latter. Events of this category contain the factor of surprise that leads to potentially dramatic results, and therefore should be avoided. Grieves and Vickers argue that the application of Digital Twins throughout the lifecycle of a complex system improves the opportunities to recognize unpredicted undesirable behavior before it actually appears. Even though the authors mainly explain their approach with a physical simulation view in mind, their insights can also transferred to digital systems and IIoT networks. Still, the approach of simulating the behavior of the control flow of an Asset in a system is not raised at all.

In the context of the IIoT, Gundall, Glas and Schotten examined an architecture of virtualization components in an IIoT network [179]. They distinguish three main phases - bootstrapping, regular operations, and replacement - and outline possible strategies. The core enabler of the architecture is a central orchestration component to manage the virtual services and react to failures. The proposal does not regard any command or operation semantics, neither does it state any requirements on the functionality or behaviour of the virtual services. Therefore, significant further enhancements are necessary before the proposed solution can be applied in productive IIoT settings.

In order to automatically combine components, existing approaches [180][181] mostly use centralistic optimization during the design phase. Contrary, Web components are not static elements but can and do change over time. In general, they follow a lifecycle as shown by Wittern and Fischer [182]. Mayer et al. [183] extend RESTdesc to cope with a dynamic environment by introducing states. Similarly, Alaya et al. suggest oneM2M, a IIoT approach to gain machine-to-machine (M2M) interoperability with a semantic reasoner [184]. But still, a central organizer with knowledge about all available components of the network is necessary.

One way to enable a more flexible way to determine component compositions are policies. The non-functional characteristics of available components are regarded with semantic reasoners [185] in order to match and rank (Palmonari et al. [186]) them against predefined requirements. La Torre et al [187] propose the dynamic context of the consuming client as a selection criteria for components. Context here is regarded as social media information of e.g. Facebook but also physical data like GPS coordinates. Although only human clients have been regarded, it should be possible to transfer the approach to automated components, having context like the location of the hosted server or the company

running it.

The complete potential of Digital Twins however is the seamless connection throughout the islands and silos, and cross networks, domains and companies. OPC UA networks usually require gateways and proxies, thereby forming bottlenecks and enforcing deep understanding of the protocol's abilities and restrictions. In addition, as both Perzylo et al. [188] and Schiekofer et al. [189] have mentioned, the lack of formal concept definitions hampers the further integration with higher-level applications.

Earlier approaches to transfer established web technologies to the industrial setting combine service-oriented architectures [190, 191]. The method-driven semantic allows the integration through rich service functions using SOAP and the standardized WS-* stack. However, the complex requirements and side effects led to redesign of the CBS, less viewed as a collection of method calls and more as a virtual resource with distinct state transitions.

The WS-* stack of standards was a first approach to enable interoperable compositions, as already described in Section 2.3.1. Oberle et al. [192] extended WSDL with modules to describe business-related, non-functional aspects like payments, SLAs, as well as usage rights, restrictions and a role management [192]. The resulting Unified Service Description Language (USDL) has been published after the shutdown of the central UDDI service[12], which may indicate that the vision of a WS-* infrastructure was not reached. Pedrinaci, Cardoso and Leidig [132] further lift the USDL concepts to the Linked Data principles and integrate them with well-known vocabularies to support their integration in Linked Data applications. Still, neither Pedrinaci, Cardoso and Leidig nor Oberle et al. intent to integrate their contributions into running Web Service infrastructures. Therefore, the resulting languages rather serve as references and inspirations for later approaches than directly deployable solutions.

Labbaci et al.[193] examine the changes to public Web Services and their description in order to recommend composition and substitution candidates [193]. They use DBpedia Spotlight to annotate natural language service descriptions, and use these annotations to calculate the semantic similarity between the services themselves. Changes in the descriptions, in combination with annotated user comments, are then used to react to updates of the services. Unfortunately, the described approach is hardly reproducible and the claimed results (more than 50% of recommended relations shall be correct) are therefore only partly useful to the community. Still, this sounds promising to solve the Web Service composition challenge in open environments. Nevertheless, successful recommendations in the area of 50% are still far below the required quality rates for productive IIoT environments with safety and business critical implications in case of failures.

3.7 Data Sovereignty

Control over digital information or *data* becomes more and more crucial for digital business models [17]. While physical goods know the concept of ownership, and therefore have a clear definition of possession, buying and selling, digital goods are inherently different. While physical Assets have exactly one representation, data can be represented in an infinite amount of forms and varieties and shared without limits.

The General Data Protection Regulation (General Data Protection Regulation) of the European Union has specified nine key rights for person-related data: the right to have transparent overview about the own data (*transparency*), about the *processing* intentions, *access* it, correct if necessary (*rectification*), or even delete it (*erasure*), to move the data (*portability*), *object* usages, and to prevent consequences of *automated decision* making [194]. The Data Privacy Vocabulary[13] (DPV) provides terms to annotate and

[12] https://www.computerwoche.de/a/microsoft-ibm-sap-discontinue-uddi-registry-effort,570059, accessed 26.11.2020

[13] https://www.w3.org/ns/dpv

categorise instances of legally compliant personal data handling according to the GDPR, including the notions *data categories*, *data controllers*, *purposes of processing data*, etc. The focus is on the description and annotation of privacy constraints. The Open Digital Rights Language (ODRL) [108] is equally able to express usage concepts through its RDF vocabulary. The specification allows expressive statements but the implications of many supported constructs are not yet sufficiently understood. The challenge is not the description of the policies but their interpretation in an enforcing system. The proposed terms lack a sufficient definition of their context, side-effects and implicit dependencies.

Several developments into this direction can observed also on an international level. The blocking for third-party cookies to prevent further tracking of Web users announced by Apple[14], Mozilla[15], and Google[16] expresses the growing awareness of digital privacy. Decentralized data aggregation my be one building block to find valid trade-offs between data usability and privacy protection. The FLoC API is one example to promote data analysis in systems which are controlled by the creator of the data, while the business partners only receive the information they really need to execute their business process [195]. Similar to GDPR however, these developments solely target person related data. For instance, Flouris et al. present a marketplace approach by enabling semantic tags of privacy features and concerns [196].

An even further distinction is necessary when regarding data related to states and governance agencies. In the European Union, the privacy regulations contained in the GDPR do not apply for governmental processes. The US Patriot Act and CLOUD Act[17] as well as China's Cybersecurity Law[18] express developments to give state agencies unlimited access to data stored or passing through their countries authority boundaries [197, 198].

Neither GDPR nor the usage of standard licenses however solve the problem of data ownership in business-to-business scenarios. As Duisberg and Camilli point out, "there is no unique proprietary concept of ownership and data as such" [18]. This underlines the need for expressive and automatically understandable usage languages, which can encode the intended usage from the data provider and outline them to potential consumers.

The OASIS standard XACML proposes a component model to access such context information, which is not necessarily known at the design time of the affected devices. The so-called Policy Information Points (Policy Information Point) encapsulate the access to the environment, and thereby act as interfaces for context attributes [106]. The core challenge however is obviously not how to make any context attribute available, but to understand (1) which previously unknown information are *available* at a certain point in time, (2) how they *affect* the target system, (3) which information is actually *required*, and (4) how to *discover* those in a dynamic environment.

XACML applies very well in Attribute-based Access Control (ABAC) scenarios. The *object* refers to the attribute of the resource under consideration, which a *subject* wants to access. ABAC rules can be stated in plain text and interpreted by the hosting system. Yu et al. [199] also show an approach to encode the access permission in cryptographic access key structures, hiding the content also from the hosting service. This is especially relevant if a third party is used as a cloud provider. Nevertheless, while this supports the downstream usage of ABAC rules, this approach is bound to the syntax of the data instead of its meaning.

Stronger focus on execution environments are the case by access control approaches and respective frameworks. The terminology of authorisations, obligations, and conditions introduced by the influential

[14] https://webkit.org/blog/10218/full-third-party-cookie-blocking-and-more/, accessed 15.11.2020

[15] https://blog.mozilla.org/blog/2019/09/03/todays-firefox-blocks-third-party-tracking-cookies-and-cryptomining-by-default/, accessed 15.11.2020

[16] https://blog.chromium.org/2020/01/building-more-private-web-path-towards.html, accessed 15.11.2020

[17] full name: Clarifying Lawful Overseas Use of Data Act, 2018

[18] Original title: 中华人民共和国网络安全法 (The Cybersecurity Law of the People's Republic of China), 2017

UCON_{ABC} [105] usage control model has been adopted by many later models. Together with RFC 2904 and the introduction of the different *policy points*, these two works form the theoretical foundation of usage control. However, neither proposes a vocabulary to specify distinct permissions or prohibitions. This task is, to some, degree covered by XACML [104, 200]. Still, XACML only focuses on *access* control, not on the more holistic usage control.

The terminology of permissions, obligations, and conditions introduced by the UCON_{ABC} [105] usage control model has been adopted by nearly all later models. In combination with RFC 2904 [201] and the introduction of the different *policy points*, the theoretical foundation of usage control systems and architectures are defined. However, none of them proposes a vocabulary to specify distinct permissions or prohibitions, their focus is rather on classifications and conceptualization. This issue is, to some degree, covered by XACML [104]. Still, XACML only focuses on *access* control, not on the more holistic usage control.

Further approaches to extend the descriptive expressiveness with execution logic are so-called *Smart Contracts*. Small Blockchain based programs are stored and executed in decentralized manners and therefore protected from manipulations. Loukil et al. present an ontology [202] and architecture [203] to describe privacy rules for IIoT data in the Etherium Blockchain. These approaches however have two major drawbacks, which can not be solved by the integration of the Blockchain technology. First, the public storage of any kind of business-critical metadata - even if encrypted and the business partners are hidden - provides sensitive information to competitors. The pure existence of a logged interaction is a potential security breach. Second, the complete trust architecture depends on the durability of the underlying Blockchain - and it's often unclear incentive system - which make them unacceptable for long-running IIoT applications.

Nagendra et al. use a custom, graph-based format to lift and analyze security policies for IIoT networks [204]. The VISCAR system shall be able to analyze and map vendor independent policies for internal conflicts but also for contradicting or missing instructions when composed with other policies and regarding the deployed context. VISCAR implements parts of Policy Administration Points (PAP) and Policy Decision Points (PDP). The described accuracy and performance to discover conflicts however requires high manual input efforts as it is not related to any standardized parameter format or policy language. The approach requires a closed world, while missing to define the attributes and representations of the implicit world model, for instance time, location, security features. It is therefore hard to cf. whether or not the claimed results can be applied in a setting different to the selected environment.

Another example is the IND2UCE system [107]. It combines the previously mentioned Policy Points with a SDK to integrate it into existing applications. IND2UCE further integrates into the common message-routing systems, for instance Apache Camel, and provides enforcement capabilities based on runtime information. The system works independently of the other components and Assets of its network and therefore can observe and control them without interference with the regular functionality. IND2UCE however requires a proprietary configuration language, which is bound to its use cases, and therefore has not yet the capabilities to express, and support, the technical means for a general usage control scenario.

The IDS regards the secure and trustworthy data exchange on a data-centric, domain-agnostic level. The Reference Architecture Model [35] consists of layers to establish interoperability and crosscutting perspectives for reaching its main target, to ensure end-to-end data sovereignty. The syntactic interoperability is accomplished by the IDS Connectors as a core gateway to the IDS, with standardised interfaces and exchange protocols. An IDS Connector is a hosting system for any kind of data, for instance also for AAS, and ensuring the interests of Data Owners as expressed in formal Contracts. The IDS specification is thereby defining a data protecting infrastructure and requirements for the interacting systems, while the AAS further outlines the endpoints and data model of the specific Digital Twins living in such an infrastructure. The Policy Language itself has already been presented for general data resources [21] but

went recently through major rework in order to sharpen the Usage Control clauses.

3.8 Distribute Architectures for Digital Twins in IIoT

Many standardization counsels and research groups propose reference frameworks for the IIoT, Digital Twins or both together. It is however not the goal of this thesis to add yet another architecture but to analyze the existing ones, put them into context and provide access to the necessary technical standards to implement them. Therefore, the presented research works of this section does not discuss the architecture models themselves but how other research groups contributed to their representation and which tools and approaches have been developed to tackle the thereby appearing challenges.

The overview on the related work is split into three parts. Section 3.8.1 explains the state of the art for the graphical representation of huge RDF datasets, a critical show stopper for the user-centric provisioning of graph data. Section 3.8.2 points out how previous works have contributed to make IIoT standards available for the implementers in effective and efficient ways. Section 3.8.3 broadens the view and also contains works for reference frameworks and generic architectures. Together with Section 3.9, a comprehensive overview of proposed reference frameworks and how to relate them with each other is provided.

3.8.1 Visualization of interlinked Entities in the Semantic Web

The volume and variety of already medium sized knowledge graphs makes it hardly possible for the non-expert users to efficiently find relevant information. The problem of presenting and interacting with such big structures has drawn a lot of attention, for instance, in the biomedical domain where information about genes, drugs, and drug interactions need to be displayed effectively. As a result, several studies examine best suited templates for the different kinds of information representation.

Classical knowledge engineering is usually not targeted to direct visual presentations. In a typical setting, the ontologies or knowledge graphs are maintained in back-end systems and accessed through tailored applications and clients. Nevertheless, several tools have been created with visualizers and visual editing capabilities. Protégé [205] is very likely the most prevalent editor for ontologies, featuring several plugins for ontology visualization and knowledge inferencing, among others. VoCol [206] and WebVOWL [207] are two visual approaches intended to simplify the creation, maintenance and analysis of RDF-encoded ontologies. However, most tools focus on the support of schema-level entities like classes and relations (TBox). Proper illustrations and filtering mechanisms for data instances (ABox) are hardly provided. McBrien and Poulovassilis for instance propose selection criteria for the appropriate usage of different visualization forms [208]. Still, the suggestion can only support the provider of the visualization service as each underlying SPARQL query needs to be adjusted to the domain knowledge graph.

A different approach for accessing RDF data is followed by collaborative tools like OntoWiki [209] and Semantic MediaWiki [210]. The graph structure is partly hidden behind forms and templates, allowing also non-experts to work together on the graph models. However, the presentation of the RDF graph is not possible out of the box and requires the application of further tools.

Searching for technical information in the internet is mainly executed through the established search engines. Even though more and more queries are answered by directly returning related information, for instance by displaying Wikipedia abstracts, in general only collections of web sites are provided. The user then has to manually discover the sources. Especially for technical information needs, this approach is highly inefficient as it is very time-consuming and requires considerable prior knowledge.

Lafia, Turner and Kuhn [211] show how semantic annotations and mappings on open data improves the discovery process. On the other hand, Xiong, Power and Callan [212] discuss embedding techniques on open knowledge graphs to measure the similarity between entities in a high-dimensional vector space.

Several works have been accomplished to address the challenge of structuring the landscapes of standards focused on the industrial domain. For instance Lu et al. [213] describe a landscape of Smart Manufacturing Systems. Similarly, Andreev et al. [214] provide several visual comparisons of radio connectvity standards and technologies. However, none of these surveys is published in an accessible data set as the contributions and insights are only represented as written text.

Grangel et al. [215] proposed an ontology for structuring the necessary information and providing it as open information. The contribution presented in Chapter 6.3 extends their work both on the level of schema extensions but also by significantly added content information in terms of more entities and new data dimensions to create a significantly improved picture of the IIoT landscape.

3.8.2 Technical Standards for the IIoT

Overview works of technical standards comparable to the one proposed in Chapter 6.1 usually appear in one of two forms. On the one hand, experts with an academic background collect relevant publications and comprise them in literature reviews. On the other hand, industry experts and consortia publish their views on the domain through reference frameworks and white papers. Both approaches require extensive efforts for the interested reader to discover, filter, and understand the provided content. Furthermore, the provided knowledge is only valid for a limited time around the publication date. Updates in terms of extensions and adjustments to recent developments are not common practice. Especially in the research community, updating survey papers – to reflect developments since the original publication – usually does not happen.

Still, a significant number of reviews on the IIoT emerges each year. For instance, Xu *et al.* present a comprehensive overview on the major drivers and also standardization activities [173], mentioning the key developments and concerns. Martinez *et al.* outline the relations between cyber-physical systems and the IIoT [216]. However, as typical for academic reviews, references to technical standards are omitted. This shortcoming does not reflect the actual relevance of standards and norms for the engineering and implementation processes.

Searching for technical information in the internet is mainly executed through the established search engines. Even though more and more search queries can be answered directly returning related information, for instance by displaying Wikipedia abstracts, in general only ranked lists of web sites are provided. The user then has to manually discover and examine the sources. Especially for technical information needs, this approach is highly inefficient as it is time-consuming and requires considerable prior knowledge of the target domain. Lafia, Turner and Kuhn [211] show how semantic annotations and mappings on open data improves the discovery process. A different approach relying on AI techniques is proposed by Xiong, Power and Callan [212] who discuss embedding techniques on open knowledge graphs to measure the similarity between entities in a high-dimensional vector space. Nevertheless, the search for targeted, domain-specific information as regarded in this work, presents a significant burden.

Several works address the challenge of structuring industrial standards as a graphical representation, or so-called landscapes, of the domain. For instance, Lu *et al.* [213] describe a landscape of Smart Manufacturing Systems. Similarly, Andreev *et al.* [214] provide several visual comparisons of radio connectivity standards and technologies. However, none of these surveys are published in an accessible data set as the contributions and insights are only represented in written text and cannot be processed by further tools and applications.

3.8.3 IIoT Landscape

The targeted challenge – to support newcomers, domain experts and any other stakeholder to establish and curate a proper overview on the published standards, frameworks, and concerns is one of the key obstacles hindering the wider adoption and successful fulfilment of the potential of the IIoT vision. Grangel-González *et al.* [126] introduced a first ontology for IIoT components, in particular for the Asset Administration Shell model. Extending this work, the basic structure and scheme of the graph has been developed, together with a first approach to structure the IIoT-related standards and norms in terms of a unified landscape [215]. These publications introduced the initial definitions of the standard and standardization framework concepts.

The presented knowledge graph on IIoT standards on frameworks of Chapter 6 is the first structured approach applying machine-readable data interlinking the textual, normative and informative resources containing the knowledge of IIoT standardization. In comparison to the earlier evolution steps, the hereby presented graph has been significantly extended in terms of contained entities, from less than 80 as presented by Grangel-González *et al.* to more than 300 described instances. Furthermore, a vast number of affecting requirements has been introduced and implemented in order to allow use case-driven filtering and context-dependent discovery of relevant entities.

The graph constitutes a machine-readable resource of interrelated standards, reference frameworks, and concerns. It thereby comprises an extendable representation of the whole topic. In contrast to the more common format of literature reviews, the presented knowledge graph is a semantically enriched and openly accessible resource, which represents the state of the domain at its publication date and beyond.

As the IIoT knowledge graph follows the principles for provisioning Linked Data, it also may serve as a way to spread semantic technologies to other communities. The recommendations and guidelines as for instance formulated by Noy *et al.* have been followed to ensure the quality of the graph [217]. The target groups are usually not too familiar with the Semantic Web in general and RDF-based knowledge graphs in particular, therefore the adaption can further support the dissemination of the mature practices of the Semantic Web and Linked Open Data.

3.9 Reference Architectures for IIoT

Several works aim to create a framework for software architecture descriptions. ISO/IEC 42010 [218] proposes Architecture Descriptions structured by a list of so-called *concerns* being addressed by several *architecture views*. An architecture view is a projection and therefore a simplification, of the abstract architecture in order to describe specific topics. For instance many IIoT reference architectures cover both interoperability and security related aspects. Though there are many inter-dependencies, describing both concerns in one view decreases readability and significantly increases the complexity. In contrast to the general agreement on the classification into views, only a minority explicitly states the regarded concerns and the followed conventions throughout the presented views. Therefore, it must be stated that the proposed structure of ISO 42010 is not followed—a development, which significantly hampers the comprehension of core aspects and limits effective comparisons. Nord et al. [219] further strengthen this fact. They also demand a first-class treatment of stakeholders and their concerns.

Boyes et al. provide taxonomies for the industry sector and its various domains and subdomains [70]. Additionally, the authors create hierarchies for connectivity, characteristics of IT, IIoT devices and user interactions. However, the proposed terms are not connected with the frameworks, which are currently developed. Therefore, the proposed taxonomies can be seen more as parallel activities and less as compliant to the prominent IIoT reference architectures.

A number of surveys on IIoT and related architectures have been published recently. Weyrich and Ebert [220] provide a concise overview of the most relevant frameworks and initiatives. While outlining the main approaches from an industry-based view, an in-depth analysis is missing. Sethi and Sarangi [221] outline a collection of IIoT topics, mentioning key drivers and enabling technologies but lack the link to guidelines how to use those. Even though a short wrap-up on lessons learned is outlined, the important organizational players especially for industrial standardization are not covered.

Zhong et al. outline the key requirements and technologies for IIoT applications [222]. They present the relations between IIoT-enabled manufacturing, cyber-physical systems, cloud manufacturing and intelligent manufacturing mainly through AI applications. They briefly compare the main efforts from the US, Germany/EU, Japan and China and argue for an agent-based, generic framework covering all previous frameworks. However, they do not present a unifying framework that could fulfill this demand.

Thoben et al. group the developments and discussions in their review mainly according to smart manufacturing and the IIoT. In addition to Reference [222], they also discuss human-machine interaction with a special focus on safety and the prevention of hazards for the involved workers. Furthermore, they draw the conclusion that the description of the "variety of technical standards from various disciplines" [223] requires clear, widely-known reference models. However, the authors only refer to the Reference Architecture Model for Industrie 4.0 but do not make the link to other ones.

Similarly to the previously mentioned reviews, Strange and Zucchella highlight comparable issues and implications as the others—for instance cyber-security and data privacy as new challenges. However, in contrast to the others, Strange and Zucchella examine the IIoT from business-focus view. They emphasize less on the connectivity and interoperability topics but outline the effects on inter-organizational and international cooperations. In this context, they forecast the continuous decentralization of IT networks but also of supply chains and production networks in general. Unfortunately, they also finish their discussions on the thereby created new business-models by pointing to the IIoT reference architectures and do not further elaborate this point [224].

More innovative approaches for security aspects in the IIoT domain are presented by Aloqaily et al. The authors show how deep learning can detect intrusion attacks in the highly dynamic area of connected smart vehicles [225]. The gained insights are also transferable to the manufacturing domain as the general setting of high numbers of dynamically interconnected devices is one important characteristic of the IIoT. In a similar direction go Otoum et al. [226]. They outline an hybrid intrusion detection approach using Restricted Boltzman Machines. However, these works focus on very specific security concerns and need to be combined with a set of communication, encryption and authentication mechanisms in order to achieve a sufficient level of end-to-end security.

Cooperations between autonomous devices have been analyzed by Kotb et al. [227]. The authors model workflows as Petri-Nets and demonstrate the benefits of cooperative behavior in distributed settings. The sharing of resources and dynamic negotiation of services optimizes the overall performance. Similarily, Al-Khafajiy et al. promote a cooperative load-balancing model for fog computing [228]. The gained autonomy in the network and the cooperative aspect hardens the system regarding local overloads and federates the computational requirements among the IIoT network.

While a vast amount of literature examines the necessary technologies and implications for the IIoT, the industrial initiatives act as the key players for the actual realization of the IIoT concepts. Although the ongoing developments are considered by the respective literature, a comprehensive overview of the relevant frameworks and guidelines is still missing.

3.10 Summary

Many publications and research attempts have been made to tackle the identified challenges in the last decades. Obviously, as the intended flexible and reliable systems are not in place yet, some aspects have been missing. Primary IT-oriented approaches like Web Services, or OT-focused architectures like OPC UA only partly solved the issues. Furthermore, one can note a periodically reoccurring interest from varying communities, resulting in slightly different approaches, but struggling with the same problems. Among others, the appearing approaches tend to become more and more extensive during their development, quickly creating too high entrance barriers for newcomers due to the entailed complexity. In addition, previous approaches are only barely studied, and valuable insights not sufficiently adopted. A comprehensive view, combining descriptions, operations, and standardization is necessary to leverage the existing opportunities.

Most of all, many previous approaches require high starting investments, to reach the necessary interoperability and standardization level, before providing their benefits. A feasible process however must pay back its investments at every step, and provide enough value to justify the associated costs. Each part must furthermore be independently implementable and iteratively extendable, so a step-wise introduction is possible. Only if these conditions are met, it is possible to reach scenarios where the distribution reaches the critical mass, and network effects drive the IIoT into a self-enforcing development.

CHAPTER 4

The Semantic Digital Twin

Interoperability in distributed systems is one of the critical challenges also for the IIoT. As discussed in Chapter 1, the *Unpredictable Requirements* (CH1) of IIoT settings demand the generic, use case independent view on the representation and description of the involved Assets. Otherwise, the integration efforts triggered by each change of the application scenarios lead to unjustifiable update costs. The relevance of this challenge is already visible regarding current IIoT landscapes. The integration and digitization efforts from previous phases form today's heterogeneous environments, explaining why facilities that have grown over time form the *Brownfield settings* (CH2) of today.

While the previously made design decisions have been justified by the requirements of the time, investing the additional effort to reach generic, future-proven representations pays off in the mid- or long-term. One of the biggest struggles related to reaching this goal is the challenge to understand implicit *Design Decisions* (CH4), which define the behavior of an IIoT system. The gap between the design and deployment time, and the phases at which a reconfiguration takes place, makes it hard for the latter involved developers to foresee the side effects and constraints, which influence the proper operation. Therefore, it is necessary to create each IIoT component as an independent, self-contained unit in an unknown environment, to encapsulate every interaction in well-defined interfaces, and to minimize the dependencies to any – potentially disappearing – external resources.

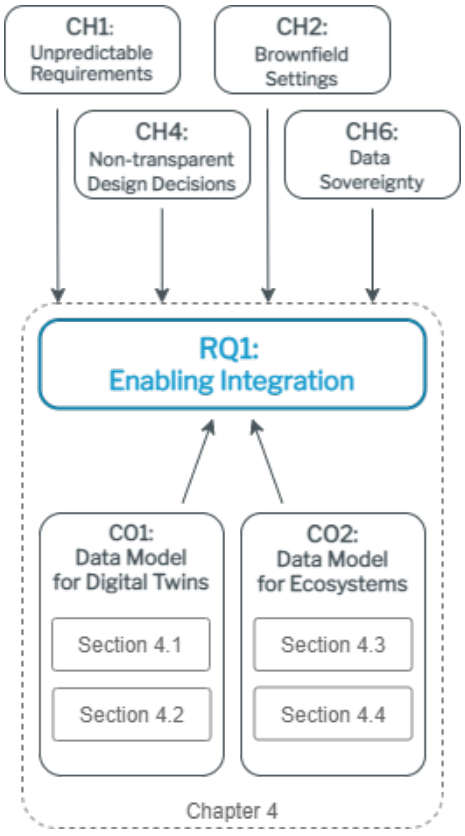


Figure 4.1: Contributions for RQ1.

Solving these interoperability challenges, however, is useless if the business concerns are not regarded. One of the core barriers that prevents deciders from implementing and opening their systems to other IIoT-driven organizations is their anticipation of losing their competitive advantage due to the sharing of business-critical data. Mechanisms to enable *Data Sovereignty* (CH6) are a must-have to establish and operate cross-organizational IIoT networks.

Semantically unambiguous descriptions and data exchange using well-standardized vocabularies can certainly solve parts of these challenges for IIoT Digital Twins. However, the proven

techniques of the (Semantic) Web are not regarded by state-of-the-art specifications, such as in the original data model of the AAS or the IIC Digital Twin [9]. In particular, formally defined data models must apply and integrate data in unforeseen scenarios with unexpected requirements. While many standardization initiatives have recognized this need, the lacking awareness of the existing, mature technologies hampers their adoption in the Digital Twin frameworks.

The contributions outlined in this chapter target this gap and examine how the established Semantic Web approaches the description of IIoT Assets. They thereby answer the first research question stated in Chapter 1:

RQ1: Enabling Integration – How should IIoT Assets be represented in order to enable their seamless integration in IIoT settings?

A general overview of this topic, its terms and requirements, has been published in chapters of the books 'Smart Service Management' [40] and 'Handbuch Industrie 4.0' [41]. The research question is answered by proposing two main contributions to the state of the art (cf. Fig 4.1). The path from Assets to Digital Twins is presented, the *data model for the standardized Digital Twin* in IIoT explained (CO1), and the required *data models for IIoT ecosystems* (CO2) described and how policies for data sovereign information usage can be formulated.

A brief overview of the state of the art and relevant publications is discussed in Chapter 3, mainly in Sections 3.1 and 3.4. Sections 4.1 and 4.1.3 of this chapter extend the state of the art and explain the necessary steps to reach a comprehensive understanding and consistent semantic model for Digital Twins in the IIoT (presented at SEMANTiCS'17 [30]), which leads to the *Semantic Asset Administration Shell* concept as outlined in Section 4.2 (CO1). This work has been published at SEMANTiCS'19 [19] and as part of the specification of the Asset Administration Shell [12].

The following sections explain how data sovereignty descriptions and formal contracts apply in a distributed ecosystem (CO2), which needs to integrate technical, legal, security, and liability concerns into a usable context (Chapter 4.3) and how they are integrated (Chapter 4.4). The corresponding papers have been presented at ISWC'20 [20] and in peer-reviewed workshops [21, 22], in the IDS Reference Architecture Model 3.0 [35], as well as in a White Paper [36]. Finally, the stated contributions are summarized and explained how they all together answer RQ1 (Section 4.5).

4.1 From Assets to Digital Twins

While IIoT Digital Twins require an appropriate internet connection, their representations are not mandatorily limited to 'equipped' Assets. There is no reason why physical objects which cannot be enhanced with network components cannot be represented accordingly. For instance, components of legacy machines can still have a virtual representation even though a digital module cannot be installed due to lacking network devices, high investment costs, or technical incompatibilities.

Even intangible Assets, like information content or data sets, are Assets and can have representations. The only requirement is that the related Asset and its Digital Twin are in some way synchronized. In particular, all valid facts for the Asset are also valid for the virtual representation and vice versa, considering a particular buffer in time and network communication. In addition, any operation applied or conducted by the Asset is mirrored to the virtual representation of the IIoT Digital Twin and vice versa. If the Asset's state changes, the virtual representation changes its state accordingly. This connection, however, can be in both ways, that a changed state of the virtual representation leads to an according transition for the represented Asset, allowing manipulations of the 'reality' by changing the digital

resource.

The ongoing digitization of industrial manufacturing has enabled new possibilities for data analysis and the development of new business processes, especially in industrial maintenance. So far, these opportunities have only resulted in minimal operative business value. As of 2019, only less than 10% of the German machine industry operates IIoT solutions on more than a prototype level [53]. One of the main challenges, for instance, for predictive maintenance solutions is that downstream processes have not yet been able to manage the inherent complexity of real-world maintenance delivery.

In order to structure the underlying challenges, a research agenda towards new business processes through integration procedures for heterogeneous data and analytic components based on IIoT compliant interfaces is presented. Furthermore, a decision support system for knowledge workers is outlined that can substantiate tacit knowledge by evaluating how different planning strategies and external effects influence a modeled field service network.

4.1.1 The IIoT from an Integration Viewpoint

The industrial maintenance services industry is used as the target domain, where companies rely on third-party providers to repair and maintain their machines. Generally, these service providers are obligated to deliver support within predefined response times. To avoid machine failure and the resulting short-term scheduling demand, providers generally inspect and maintain machines regularly. As a consequence, the regarded industrial maintenance is not only about delivering technical support but also about failure risk management. Therefore, detailed insights into the real-time status of the target devices present a significant advantage to schedule and execute maintenance activities.

Furthermore, it is the maintenance contractor's task to regularly inspect the machines but also to deliver support in certain response times in case of breakdowns. Therefore, industrial maintenance is not only limited to technical support but also to manage the failure risks. It is important to note that the maintenance contractor and the machine operator are usually different companies, which need to protect their respective competitive knowledge from each other.

Traditionally dispatchers manually coordinate field service technicians based on their individual experience. This process works sufficiently when the amount of incoming maintenance and repair tasks is low, and the requests do not exceed the available capacity. Nevertheless, especially urgent short-notice repairs force dispatchers – as the available staff is limited – to reschedule on short notice. Often these adjustments result in sub-optimal tours and in-efficient allocation of technicians, leading to a backlog of untreated tasks. However, this development also raises several challenges. Since data and functionalities originate from various vendors, they are heterogeneous, formatted in non-standardized schemes, and have inconsistent communication methods.

Furthermore, in a multi-partner business network, security concerns, compatibility issues, and differing views on objects and events make data exchange a non-trivial challenge. Consequently, the need for fast and adaptable support systems arises as external services have to be combined with internal applications, the IT infrastructure of customers needs to talk with the maintenance support center, and decisions based on data-driven analytic to empower maintenance organizations for the future. As a result, companies are looking for fast and adaptable support systems to support their dispatchers. The ongoing digitization of manufacturing and the growing amount of third-party advanced analytic services have significantly extended the capabilities of these tools. The continuous observation of manufacturing facilities leads to valuable insights into their current state and can even predict future failures. Combined with information on already planned maintenance intervals, dispatchers can utilize this information and schedule necessary repairs before failures happen and production is disrupted. These activities, commonly called predictive maintenance, are the top-ranked motivation for German manufacturing companies regarding the IIoT in

2020 [229].

Nevertheless, the described development further heavily increases the cognitive workload for the maintenance organization. The frequency and amount of incoming messages need to be handled, whereas at the same time the coordination effort with customers and technicians grows further on. At the position of the maintenance dispatcher, the thereby produced complexity reaches a level that is no longer efficiently and sufficiently manageable by human actors.

In order to meet the required demand, there are multiple adjustment possibilities: The number of incoming service requests (e.g. the service demand) can be reduced, the number of available technicians (e.g. the service supply) can be increased, or the service throughput can be increased, while keeping the service supply at the current level. Currently, increasing the throughput is the main objective for many companies. That is possible in two ways: (a) increasing scheduling efficiency and (b) increasing technician efficiency.

Interviews conducted with domain experts of multiple companies have shown that increasing technician's time at a machine – thus decreasing overhead – will lead to a situation in which the current service demand can already be met with current service capacities. In order to decrease overhead, the challenge is to support the dispatchers with decision support tools that can make justified decisions under complex circumstances. For a system to improve the overall situation, it must be able to reflect the dispatcher's "know-how" in its decision making while at the same time integrate all relevant currently – and future – involved systems.

In order to empower the maintenance provider for the IIoT, and at the same time make use of already available data and assisting third-party systems, a flexible data and system integration approach is an indispensable necessity. The target is a reusable and customizable concept where all participating modules communicate through consistent methods to allow the fast implementation of new functionalities. Based on semantic integration, state-of-the-art systems for tour planning and process simulation can further digitalize the scheduling process.

4.1.2 IIoT Assets in Distributed Architectures

A shared understanding of the meaning of data can be reached through semantically encoded data in RDF. That allows the publishing and exchanging of human and machine-readable information in standardized methods. The technical interoperability is ensured with RESTful Web APIs for requesting required information, its functionalities, and to control the involved components. Wherever different access protocols are already implemented by the component, a 'wrapper component' hides the required APIs and provides the functionalities in the desired form of RESTful APIs and semantic data. Thereby, the various existing communication protocols are substituted with one single mechanism. With the hiding of predefined specifications, suitable data exchange is facilitated and an integrated view of all involved components accomplished. Consequently, the integration of new components is simplified – an essential characteristic for the proposed scenario with a strong focus on continuous adjustments.

The described principles have already been implemented in a prototypical setup. GPS coordinates of field technicians are analyzed by remote services, and their output is used to visualize the current movements of technicians (Figure 4.2). As a demonstrator, the functionalities from a cloud service for geospatial analytics[1], a public map provider[2], and a commercial tour planning service showing how loose coupling enables seamless usage of components from different backgrounds and communication patterns.

Each service was transformed into suitable Web components during the integration process by translating the predefined interaction methods into the target scheme. So is the geospatial analytics service

[1] https://console.ng.bluemix.net/catalog/services/geospatial-analytics

[2] https://developers.google.com/maps/documentation/javascript/streetview

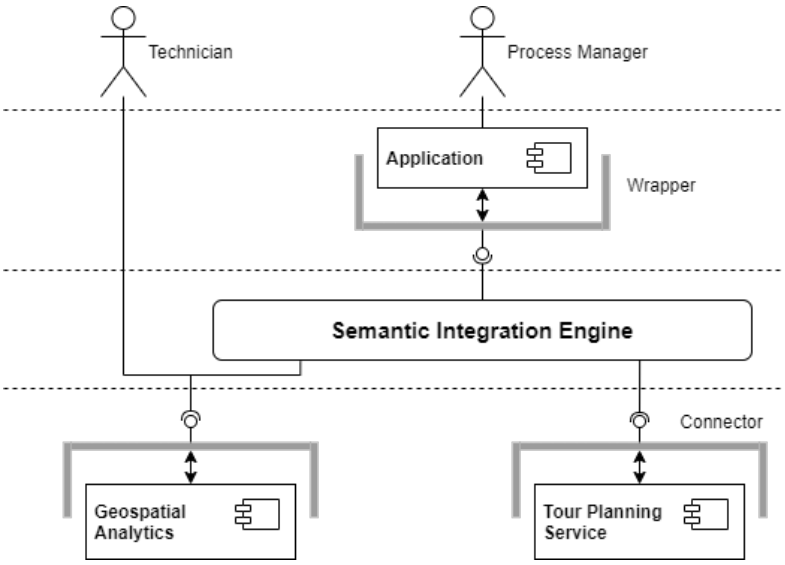


Figure 4.2: Prototypical implementation of an IIoT-compliant dispatching module

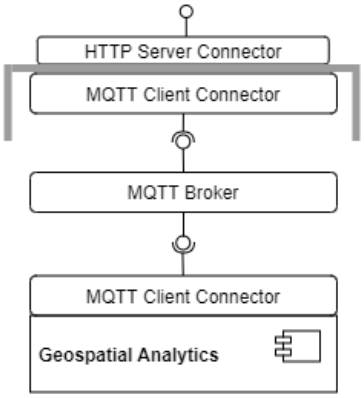


Figure 4.3: HTTP wrapper component, presenting an Asset into the IIoT

dependent on the publish/subscribe-based MQTT protocol as its input and output channel (figure 4.3). The implemented wrapper[3] offers an HTTP server connector that waits for HTTP requests. In the case of incoming HTTP data, the wrapper translates the statement into an MQTT message and calls its MQTT client connector to publish the message towards the service via a broker. In addition, the wrapper's MQTT client is subscribed to the broker for the output of the geospatial analytics. Whenever a message arrives at this connector, it is directly translated into RDF and represented as a Web resource hosted by the wrapper's HTTP server connector.

The SOAP-dependent tour planning service and the API of the map provider are treated in similar ways. The offered API methods are implemented by an according client connector at the individual wrapper and made available as Web resources offering data in RDF. Utilizing the HTTP APIs, a rule-based orchestration service can easily connect all components into one application. The prototypical use case combines scheduling information from the tour planning service and transfers them to a cloud service for geospatial analytics. The analytics service observes when the technician enters a certain, by the tour planning service defined region and notifies the visualization service containing the maps. This workflow can be accomplished with seven instructions which determine the whole data flow between the different services[4].

4.1.3 Self-Descriptive APIs

Currently, the World Wide Web is shifting from a human-centered to a more and more machine-processable environment. Especially global trends like the Internet of Things demonstrate how the established principles of the Web provide an infrastructure for basically any kind of information exchange. In addition to traditional, human-oriented websites, an increasing amount of Web APIs offer remote functionalities and data access composed of powerful federated applications. However, the integration process is a sophisticated task that requires a profound knowledge of the APIs and their specific constraints.

[3] https://github.com/sebbader/BlueWrapper
[4] https://github.com/sebbader/Industrie40Demo

The functionalities and operation semantics of remote interfaces differ significantly, making each integration or update activity a challenging and time-intensive task. At the same time, the World Wide Web is shifting from a human-centered to a more and more machine-processable environment. Modular, autonomous IIoT services relying on standard Web technologies can independently select and realize actions. Based on the semantic data format RDF, the proposed components consume, manipulate and send both data and program logic in one single consistent manner. Interfaces built on Linked Data standards combine data handling with service invocations. Thus, they can invoke, trigger and even reprogram each other autonomously and thereby create complex distributed systems.

The semantic API contains descriptions of input and output data and how functional and non-functional aspects assist in configuring the data exchange between IIoT compliant services. In order to increase the possibilities to find suitable functionality providers, the services are enabled to relax and even neglect conflicting aspects. Therefore, the outlined components encapsulate aspects of the integration problem and reduce the required manual implementation effort. However, these services still lack a common data model (IM for Digital Twins) and governance standards, for instance, on access or usage policy descriptions and instructions (cf. CH6: Data Sovereignty), in order to present a valid IIoT Digital Twin.

The internet, and in particular the World Wide Web, offers already a well understood and widely accepted infrastructure to exchange data. Standards like HTTP and URI have proven to allow reliable and straightforward communication mechanisms in a decentralized manner. As a consequence of the broad application of these technologies, more and more business-critical applications are shifted to or depend on the Web. The general underlying driver is the “winner takes it all” attitude of digital networks. This implies that whenever a newcomer invests in IIoT technologies, it will most likely choose the most widespread technology stack to maximize its individual value and minimize further integration costs. By joining the already dominant technology stack, its ratio increases, leading to an even higher motivation for the next newcomer to also invest accordingly. OPC UA-compliant systems present the de facto standard for OT devices while at the same time a development towards HTTP-based applications in the IT area can be observed. As the IT domain exceeds the OT by orders of magnitudes, the introduction of also HTTP-capable devices into the OT can be seen as an allowed assumption.

In addition, cloud services and on-demand solutions offer fast and flexible deployment of applications. Distributed services, running and connected through Web standards, enable the modularization and reuse of software artifacts and thereby increase the efficiency of IT solutions. Distributed applications integrate Web Services from various providers to benefit from already existing functionalities or share computation load. In theory, the clear separation between the single services allows fast integration and, at the same time, an easier maintainability. Nevertheless, a developer still needs to design the application’s architecture, understand the interaction patterns of each service and connect them together. Inconsistent and incomplete documentation in various formats, the absence of standardized interaction rules, and missing descriptions of important characteristics make this a challenging task. Furthermore, as Web Services potentially getting updated or even switched off, continuous maintenance is necessary. Nevertheless, especially small and medium companies can not effort to provide expensive support capabilities for a potentially unlimited period.

Therefore, to establish and use them in distributed applications, an adaptive control mechanism is necessary to cope with changes in the architecture, understand the requirements of Web Services of interest, and trigger and execute appropriate steps. In order to do so, formalized Web Service descriptions in a shared vocabulary are necessary, and a sufficient understanding of the practical requirements and decision criteria needs to be accomplished.

4.1.4 Self-governed IIoT Services

In order to create decentralized networks, a single controlling instance needs to be avoided. A central instance poses the risk of a single point of failure as well as it also combines the decision power and the single authority of the network. This is not aligning with the idea of a network of equal players, self-coordinating by shared principles (CH6: Data Sovereignty). The great success of the Web has already proven that a decentralized architecture without a powerful coordinator provides sufficient answers in terms of scalability and adaptability to unforeseen use cases.

The previously described concepts enable a technical integration through Web standards, a common interpretation of data through semantic definitions, and can change their behavior according to certain circumstances. Nevertheless, they are not yet capable of acting autonomously as they have neither the information nor the capability to reconfigure themselves on their own. Autonomous IIoT services are reached when abstract policies describing human motives are interpreted and directly applied not by a human controller but by the service itself. To enable the self-governed IIoT services to decide on behalf of their user or creator, it needs to translate vague specifications (e.g. “only use secure connections”) into technical requirements (“require SSL encryption at all times”).

4.1.5 Summary: IIoT Assets and Digital Twins

The prototypical realization of an exemplary IIoT scenario shows how fast heterogeneous components can be linked together when the data models are interoperable. The utilized Semantic Web principles lay the foundation for data-driven support tools, as shown with the proposed dispatching simulation functionality. Based on the feedback from practitioners of the domain, one can expect that these developments will drastically increase the efficiency of the whole industrial maintenance process.

With this progress, the role of the dispatcher is shifted from being a single organizer of technicians and their tours towards becoming a manager of the dispatching process as a whole. Thus, the dispatcher will monitor global target parameters and do appropriate situational scheduling improvements, making him capable to fully exploit the potentials of the IIoT.

The variety and amount of available Assets are a significant advantage of applications running in the IIoT. A well-known set of standard protocols and modeling principles enables the fast reuse of software functionalities. However, existing Web-based services show that the heterogeneity of their behavior and unforeseeable changes in the implementations require constant manual adjustments. Therefore, a stronger utilization is prohibited.

Digital Twins are one method to enhance the degree of automatization in such distributed architectures. The proposed methods target the local decision-making to empower the single Asset further. The ability to independently react to changes minimizes required maintenance of the Digital Twins and decreases the barriers of component reuse. This results in faster deployments, decreasing maintenance efforts, and reduced complexity. The additionally provided evaluation environment makes the proposed approaches comparable and opens the paths to continuous improvements.

The outlined approach describes the necessary components and techniques to gain more flexibility by combining heterogeneous Assets in distributed environments. Still, whereas the core technologies of the internet, the Web and the Semantic Web Stack have proven themselves, the applicability of adaptive components and decentralized, self-responsible coordination has still to be shown. Nevertheless, the foundation for any promising integration approach is a defined data model for the Assets, which is presented in the next section.

4.2 A Data Model for the Digital Twin: The AAS and Virtual Representations

The disruptive potential of the upcoming digital transformations for the industrial manufacturing domain has led to several reference frameworks and numerous standardization approaches. On the other hand, the Semantic Web community has made significant contributions in this field, such as data and service descriptions, integration of heterogeneous sources and devices, and AI techniques in distributed systems. However, these two workstreams are primarily unrelated and only briefly regard each other's requirements, practices, and terminology.

The contribution to this gap is the definition of the Semantic Asset Administration Shell (SAAS), an RDF-based representation of the IIoT Digital Twin. The concept of the Asset Administration Shell and its semantic representation has been developed as an industry standard to reach a common grammar for representing industrial Assets [19] [41] [12]. An ontology for the latest data model specification is provided together with an RML mapping, supplying resources to validate the RDF entities, and basic reasoning capabilities on the Asset Administration Shell data model are introduced. Furthermore, a discussion of the different assumptions and presentation patterns is applied, and an analysis of the implications of a semantic representation on the original data is conducted. The approach is evaluated and the thereby created overhead measured, which allows the conclusion that the semantic lifting is manageable, also for restricted or embedded devices, and therefore meets the conditions of IIoT scenarios.

4.2.1 Introduction

IIoT data is currently mainly exchanged in either JSON or XML. These commonly used data formats ease the serialization and parsing by providing specifications for the syntactic structure of the data objects. Additional information on the meaning of keys/values is usually specified in customized data models and schemata. The latest specification of the AAS also follows this convention [12]. The AAS is promoted as the Digital Twin for the German Plattform Industrie 4.0 and encompasses the interpretation of the digital representation of any production-related Asset. As such, materials and products, devices and machines but also software and digital services have a respective digital version.

While the predefined structure and the usage of the general terms of the AAS data model already reduce the heterogeneity inherent in the data exchange processes of current industrial scenarios, real-world scenarios still require a thorough understanding of their meaning, relations and requirements, as well as domain-specific terms. However, the data model itself is a plain catalog for core classes and attributes with limited class hierarchy information. Therefore, AAS implementations are dependent on extensive manual work and understanding of the extended AAS model through their developers, leading to time-consuming data mapping.

A semantic formalization of entities and data objects has several advantages in this context. The mature Semantic Web technology stack around RDF enables explicit references to classes, properties, and instances in the form of URIs, beyond the scope of single AAS objects but also across applications, domains, and organizations. The defined meaning of the used entities further allows its combination with predefined logical axioms, which allow the automatic derivation of new knowledge.

This contributes to the state of the art by presenting a mapping from the latest AAS data model to RDF. It contains a data model as an openly accessible ontology and creates SHACL shapes for all classes to enable schema validation. The various pitfalls of its usage are outlined, especially the different patterns to identify and refer to encoded entities, and to links to remote resources. Based on the inherent Web nature of RDF, it is shown how the transformation to the semantic data model decreases the amount of required storage space. Furthermore, the patterns to directly insert the RDF translation into the original XML and

JSON files are presented and their implications discussed. Relying on the RDF/XML and JSON-LD serializations, it is possible to merge the predefined data structure with the semantically defined data. It is shown that the provided extension points in the form of submodel elements are suitable for this task and that the output AAS files are still processable by existing software. Therefore the risk of compatibility issues is manageable.

In addition, the contribution is inserted into the process of the core AAS standardization working groups by designing, establishing, and maintaining the AAS data model identifier catalog [5].

The applicability of the presented approach is evaluated by determining the necessary overhead in terms of both storage and computation effort and by a detailed discussion of the restrictions of the RDF version. In addition, several semantic constructs require less space than the initially specified ones, whereas others are not directly compatible with the data structure of RDF, and some are even not expressible at all. In this context, the approach contains the following contributions:

(1) an RDF data model of the Semantic Asset Administration Shell $SAAS$,

(2) a mapping from XML Asset Administration Shell representations to $SAAS$,

(3) a set of preliminary reasoning axioms in order to explicitly derive implicitly encoded information from the data model, and

(4) a validation model for this data model encoded through SHACL shapes.

In the following, a formalization of the regarded domain is presented, followed by the presentation of the RAMI ontology and an RML mapping from the XML serialization. Several axioms for automated reasoning on top of the SAAS are examined, followed by a discussion of the suitable SHACL Shapes for the schema validation. Several use cases are used to evaluate the approach and to describe the characteristics and implications of the discussed model.

4.2.2 Methodology

The data model for the IIoT Asset aims to provide high coverage of the different modeling variants. RDF, on the other hand, has specific conditions on how data is presented (triple-based structure, URIs as identifiers). In order to structure the presented contributions, the parts of the respective data models are defined as follows:

AAS captures the information about the Administration Asset Shell itself. In this regard, *AAS* is the digital representation or Digital Twin of the Asset. Information from *AAS*, therefore, refers to the information object or document and only indirectly to the original Asset. Examples are the creation date of the digital representation, manuals, or how the AAS was generated or modified. It is important to note that the same reference is used to denote both the Administration Asset Shell itself and the set of information contained by it.

A captures the information about the actual Asset. The Asset can be anything of interest in the context of a digital production setting. Even though Assets are usually embedded devices or internet-capable components, any physical object, such as materials, production goods, or machines, can be seen as an Asset too. In addition, Assets also include software components and any digital service or intangible Asset, which is necessary to model a manufacturing use case.

S denotes the submodel of the Asset Shell. Submodels partition the provided information and categorize facts according to their usage, for instance, as part of a documentation submodel or a submodel for quality testing. Submodels are further separated into SubmodelElements, which are either themselves collections of SubmodelElements or the final bearer of key-value-encoded facts. As any combination of

[5] The catalog is provided as a GitHub repository of markdown files to supply a first, lightweight reference point for future provisioning of more expressive information at https://github.com/admin-shell-io/id)

Figure 4.4: Process steps through the provided modules.

different submodels can be included in the Asset Administration Shell, the set S^k represents the superset, including all possible submodels.

I is the set of identifiers for data objects. Specifically, $I = I_{glob} \cup I_{loc}$, where I_{glob} contains all globally valid identifiers, while the elements of I_{loc} are only valid in their context, in particular inside the AAS, which uses them.

The concept descriptions denoted with ***CD*** may provide further definitions about the used concepts, mainly attributes and data types. While concept descriptions are optional components of an AAS, they give the ability to place necessary explanations, especially for entities with local identifiers close to the data. Similar to submodels, concept descriptions are not limited in their appearance. Therefore the superset CD^l is used.

An instance *aas* of an AAS is, therefore, defined by the union of the mentioned sets:

$$aas \in AAS \cup A \cup S^k \cup CD^l \tag{4.1}$$

The identifiers appear in all sets and are therefore not mentioned separately. They connect the objects of the different sets with each other. However, the nature of identifiers in the AAS data model is mostly expressed by foreign keys, which do not link directly to the intended object. Two types of functions are defined on the Asset Administration Shell. First, a serialization *ser* transforms each instance to a representation in a data format, in particular JSON and XML: $ser : AAS \rightarrow D = \{XML, JSON, ...\}$
Second, a mapping is a transformation *m* from the data model *AAS* to the Semantic Asset Administration Shell $SAAS$. $SAAS$ is defined as

$$SAAS = AAS_{RDF} \cup A_{RDF} \cup S^k_{RDF} \cup CD^l_{RDF} \tag{4.2}$$

Using these definitions, an AAS in XML undergoes several steps (cf. Fig. 4.4). A created SAAS object using the provided mapping can be sent to a reasoning engine to enrich it with additional facts. Both the native $SAAS_{RDF}$ and the enriched $SAAS^{+}_{RDF}$ can be forwarded to a validation module. The validation module creates a validation report containing the errors and inconsistencies against the SAAS schema. Of course, also otherwise created SAAS objects can be sent to the reasoning or validation modules (bottom lane).

4.2.3 Asset Administration Shell as the Digital Twin

The Asset Administration Shell concept is a collection of several specification modules. The data model part specifies the core structure and different elements of the AAS, mainly administrative information about the AAS itself, the Asset, and use case-specific data collected and sorted in so-called Submodels. The serialized data can be transmitted in several data formats, for instance, JSON, XML, or any RDF serialization. Another part of the AAS specification targets the interaction in a remote environment. API calls for OPC UA, MQTT, and REST (based on HTTP) are currently under development.

Another specification targets the infrastructure components in an AAS-driven network. Different systems for search and discovery but also identity provision are required. For this challenge, two different aspects need to be distinguished. The identifiers of Assets and their AAS will always be in the authority of their owners, following their company-specific schemes. As the Digital Twin of the Asset is intended as a product itself, no manufacturer can be expected to rely on a third party for naming its products. The attributes, properties, and values contained in the AAS, however, need to be known to the intended users. Otherwise, no interoperability can be obtained. Rich and extensive vocabularies such as ECLASS[6] or the Common Data Dictionary (IEC CDD[7]) shall provide the necessary concepts. In the example, the Manufacturer can annotate the sensor streams using the ECLASS property '0173-1#02-BAJ001#006'. Thereby, the value is fixed to integer, and the Data Analyst directly knows that the values have the unit liters per minute (l/min).

These approaches, however, only marginally reflect the potential of a comprehensively Web-oriented approach. The catalogs only provide basic identifiers reflected in a taxonomy structure. Hypermedia links or machine-readable annotations are not provided out of the box. Identifiers have to be dereferenced using catalog-specific methods, following the assumption that Web technologies (cf. Ass. 1) are a uniting technology and in place in every IIoT scenario. Note that this does not mean that non-Web interactions shall be neglected. For instance, in machine-to-machine interactions, HTTP or Web browsers are usually not an appropriate choice. However, all developers, operators, or any other involved stakeholders are capable of accessing and exchanging Web-based information using the *currently* available tools, different from any other available technology set. This convenience advantage can not be underestimated since the acceptance by the target community is absolutely crucial for the success of any Industry 4.0 proposal.

Assumption 1 *Web technologies are the common denominator known to every Industry 4.0 stakeholder. All involved parties are familiar with URIs, HTTP, DNS, and Web Browsers and can use this technology set to establish Industry 4.0 use cases based on it.*

Linked Data and the Linked Data vocabularies can provide terms and concepts with rich annotations and machine-readable relations. Nevertheless, several reasons still severely hamper the broader adoption of Linked Data-based approaches for Industry 4.0 vocabularies. First, the requirements of industrial applications are **long-term** and for **formal and guaranteed maturity**. As legal liabilities arise from the implementation of concepts in productive facilities, a community-based approach such as DBpedia is indefensible. Established and well-reputed agencies need to back up such a vocabulary with clearly defined process and decision criteria. This is in contrast to the less formal procedures currently in place in the Semantic Web community, which is still driven mainly from an academic background.

Statement 1 *Long-term support and formal liability are crucial factors for the adoption of any digital resource through the manufacturing industry. Preferred are established standardization agencies such as ISO, IEC, NIST, or DIN.*

The second reason is certainly the separation of communities. Industrial developments are, by their nature, driven and managed by people with an engineering background. Web developers, on the other hand, do not understand the specific requirements or speak the used language appropriately. For instance, using the running example, a software application ran at the Data Analyst can quite easily be updated, relocated, or completely replaced. This is a common development for Web services. A physical device or facility needs to stick to safety regulations and certification processes as otherwise people could get harmed. This leads to significantly longer and more complex design and decision

[6] http://www.eclasscontent.com/

[7] https://cdd.iec.ch/cdd/iec61360/iec61360.nsf

processes. Combining digital services with physical Assets leads to clashes between both approaches and communities. Fortunately, a steady progress can be identified. This is driven by the identified business potential and the widespread conviction that neither non-digital devices nor plain software present future-proven products.

Statement 2 *The current solutions proposed by Industry 4.0 consortia hardly reflect the potential of Web technologies. The (Semantic) Web community, on the other hand, needs to reflect the formal requirements of industrial applications and potentially harmful consequences.*

An entity, as understood in this thesis, aggregates physical or tangible objects, information resources or digital data objects, software applications, and any other identifiable Asset. It becomes a Digital Twin when it is extended with a digital form, representing it in digital networks.

Note that the relation between an Asset and a representing AAS is not necessarily a one-to-one connection. For instance, the device in the example has an assigned AAS with all information for the Data Analyst and another one with relevant data for the Manufacturer's engineering department. They have different content and different identifiers. That means that not one single Digital Twin exists but rather overlapping sets.

Statement 3 *As Digital Twins go through their own but also reflect their Asset's lifecycles across companies and phases, one single, universal Digital Twin instance is usually not possible. Therefore, the AAS must be regarded as a fragmented set of information of the actual Asset.*

In particular, this implies that the Open World Assumption holds for the AAS. Potential reasoning or other examinations of their content always takes place with incomplete information. This fact is partly regarded by the efforts to standardize Submodels as use case-dependent information carriers, which fix the set of possible attributes and features.

All these models and data provisioning need to be based on a well-defined and consistent identification method. Currently, an incomprehensible set of legacy patterns is used in the domain, challenging a unifying approach. Furthermore, the ability and authority to select own identifiers is an essential characteristic for the public appearance of companies. However, proprietary schemes – sometimes implicitly encoding product families or versions – can lead to misunderstandings and errors at downstream services. Following the recommendations of the (Semantic) Web – using URIs – is a proven and technically easily manageable convention. This is formulated by the following assumption:

Assumption 2 *Every relevant actor in Industry 4.0 ecosystems has his own internet domain. Direct mappings between the domain and the related company are always possible, and vice versa.*

Due to the consideration that every company nowadays needs its own website – and thereby protects its DNS entry – the validity of this claim seems reasonable. The internet domain can be regarded as a valuable, well-protected resource. Even though one can construct use cases where companies lose control over their domain, this will only happen in very few cases. Consequently, a sufficiently durable namespace for identifiers is directly available for every company. Even though this assumption might be convincing at first glance, the legal obligations of stable industrial identifiers are not necessarily met yet. However, this gap might be resolved through new service providers, guaranteeing their sustainability as a new business model.

Assumption 3 *Internet domains are maintained over the whole lifecycle of a Digital Twin. A company will take measures to keep control over its domain entry at any time.*

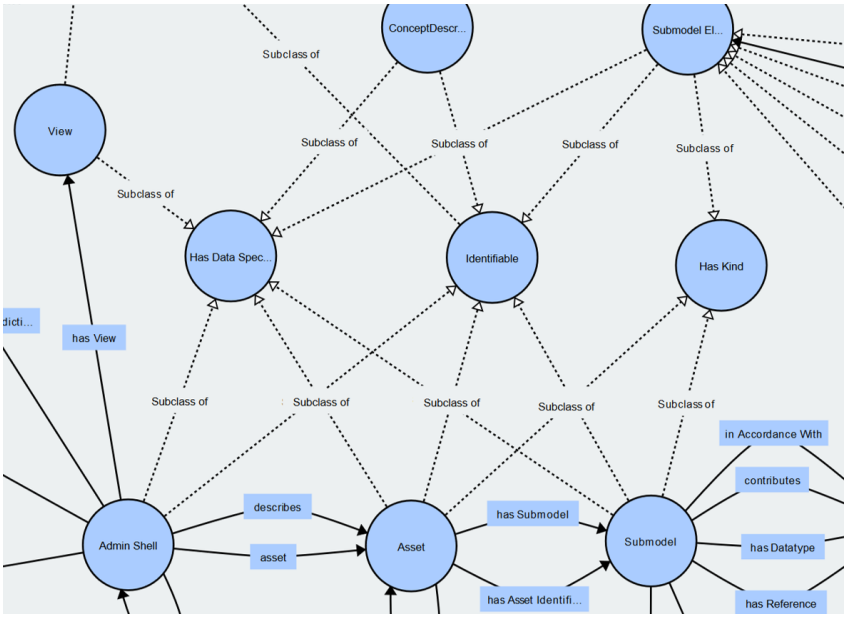


Figure 4.5: Overview of the most important classes and properties of the SAAS[8].

Statement 4 *Constructing identifiers by concatenating a company's internet authority with its dedicated product-naming scheme automatically creates globally unique identifiers. The expressiveness of URIs is suitable for transporting this information in a syntactically feasible manner.*

Statement 5 *URIs are the most commonly used identifiers throughout digital applications. Their established dissemination alone makes them superior to any other scheme.*

Note that the convenience of the target group is again the dominant argument in favor of URIs. While this appears as a non-functional argument, the fact that missing dissemination and adoption are the key obstacles of any Digital Twin concept demonstrates its importance.

4.2.4 The SAAS Data Model

In the following, the core *SAAS* data model as an RDF ontology[9] is presented (cf. Fig. 4.5). As mentioned, the ontology is an advanced version of the RAMI ontology [125] and, therefore, the namespace *rami* is used. For each class of the AAS data model [12], a corresponding OWL Class has been created, and every attribute has been mirrored with either an ObjectProperty or a DataProperty, except for the 'semanticId'. The reason for the latter is that 'semanticId' links to the unique identifier for the entity. In RDF, this is the entity URI itself and therefore does not need to be repeated.

All RDF entities are supplied with (sub)class assertions, labels, and comments. The SAAS classes reflect the original ones in most cases and form a subclass hierarchy based on the inheritance specification of the AAS data model. However, neither RDF nor OWL knows abstract classes. AAS uses abstract class constructs to partition specific attribute requirements and characteristics. For instance, the 'Has Kind' class covers all realizations, which contain a 'kind' attribute. This attribute encodes whether a certain

[8]For full visualization see http://www.visualdataweb.de/webvowl/#iri=https://raw.githubusercontent.com/i40-Tools/RAMIOntology/master/rami.ttl

[9]https://github.com/i40-Tools/RAMIOntology

entity refers to a concrete instance (the explicit machine installed in a shop floor) or is related to a whole type (machine type A can be installed in a certain setting). The data model reflects the abstract nature through *:class skos:note "abstract"* statements.

While the existing schemes for XML and JSON are based on a tree structure, the RDF data model supports a more generic graph structure. While this might lead to the conclusion that for every model from AAS_{xml} or AAS_{json}, a corresponding RDF serialization must be possible, therefore $AAS \subseteq SAAS$, the existence of several limitations is outlined, showing that actually, $AAS \supset SAAS$ is the case.

4.2.5 Mapping to RDF: The Digital Twin from a Data Viewpoint

The Administration Shell object (AAS) is the root of every Asset Administration Shell. Listing 4.1 shows an example XML snippet. As it is the root entity, it is also the entry point for traversing the SAAS graph. A native mapping is always possible if the identifier is already applied in the form of a URI. However, also International Registration Data Identifiers (IRDI) and any other custom format are allowed. While IRDIs in the case of the widespread ECLASS system can – with significant additional efforts – be mapped to URIs, this is, in general, a very hard and error-prone challenge[10]. This becomes even harder when regarding proprietary or custom identifiers. In addition, custom identifiers may contain special characters as spaces or several hash signs. These characters are percent-encoded (# → *%23*), changing the appearance of identifiers. As a result, only native URI identifiers can be mapped without risk, not only for AAS identifiers but also for the other sets in the following.

A consequence of this decision is also that the 'Has Semantics' class and the 'semanticId' property of the AAS data model become native to all objects. Moreover, it implies that all URIs are not only uniquely identifying their data object but also supply the semantic definition of their meaning. This rather strict requirement can be further aligned with the Linked Data Principles if URIs are also enforced to point to actual resources. However, dereferenceable URIs are not a requirement for now but should be seen as a preferable best practice.


Figure 4.6: The Web resource containing the mappings[11].

The Asset objects (A) constitute the link from the AAS to the real-world Asset. As Assets themselves only contain a very brief description, only the class assertions (rdf:type), the name (rdfs:label), descriptions (rdfs:comment), and the kind attribute are translated to A_{RDF}.

Submodels (S) and SubmodelElements are the core information carrier of the Asset Administration Shell. The basic structure of the submodel serves as a bracket for several SubmodelElements. Abstract SubmodelElements can be realized by Operations, ReferenceElements, Files, binary objects (Blob), and Properties. Properties have further attributes such as a key, value, value type, and several others. In order to align the Property class with the graph model of RDF, each instance is transformed to a respective rdf:Property. Therefore, a distinct class 'Property' does not exist in SAAS. The alternative usage of n-ary relations, which would further allow the linking of more attributes to the relation, was discarded in order to sustain cleaner graphs. Consequently, not all Property objects can be translated to the SAAS model.

Mainly, attributes and properties are converted to triples, and identifiers are restricted to URIs. Therefore, all identifiers of attributes become globally valid, as URIs are globally valid. It has been deliberately decided against n-ary constructs with blank nodes and an explicit property class, which would have been closer to the XML and JSON influenced data model. The reason is that a thereby created graph increases

[10] For instance, templates for ECLASS IDs, e.g. 26-04-07-02 (High-voltage current), may map to https://www.eclasscontent.com/index.php?id=26040702

[11] Full example: https://github.com/i40-Tools/RAMIOntology/tree/master/rml_mapping/mapping_examples

Listing 4.1: XML serialization of the Robot Gripper Arm AAS.

```
<?xml version="1.0"?>
<aas:aasenv xmlns:IEC61360="http://www.admin-shell.io/...">
  <aas:assetAdministrationShells>
    <aas:assetAdministrationShell>
      <aas:idShort>RobotGripperArm</aas:idShort>
      <aas:identification idType="URI">
        http://manufacturer3.com/aas/robot/
      </aas:identification>
      <aas:assetRef>
        <aas:keys>
          <aas:key type="Asset" local="true" idType="URI">
          http://manufacturer3.com/asset/gripper/755003377
          </aas:key>
        </aas:keys>
      </aas:assetRef>
      ...
    </aas:assetAdministrationShell>
  </aas:assetAdministrationShells>
</aas:aasenv>
```

Listing 4.2: Example RML TriplesMap excerpt.

```
   _:AssetShellMap a rr:TriplesMap ;
      ...
      rr:subjectMap  [
            rml:reference "identification" ;
            rr:class aas:AssetAdministrationShell ] ;
      rr:predicateObjectMap  [
            rr:predicateMap [ rr:constant rdfs:label ] ;
            rr:objectMap  [
                  rml:reference "idShort" ;
                  rr:termType rr:Literal ;
                  rr:datatype xsd:string ]
      ] ; ...
```

in complexity while its comprehensibility significantly decreases and the information content stays the same.

Concept description objects (*CD*) serve as local dictionaries for used entities. As the proliferation of definitions and metadata directly with the productive data eases its interpretation, Concept Descriptions increase the degree of interoperability between AAS providing and consuming components. However, RDF and Linked Data propagate the usage of dereferencing URIs in order to retrieve metadata. In that sense, Linked Data conventions can reduce the amount of transmitted data. On the other hand, not all relevant IIoT Assets are able to actively request such metadata. The possibility to independently open outgoing interactions beyond the restricted shop floor network is usually also a security risk and is not a

Listing 4.3: Equivalent representation to Listing 4.1 as RDF/Turtle.

```
<http://manufacturer3.com/aas/robot/> a aas:AssetAdministrationShell;
  rdfs:label "Robot Gripper Arm"^^xsd:string;
  aas:asset <http://manufacturer3.com/asset/gripper/755003377>; ...
```

good practice. Therefore, Concept Descriptions are a valuable feature to ship metadata and to ensure a common understanding of the shipped AAS. The mapping itself is provided as RML TripleMaps (cf. Listing 4.2) and can be executed with the open-source tool RMLMapper[12] (cf. Listing 4.3).

4.2.6 Reasoning

RDF and RDFS already contain trivial entailment rule sets[13]. As RDF and RDFS are very general vocabularies, the allowed reasoning focuses on the syntactic position (subject, predicate, object) of entities in RDF graphs. For instance, the information that p is an instance of the class Property can be inferred from the fact that a triple with p at the predicate position exists. Although rule entailments of this kind are certainly correct, the created amount of explicit data increases significantly while the information content stays nearly the same.

In order to illustrate the power of reasoning based on the SAAS, selected rule sets using owl:sameAs and rdfs:subClassOf properties have been prepared. The rules are encoded in N3 according to Stadtmüller et al. in order to use their Linked Data Integration and Reasoning Engine [100]. In addition to the two entailment regimes, both consisting of several single rules[14], the SAAS ontology with its inherent axioms is integrated on the fly.

4.2.7 Schema Validation

The AAS presents a closed-world model. As such, the definitions of classes and properties must be regarded as restrictions. Simply reusing properties for class A that have been introduced for class B usually causes a violation of the model. RDF, on the other hand, allows by default all not excluded patterns. Nevertheless, industrial use cases require verifiable statements on the data content but also its structure.

The Shapes Constraint Language (SHACL) [61] introduces a W3C Recommendation for validation mechanisms on RDF graphs. The definition of required attributes, the cardinality of relations, or datatype restrictions in the form of shapes is an important aspect to enable data quality assurance in any productive system. Some tools are already created to assist the creation of SHACL shapes, for instance, a Protégé plugin and as a part of TopBraid Composer. As SHACL shapes are also defined in RDF, they share the same format as the validated data in contrast to plain SPARQL Rules. This eases the required technology stack and reduces the number of used libraries.

The SAAS supplies respective shapes for all its classes[15]. These shapes mainly check for mandatory properties but also check the existence of label and comment annotations. In addition, the shapes are essential in order to check the incoming data during the exchange of Asset Administration Shells. Furthermore, the shapes can also be used to describe input and output specifications. For instance, an

[12] accessible at https://github.com/RMLio/rmlmapper-java

[13] https://www.w3.org/TR/rdf11-mt/

[14] rdfs9 and rdfs11 from [59], transitivity, symmetry and replaceability characteristic for owl:sameAs

[15] https://github.com/i40-Tools/RAMIOntology/tree/master/schema

	#XML Leaves / #XML Nodes	AAS (XML)	#Triples	SAAS (XML)	SAAS (nquad)	SAAS (turtle)	SAAS (JSON-LD)
RaspberryPi	1161/2864	148 KB	510	40 KB	86 KB	32 KB	51 KB
AAS2	925/2604	91 KB	459	17 KB	58 KB	12 KB	20 KB
AAS3	2651/6743	313 KB	1154	43 KB	156 KB	31 KB	52 KB

Table 4.1: Results of the SAAS mapping and RDF serialization.

IIoT Asset can postulate that its API requires data objects conforming to the Asset Shape and will output Submodel objects as defined by the Submodel Shape.

4.2.8 Use Cases

Three different Asset Administration Shells are used to evaluate the approach. All of them are reflecting the specifications from [12] and are in the AASX file format. The corresponding descriptions are included in XML files contained in the AASX files.

Robot Gripper Arm. The first Asset Administration Shell represents Robot Gripper Arm[16] (cf. Listing 4.1). Three Submodels are included, namely one for the technical characteristics, one containing documentation material as the product sheet and a usage manual, as well as one submodel explaining the Asset itself. Here, the Asset is one specific Robot Gripper Arm (kind=instance, part number 755003377) and not referring to the type of product of all Robot Arms, which have been produced or will ever be produced (kind=type). Therefore, the description is only valid for the one Robot Gripper Arm that has been delivered to the operator O. This AAS delivers a total of 52 SubmodelElements.

Automation Controller. AAS2 describes an electronic controller for automation facilities. As it is not approved as an official artifact, the providing company as well as its details, can unfortunately not be published. AAS2 contains one Asset, three submodels, and more than 100 SubmodelElements.

Multi-protocol Controller. The third use case (AAS3) represents an internet-capable controller unit with multiple protocol support. Like AAS2, this Asset Administration Shell is not officially published yet. However, the creation of either AAS2 or AAS3 happened independently of this thesis. The third use case includes one Asset with eight Submodels and more than 150 SubmodelElements.

4.2.9 Experimental Evaluation

The evaluation of the AAS to SAAS mapping examines the results and the performance of the three use cases (cf. Table 4.1). As a reference to estimate the information coverage, the number of XML nodes of the AAS serializations is provided. In addition, the number of unique leaves of the three XML trees is noted, as these numbers better reflect the single information content of the AAS. Table 4.1 also presents the numbers of generated triples by the RML Mapper. The comparison indicates that not the whole expressiveness of AAS can be transported to the SAAS version. This is due to the fact that some constructs can not be represented sufficiently in RDF (for instance the Property class), but also many original entities contain redundant information. Especially the ConceptDescriptions repeat many attributes, which are collapsed by the mapping process and only added once.

The necessary overhead in terms of computation time measured in milliseconds is presented in Fig. 4.7, in addition to the average mapping times outlined in the last column of Table 4.1. The time was measured on a regular laptop (Win10, 16GB, Intel i5-7300 2,60 GHz) using a bash emulation. The different RDF

[16]The initial evaluation in the paper [19] used a Raspberry Pi 3B+ as the first example. The renaming shall keep the readability while all examined aspects have stayed the same.

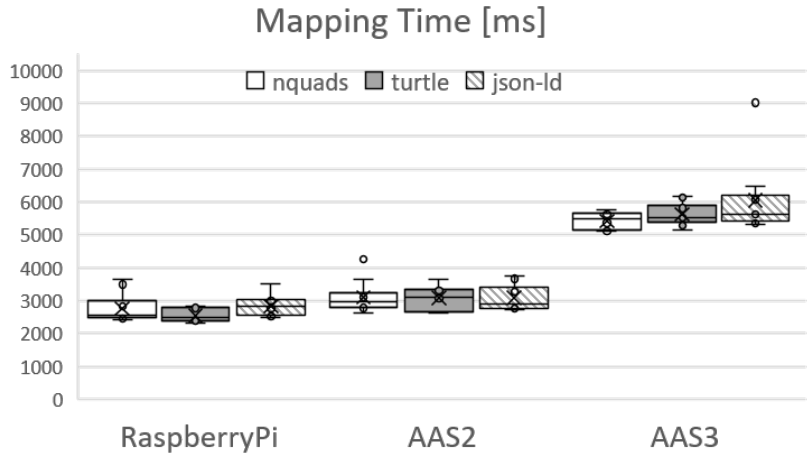


Figure 4.7: Mapping times for the three Asset Administration Shells.

serializations do influence the execution time, indicating that the writing is not the bottleneck. While the average mapping time of the Raspberry Pi AAS (2,7 sec) and AAS2 (3,1 sec) are relatively close, the duration for AAS3 (5,7 sec) is significantly higher. The variation between the selected use cases reflects the differences in their XML file size. This could indicate that the overall behavior is nearly linear. However, each of the 19 TripleMaps leads to reloading and the reiteration of the whole XML file. Overcoming this expensive process would speed up the process significantly but is out of the scope for this thesis.

RDF is, in general, not an effective data format in terms of storage efficiency. Nevertheless, the syntax requirements of the AAS and especially its XML schema create already significant overhead for the original AAS model. As depicted in Table 4.1, all RDF serializations reduce the necessary storage size. Especially noteworthy is the difference between the original XML file size and the RDF/XML serialization. This is mostly due to the usage of namespaces in the RDF/XML version, which reduces the noted URIs. It should be mentioned that for all serializations, the mapping step (m) and the serialization (ser) were executed directly by the mapping engine.

Nevertheless, the resulting costs in terms of storage requirements and communication bandwidth do not exceed the ones created by the original Asset Administration Shells. Consequently, all devices and scenarios capable of handling AAS are also sufficient for the operation of SAAS. Furthermore, the possible serialization of SAAS as both XML and JSON should enable AAS implementations to quickly adapt to SAAS objects in their original file format.

Three different rule sets have been applied to all use cases. All rulesets contain a Web request to the ontology source file in order to load the class hierarchy and any other relevant axioms of the data model itself. The first one also adds several rules reflecting the symmetry and transitivity of *owl:sameAs*, as well as the fact that the same instances share all properties and annotations of each other. The second rule set contains subclass statements as encoded by the rules rdfs9 and rdfs11 [59]. The third set combines both to the most expressive reasoning set. Table 4.2 gives an overview of the amount of created triples. *rdfs:subClassOf*, *owl:sameAs* and the combination of both entailments are shown with the amount of uniquely added triples and the average reasoning time.

The semantic rule engine Linked Data-Fu [100] is used to execute the reasoning tasks. The preparation of the reasoning engine, involving the parsing of the rule files, takes around 1 second. The following Web request, the download of the ontology, the evaluation of the rules, and the serialization to the n-triple file

	Triples (original)	sameAs (triples)	sameAs (time)	subClassOf (triples)	subClassOf (time)	both (triples)	both (time)
RaspberryPi	510	959	2,760 ms	771	2,719 ms	1,217	2,808 ms
AAS2	459	452	3,057 ms	367	2,368 ms	570	2,313 ms
AAS3	1154	1,115	2,776 ms	818	2,677 ms	1,343	2,668 ms

Table 4.2: Added triples by the different rule sets.

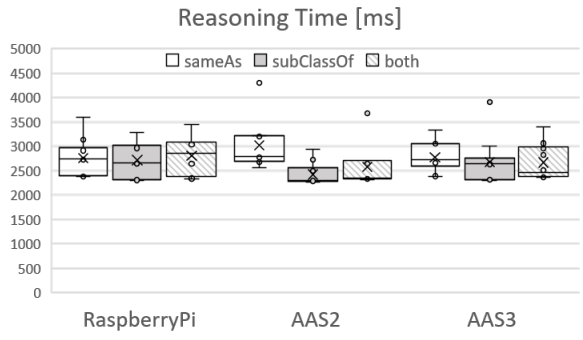


Figure 4.8: SAAS Reasoning duration.

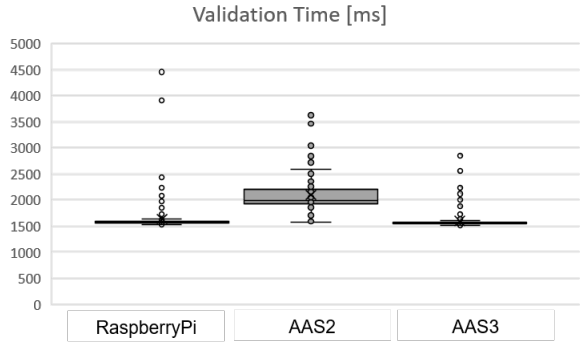


Figure 4.9: Schema validation performance.

are then executed. The duration distribution of ten repetitions is shown in Fig. 4.8. One can see that the whole process takes between 2,3 and 3,3 seconds, nearly independently of the amount of the input size (AAS3 is significantly larger than the graph for the robot gripper arm) and the expressiveness of the rule sets (the second set is leading to significantly fewer results than the others).

As the rulesets are only regarding the structure of the ontology, the inferencing of context-dependent knowledge is not yet possible. In order to reach productively usable information, domain-specific axioms tailored to the actually contained or expected data are necessary. However, it is shown that the reasoning process with complex rules is applicable in an acceptable amount of time.

The evaluation times of the SHACL shapes are shown in Fig. 4.9. On average, the execution of all shapes takes 46,2 seconds and the execution of one single shape 1,8 seconds. All shapes have been executed a total of ten times.

About 2 seconds are required for setting up the validation tool and parsing the data shape (the Asset Administration Shell) and the single class shape. The size of the Asset Administration Shell has no significant impact on the achieved results. Regarding these conditions, the necessary effort is acceptable for a typical IIoT scenario as the validation itself is not necessary for every restricted device. This is due to the fact that the validation of data takes either place at development or deployment time, where time is not critical. In addition, the validation is essential for the higher-level data analytical services that usually run on more powerful machines or are even hosted in the cloud.

4.2.10 Summary: A Data Model for the Semantic Digital Twin

This contribution presents a semantic version of the Administration Admin Shell, a mapping from its XML serialization to any RDF serialization, schema validation shapes, and a brief set of reasoning rules. In that context, the lifting process of the AAS data to a semantic integration layer (CO1) has been shown.

This is one step to an automated integration of IIoT components. Existing, non-customized tools can work with the RDF model of the AAS and execute their task without prior configuration (CH2). This

enables the implementation of real interoperable pipelines and data-driven workflows, not only on the data format and syntax level but also regarding the meaning of the data. Furthermore, the examined overhead of the SAAS model has been examined and showed that the requirements do not exceed the requirements set by the original AAS model.

The mapping outlines the data lifting to the SAAS RDF model. The lowering of RDF to the original AAS data model has not yet been achieved. Furthermore, the main benefit of the semantic model is, besides its formalized meaning, the interlinking with other definitions and the integration of additional sources.

For now, only the data provisioning capabilities of the AAS are defined. As a next step, the provisioning and invocation of operations through such Digital Twins are currently specified. Using semantically defined descriptions of the respective interfaces, their input and output parameters, and the provided operations will allow the IIoT community to rely on the vast amount of expertise and experience with Web Services and Semantic Web Services in particular. This way, the goal of truly interoperable and flexible manufacturing workflows, where software and hardware, materials and products, customers and suppliers form on-demand information chains, benefits from the existing research in the area. This combination paves the way to continuous reorganizations and reconfiguration for any kind of appearing use case (CH1).

The presented modules need further maintenance to keep the semantic models aligned with the progress of the Asset Shell specification. Furthermore, two main challenges appear which must be tackled by the semantic community. First, the core potential of the Semantic Web – the seamless integration of heterogeneous devices, services and data sources – still lacks sufficient numbers of implemented use cases and deployed scenarios in practice. Second, the reoccurring discussion on identifiers in distributed settings is a huge chance for the established practices of the Semantic Web and Linked Data in particular. However, the benefits of (dereferenceable) URIs are still underestimated in the manufacturing community, primarily because of missing experiences.

4.3 A Data Model for IIoT Ecosystems: The IDS Information Model

There is currently no standardized and widely accepted way for a trustful exchange of business data that ensures traceability, data owner's privacy, and sovereignty. Privacy concerns and the protection of proprietary information are critical factors of future data infrastructures [230]. Such an infrastructure is a key prerequisite for a secure, standardized, and fine-grained sharing of sensitive business data, unlocking the potential for novel value creation chains and the inception of intermediation platforms [231].

While private communication channels mainly have to provide simple access and sufficient quality of experience, business-related data exchange must go further. The IDS is based on the detailed understanding and need for technically enforced *trustworthiness* of the involved actors, components, and interaction patterns. *Security* and *Data Sovereignty* need to be considered from end to end in order to create a feasible ecosystem for digital business interactions. This section outlines the motivation and capabilities behind this ecosystem, and in particular, presents its data model as the common denominator of all aspects.

4.3.1 Introduction

The IDS proposes a message-based approach to bridge syntactic differences. Still, a successful exchange of data objects requires a sufficient understanding of its content and meaning. A shared information model is therefore needed. The *IDS Information Model* (IDS IM) is an RDFS/OWL ontology, which defines the

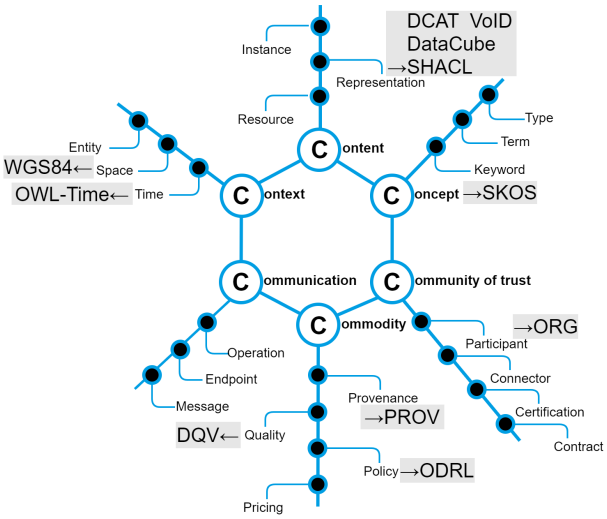


Figure 4.10: Partitions of the ontology by concern (pointing to standards reused).

general concepts depicted in Fig. 4.10 along with roles required to describe actors, components, roles, and interactions in a data space. This ontology serves two purposes, (1) as a catalog of machine-readable terms and data schema for IDS components and (2) as a shared language for all stakeholders. Each involved player needs to understand and be able to interpret this set of terms, thus enabling semantic interoperability in federated environments. The IDS IM presents the backbone and common denominator for the data-sovereign ecosystem as envisioned by the IDS.

The contribution specifies declarative definitions that enable sovereign data exchange in federated spaces. It is shown how the latest developments in standard Semantic Web vocabularies have been assembled together to enable interoperability in federated data ecosystems.

4.3.2 Governance and Context of the IDS Information Model

The IDS has been designed in a systematic process with the broad involvement of industrial stakeholders [232]. Its specification and reference implementations are maintained and supported by the International Data Spaces Association (IDSA), a non-profit organization to disseminate and evolve the IDS views and principles. The IDSA, with more than 100 member organizations meanwhile, serves as the institutional body for promoting the IDS in research projects and industrial applications. In particular, via its sub-working group (SWG) 4 “Information Model”, the IDSA ensures the sustainability of the ontology and provides the resources for future extensions.

The IDS Reference Architecture Model (RAM) defines the roles assumed and the responsibilities of organizations interacting in a data space [35]. These definitions are given on multiple layers (business, functional, process, information, system) and from multiple perspectives (security, certification, governance). Fig. 4.11 shows, for a broad initial overview, the core *interactions* and *roles* in the IDS.[17] Data Providers exchange messages with Data Consumers via standardized software interfaces and use multiple services to support this. They can, for example, publish metadata about resources to a directory (“broker”) and thus allow others to find these. At the heart of every IDS interaction is the adherence to the usage rules – accomplished by the connection of machine-readable usage policies with each interaction and the application of certified, trustworthy execution environments. The so-called IDS Connectors interpret and

[17]The “certification/membership mandatory” badges refer to receiving official IDSA recognition and support.)

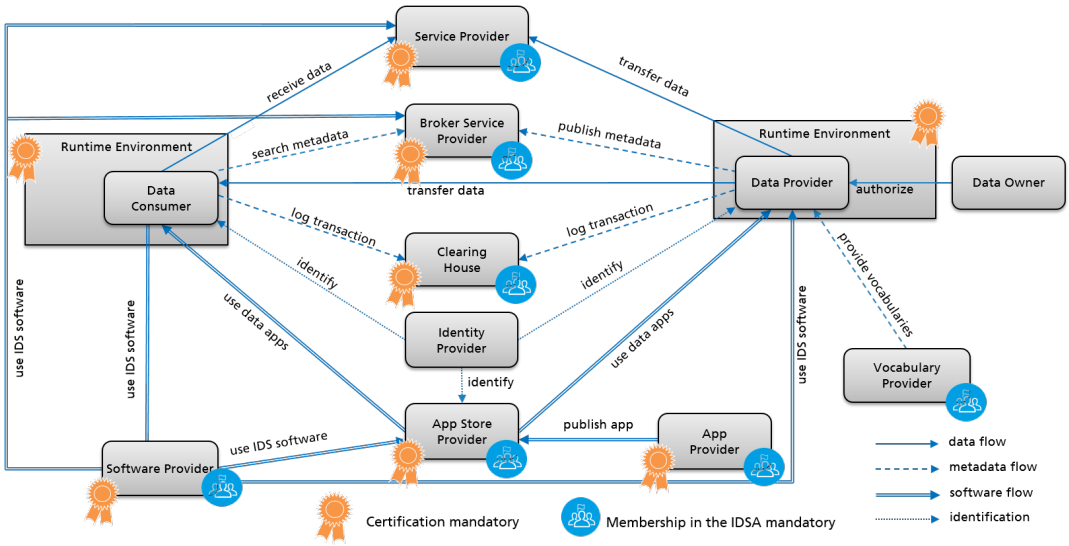


Figure 4.11: IDS Reference Architecture with its main roles and interactions (source [35]).

enforce the applied policies, thus creating a federated network for a trustworthy data exchange. A Data Provider is authorized by a Data Owner, and a participant might as well be both a Data Owner and Data Provider at the same time. The different Providers can participate on behalf of other actors who cannot host a technical IDS infrastructure for providing data.

The IDS IM specifies the domain-agnostic common language of the IDS. The IM is the essential agreement shared by the participants and components of the IDS, facilitating compatibility and interoperability. It serves the stakeholders' requirement "that metadata should not be limited to syntactical information about data, but also include data ownership information, general usage conditions, prices for data use, and information about where and how the data can be accessed" [232] by supporting the description, publication and identification of (digital) resources. Beyond these core commodities, the IM describes essential constituents of a data space, its participants, its infrastructure components, and its processes. It is, like other elementary IDS software components, available as open source to foster adoption (cf. Tab. 4.3). The IM is specified on three formalization levels (conceptual, declarative, and programmatic representation). The ontology, the normative implementation of the declarative UML representation in the IDS RAM, was initially created in 2017 and first released in 2018. [18]

Digital content is represented by several abstraction layers in the ontology. *Resources* describe aspects of digital content that is shared equally by all of its *Representations*. A *Resource's Representation* adds serialization details regarding the physical manifestation of a Resource and may refer to the actual contents. Contents can be simple values or complex data structures, called *Artifacts*. The *Connector* is the core building block of any International Data Space, a communication server providing and consuming digital content via a number of resource endpoints. Additional infrastructure components like *Identity Providers*, *Metadata Brokers*, and *AppStores* supplement the connectors with various required services. The overall objective is to design an interoperable ecosystem of trustworthy components, enabling sovereign data exchange. The various aspects must be unambiguously defined with and through the IDS Information Model. The *Broker* component is a meta-data registry of *Resource* offerings, whereas the *AppStore* is a registry of software offerings and a secure registry for distribution. The *Vocabulary Hub* serves the maintenance of shared vocabularies and related (schema) documents. The *Identity Provider*

[18] https://github.com/International-Data-Spaces-Association/InformationModel/releases/tag/v1.0.0

Table 4.3: Key facts about the IDS Information Model and related resources.

General	**License**	Apache License 2.0
	Size	278 classes, 149 object properties, 115 data properties, 684 individuals
	Total size	3912 triples
Reuse	**Reused ontologies**	CC, DCAT, DCMI Terms, FOAF, ODRL, OWL-Time, VoID, etc.
Documentation	**Ontology documentation**	https://w3id.org/idsa/core/
	Element description	Using `rdfs:label`, `rdfs:comment`
Availability	**Namespace**	ids: https://w3id.org/idsa/core/
		idsc: https://w3id.org/idsa/code/
	Serializations	Turtle, RDF/XML, JSON-LD, N-Triples
	GitHub	https://github.com/International-Data-Spaces-Association/InformationModel/
	VoCol Instance	http://vocol.iais.fraunhofer.de/ids/

manages and validates the digital identity of Industrial Data Space participants. The *Clearing House* provides clearing and settlement services B2B interactions within International Data Spaces. All these components need to be unambiguously described for all stakeholders.

The further usage is illustrated using the Digital Twin of the robot gripper arm, supplied by *O*, to explain the data model's core classes (cf. Fig. 4.12), the characteristics of the model, and how they can be used to tackle the given requirements. The example consists of the AssetAdministrationShell of the robot that the Operator *O* provides through its IDS Connector to its partners. The Self-Description (cf. Listing 4.4) connects the resource with its representation and further links to the serialized instance. As *O* needs to protect the access and the usage of the Digital Twin, it also formulated a machine-readable usage offer in the form of an ids:ContractOffer. This contract contains the specifications on which activities are allowed and which prerequisites have been met before the permission for an intended action is granted. The IDS Connector as the execution environment of these evaluation steps ensures that the contract details are followed, both on the provider side (*O*) and the consumer side, for instance, in the IDS Connector of the Data Analyst *D*.

1. transforming existing data into economic value while
2. restricting access and subsequent usage, and thus
3. ensuring his sovereignty over its data.

These translate to four requirements:

(R1) Describe the data resource to make it discoverable (to potential, still unknown customers)

(R2) Create business value through data exchange

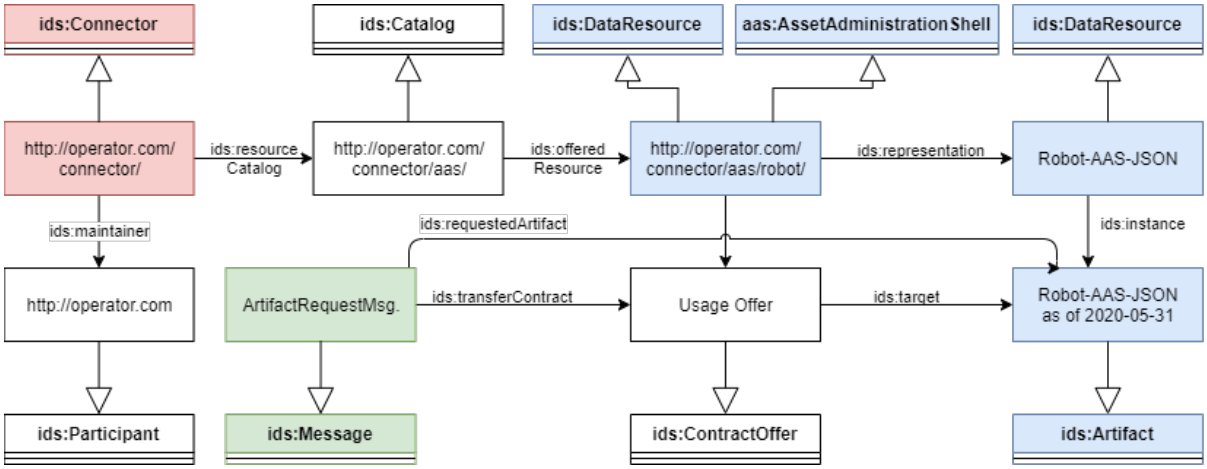


Figure 4.12: IDS core classes and their instances in the running example.

(R3) Describe intended and prevent unintended usages

(R4) Control the usage over the complete digital lifecycle

The description and announcement of the data resource (R1) are shown in Listing 4.4.[19] The unambiguous metadata is understood by every other participant in the IDS. In addition, as Attard et al. explain, the added value from digital data can only be created through value co-creation [233]. Therefore, data resources must be made available at the right time to the right consumer. The requirement (R2) is fulfilled by the IDS infrastructure and the involved components, which are able to interpret a data resource based on its self-description and understand the described relations (cf. Fig. 4.12).

Listing 4.4: The robot gripper arm modeled as an IDS DataResource.

```
_:Robot a ids:DataResource ;
  ids:title "Robot Gripper Arm"@en ;
  ids:description "This data resource contains the IIoT Digital Twin
    of the Robot Gripper Arm with all its Submodels."@en ;
  ids:keyword "asset shell", "robot", "digital twin" ;
  ids:publisher <http://operator.com/> ;
  ids:temporalCoverage [ a ids:Interval ;
    ids:begin [ a ids:Instant ;
      ids:dateTime "2021-01-01T00:00:00.000-04:00"^^xsd:dateTimeStamp ];
    ids:end [ a ids:Instant ;
      ids:dateTime "2021-12-31T23:59:59.999-04:00"^^xsd:dateTimeStamp]];
  ids:language idsc:EN ;
  ids:representation [ ids:instance _:Robot-AAS-JSON ; ids:mediaType
    <https://www.iana.org/assignments/media-types/application/json>];
  ids:resourceEndpoint [ a ids:ConnectorEndpoint ; ids:accessURL
    "https://opertor.com/connector/aas/robot.json"^^xsd:anyURI];
  ids:contractOffer _:DigitalTwinDataOffer ;
.

_:Robot-AAS-JSON a ids:Artifact ;
    ids:byteSize "2923497" ;
    ids:fileName "robot.json";
    ids:creationDate "2020-05-31" .
```

[19] URIs are abbreviated following http://prefix.cc/.

In order to uniformly identify, describe and implement interactions among participants and infrastructure components, these interactions were modeled in terms of typed messages being exchanged. The Information Model ontology describes messages for defined phases or aspects of interactions in an extensive taxonomy. The class *Message* is at the top level in the taxonomy and describes all common properties of the messages. The actual messages are classified into three main categories. Each category defines the semantics, attributes, and optionally the type of transported payload of the messages. The *RequestMessage*s initiates a communication, motivated by a defined purpose expecting a *ResponseMessage*s to occur. The response contains information about the reaction (result, status, or fault) to a former, correlated request. *NotificationMessage*s represent unsolicited informative events broadcast to a configurable set of recipients.

4.3.3 Methodology

The IDS overall has been designed as an alliance-driven multi-sided platform [232]. The basic process is aligned with the *eXtreme Design* method [234], with a strong focus on agile and collaborative workflows. The role of a customer is filled through a dedicated *ontology owner*, an experienced ontology expert who acts as the link to the developer community. In addition, the IDS IM is driven by the initial requirements originally collected and described in the RAM [35] and later on represented through publicly accessible *issues*. Furthermore, the *eXtreme Design* proposal to use separate ontology modules has led to the partitions shown in Fig. 4.10. As demanded by [235], the ontology development process needs to be test-driven, which is implemented by an automated syntax validation together with a semi-automated code generation pipeline. This code is integrated into several runtime components, in particular IDS Connectors, serving as a test environment for each and every update.

Deep integration with state-of-the-art software development platforms (Git, continuous integration, build agents, sprint-based development) enables agile, iterative release management. Combining these characteristics with Semantic Web best practices led to the core design principles of the IDS IM:

Reuse: The body of existing work is evaluated and reused by refining terms of standard vocabularies, many of them being W3C Recommendations.

Linked Data: The IM is published under a stable namespace, in common RDF serializations together with human-readable documentation and interlinked with external resources.

FAIR: The ontology as a whole follows the FAIR principles (findable, accessible, interoperable, reusable [28]). The respective aspects are in the following stated through references in brackets, for instance (F1) depicting “the globally unique and persistent identifier” (page 4 [28]).

Separation of concerns: Each module of the ontology addresses a dedicated concern that applies to a digital resource (cf. Fig. 4.10).

Instances of the IM can also be validated against its schema by using SHACL shapes. Every class has its corresponding shape, stating the required properties, their cardinality requirements, and value types[20]. Furthermore, all shapes have a human-readable explanation message, including a reference to the original shape. The shapes are used to (1) validate incoming data objects but also (2) to describe the restrictions on class attributes. Thereby, the SHACL representations are used both as the enabler for validation purposes and as a further extension to the schema description, for instance, for cardinality restrictions.

The IDS IM is independent of concrete application domains and thus does not provide terminology for the *content* of data resources. However, as the IDS encourages interoperability and extensible ecosystems,

[20] https://github.com/International-Data-Spaces-Association/InformationModel/tree/master/testing

it encourages the use of RDF and domain ontologies for Representations (cf. [236] for a sample scenario using a taxonomy of steel grades). In this context, it is desirable to include information about the domain-specific semantics and, similarly, the structure of content into the metadata of a Resource or some of its Representations – for example, to be able to retrieve more relevant data resources. To this end, the IM reuses VoID, the Data Cube Vocabulary, and SHACL[21], as detailed by examples in the GitHub repository.

4.3.4 Evaluation

The feasibility of the data model and the underlying concepts are hard to compare against different models independent of its usage environment. Its quality is determined by its suitability to cover the demands of the implementation scenarios. Therefore, this section gives a brief overview of common use cases in which the IDS IM enables semantic interoperability in data spaces, following the assumption that a broad adoption is the best quality measure for this kind of resource. The adoption processes, in general, are organized in five main verticalization initiatives, which map the generic IDS specifications with the domain-specific requirements. These initiatives involve the industrial manufacturing community, which is strongly related to the Plattform Industrie 4.0 and the Industrial Internet Consortium, the medical, energy, and material data space, as well as IDS in Smart Cities. In addition, at least seven commercially driven implementation processes are known to the authors. For instance, the public tender on re-implementing the German national mobility data platform explicitly enforces the IDS specifications[22] to ensure a self-sovereign landscape of equally empowered participants.

The purpose of the IDS IM is to act as a reference ontology for trustworthy, data-driven architectures. It is a cornerstone of any IDS-related implementation and thus used in all core IDS scenarios, related publicly funded projects, and impacts several industry platforms. The IDSA highlights 14 real-world use cases, the majority of them being realized with an investment from companies and contributing to their business success; furthermore, 10 EU research projects alone involve the IDSA (plus several of its members).[23] On a more technical level, an IDSA-internal overview[24] of 14 available implementations lists 11 implementations that explicitly support the IM. Further adoption of the IDS IM among component developers is fostered at the quarterly IDSA Plugfest[25].

Technically implemented trust and data sovereignty are at the heart of the IDS. Unambiguous description of usage restrictions and definition of the required attributes are therefore one of the most critical use cases of the IDS IM. As the vocabulary presents the shared understanding of all involved parties – combining the different domains to one consistent ecosystem – it connects their security, certification, governance, and interoperability models with each other.

Key challenges in the context of data usage control are the formal description of permissions and obligations. In the example, the Business Intel Inc. is able to present its intended restrictions in terms of a machine-readable policy (cf. R3). The Open Digital Rights Language (ODRL [108]) provides the terms and concepts for these statements. The IDS IM further details these constructs and defines their implications, focusing on their publication, negotiation, acknowledgment, and enforcement. These additional steps enhance the solely *descriptive* ODRL vocabulary to legally *binding and enforceable* statements. Thus, the IDS IM not only allows to state permissions, e.g., that an Asset can be read

[21] `https://www.w3.org/TR/{void,vocab-data-cube,shacl}/`

[22] https://www.evergabe-online.de/tenderdetails.html?0&id=322425

[23] https://www.internationaldataspaces.org/success-stories/

[24] Document accessible to IDSA members via https://www.internationaldataspaces.org/community/; latest version of 2019-10-16.

[25] https://www.internationaldataspaces.org/get-involved/#plugfest

Listing 4.5: Exemplary policy (*ids:ContractOffer* class; cf. Fig. 4.12) that grants read access to members of the Data Analyst company.

```
_:DigitalTwinDataOffer a ids:ContractOffer ;
   ids:permission [ ids:target _:Robot ;
      ids:action idsc:READ ;
      ids:constraint [ ids:leftOperand idsc:USER;
          ids:operator idsc:MEMBER_OF;
          ids:rightOperandReference <http://data-analyst.com/> ];
      ids:postDuty [
         ids:action [ ids:includedIn idsc:COMPENSATE ;
            ids:actionRefinement [ ids:leftOperand idsc:PAY_AMOUNT ;
               ids:operator idsc:EQ ;
               ids:rightOperand "5"^^xsd:double ] ] ] [...] ] .
```

(*ids:Permission* to *idsc:READ*) by certain users, but it can also express them in decidable terms for usage control engines such as MyData[26]. These tools independently evaluate the agreed usage policies and, for instance, grant or deny access to individual resources. They are the foundation of a *Contract* specification. Modeling usage policies, contracts, and the mappings between declarative and technically enforceable policies is a crucial prerequisite for the implementation of the IDS value proposition to maintain the complete *sovereignty* of data owners with regard to their content.

Listing 4.5 shows a policy for the Digital Twin as a data resource. The IDS IM is the only vocabulary to cover the actual enforcement of usage restrictions. Established standard languages, for instance XACML, only focus on access control or, as for instance ODRL, only allow the description and exchange of policies. The IDS IM closes this gap with detailed instructions on how to interpret each attribute, how to resolve statements and how to relate given policies to a system environment [21]. This is one aspect of solving R4. Listing 4.5 further shows how R2 (Business Value) is expressed. The *postDuty* clause describes and enforces a compensation for using the dataset, thereby combining business and data security statements in one representation.

Furthermore, the clear semantic of the allowed Action (READ) tells every interested buyer that the usage in its own IT landscape is covered (R3.1: describe intended usages), while any further distribution or reselling will and must be prohibited by the Usage Control Framework (R3.2: prevent unintended usage). The contract gives the Business Intel Inc. the tool to enforce its business model in the technical landscape of the customer (R4: control over the complete lifecycle), which must also be supported by the execution environment of the customer. This is ensured by the signed certification claims and can be checked on the fly.

4.3.5 Summary: A Data Model for IIoT Ecosystems

The IDS *Information Model enables data space ecosystems* (CO2) with a strong focus on supporting data sovereignty. The model supports the development, documentation, and usage of different representations for various groups of stakeholders. The model is available openly on GitHub and comprises the patterns and features necessary to describe and implement digital sovereignty in a federated ecosystem. It shows how semantic technologies can be enhanced with security and trust to pave the way for enforceable, self-determined, i.e., sovereign data management across organizations. The comprehensive view on the challenge of addressing data owners' legitimate concerns while enabling productive data usage by other parties is a requirement for upcoming data-driven business cases.

[26] https://www.mydata-control.de/

Following the described contribution methodology, the IM is continuously evolved with industry stakeholders via the IDSA. This ensures that it is in line with emerging requirements of data ecosystems concerned with maintaining data sovereignty down to implementation specifications. Thus, the IM also promotes Semantic Web standards in disciplines where there is little awareness so far.

The data model, in combination with the reference architecture [35], defines a generic ecosystem with multiple roles and infrastructure services. The proprietary, vendor-specific details and the heterogeneous IT landscape are encapsulated and thereby hidden for the communication partners. This procedure relieves the other participants to understand the *Design Decisions* (CH4) behind the existing IT environments of potential partners and simplifies the establishment of data interactions.

The data model also defines the constructs and terms to express data usage restrictions. Together with other activities in the IDS, namely the evaluation and certification of trustworthy software components and the delivery of practical access and usage control engines, the foundation for applicable *data sovereignty* (CH6) solutions. The according definitions to unambiguously define data contracts are further detailed in the next section.

4.4 A Usage Control Language for Data Sovereignty

The requirement of the IIoT to enable trustworthily, not harming data exchange patterns demands an appropriate vocabulary to formulate and exchange the intended restrictions. Fundamental for every serious business case is an in-depth understanding of the dependencies and implications of *data ownership*. In general, the legal possession of information is not possible, distinguishing data businesses dramatically from traditional economic activities based on physical goods. Still, trading data is already an everyday business activity and therefore must be based on a solid foundation. One exception appears when personal data is affected, where the legal perspective is more or less defined in the EU through the GDPR regulations.

In addition, creative work like newspaper articles or scientific papers is protected by the copyright of their authors. Industrial data, which to a huge degree is neither creative work nor directly related to an individual person, for instance, sensor streams or production plans, are by default not covered through a legal ‘ownership-like’ mechanism. Still, the freedom to agree on custom contracts enables the expressiveness of binding intentions and obligations. Therefore, specific contracts have to be created whenever a provider wants to ensure getting a) any kind of compensation and b) restrict the further usage of his data. As mentioned, no explicit ownership of data is possible, and distribution of digital data is usually at no (technical) cost. Therefore, any data provider has the interest to limit the dissemination of its business asset even after it was bought and delivered.

Following this argumentation, a vocabulary to describe intended and unintended usages of data resources is necessary. The Open Digital Rights Language (ODRL) [108] is a W3C recommendation providing terms and concepts for permissions of digital resources. However, as the focus of ODRL goes beyond the regarded IIoT use cases of this thesis, the definitions are too general and not specific enough for the regarded scenarios. The IIoT Digital Twins do not only need to state permissions but also must formulate them in patterns that are as decidable as possible. For instance, it is already possible to state that an Asset must not be distributed (odrl:Prohibition to odrl:distribute) to external entities using plain ODRL terms. Still, IIoT Digital Twins also need to know how and where to enforce this, how to identify external parties, when the prohibition is valid, and so on.

The requirements and technical solutions of describing, evaluating, and enforcing data ownership in self-sovereign ecosystems have been examined [21] [22]. The gained insights are explained in the following.

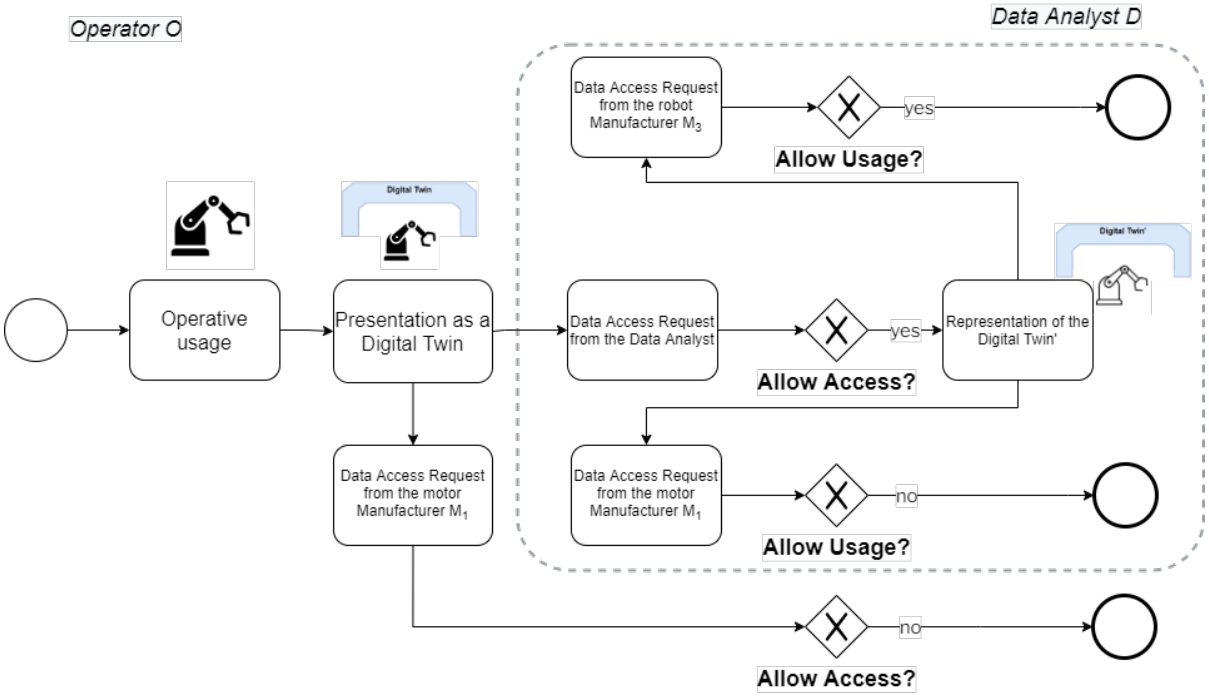


Figure 4.13: Data Access and Usage decisions from the operator perspective.

4.4.1 Introduction to a Usage Policy for IIoT Digital Twins

One of the most critical concerns of IIoT business cases is the ensuring of data security and privacy. Part of that is the protection of business-critical information and know-how. The desired opening of interfaces, systems, and devices contains the risk of losing control of these resources. The presented contribution in this section targets the challenge by outlining ways and processes how digital data can be exchanged through the Semantic Asset Administration Shell, ensuring that the containing information stays protected. New mechanisms are proposed which, relying on current standards, integrate the concept of data sovereignty from Section 4.3 and incorporating the conventions of the Semantic Web.

4.4.2 The IIoT from a Data Sovereignty Viewpoint I

The section regards the scenario from a data sovereignty viewpoint to outline the ideas and approaches for an integrated Usage Control of an IIoT Digital Twin (cf. Fig. 4.13). It extends the descriptons from Listings 4.4 and 4.5. The Operator O collects static master data together with sensor observations of an operating device inside the Digital Twin. It mandates the Data Analyst D to access the AAS instance and compute performance KPIs and failure predictions in order to optimize its production processes. O is also willing to share these insights contained in the Data Analyst's copy (Digital Twin') with the robot manufacturer M_3, as early information on potential material fatigue can help him improve the gripper arm. However, it is crucial for O to prohibit any access of other participants in the supply chain, in particular M_1, as they closely cooperate with O's competitors.

Figure 4.13 outlines the information flow in this simplified setting. The Asset, enriched to a Digital Twin, provides static master data and dynamic sensor streams. This interaction is managed by an access control engine deployed directly at the hosting system at O. The Data Analyst, as an intermediary, requests the data and enriches it with the result of its prediction algorithm. Note that this step results in a

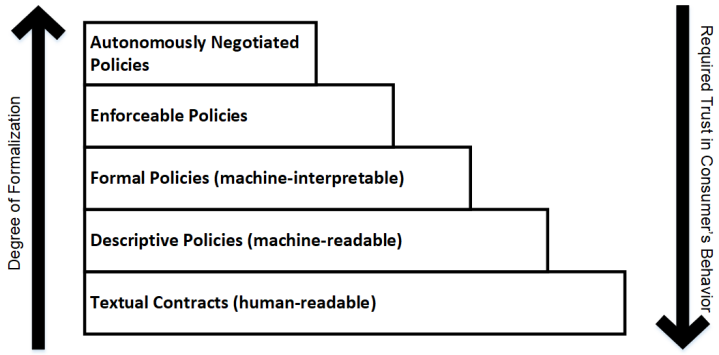


Figure 4.14: Stairway of Usage Policy Categories by required formalization.

newly created Digital Twin', which is not directly located in O's network but on the Data Analyst's server. The uniform representation conforming to the Digital Twin metadata model allows M_3 to automatically request and understand the Digital Twin', independently of the data model used internally by the Data Analyst.

However – as the Digital Twin' is now stored at the Data Analyst – the contained information is not under the control of O anymore but still contains his critical data. While O can prohibit download requests from M_1 when targeting its own server, it has no influence on the decision for the Digital Twin'. Therefore, the Data Analyst needs to evaluate the access requests from both M_1 and M_3 according to the formalized usage restrictions originating from O. The required instructions and descriptions must be contained in the Digital Twin, and through this also in the Digital Twin'.

4.4.3 Policies and Machine-readable Restrictions

The expressiveness of the policies does, in general, not depend on the domain of their target Asset but on the degree of formalization and inherent complexity. Figure 4.14 illustrates a proposal for an according categorization scheme. A human operator can understand but also formulate complex constructs, quickly creating insolvable issues for any current AI by modeling clauses outside of its previously configured application domain. A control language stretching an unrestricted space is therefore not beneficial, as it requires direct human intervention each time a (new) policy is regarded. In order to cope with most of the relevant use cases, a relatively small number of standardized feature dimensions (counting, time, space, role/membership cf. [21]) is sufficient.

Those dimensions need to be unambiguously defined, limit the variety of interpretations and specify implementable logic for their evaluation. While, for instance, XACML or the security module of the AAS and the general IDS Usage Contract classes provide a structure, the implications for evaluation systems are not yet sufficiently defined. In the presented example, the Data Analyst must not only be able to request and receive the Digital Twin but also understand and evaluate the clauses which are bound to its usage later on. While this is solvable in point-to-point scenarios by manually examining the contract semantics and configuring the target environment, a truly federated IIoT framework is not reached yet. In scalable use cases, the target environments need to have the necessary autonomy to ensure the compliance of the contracts. This autonomy needs to be controlled and verified to the data sovereign, leading to the following finding, as the Usage Control of any IIoT Digital Twin can only be as reliable as its executing system:

Statement 6 *A reliable end-to-end Usage Control scenario for Digital Twins requires trustworthy systems. Their state needs to be verifiable through certification processes, cryptographic signatures, and*

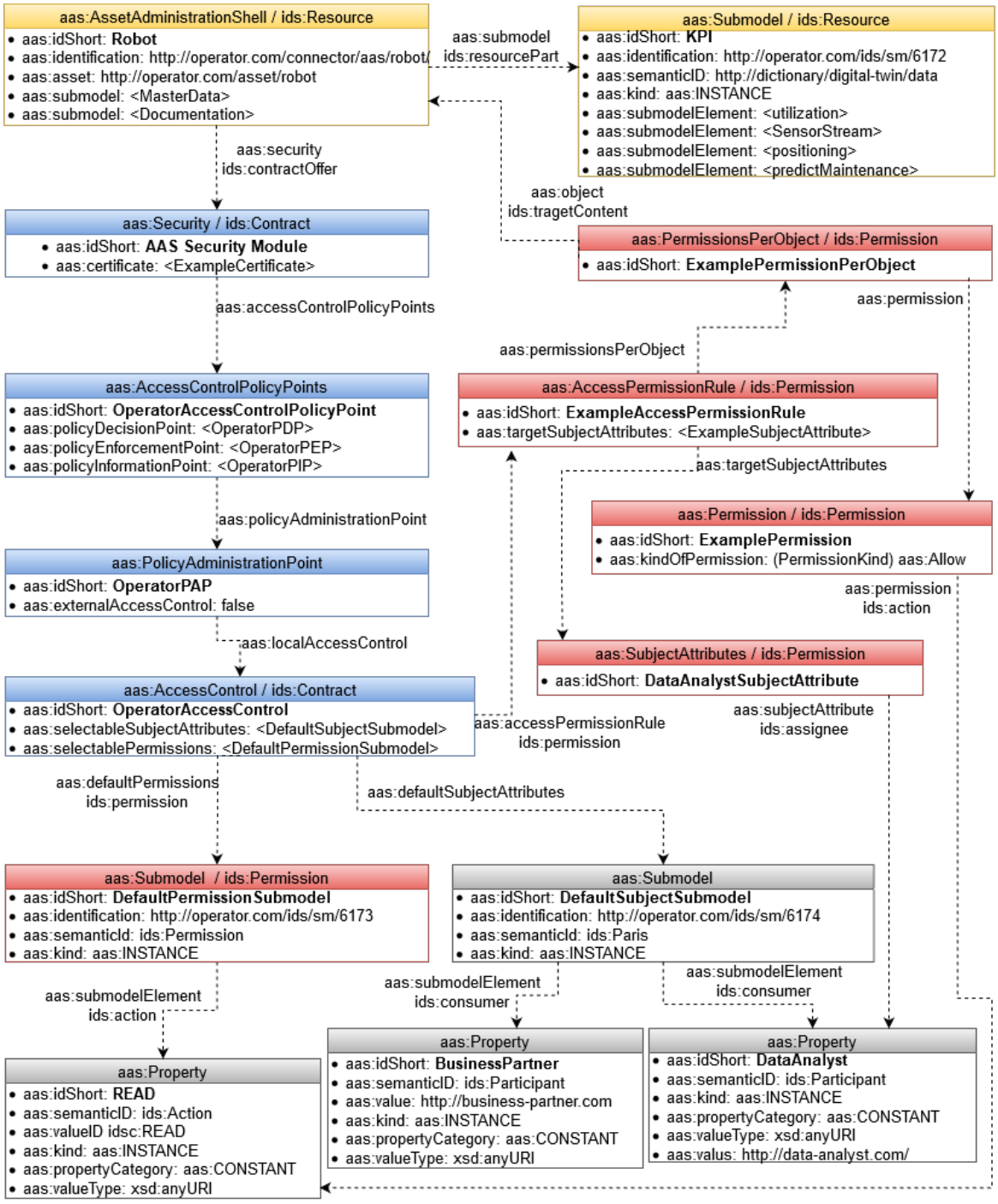


Figure 4.15: Basic model of a Digital Twin with Usage Policies according to the SAAS and IDS specifications.

controlled environments. As soon as the Digital Twin leaves such a controlled environment, the Usage Control commitments become obsolete.

Figure 4.15 shows the example expressed as an Asset Administration Shell snippet focusing on the description of the usage restrictions. The yellow top classes represent the actual Digital Twin and the features of interest. According to the metadata model, those are represented by the AssetAdministrationShell and Submodel classes. The core data control sections are outlined, starting with the Security to the AccessControl class (blue), followed by the definition of relevant entities and interactions (grey). The red Permission-related classes close the circle and link back to the actual protected features. Note that the same pattern can be expressed through a Usage Contract. The respective classes and properties are directly added but in different namespaces (*aas* vs. *ids*).

The AAS is driven by the understanding of the host as the authority to define security rules. The focus is twofold, (a) to describe the requesting parties (called 'subjects') and their relations to certain attributes ('objects'). Furthermore, the relevant subcomponents of the control system are explicitly modeled (b).

The IDS aims to provide such a controlled, trustworthy ecosystem. While making no difference in terms of the nature of the exchanged data, the IDS Connectors ensure a certain degree of control through certified execution environments, partly independent of the operating organization. Combining the AAS specification about the data itself and the IDS as the surrounding ecosystem, several of the mentioned challenges can be addressed. Therefore, the IDS locates the definition of those Policy Points in the responsibility of the hosting system, without being mentioned in the policy at all. While this behavior allows for the shipment of both the AAS and the Policies, the interpretation of the described policy is harder and requires higher degrees of formalization. For instance, as soon as the AAS' is evaluated by the Data Analyst, its enforcement engine needs to interpret the Policy in accordance with its own local environment. This implies the need to guess appropriate Policy Points independently. This very challenging task is further complicated by the prohibition of the direct manual configuration since the hosting system must ensure the Data Producer's interests and resist these kinds of manipulations.

Statement 7 *Derived Digital Twins must transport the original usage constraints with them. Downstream activities must be compliant with all previously stated restrictions.*

This is difficult to maintain, as at some point in any workflow, the input components merge to a new product, with the assembler (previously consumer) as the new data sovereign. A simple example is presented by the manuals shipped with each component of a machine. The Manufacturer in the example combines not only the device components into its product but also the manuals and safety documentation. His customers do not get a single document set for each and every component but one single file. The resulting device's documentation, even though reusing content from several suppliers, is the intellectual property of the Manufacturer. Defining at which stage a new product appears – and who has the authority over its further use – is obviously one of the crucial questions. This is especially true for digital data in general and Digital Twins in particular. Despite the relevance of this challenge, it is still open for clarification. There is no clear answer in the European legislation, and decisions of the respective high courts are also missing. As far as this thesis is affected, the particular question is regarded as out of scope and open for further research.

Independent of the open issues regarding the authority on data and information, the structure, and necessary parts of a Usage Contract can be defined independently. In this sense, a contract is a set of associated rules, which affect the relationship between two particular parties. It is noteworthy that this model ignores the existence of additional stakeholders, for instance, the government/legislation or the society in general. In particular, it is assumed that only the two named parties are relevant and sufficient to model the regarded scenarios. Relations and obligations imposed by external actors, for instance,

fiscal authorities for appearing taxes or competent courts for potential legal actions, need to be covered in framing declarations, which both parties need to accept beforehand. The details of these declarations, which might be seen as similar to the commonly used *terms and conditions* of human-centered digital business models, remain open tasks for the respective research domains.

Definition 4.4.1 *A **Usage Contract** is a set of Rules $\{R_1, ..., R_n\}$ between exactly two parties P_0 and P_1. A **Rule R** is defined by its type T (Permission, Prohibition, Obligation), an **Action** A on a target object O, and a set of Constraints $\{C_1, ..., C_m\}$.*

Most varieties are contained by the Constraint clauses C_j^i. A Constraint models a statement, which needs to be checked at the evaluation time of the contract. That means that a Constraint presents an instruction that the enforcement system must be able to interpret at runtime and that needs to evaluate to a boolean value.

Definition 4.4.2 *Each **Constraint C** is a condition under which the Rule is valid and defined through its **Feature of interest F** the comparing Operator $\diamond$, and a **realization F^0** of the feature of interest:*
$C : F \diamond F^0 \rightarrow \{true, false\}$

Combining both definitions brings us to the following model of a Rule:

$R := T \times O \times A \times C^n = T \times O \times A \times \{F, \diamond, F^0\}^n$

This pattern is promoted in similar ways by nearly all Access or Usage Control vocabularies or frameworks. Regarding the example, the Manufacturer (P_0) allows (R_1 =Permission) the Data Analyst (P_1) to access (A_i =READ) its AAS (O) until the end of 2020 (C: TIME $\diamond_{BEFORE}$ 31.12.2020). A proper Usage Control needs a definition of all mentioned sets. A set of actions is proposed by ODRL 2.2 but lacks clear specifications on how they have to be treated. For instance, the action ANONYMIZE does not include the required algorithm, neither does it state whether the P_0 or P_1 are the responsible parties to ensure it.

Two additional aspects add further dependencies and complexity to the evaluation of rules. The Constraints can not only be linked to Permissions, for which they model the respective preconditions. Constraints can also be used together with Actions, which results in customized action refinements. In addition, obligation rules are not conditioned through Constraints. Instead, Constraints impose instructions for Obligations. Both aspects are not regarded in this thesis to keep the readability of the contract models. Nevertheless, the related contributions, in particular the IDS Information Model and its Usage Contract specifications, contain both features and explain their implications.

Statement 8 *There is currently no adequate definition for Actions, Features of Interest or their datatypes as well as the required Operators.*

Table 4.4 contains the identified core building blocks of a data protection description. Certainly, the list can be continued easily with more feature dimensions. Still, this list covers a sufficiently high amount of relevant scenarios in order to serve as a valid starting point. It is also worth mentioning that no single implementation can be expected to implement all described feature dimensions. The community agreement on the proposed list allows the description of capabilities and the usage of commonly known clauses. It also enables the involved systems to state missing support to an opposite party explicitly. For instance, the Data Consumer can state that its Digital Twin is able to enforce logical, arithmetical, and temporal usage restrictions but not spatial features, and therefore can adhere to a policy limited on the clauses of the first ones.

4.4.4 Dimensions of Usage Control Aspects

The general syntax of Usage Control contracts, as proposed in the section before, enables systems to form expectations about the order and relations of their clauses and parts. The formulation in RDF further defines serialization formats and allows their exchange via messages. Nevertheless, the involved systems also need to understand their content, their implications, as well as the limitations of their own capabilities. A pure syntactic or data-oriented view on the contracts directly leads to situations in which undesired clauses are acknowledged.

For instance, in the scenario of Section 4.4.2, the Data Analyst could ask for a high fee for every time the Business Partner even reads his results. If the Business Partner's Usage Control system receives the contract and acknowledges it just because the parsing was successful, the Business Partner is at risk of losing a significant amount of money. The creator of the contract can regard a proper response message as an acceptance of the conditions, therefore implying legal obligations for both. The involved parties obviously need to have the same understanding not only about the syntax and data formats but also the used terms and their meaning. Furthermore, standardized operation semantics is required to avoid ambiguity also on the interaction level.

Remark *The contents of this section have been created as part of the IDS Usage Control activities and incorporated in the IDS Usage Control Specification and the respective module of the IDS Information Model (Section 4.3). The quality of the content was therefore checked by the IDS community and not by a peer-review process.*

Table 4.4 presents the sorted list of agreed feature dimensions through the IDS community. Higher ranked features are assumed to be slightly more relevant or more commonly used than lower-ranked ones. For instance, nearly every Usage Control use case requires basic logical statements (something is *true* or *false*), while geospatial relations are less important. In addition, higher-ranked features tend to be better understood and their definitions more mature than lower-ranked ones. Due to their wider usage, more research works have treated them and examined their implications. This assumption is further supported by the observation that the 'Network Space' and 'Events' are crucial for many IDS use cases. Still, no widely accepted modeling scheme or data format can be found in the literature.

This varying complexity of the different features is a further gap in the literature. No other research work currently available aims to examine the different requirements on an abstract level. It is essential to understand the different kinds of input requirements for each dimension. For instance, basic temporal comparisons like *X is before Y* or *X is after Y* are, given that Y is a known and fixed value, dependent on the one free variable X. Questions like *X is in the three-month testing period*, however, requires not only the given period length but also its starting date. Consequently, two input factors need to be known before the statement can be evaluated. The column *Degree of Freedom* (Degr.) contains the analysis for each feature dimension and gives the requirements to the actually usable contract clauses.

Definition 4.4.3 *A **Degree of Freedom** is the minimal number of different input parameters that are necessary to define a statement of the respective Feature of Interest unambiguously.*

It is important to understand why certain contract clauses, in particular Constraints in this work, are complete and others not. Continuing the example of a testing period, a clause like *use only for three months* is not sufficiently defined (start date of the period is missing) and must be rejected by every enforcement system. Understanding such requirements, and linking them to the defined features of interest, is the only way to prevent unsolvable situations at the evaluation time of a contract.

Similar important to the Degrees of Freedom are the anchors of each scale. Intuitive examples are the DateTime scheme defined by XML Schema 1.1. The anchor is the Coordinated Universal Time

Feature of Interest	Degr.	Reference (Anchor)	Datatype/Vocabulary
Logic	1	absolute (Boolean algebra)	xsd:boolean
Set	1	relative	rdf:List
Arithmetic	1	absolute (0)	xsd:double
	2	relative (counter)	xsd:double
Text	1	absolute	xsd:string
	2	relative (pattern matching)	xsd:string incl. RegEx pattern
Time	1	absolute (UTC)	xsd:dateTimeStamp
	1	relative	xsd:duration
	2	absolute (UTC)	Interval (xsd:dateTimeStamp × xsd:dateTimeStamp)
SpatialEntity	1	qualitative	Wikidata Entities
Geometry (Point)	2	relative (local origin)	Coordinates (xsd:float × xsd:float)
Geometry (n-edge Polygon)	2^n	relative (local origin)	Point ×...× Point
GeoSpatial (SpatialThing)	2	absolute (WGS 84)	WGS84 Geo Positioning
GeoSpatial (n-edge Polygon)	2^n	absolute (WGS 84)	SpatialThing ×...× SpatialThing
Organization	1	hierarchical	ORG Ontology
Org. & Membership	2	hierarchical	ORG Ontology
Org & Memb. & Site	3	hierarchical	ORG Ontology
Network Space	1	IP address (relative)	xsd:string
	1	domain (absolute)	xsd:string (DNS scheme)
	1	MAC (relative)	xsd:string
Events	?	relative	Cloud Events
...			

Table 4.4: Building blocks for Feature Dimensions (incomplete list).

(UTC) based on the Gregorian calendar. While this is a universal, worldwide identical anchor, this is not necessarily the case for other features. Organizational relations, most notably the role of an employee in the organizational hierarchy, are different in every company. It is therefore impossible to define such anchors at the design time of a system, and the related information needs to be supplied at the evaluation time. Somehow in between are clauses about geolocations. While there is a widely used coordinate system (WGS 84) based on the GRS 80 reference ellipsoid, other coordinate systems like for instance ETRS89 can have deviations and need to be mapped beforehand. That implies that – even though coordinates are syntactically valid, and the evaluations on top can be calculated – also the anchors must be the same to gain the same results. Relying on a shared reference system, further extensions like the introduction of Points Of Interests and Areas Of Interest as shown by Dadwal et al. can be integrated [237].

The last column of Tab. 4.4 contains the specified datatypes for simple types and the used reference for complex ones. Whenever possible, the RDF recommendations have been used. Where necessary, well-defined reference ontologies provide the clauses, for instance, the ORG ontology[238] for organizational relations and roles. The table only represents a first common basis that will certainly need extensions in the future.

4.4.5 Operators for Usage Control Rules

The presented feature dimensions from the previous section define the space in which the Usage Contracts can be applied. The individual clauses in this space are built using operators to create meaningful statements. These operators are limited to two input parameters per clause. While certainly more parameters are possible without affecting the applicability of the operators in general, this limitation reduces the implementation load for applications and has been proven as a suitable trade-off between expressiveness and implementation complexity.

Definition 4.4.4 *A **Usage Control Operator** $\diamond$ is a Binary Operator identified by a URI, deriving a Boolean value from exactly two input parameter sets A and B:*
$\diamond : A \times B \rightarrow \{true, false\}$

An operator is therefore defined on the input dimensions A and B and its derivation function. The identifier, a character sequence forming a valid URI, is required for the reference in the rules. As far as IIoT use cases are affected, URIs in the HTTP scheme have proven their applicability and are the preferred pattern.

The IDS has determined that the following operators are necessary to implement Usage Control scenarios:

Table 4.5 aligns the operators with their corresponding feature dimensions. The previously mentioned clause – *X is before Y* – can be modeled using the $\diamond_{BEFORE}$ operator and X and Y as time instances. Using graph patterns, one can note this in the following form:

$?x \diamond_{BEFORE} ?y \rightarrow \{true, false\}, ?x, ?y \in TemporalInstance$

A definite value using a dateTimeStamp is assigned for $?y$, for instance *"2020-12-31T23:59:59+01:00"^^xsd:dateTimeStamp*, or any other unambiguous reference to a TemporalInstance. Reflecting its position in the statement and according to the ODRL terminology, $?y$ is called a *RightOperand*. The *LeftOperand*, $?x$, may also contain a fixed dateTimeStamp, for instance *"2021-01-01T00:00:00+01:00"^^xsd:dateTimeStamp*. In that case, however, the clause is trivial and therefore uninteresting, as it always evaluates to *true*. More relevant for real-world use cases are references to the current point in time, for instance, to express that a *now* must be *before* a certain time instance as the contract shall limit the usage time. It is necessary to define a set of references, which the Usage Control system is aware of and can use to evaluate the applicability of the contracts.

This demand is reflected by the list of *LeftOperands*. *Now* is represented through the *POLICY_EVALUATION_TIME* instance. Additional instances, like *PAY_AMOUNT*, allow the description of usage fees or make references to certified security characteristics (*SECURITY_LEVEL*) of the regarded application. Combining all these parts allows the data sovereign to state its intentions and gives the downstream consuming systems the information on how they need to handle the incoming data. The according behavior of their applications can be certified and proven to remote parties using trustworthy tokens. The IDS promoted Dynamic Attribute Tokens (DAT) contain such claims, combining the results of certification processes with cryptographic signatures. The result is an IIoT ecosystem, where usage intentions can be expressed, exchanged, and implemented on a technical level.

4.4.6 Summary: Sovereignty of Digital Twins

The outlined vision to merge Usage Control and *Data Sovereignty* (CH6) with the Asset Administration Shell shows a way how to create an interoperable and at the same time protected Digital Twin for IIoT purposes. The provided assumptions and statements are intended as starting points for further discussions.

Feature of Interest	Compared Values	Operators
Logic	Boolean ⋄ Boolean	EQUALS, NOT
Set	Element ⋄ Collection	IN
Arithmetic	Double ⋄ Double	EQ, LT, GT, LTEQ, GTEQ
Text	String ⋄ String	STRING_EQ, STRING_CONTAINS, STRING_IS_CONTAINED
RegEx	STRING ⋄ RegEx	MATCHES
Time	Instant ⋄ Instant	BEFORE, TEMPORAL_EQUALS, AFTER
	Instant ⋄ Interval	BEFORE, MEETS (= STARTS), TEMPORAL_EQUALS (= CONTAINS), MET_BY (= FINISHES), AFTER
	Interval ⋄ Duration	LONGER, LONGER_EQ, AS_LONG, SHORTER_EQ, SHORTER
	Interval ⋄ Interval	BEFORE, MEETS, OVERLAPS, DURING, STARTS, TEMPORAL_EQUALS, CONTAINS, OVERLAPPED_BY, MET_BY, FINISHES, AFTER
	Duration ⋄ Duration	LONGER, LONGER_EQ, AS_LONG, SHORTER_EQ, SHORTER
GeoSpatial	Point ⋄ Point	DISJOINT, SPATIAL_EQUALS
	Polygon ⋄ Point	DISJOINT, SPATIAL_MEET, CONTAINS
	Polygon ⋄ Polygon	DISJOINT, SPATIAL_MEET, SPATIAL_OVERLAPS, INSIDE, COVERED_BY, SPATIAL_EQUALS, COVERS, CONTAINS
Organization	User ⋄ Organization	MEMBER_OF
	User ⋄ Org. × Role	HAS_MEMBERSHIP (includes MEMBER_OF)
	User ⋄ Org. × Role × Site	HAS_SITE (includes HAS_MEMBERSHIP)
Network	Network Location ⋄ Network	INSIDE_NETWORK (?), ...
...		
Event	Instant ⋄ Event	?
...		

Table 4.5: Binary operators for Usage Control Constraints. Operators represent URIs with the *idsc:* namespace prefix.

A consolidation of the thereby affected approaches and concepts is necessary, and a trade-off between formalization and expressible details on the one hand and adoption on the other is indispensable.

Still, the demand for more and more autonomously acting systems enforces overhead in terms of data models and implementations. The Usage Policies explained in the AAS show how the different specifications could be combined to a comprehensive interpretation. The combination of identification and interaction patterns with the standardization efforts and domain requirements of manufacturers requires efforts from all parties. However, the result has the potential to disrupt the way digital information and data in the intersection of the physical and virtual world is regarded.

4.5 Summary

The contributions presented in this chapter – the standardized *data model for IIoT Digital Twins* (CO1, Section 4.2), the *data model for IIoT ecosystems* (CO2, Section 4.3), and its extension for data sovereignty (Section 4.4) – define a presentation layer for the relevant aspects of Digital Twins and their surrounding. All three parts provide the expressiveness needed to describe the current view of Assets in the world and provide generic terms to cover further updates and thereby reflect the changes in the Assets themselves but also of their context.

The support of both contributions through well-established consortia ensures their adoption and continuous development. This is, at the time of writing this thesis, further reflected by the activities around the GAIA-X initiative for the Cloud Computing domain. Such applications might already serve as a supporting argument for coping with *Unpredictable Requirements* (CH1). At the time when the first versions of the Semantic Asset Administration Shell and the International Data Spaces data models have been created, the GAIA-X requirements have not been known. Nevertheless, the generic nature of both models allows their integration without changing their definitions.

As expressed in the previous sections already, the contributions are also applicable in Brownfield scenarios (CH2) and can be introduced at any point of time into an existing shop or office floor. Section 4.1 explains how current applications can be extended with the semantic descriptions and capabilities for on-the-fly orchestration. Especially the retrofitting of Assets and the therefore necessary subsequent addition of data to their Digital Twins is no problem with the presented data models.

Both the IDS as an IIoT ecosystem and the Semantic Asset Administration Shell require the explicit declaration of attributes and declarations. They force the providing organizations to annotate their Assets according to given standards, for instance DCAT or VDI 2770, but also extend them where necessary. This generic approach already guarantees the fundamental basis of available information and reduces the number of implicit *Design Decisions* (CH4). Further attempts to cope with this challenge are given by the generic infrastructure components, and their defined roles and capabilities that allow the establishment of IIoT connections at runtime.

The Usage Control Language of Section 4.4 expresses the foundations and limitations of descriptive usage contracts of the Digital Twin in IIoT ecosystems. The *Data Sovereignty* (CH6) of the Digital Twin, expressed as a Semantic Asset Administration Shell, in an IDS ecosystem is outlined through several predefined, automatically decidable dimensions that can further be extended with additional categories. These modules pave the way to end-to-end automatized data control processes in future applications while already presenting added value for each of the necessary development steps.

The self-descriptiveness, machine-readable characteristic of the two contributions in combination with their extensibility based on Semantic Web Technologies qualifies CO1 and CO2 to fulfill the requirements leading to RQ1 and therefore describe a suitable answer to how IIoT Assets need to be represented as Digital Twins. Their adaption and the assured maintenance through their supporting organizations, as

well as their growing adoption in industrial use cases, indicates their suitability and their soundness. Building on this descriptive layer, the interaction of the thereby modeled Digital Twins needs to be examined further. This is targeted by the contributions of the next chapter.

CHAPTER 5

Interaction Patterns for the IIoT

Consistently modeled and commonly understood information models of IIoT Assets, as presented in Chapter 4, are the foundation for any kind of interaction and data exchange. Still, the used communication patterns, protocols, as well as their underlying assumptions and design decisions, also need to be standardized and aligned with the requirements of IIoT environments.

Interoperability in distributed systems is more and more regarded as the crucial challenge for IIoT networks. However, the mature specifications of the Semantic Web Technologies are not regarded by the dominant IIoT and Digital Twin specifications. In particular, the unpredictable and changing requirements (CH1) of future IIoT applications have to deal with proprietary, non-standardized interfaces of current and future Brownfield environments (CH2), which originate from the point-to-point connections of current control networks. The resulting heterogeneous communication patterns (CH3) make every update or change in the involved systems a cumbersome and error-prone task. As typical shop floors consist of a great variety of machines and devices from different manufacturers, their communication patterns rely on different, many times even conflicting, design decisions (CH4).

The contributions outlined in this chapter (cf. Fig. 5.1) target this problem and examine how semantic approaches and Web-inspired interaction patterns can be extended to enable IIoT operations with predictable behavior (CO3). The requirements of communications among IIoT Digital Twins and external applications are analyzed (CO4) and mapped to reusable interaction patterns with clear and transparent operation semantics. The presented set of guidelines and defined operations answers the second research question, as stated in Section 1.4:

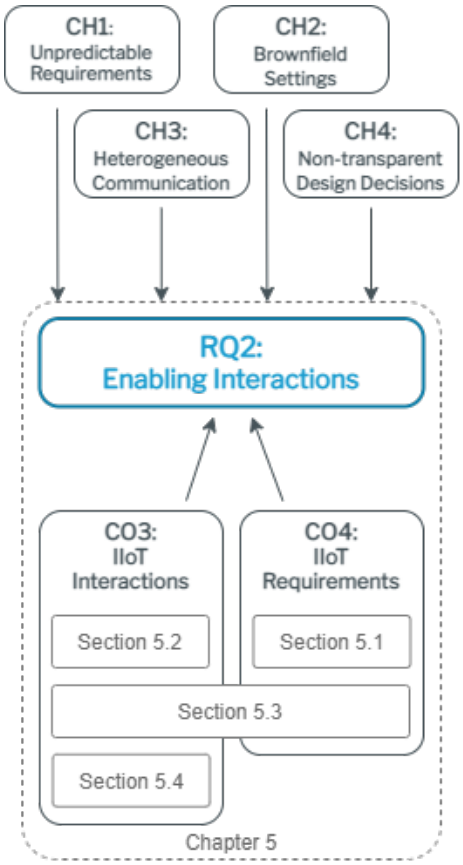


Figure 5.1: Contributions for RQ2.

RQ2: Enabling Interaction – How do IIoT Assets and their Digital Twins interact with each other in dynamic and unpredictable environments?

The contributions answering this research question have been published in an article in the journal *Future Internet* [23] and discussed in a series of peer-reviewed workshop papers co-located at well-known conferences [21, 25, 26, 34, 239]. Further presentations have been conducted at the SEMANTiCS 2016 conference as well as at the annual conference of the German Operations Research Society 2017.

The related work and the state of the community discussion for the contributions CO3 and CO4 are already presented in Sections 3.5, 3.6, and 3.6 of Chapter 3. The content of this chapter builds on these explanations and contains the following propositions. Section 5.2 outlines the proposition of how to integrate executable logic and a consistent interaction scheme (CH3) on top of RESTful Web interfaces and semantically described data and operations on state-based resources. This approach is then extended in Section 5.3 to *Brownfield Environments* (CH2), incorporating the dominant IIoT protocols into a generic framework of operations. The Web-native representation of Linked Data Platform Containers and Resources is combined with on-the-fly composition and orchestration mechanisms for IIoT Assets and cloud services.

Section 5.4 uses the previously elaborated concepts and combines them with the approaches of the Solid initiative to interactive Digital Twins. The focus of Solid, the human user and its data as a social entity, is transferred and mapped to the IIoT Digital Twin and its characteristics. Especially the transparent interaction paradigm important in IIoT networks is aligned with the atomic operations and resource calls of Solid to reduce the interpretation effort of the clients which *Design Decisions* (CH4) determine the behavior of the interfaces.

5.1 IIoT Requirements Model

Many publications from research but also industrial organizations propose their own requirement analysis or categorization [74] [240] [4] [241]. Each of them has its justification in its regarded application environment. Still, one can extract a certain set of similar views and approaches. For instance, most publications distinguish between functional characteristics and information modeling aspects, promote the importance of security features, and connect a virtual representation with an Asset. However, the differences are still remarkable, and the frameworks are usually not directly convertible. For this analysis, the extensive models of Bassi et al. [242] and Kovatsch et al. [62] (cf. Fig. 2.6) are most suitable in order to explain the situation in the example as both models suggest a clear classification of requirements, outline a generic data model and propose protocol bindings based on a defined list of necessary interaction features. The third contribution, CO3, which is explained in this section, builds on these works but extends them to a comprehensive list of categories and criteria for the design and evaluation of IIoT solutions.

Still, neither the terminology, the structure of the underlying system architecture, or the data models are completely congruent. Nevertheless, a combination of both models is feasible and creates a reasonable structure for the further analysis. In the following, the relations of the respective requirements are outlined, using the layered model of Figure 5.2 and the similar one of Figure 5.4 as a simplified reference model.

The four main categories, *Functional* and *Non-functional* requirements, *Protocol Bindings*, and *Data Models*, group the targeted concerns into a basic structure. The also commonly used category of 'Composition' – sometimes also referred to as 'Orchestration' – of different components is explicitly not regarded. Other publications targeting this category and potential future work can explain how distributed architectures with Digital Twins can be organized and controlled independently, an open challenge for future work. In order to increase readability, each stated requirement is identified by its category identifier, for instance, 'F' for a functional requirement, followed by a number. This notation is used throughout

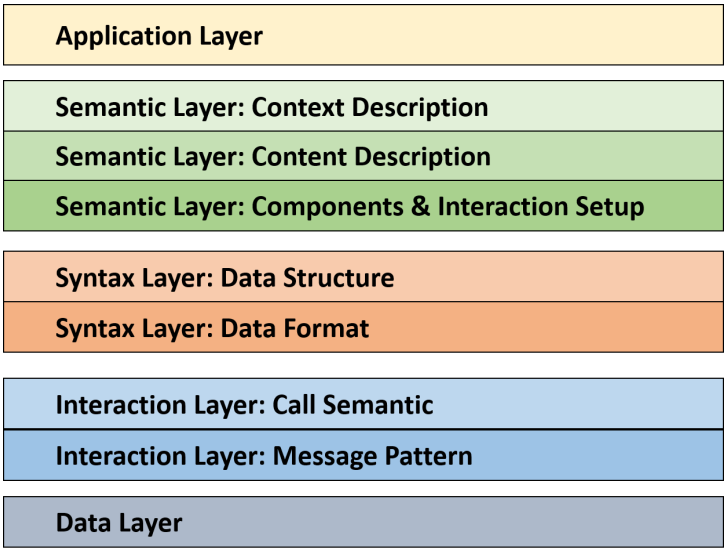


Figure 5.2: Simplified IIoT Layer Architecture

this thesis and is intended to follow better the mapping of the respective requirements with the respective system capabilities.

5.1.1 Functional Requirements

The functional requirements of an IIoT system collect the core functions and capabilities of a system to make it work. Following Bassi et al., this category frames "the system's runtime Functional Components, including the components' responsibilities, their default functions, their interfaces, and their primary interactions" (page 133 [242]). In this sense, these aspects are regarded independently of the protocol implementations outlined later, although the distinctions can certainly differ.

Kovatsch et al. specify a set of required principles for this category. Even though some of the mentioned aspects are actually non-functional characteristics (scalability or flexibility), a core set of necessary features can be extracted. Most relevant is certainly *interoperability* (F1) to manage the heterogeneity of devices and enable the exchange of digital data. Kovatsch et al. further distinguish Thing-to-Thing Interaction (F1.1), Thing-to-Gateway (F1.2), Thing-to-Cloud (F1.3), and Web Integration (F1.4), The abilities to *read and interact with resource states* (F2), manage notifications (F3), and invoke operations (F4) are also stated (cf. Fig. 5.3).

A particular challenge for IIoT frameworks is a durable and scalable *identification* (F5) mechanism [242]. A suitable method needs to be able to cope with the different existing identification patterns but also allow the seamless integration of new approaches. Basically, following the outlined example, the Manufacturer must have the necessary freedom to find a *globally unique* character sequence, which is *understood by sufficiently many* current and future systems and combines identification with the *provisioning of metadata* in federated environments. The Business Partner must be able to integrate the Manufacturer's identifier into its own systems without further adaptions. As central identity providers impose single points of failure but can also present undesired concentration of power in an IIoT network, *decentralized approaches* are necessary. Therefore, identification frames everything related to the ability to assign unique, non-clashing identifiers and to provide reliable and durable mechanisms to resolve the location of the (digital) entity by its identifier.

The RFC 4151 [243] formulates the following requirements for identifiers. Even though the RFC already restricts the scope to URIs and disregards other identifier types, the listed requirements give a good overview:

- "Identifiers are likely to be unique across space and time, and come from a practically inexhaustible supply.
- Identifiers are relatively convenient for humans to mint (create), read, type, remember etc.
- No central registration is necessary, at least for holders of domain names or email addresses; and there is negligible cost to mint each new identifier.
- The identifiers are independent of any particular resolution scheme." (RFC 4151, page 1)

The authors of the RFC further conclude that "UUIDs [...] are hard for humans to read" (RFC 4151 page 2 [244]) while Object Identifiers (OID) and Digital Object Identifiers (DOI) rely on central registration services. Decentralized Identifiers (DID) [245] or Uniform Resource Names (URN) are further approaches to identify information entities.

However, all mentioned approaches face the challenge of how a reader can find further information about the resource, given that it only knows the identifier itself. In the case of UUIDs, OIDs, and DOIs, the client would require initial knowledge of possible lookup services. The DID somehow encode their lookup infrastructure in their identifiers. Still, the client needs to know the location of such services in advance.

Uniform Resource Locators (URLs), and in particular URLs in the HTTP scheme, solve this challenge by relying on the DNS infrastructure for lookups. A commonly mentioned argument against URL identifiers, that DNS entries can change owners over time and therefore are not reliable, is only partly applicable. As this possibility is theoretically given, the DNS entries have become so important for any current organization that keeping control over them justifies nearly any cost. Consequently, the chance of disappearing or misused DNS entries might be in general regarded as similar or even lower than the chance of the disappearance of central lookup services like the one for DOIs or IODs. URLs, therefore, constitute a practicable and reliable mechanism for resource identification and lookup resolution.

Companies usually organize permissions and authority according to roles and the membership of certain teams. For instance, the Manufacturer's IT administrators usually have more access rights than its machine operators. The administrators are usually not allowed to execute cash flow-related operations, which instead are linked to the accounts of managers. This example outlines that a role-based permission model for *authorization* (F6) is a necessity for any real-world system. While Bassi et al. and Kovatsch et al. argue for extending this view to *access policies* on attribute level (F7), more fain-grained distinctions as imposed by a role-based model are a must-have.

Different from personal data, where the referent (the pod user or the person described by the data, for instance, through a social media profile) is controlling the complete dataset, two major differences appear. First, the risk of misuse of critical information does not decrease by the distance to the information source. For instance, the value of knowing a user's clickstream in an application is high for the operator of this application but already decreases for the cloud hosting company serving the application back end. However, a machine configuration might not be of value to any direct Business Partners but to the Competitors of the Manufacturer also collaborating with them. That implies that the data creator (Manufacturer) must not only consider who is *directly accessing* its provided information but also how this *is used* later on. This is also true to certain person-related information, for instance, credit card numbers, but obviously acceptable for most human users while companies in general regard such risks as fundamental obstacles for any data exchange at all.

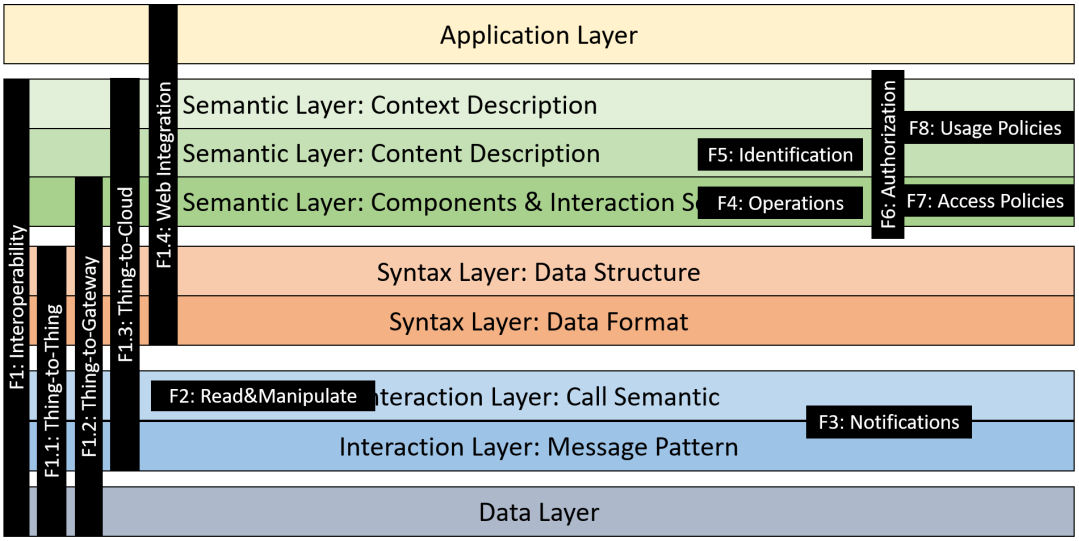


Figure 5.3: Overview of Functional Requirements positioned in the IIoT layer model.

The second crucial difference is the different protection through legislative rules. Person-related data is, especially in the EU through the General Data Protection Regulation (GDPR), protected by itself. The usage of names, addresses, or pictures is always bound to the commitment of the referred person. This is not the fact for business-critical data of companies. Know-how and intellectual property need to be protected through licenses or contracts. As soon as the concerned data left the controlled area, the damage can not be prevented anymore. Same as above, this can also happen to sensitive personal data. However, the frequent usage of social media indicates that the contained risk is regarded as acceptable for most individuals while the risk assessment of companies looks differently.

These characteristics make the hereby regarded scenarios fundamentally different from person-related data, consequently requiring a fundamentally different treatment. Furthermore, as stated by Otto et al., access control is only one component of a future-proven data control architecture. As IIoT applications are usually organized in workflows and complex communication networks, access control alone only observes the data exchange from one device to another. Further upstream usage of the data is not restricted anymore. Usage Control (F8) as a superset of access control is therefore required, especially in scenarios where different organizations pass on sensitive data in their supply chain [35].

5.1.2 Non-functional Requirements

Security requirements for IIoT applications are collected by many publications from the research community ([246][242]) but also from industrial consortia ([4][74][35], cf. Fig. 5.4). Different from Bassi et al. and others, security and privacy requirements are regarded as *non*-functional, as a non-secure system can still operate – even though the value to its users is quite limited. This is due to the understanding that non-functional features state *how* a system behaves, and acting in a secure manner is therefore grouped here.

Trustworthiness (N1) can be regarded as the root concern using this understanding [35][4]. In this thesis, the trustworthiness of a system is seen as its ability to perform as expected, which is strongly influenced by its environment and purpose. Still, the required or achieved trust level directly depends on the lower level characteristics but also on the regarded context. For instance, a weakly secured device in a controlled environment can have a sufficient trust level for its use case, whereas a more hardened one in

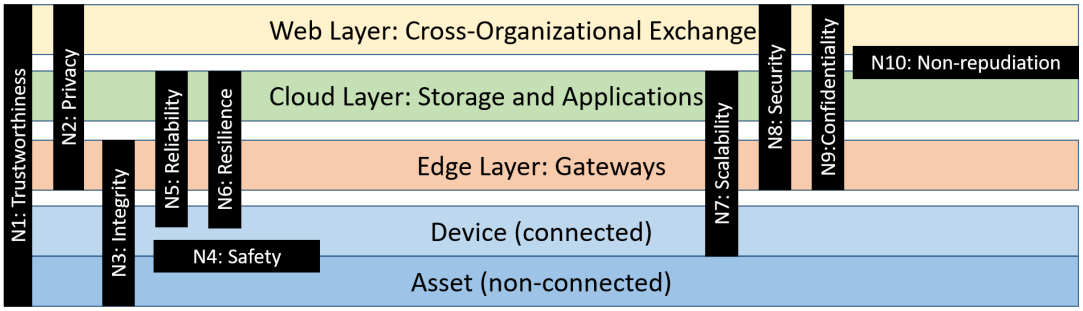


Figure 5.4: Overview of Non-Functional Requirements by their location in the network architecture.

an open network might still not be sufficiently trusted.

IIoT applications form real-world workflows and can physically affect machines and humans. Therefore, the reliability and trustworthiness of the devices and their behavior are crucial. Security in terms of *data privacy* (N2) of both personal and business-critical data, controlled *integrity* (N3) but also the guarantee of physical *safety* (N4) are fundamental requirements. *Integrity* ([246][4], N3) means that information is not altered. In addition, *reliability* (N5) in terms of expected accessibility ratio, responding times and general availability of a system [246] [4], and resilience (N6) against intrusion and denial of service attacks are crucial as critical processes may depend on the devices.

These traditional requirements of automated manufacturing now need to be extended as the shop floor loses its boundaries and the local communication networks become open to the global internet. Consequently, the very well-known requirements of Information Technology (IT) networks need to be addressed as well. *Scalability* [246] (N7) further adds the characteristic to grow and expand without facing sudden limitations easily. *Security* (N8) in the specific definition of Lin et al. is interpreted as the protection 'from unintended or unauthorized access, change or destruction' ([4] page 16), different from the previously mentioned more general understanding of security as an overall requirement, and contains the following aspects. *Confidentiality* (N9) frames ([246]) the secrecy of stored or transferred data. *Anonymity* [246] is related to that, hiding the identity of the related real-world entity of a data object also to the receiver of the communication. *Non-repudiation* (N10) [246] gives the players in a network the assurance that executed actions by a certain party cannot be denied afterwards, allowing to derive the necessary liabilities for serious business interactions.

5.1.3 Protocol Characteristics

Crucial for any information exchange is the grounding of the outlined characteristics towards well-known protocols (cf. Fig. 5.5). Only this step makes the requirements implementable. A common practice is protocol bindings which state the translation of the required functionalities into the abilities of already existing protocols. In accordance with the IOT-A requirement analysis [247], several criteria are distinguished defining the *type of a protocol* (P1), namely whether it supports synchronous or asynchronous interactions and how it copes with events like notification and alarms (cf. F3). Furthermore, the interaction pattern between client and servers is stated (request/response, publish/subscribe, flooding) similar to Joshi, Didier, Jimenez and Carey [84] and others.

Closely related to the interaction patterns is the provisioning of *control information* (P2). The link-based Hypermedia approach exposes *interaction options* (P3) with the information document itself, while Web Services require special interface descriptions. Similarly, the *coupling between client and server connectors* ([62], P4) and the required configuration effort at the client-side (P5). While the first is relevant

for the scalability of a network (cf. N7), the latter targets the issue of whether interaction parameters are explicitly presented at the resource, externally defined by, for instance, a standard document, or even implicitly stated and therefore hard to implement at the client connector.

Generally speaking, a binding must state how a specific protocol implements the demands of the functional requirements while complying with the non-functional ones. As such, the interactions with resources (F2 to F4) are crucial. However, IIoT applications have several unique requirements as they are usually deployed on small, resource-constraint devices. *Network bandwidth* (P6) is the amount of binary data needed to transport an information unit over the network, closely related to its *energy consumption* (P7).

Especially for real-time applications, for instance, process control over the network, low *latency* (P8) is crucial. In some scenarios, losing some data objects might even be a valid trade-off for lower latency. Latency is certainly affected by the selected data format. While the data model and its semantic are described in the following section, the syntactical aspects and especially the supported *Media Types* (P9) are specified by the protocol binding[62].

A significant amount of IIoT protocols have been introduced for the varying domains and scenarios. While usually none of the presented protocols in the following is by design limited to IIoT use cases, in general, people use them in IIoT-related settings. Therefore, regarding the selected list as 'IoT protocols' is certainly justified as they have obvious conformance. Both TCP and UDP appear as transport protocols on top of IP. Most share the motivation of reducing resource consumption in terms of computational power, network bandwidth, or energy consumption. As a consequence, the interaction semantic is to some degree implicitly defined by either the protocol specification or open to interpretation. Consequently, while reducing the resource load, the interpretational load – mostly at the client-side – is increased.

CoAP follows a similar approach as HTTP but with smaller messages. Methods, header, and status codes are easily mapped to their HTTP counterparts, while the counter direction is not always possible. Similar to HTTP, CoAP is a request/response-based protocol, relying on resources and supporting RESTful interactions. However, instead of TCP, CoAP uses UDP as the transport protocol. Consequently, TLS is not supported. Instead, the Datagram Transport Layer Security (DTLS) protocol can be used [248].

MQTT is a lightweight protocol used for publishing messages to many targets. A central message broker is used to receive messages from a publishing client and forward them to previously subscribed clients. Messages are tagged with topics, allowing the sender to describe the message intention and the receiver to filter its subscription accordingly. Even though MQTT is relying on TCP/IP, the delivery of a message is not guaranteed but can be requested by setting one out of three Quality of Service levels, whereby higher service levels lead to higher network overhead. MQTT can be encrypted by applying TLS.

Apart from CoAP and MQTT, many other protocols have been introduced. OPC UA is intended for machine-to-machine communication in industrial settings. A binary version of OPC UA requires fewer resources while another specification for SOAP on top exists for interactions with Web Services.

5.1.4 Data Representation

While the serialized format of the regarded data object is part of the syntactical aspects, its representation, scheme, and meaning of the used vocabulary are generally defined in an information model (as in [62] [242] [35] [74] [4]). First of all, a consistent *identification pattern* (D1) needs to be defined. Some approaches are IRDIs, UUIDs, or URIs. While both IRDIs and UUIDs usually require a certain dereferencing service, URIs in the form of URLs can also directly serve as *resource locators* (D2, cf. Fig. 5.5).

An IIoT *information model* needs to define the physical Asset, its virtual representation, the relations

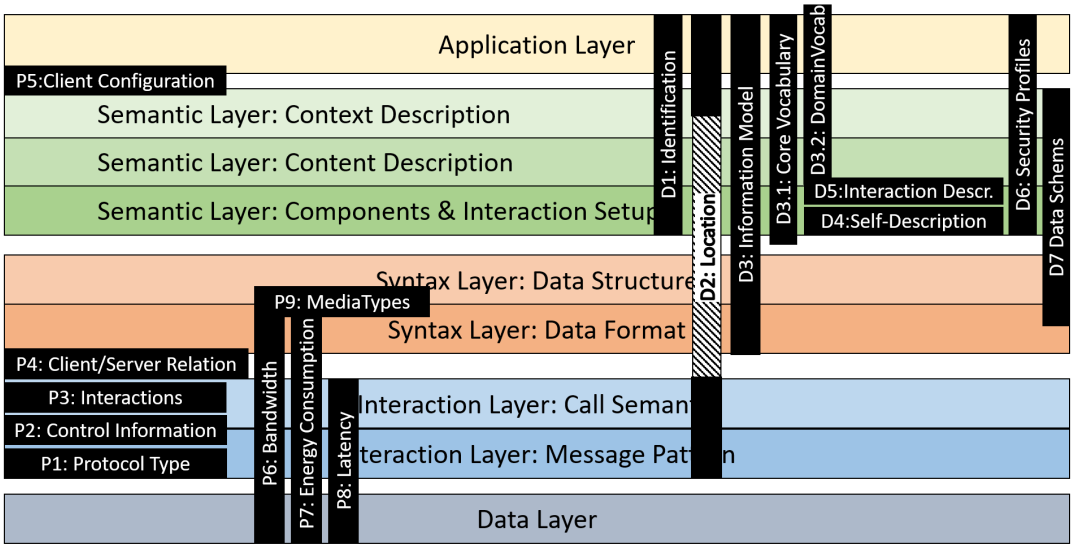


Figure 5.5: Overview of Protocol (left) and Data (right) Requirements, positioned in the communication stack.

between them, its attributes, and the operations on top of it [101] [74] [242] (D3). This *core vocabulary* (D3.1) is then extended with optional *domain-specific extensions* (D3.2) in order to describe specific features which are not relevant in all use cases. Still, as in the regarded settings a discovery at runtime is required, the resources need to provide *self-descriptive information* (D4) [62] and outline which of the interaction patterns of F2 to F4 are available for the specific resource (D5) [62]. Potential configuration of the access behavior of a hosting system can also be represented and changed through *security configurations* or profiles (D6), which might be exposed to users according to their permission setting [247][35].

In addition to the publishing of the resource itself and available operation, Kovatsch et al. distinguish between the format of such metadata and payload data and argue for the provisioning of *data schemes* (D7). In contrast to the metadata, which has to follow the specification of the IIoT framework, the payload data might be sent in binary or proprietary serializations [62][35].

5.1.5 Summary: An Overview of IIoT Requirements

The contribution of this section presents the main parts of the model for *IIoT Requirements* (CO4). It introduces several categories and explains the contained aspects and relevant topics. The gained insights from CO3 are the foundation for the in-depth analysis of IIoT Reference Frameworks in Chapter 6 but also for the interaction model for Digital Twins (CO3), which is introduced beginning with the next section.

Obviously, the presented list of requirements and concerns can not be complete, neither can it cover each and every use case. Still, it comprises a basic model being sufficient for many relevant IIoT scenarios. Having the outlined categories and topics as references helps to understand upcoming challenges better and establishes a common ground to design reliable solutions.

In particular, the location of the named IIoT topics in the layered structure helps the stakeholders of IIoT applications to find unnoticed areas and blind spots in the early project phases. This is especially relevant as the subsequent introduction of necessary features is cumbersome and requires significantly more resources. The following sections of this chapter use these hypotheses as their underlying assumptions,

and argue that a generic - and therefore adaptable - communication pattern ensures the simple adjustment to new conditions.

5.2 State-based Integration Through Resource-orientation

Especially in the context of smart factories, the connection of the different, heterogeneous production facilities through a digital shop floor promises faster conversion rates, data-driven maintenance, or automated machine configurations for use cases, which have not been known at design time. Nevertheless, these scenarios demand IIoT representations of all participating machines and components, which requires high installation efforts and hardware adjustments. A strict resource-based view on IIoT interfaces allows on-the-fly integration of any kind of resources or information Asset, as for instance outlined by Bader and Maleshkova [25].

However, that is only possible through an incremental process for bringing the shop floor closer to the IIoT vision. Each modification step needs to justify its investment costs, independently of the following ones. Currently, the majority of systems, components, or parts are not yet connected with the internet and might not even provide the possibility to be technically equipped with sensors. However, those could be essential parts for a realistic digital shop floor representation. In the following, a new approach for IIoT Digital Twins is presented, which is capable of independently calculating a physical object's condition by dynamically collecting and interpreting already available data (cf. CH2) through RESTful Web APIs. The internal logic is adjustable at runtime since changes to its respective physical object, its environment, or updates to the resource itself should not cause any downtime (cf. CH1). Thereby, the state-based view promoted in this section marks the first part of CO3, a generic *interaction model for IIoT Digital Twins*.

Especially in the context of smart factories, notable efforts are conducted to bring the diverse and heterogeneous components of the shop floor to a consistent digital layer. In order to keep the integration efforts sustainable, an easy-to-understand data and interaction model is required. The self-explaining characteristics and the high maturity level of the Semantic Web make it a suitable candidate for a future-proven solution. Such an integration layer encapsulates the complexity of underlying network implementation.

Even though IIoT technologies support the integration process, the IIoT is not only about increasing the flexibility or new insights into the behavior of one machine but of the production line or even supply chain as a whole. Applications achieving these aims need to be deployed rapidly, if not automatically, with minimal effort. A heterogeneous landscape with a use case-driven integration approach requires continuous efforts, contradicting any potentially acquired efficiency gains. The main reason is that the configuration of interfaces is usually oriented towards scenarios relevant at the respective point in time but not for generic requirements of such later deployed systems [249]. As the input needs for future applications are per se unknown at design time, a consistent and comprehensive digital reflection of the real world is more efficient and sustainable. Achieving this flexibility is the main benefit where the IIoT can contribute to the manufacturing industry.

The approach outlined in this thesis is an incremental process for bringing the shop floor to the IIoT. Many machines and devices nowadays are already (partly) IIoT-ready. However, systems, modules or parts, which cannot be connected to the internet or promptly equipped with sensors are also required for a valid digital representation of the shop floor. Therefore, it is necessary to come up with a suitable digital representation, which does not necessarily require a direct network connection to the Asset itself. The Digital Twin is responsible for independently calculating the object's condition by collecting and interpreting already available environment data. It relies on a RESTful interaction model to deploy and scale Digital Twins. Furthermore, it is shown how the configuration specifying a Digital Twin can

be adapted at runtime by RESTful Web APIs since changes in the real world need to be reflected in a transparent and standardized manner. The implementation of the approach is based on a Linked Data Platform server, thus also enabling future extensions. A demonstrating showcase is also available to illustrate the concept. The significantly lower cost of a Digital Twin based on omitted hardware updates (if those are possible at all) as well as the newly gained opportunities to monitor, control, and optimize the digital shop floor motivates the use of Digital Twins to gain a coherent and scalable IIoT landscape.

One example application scenario is a robot gripper arm, as one unit in an industrial production line. Its motors and the supplied materials are directly observable by appropriate analytic components. In contrast, the shafts transmitting the force from the motor to the jaws and the jaws themselves cannot be monitored. Nevertheless, the abrasion state of both components might be a critical factor for a maintenance strategy. A Digital Twin can encapsulate the necessary data collection (number and duration of loads, applied material type, etc.) and the currently best-performing evaluation logic. The thereby encapsulated Digital Twin can be shifted easily, connected to other IIoT systems, or updated easily as it conforms to well-established Web standards. This supports the development of a sufficiently detailed and flexibly modeled smart factory while reducing the required investment.

5.2.1 The IIoT from an Interaction Viewpoint

The implementation of the here described scenario is openly accessible. A Linked Data Platform server hosts the IIoT representation of the robot arm as an LDP RDF Source together with its components (Fig. 5.7). Requesting the Digital Twin of the robot as a web resource, as well as its sub-resources, is possible via GET requests, returning the current state of the physical object as far as observed together with some metadata. RESTful interactions with the Digital Twin's Web API are enabled by overwriting the current resource state of the robot's motor and joint by sending an RDF statement with the predicate *saref:hasState* and the new state ('active'/'inactive' and 'moving'/'stopped') at the object position. Example requests[1] illustrate the use case.

Figure 5.6: Online resource[1] for the use cases.

Even though the robot itself and two components are digitally accessible, these represent only a fraction of the actually installed components. In order to, for instance, decide on the optimal maintenance strategy, further information about other parts is also necessary. Two Digital Twins are hosted on an Apache Marmotta[2] server representing the shaft and the jaws.

Their Digital Twins simulate the physical Asset's state to cover also the shaft and the jaws. To do so, the relation between the already available input data and the required features needs to be determined. Such an algorithm or heuristic is based on the currently most accurately available knowledge. Furthermore, in the same manner as the setup in the physical world changes or new information on the performance of the regarded components becomes known, the deployed logic of the Digital Twin needs transparent update functionalities as well.

Assuming that the correlation between the shaft's abrasion state and the conducted number of actions can be captured via a linear function, the component shall keep its original quality after 10 usages but be broken after 20. A straightforward heuristic would be:

[1] https://github.com/aifb/virtrep/tree/master/requests/
[2] http://marmotta.apache.org/

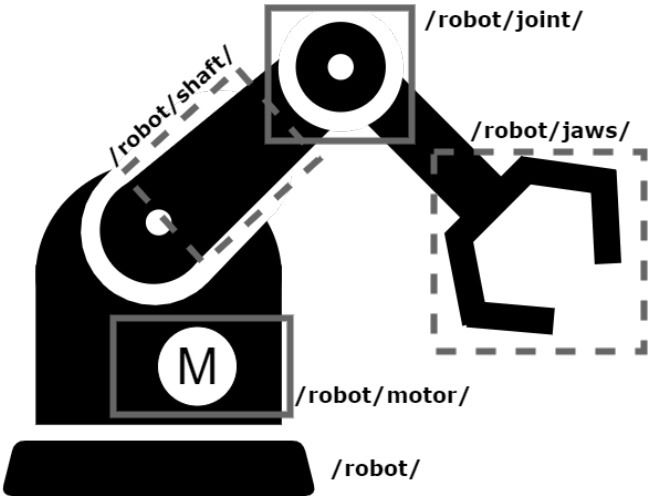


Figure 5.7: Illustration of the robot components with a direct network connection (solid rectangles) and derived Digital Twins (dotted rectangles).

$$abrasion(actions) = \begin{cases} 0 & , |actions| < 10 \\ min\{\frac{1}{10} * (|actions| - 10), 1\} & , else \end{cases} \tag{5.1}$$

This simple function is most probably not very accurate. After better insights into the abrasion process, a more fine-grained relationship can be defined, reflected in Eq. 5.2. Hence the formerly deployed heuristic must be adjusted. The Digital Twin can be initialized with Eq. 5.1 and then be easily updated with Eq. 5.2.

$$abrasion(actions) = \begin{cases} 0 & , |actions| < 10 \\ min\{\frac{1}{1000} * (|actions| - 10)^3, 1\} & , else \end{cases} \tag{5.2}$$

5.2.2 The Virtual Representation of the Digital Twin

The Digital Twin at its core contains a) some descriptions of its physical *Asset*, b) its current (simulated) state, c) the function/algorithm responsible for the calculation of derived features, and a RESTful pattern to request it. A Digital Twin is a Web resource identified by a globally unique URI. At this point, the focus is on the interaction pattern and how the derivation function can be implemented, its characteristics, and how to interact with it.


Figure 5.8: Source code[3] of the prototype implementation.

The Digital Twin is modeled as a RESTful Web Resource producing RDF, in particular Linked Data[3]. RDF, as the common data format, has the advantage of being a very mature and widespread standard as it serves as a cornerstone of the Semantic Web Stack. Its native connection to formal knowledge representations makes RDF the data format of choice for loosely coupled environments [250], such as the here considered shop floor, with many heterogeneous machines from different vendors.

[3] source code available at https://github.com/aifb/virtrep/

```
1  # request necessary information
2  {   [] http:mthd httpm:GET ;
3         http:requestURI <https://18.157.197.66:8443/assets/robot/motor/> . }
4  {   [] http:mthd httpm:GET ;
5         http:requestURI <https://18.157.197.66:8443/assets/robot/joint/> . }
6
7  # calculate abrasion state
8  {
9    ?motor ex:numberOfRequests ?r .
10   ?r math:greaterThan "10" .
11   ?r math:notGreaterThan "20" .
12   (?r "10") math:difference ?diff .
13   (?diff "3") math:exponentiation ?exp .
14   (?exp "0.001") math:product ?abrasion .
15 } => {
16   <#shaft> ex:hasAbrasion ?abrasion .
17 } .
```

Figure 5.9: Integrating data from external sources and deriving new information (simplified)

A configuration function f of a Digital Twin is a relation of derived features y (e.g. the current abrasion state) by evaluating currently available input data x (e.g. the number of movements). Its representation as Linked Data as the data format of choice simplifies the syntactical and semantic interoperability at the same time. Therefore, both input data and derived features are Linked Data triples ($x_i, y_j \in L$), with L being the set of RDF triples compliant to the Linked Data principles.

As the Linked Data-Fu engine [171] is used to collect and process the input data x in the demonstration implementation of Digital Twins, the function f needs to be a two-stage process. First, a set of declarative rules in N3 syntax defines the initial data and processing steps. According to [171], these rules are denoted as a *program* p. Executing a rule of p leads to either the execution of HTTP requests towards RESTful Web APIs – responding with RDF – or the derivation of new RDF statements locally. The second step involves a SPARQL Construct query q filtering the collected data and finally generating y. The set of RDF triples y representing a Digital Twin is, therefore, for the demonstration implementation:

$$y = f(x) = q \circ p(x) \tag{5.3}$$

An example program is shown in Fig. 5.9. Lines 2 to 5 specify two HTTP requests to available IIoT entities as Linked Data resources. The rule from lines 8 to 17 derives the new statement specifying the current abrasion state when the condition in the body part (lines 9 to 14) of the rule is satisfied by the above-requested RDF set. The query in Fig. 5.10 receives all statements from the program execution and constructs the description of the Digital Twin as specified from lines 2 to 10 in the form of a SPARQL Construct query.

One has to note that the chosen form of the configuration function is not a general requirement for Digital Twins but is due to the selected engine. Different implementations can require other formats such as, for example, machine learning models or scripts. The proposed configuration method has the main advantage of being familiar with the Semantic Web and, therefore, forms a single technology stack with RDF and Linked Data.

A Digital Twin is the combination of metadata triples together with the set of derived triples and a resource representing the function code. As this resource is also a Linked Data Platform Resource (LDP-R), it can be manipulated according to the Linked Data Platform specification. Even though the concept of Digital Twins is conforming to the Linked Data Platform specification, Digital Twins enhance

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX saref: <https://w3id.org/saref#>
PREFIX td: <https://www.w3.org/2019/wot/td#>
PREFIX ex: <http://example.org/>

CONSTRUCT {
  <./shaft> rdf:type td:Thing .
  <./shaft> :hasProgram <./program/> .
  <./shaft> :hasQuery <./query/> .
  <./shaft> saref:hasState <./shaft#abrasion> .
  <./shaft> saref:accomplishes ?job .

  <./shaft#abrasion> rdf:value ?abrasion .
  <./shaft#abrasion> rdf:type saref:State .

} WHERE {
  ?s ex:hasAbrasion> ?abrasion .
  ?t saref:accomplishes ?job .
}
```

Figure 5.10: Creating the Digital Twin through a SPARQL Construct query (simplified).

their regular interaction model by requiring further configuration. An adaption of a Linked Data Platform server enables the proposed deployment of Digital Twins, as shown in Fig. 5.11.

The Linked Data Platform specification defines how clients can interact with LDP Web Resources in a RESTful manner. In particular, Digital Twins and the additionally necessary configurations (p and q) are themselves Web resources. More precisely, the configuration files are LDP Non-RDF Sources, whereas the Digital Twin itself is an LDP RDF Source. The distinction is necessary as neither the SPARQL syntax of q nor Notation3 of p or any other model are necessarily RDF serializations, a requirement for an LDP RDF Source.

In general, two possible strategies can be applied to create a Digital Twin in a RESTful manner. On the one hand, a user agent can periodically calculate the data, send it to the server, which then enables access to the latest received state of the resource. Alternatively, the server waits until a client requests the resource and then calculates the data on the fly. The first approach has advantages in an environment with unreliable network access since at least some version of the requested data is available. Also, storing each state creates a time series and allows for historical analysis. Nevertheless, especially in the context of the IIoT update rates, which are often in the range of milliseconds, this would result in creating large data volumes.

To receive a lightweight representation and also to increase the scalability of the concept, the calculation of a Digital Twin's state on the fly is a feasible approach. An HTTP GET request on the Digital Twin triggers the server to load both the program and the query into the engine and executes it (Fig. 5.11). The dynamically created RDF statements are calculated, representing the related physical object in the same way as a regular Web resource.

In order to identify the desired configuration of a Digital Twin, the engine needs additional information. The relations (1) and (2) of the data model (Fig. 5.12) are part of this additional information. A naive client would expect these statements to reside at the Digital Twin itself as both include the Digital Twin as their subject. Unfortunately, the Digital Twin is calculated dynamically at request time and not stored in the server's database, and, therefore, not accessible for the engine itself. Even though one could deposit the statements (1) and (2) at any other LDP resource, it is better to keep such information as close as possible to the related Digital Twin. Therefore, the Digital Twin *Container* shall be introduced, an LDP

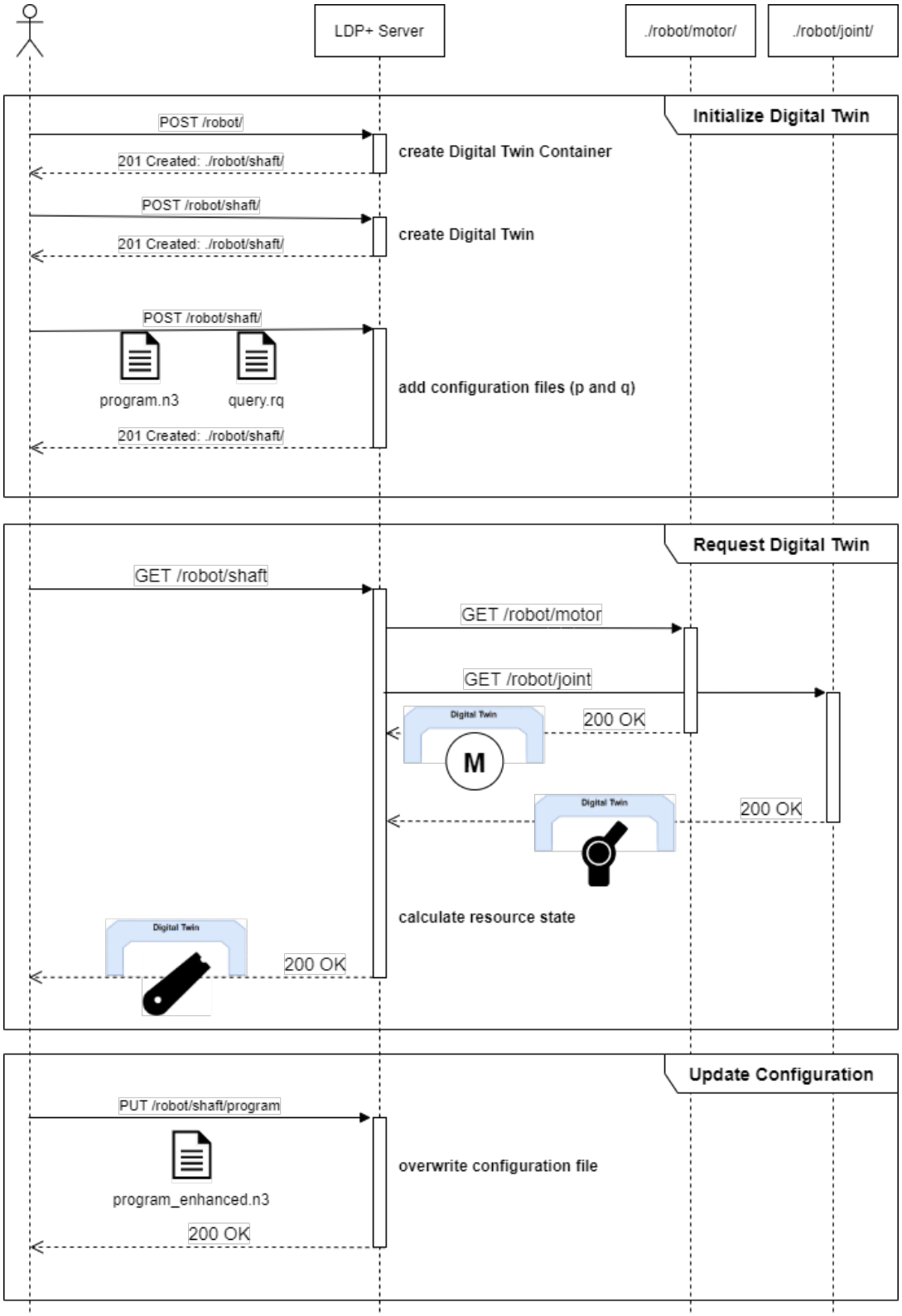


Figure 5.11: Interaction model for Digital Twins

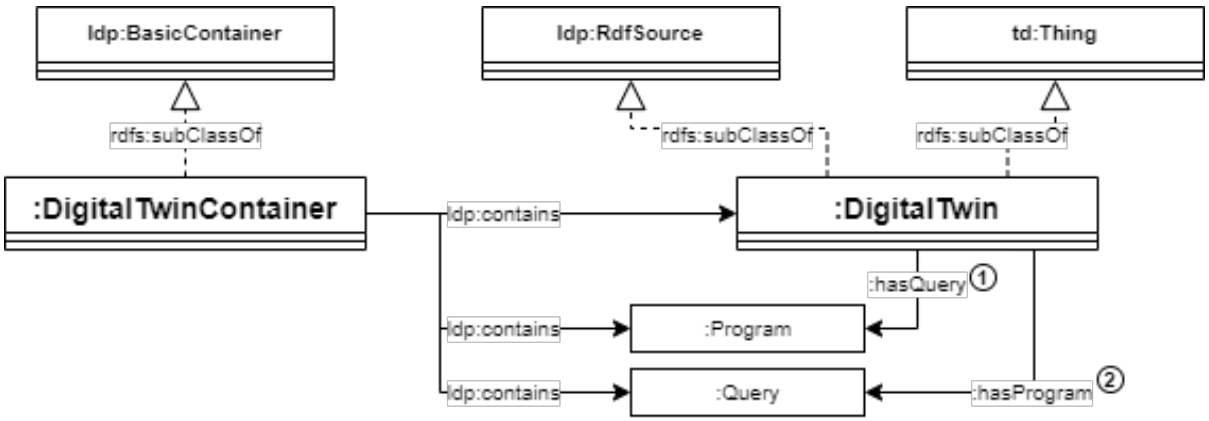


Figure 5.12: Basic classes and relations to create a Digital Twin.

Basic Container with the additional requirement of containing exactly one Digital Twin together with its configuration, here the query and program resources. The engine can locate the Web resources p and q as the container's child resources and, therefore, combine the necessary Web resources.

In REST terms, as introduced by [95], the proposed concept of Digital Twins cannot be seen as a user agent. Even though the Digital Twin can initiate requests through a client connector, it is not the source of the initial request. In the same way, the *origin server* concept does not apply as the source of the requested Asset since it is not directly connected with it. It is a general problem with the IIoT that the virtual reference is not the target of the request, which in fact, is the physical object. Therefore, any IIoT entity must be some kind of intermediary. Digital Twins have server and client components as required for a *proxy* but without allowing a client to choose whether to use it. In contrast, a *gateway* forwards the request but acts as an origin server for the client. Digital Twins act in that way, making them a *gateway* for the digital shop floor without necessarily informing the client about this role.

5.2.3 Summary: A scalable Integration Pattern

The concept of Digital Twins serves as a mediator to bring unconnected physical objects quickly to the IIoT. Still, it is obvious that a simulated model cannot be as accurate as a directly observed measurement. Nevertheless, the advantage of the Digital Twin representation is in the low-cost deployment and the flexible ways to write the configuration into an encapsulated representation of a real-world object.

The proposed interaction model does not focus on security challenges or discusses Digital Twins in other IIoT protocols such as MQTT, CoaP, or OPC-UA. Furthermore, only a state-based request/response interaction is possible, in contrast to data streams and events/notifications of many IIoT applications. Future steps include the consisting modeling and description of the Digital Twin's analytic logic, its interaction scheme, and physical behavior in a machine-processable manner. The Hydra and openAPI vocabulary will be used to further document and explain the interface. Although the implemented example execution environment is based on an HTTP Web server, the common concept is not restricted to HTTP.

Digital Twins can be one stepping stone for a faster introduction of the IIoT to real-world production lines. The described contribution of this section shows how Digital Twins fit in current reference works and specifies how integration logic can be deployed in a transparent manner. The next section extends this approach and explains its potential for Edge Computing in Brownfield scenarios.

5.3 Brownfield Integration at the Edge of IIoT

Maintenance and repairs in Brownfield environments are discussed and outlined as suitable wrapping technologies as presented in [27]. This section extends the approach from Section 5.2 towards such environments and demonstrates how the fundamental operations, independent of the used protocol, can be used to create consistent Digital Twins at the Edge. Edge Computing – the provisioning of processing capabilities close to the usually resource-restricted IIoT device – can encapsulate parts of the original data heterogeneity coming from the IIoT data source and thereby unify the appearance of the Asset through the presentation of its Digital Twin. Nevertheless, the operation semantics needs to be unified and a shared understanding established, which atomic operations are required and what their implications are.

Even though some design characteristics are generally accepted for the digitized integration of machines, applications, and surrounding components, the inherent complexity and variety of interaction protocols, data formats, and (implicit) dependencies of existing deployments in so-called Brownfield environments hampers the data-driven manufacturing of the future. The following discussion explains how a protocol-agnostic integration layer and the focus on very view operations can simplify the general setting and enable flexible and easy to manage environments.

Therefore, it is necessary to design an iterative approach where existing context data is used to encapsulate the specific complexity of each Asset in order to create a flexible integration layer at the Edge. This argumentation leads to a framework for a general IIoT integration layer based on the patterns of Section 5.2, which is presented here. Nearly all relevant resources are modeled as self-descriptive Digital Twins according to the setting of the physical production environment, therefore drastically reducing the required access barriers for rich applications on top. A reference implementation together with a discussion about its business implications by the example of industrial maintenance is conducted in the following.

5.3.1 Iterative Brownfield Deployment

Several architecture approaches are possible in order to digitally connect production machines with an organization's IT systems. In the most basic scenario, a document-based information exchange (e.g. relying on proprietary formats, emails, or even office documents) transfers jobs either directly to a customized interface. Pulling or polling data, varying protocols and data formats, differing data syntax and identifiers hamper a seamless information and command flow. For instance, a script-based application could periodically query the data from some databases and send an email to a certain account. Any change in the setup, the used databases, or new information on the abrasion process would then require a manual adjustment of such scripts, a text-based interpretation of the received email, and sufficient information on the requirements of the downstream applications.

Moreover, the connection of machines from different manufacturers requires a deep understanding of each deployed device, its characteristics, and limitations and the design, creation, and maintenance of highly customized interfaces and control software. In such a point-to-point wiring approach, any applied change in either its features or at the composition of the production line, for instance, introducing a new unit or replacing an outdated one, leads to mandatory and very complex adjustments at any related device and application. The thereby created organic growing results in inconsistent data models, varying interaction patterns, and applied protocols that are hardly maintainable. Especially small and medium-sized companies do not have the possibilities to employ the according staff to cope with the thereby created challenges.

Hybrid approaches relying on combinations of field bus architectures (e.g. the CAN bus) and Ethernet connections combine polling information from a data producer with pulling data to a consuming system.

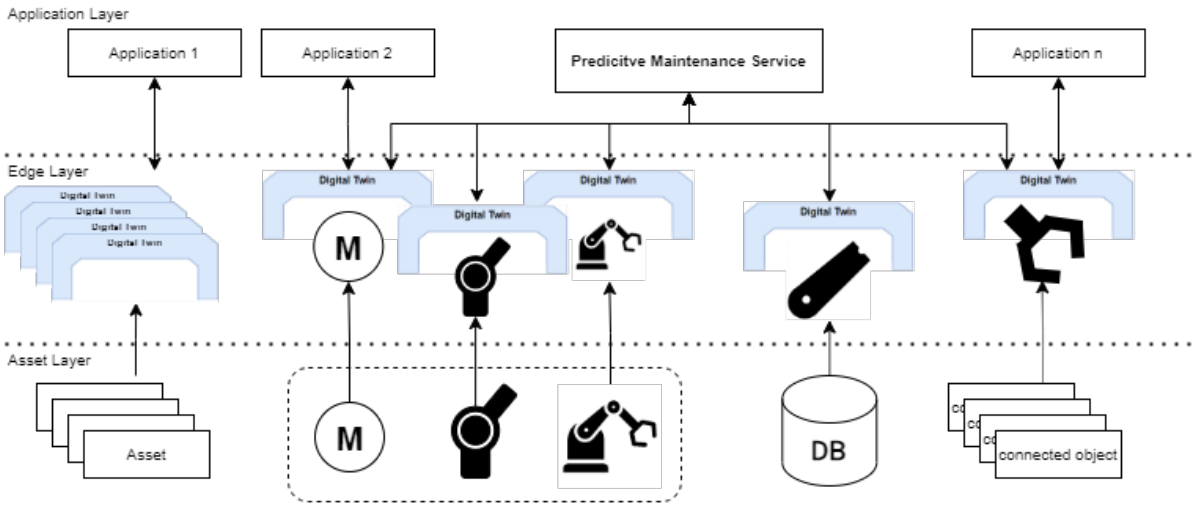


Figure 5.13: The integration layer contains both network-enabled Assets and unconnected objects as Digital Twins.

Installing both paradigms is mostly the case when one subset of machines is initially designed for bus communication, whereas other systems require a request-response pattern. The resulting environment makes it even harder to understand the existing dependencies and data flows.

On the other hand, the IIoT promises consistent, well-structured architectures with plug-and-play-like deployment patterns. Even though a majority of responsible managers agree on its relevance for future manufacturing [251], a common agreement on its technical characteristics and real-world manifestations is still missing. Especially in scenarios where existing production lines need to be upgraded to IIoT standards, well-established strategies and procedures are not in place yet. So-called Brownfield Deployments are common use cases as only in rare cases the gained benefits of a newly created fabric from scratch justifies its investment costs.

Combining IIoT with Semantic Web technologies enables a self-descriptive integration approach where each physical object is represented by a digital resource, either an active Digital Twin if the Asset is equipped with a network connection or Digital Twin based on simulations and external data otherwise. Every resource is identified and accessible through a unique URI, allowing its referencing through the local IIoT network but also worldwide without additional efforts. Every resource contains information on itself describing its type, location, functionality, and capabilities together with Web links to additional information. The data format for these descriptions is provided in RDF, which provides syntactically and semantically defined statements on the resource itself, its characteristics, and its current state. Other formats, for instance XML or JSON, lack the semantic part out of the box. The self-descriptive aspect is essential as only the close provisioning of information together with the regarded resource itself guarantees a true modular landscape (cf. Fig. 5.13).

Additionally, the restriction of interaction patterns to four basic methods, namely *Create*, *Read*, *Update* and *Delete* (CRUD), breaks down the operation complexity of higher-level business workflows and distributes the logic between clients and servers in transparent manners. Obviously, limiting remote calls to these methods significantly reduces the possibilities to implement interactions. Still, only the reduction towards basic operations enforces the explicit statement of the before implicit assumptions and design decisions. Higher-level functionality requires a deep understanding of the existing dependencies and design decisions which are not possible to explain in a suitable interface description of complex calls. However, his trade-off accepts less effective calls and higher data traffic as more requests are necessary

and the responses contain a high amount of undesired information. These disadvantages do not directly affect the resource-restricted IIoT device but the communication between the services with the integration system at the Edge.

As the integration layer forms a distributed network using internet protocols and identification and access mechanisms, it can be connected to the global internet without any adjustments. For security purposes, a gateway with appropriate access control is mandatory, but the technical communication will work without any adjustments. Furthermore, with the same mechanisms proven in the Web, new data providers and resources can be added, updated, or replaced as the loose coupling of data producers and consumers guarantees a future-proof architecture. The iterative characteristic takes effect that any necessary change can be directly introduced at the concerning resource with only a minimal effect on others. The scalability of the network is directly provided by the same features as any number of new resources, servers, or Digital Twins can be added, similar to the well-known scalable nature of the Web.

5.3.2 IIoT Framework for Digital Twins at the Edge

The proposed declarative adaptation of service functionality by utilizing Notation3 (N3) rule programs follows the Smart Components approach from Keppmann and Maleshkova [250]. These rule programs are evaluated by an interpreter against RDF graphs and enable deduction of new knowledge, data transformation, and via build-in functions HTTP interaction as well as mathematical operations. In the extending approach of this thesis, these rule programs are used to describe the internal logic between the RDF input and RDF output of services. The integration of this custom functionality and the LDP implementation is established via HTTP POST, which is explicitly marked by the LDP specification as optional and without further implications for LDP Resources. Only for LDP Containers, the LDP specification provides specific behavior for HTTP POST, in particular, the creation of child resources.

Restricting the interaction methods to CRUD operations limits but also simplifies the data management. Another relevant challenge in industrial environments is the different communication protocols. The prototype developed to examine the presented claims supports a selection of widely used protocols, namely HTTP, WebSockets, and OPC UA (cf. Fig. 5.14). The low entrance barriers and the broad dissemination of HTTP make it the protocol of choice for fast and reliable communication with many services, in particular the simple access through Web browsers. Its clear client-server separation is one of the major success factors and the foundation also for the other protocol implementations of the framework. As shown in [25], Digital Twins can solely rely on RESTful interactions for their access but also configuration, which adds the clear semantics of CRUD operations to the Edge APIs. Propagating this pattern to the other protocols keeps the interactions consistent and gains a losely coupling of producers and consumers of data throughout the shop floor but also beyond.

WebSockets rely on HTTP but allow bidirectional message exchange. Thus, the server can push information to subscribed clients whenever a noteworthy event occurs. Another advantage is the higher efficiency due to its data compression. However, even though WebSockets is an extension of HTTP, only GET requests are supported. To enable CRUD operations, the usage of WebSockets sub-protocols enables the distinction of different operations. For every operation type, one distinct connection channel is established. The operation semantic, whether to read, write or delete a resource, is encoded by the selection of the respective channel. That means that one channel is needed for every possible operation type as the operation semantic of each ‘sub’ protocol cannot be changed after connection establishment. For instance, after opening a Web Socket channel for READ, it can not be changed to WRITE but needs to be finished first, and a WRITE channel needs to be opened. Hence, it is necessary to keep the four connection channels open for each Digital Twin as long as the participants want to interact with each other.

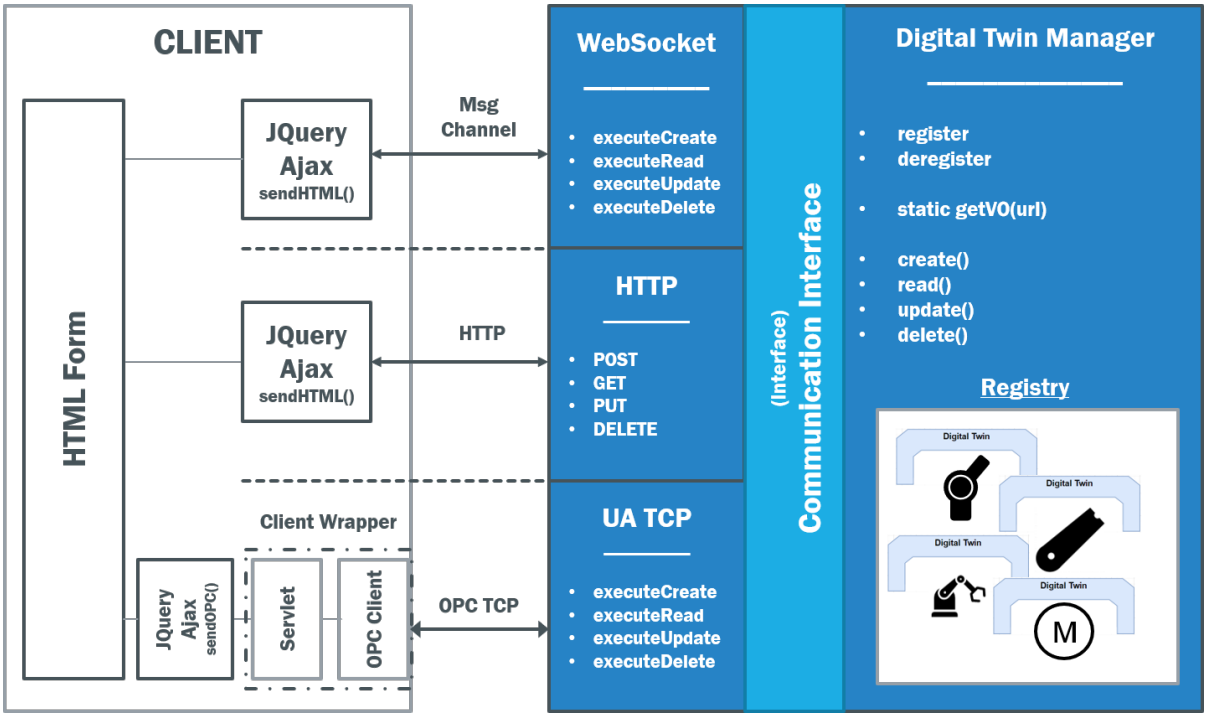


Figure 5.14: Integration Manager at the Edge for Digital Twins.

The third chosen protocol is OPC UA. As explained in Section 2.2, it is one of the most widely used techniques for machine-to-machine communication in IIoT applications [252]. OPC UA ships with two different communication protocols named HTTP/SOAP and the binary UA TCP. As the already existing HTTP sockets can easily be extended towards the first, UA TCP is examined in detail in the following. Other common IIoT protocols like CoaP, MQTT, or XMPP basically follow similar patterns and are therefore not yet part of this analysis.

The core project contains various Digital Twins that are hosted by the Digital Twin manager. The manager is responsible for every incoming operation request. The platform itself has a backend for each protocol. These backend modules implement the communication interface that ensures equal effects on the targeted resource. Every implementation has two main jobs: First, convert the request to CRUD methods on the Digital Twin manager and transfer the operation to the Asset. Second, translate given responses into protocol typical responses (e.g. exchange status codes or add header information).

If a client starts a query on a Digital Twin, the data aggregation is initiated. As stated in [25], Digital Twins should collect their data dynamically at request time to stick to a state-based view of the world. This avoids data inconsistency on different application layers and solves – among others – the challenge of time synchronization. The acquisition algorithm is configurable for now by a configuration file in N3 syntax but shall be enhanced to script-based solutions (e.g. Python) and Machine Learning models. It is also possible to define direct connections to resources, for instance, an OPC UA Server where the parts of the necessary context data are provided. Several different serializations formats are provided due to different possible requirements of the communication partner.

Following the proposed approach, it is easy to access data that is stored in the framework. Nonetheless, it is necessary to manually add these representations to the framework because one has to define where the representation should take the data from. This is accomplished using the deployment techniques of Section 5.2. Even though Digital Twins, more specifically the hosting integration manager, could

discover data sources autonomously, there is a need to make particular data and access APIs somehow accessible first.

5.3.3 Implementation

In the first phase of the deployment process, empty service stubs are deployed as part of an extended Apache Marmotta [253] Linked Data Webserver[4]. At this point, the started Marmotta server contains some basic configurations and is already accessible but has no included data or further functionalities beyond the LDP specifications. In order to add a web service, at least three resources have to be created. We will describe the procedure regarding a simple service that applies basic mathematical operations on input values and returns the result.

5.3.4 Deployment

The main control component is a Python script that coordinates the complete setup. Mandatory input parameters of the script are the number of desired instances, a seed number, and the hostname of the executing machine. Started on a system that fulfills the requirements[5], the script performs the following tasks in order:
(1) Pulling a specific docker image[6] from the public docker repository. This image is used for all containers running in the evaluation environment. It is based on tomcat7:alpine[7] and contains the enhanced version of Apache Marmotta.
(2) Based on the input parameter, the script creates a docker-compose.yml deployment descriptor. Each docker container gets port 8080 (tomcat/marmotta) mapped to an individual port of the host machine. The port forwarding uses ports in the range [9000:9000+n] by default but can be set to any range at starting time. With this information, the docker containers are deployed.
(3) Controlled through the seed parameter, pseudo-random variants of the service are selected. For each instance, a new variant of the service program is created. As the same seed will always generate the same services, this procedure assures the reproducibility of every scenario.
(4) The generated programs are pushed to the waiting containers by utilizing the LDP REST-API provided by Marmotta. First, an instance of *:DigitalTwinContainer* is required. A :DigitalTwinContainer is a subclass of the ldp:BasicContainer and serves as the Web Service root. It contains all descriptions and links to the other Web resources. Second, the service gets its execution instructions through an *:App* resource. For now, only declarations in Notation3 are supported. For further details on the rule-based syntax, see [171].

Next, a start resource gets posted to the server in the form of a new ldp: Resource. This resource must include an RDF statement declaring it an instance of the :StartAPI class. As the server treats members of this class as triggers for services – and not as ldp:Containers or ldp:RDF-Sources – this is a crucial information as it changes the LDP-defined interaction pattern.

The script can also be executed against a docker swarm. In this case, the containers are automatically distributed on physical machines, which allows scaling the evaluation environment without any further adjustments. By using this approach, the only artifact that is required to set up the evaluation environment is the script itself. Using a seed makes environments reproducible.

[4]The source code is available on GitHub: http://github.com/aifb/s2apite
[5]System requirements: python 3.5+; docker 1.13.0+; docker-compose 1.11.1+
[6]https://hub.docker.com/r/aifb/s2apite/
[7]https://hub.docker.com/_/tomcat/

Resource	Method	Description
/marmotta/ldp	GET	Marmotta Root Container
.../LinkedDataWebService	GET	Returns service descriptions
.../LinkedDataWebService/App.bin	GET	Returns the program code
.../LinkedDataWebService/App.bin	PUT	Updates the program code
.../LinkedDataWebService/InputPattern	GET	RDF meta-data
.../LinkedDataWebService/InputPattern	PUT	Updates the InputPattern
.../LinkedDataWebService/StartAPI	GET	RDF meta-data
.../LinkedDataWebService/StartAPI	POST	Excutes the functionality

Table 5.1: Elementary resources of the proposed Web API.

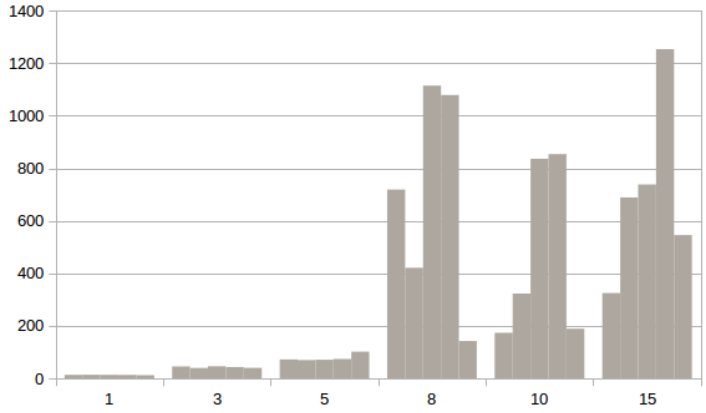


Figure 5.15: Deployment time of n Docker container in seconds. Each container hosts only one web service.

5.3.5 Service generator

The basic idea of the evaluation environment is to provide a platform for generating reproducible, quickly deployable networks of semantic services. Those networks can then be used to test aggregation, composition, and orchestration of such. After the execution of steps (1) and (2), the docker container with basic LDP Web APIs is up and running. The component is created, and the interaction patterns of Table 5.1 are available.

5.3.6 Evaluation

The environment can generate a Web-like surrounding with minimal effort. In order to give the first impression, the necessary computational overhead to establish a certain number of containers is measured. The tests are conducted on an average office laptop[8].

The results (Figure 5.15) show a significant increase in computation time for more than five instances. It is basically caused by the filled main memory of the evaluation machine. In the current state, 1 GB to 1.2 GB of main memory per container is necessary for acceptable responds times. Whenever the docker engine provides less memory, the deployment time for the corresponding instance increases drastically.

The starting process of the Linked Data server – when no sufficient resources are associated – is the main bottleneck. This is supported by recommended settings of Marmotta of about 1 GB. However, even

[8] i7-4600U with 2.10 GHz & 4 cores, 12GB RAM memory, running 64-bit Ubuntu 16.04

if spare memory is available, Docker will start to provide it to the container after an unpredictable amount of time.

As outlined, the service stubs are filled after they have been deployed. Even more, programs are modified and even replaced while the instance is active. The Linked Data-Fu [171] provides the necessary capabilities. Initially designed for Linked Data Stream integration, it is capable of performing HTTP operations in order to interact with RESTful Web resources. As all involved elements actually are of this type, one Linked Data-Fu service can select, invoke and change other services and even itself.

Whenever an HTTP POST request is sent against a Web resource of the :StartAPI class, the corresponding service is triggered. The server combines an optionally received RDF graph from the request body with the specific program file and starts the engine. The computed RDF triples are then collected and written to the response message. Therefore, only synchronous access with an instant response is possible.

5.3.7 Summary: Brownfield-Integration at the Edge

The contribution of this section, a segment for the *Interactions for Digital Twins* in the IIoT (CO3), presents a concept for an iterative deployment layer to transform production facilities to IIoT capable environments. It contains patterns to reduce existing interaction patterns and how available information can lead to a use case-driven enhancement of the virtual model of the shop floor. Additionally, it has been shown how the reached information gain can create economic benefits. Therefore, the presented work contributes to ongoing research on the introduction of IIoT within *Brownfield Environments* (CH2).

The main challenge in the approach is the consequent transformation towards a state-based model and the restriction to the CRUD operations. Existing interfaces commonly do not implement these requirements, especially when based on proprietary protocols. Even though major trends towards RESTful interactions simplify the communication (cf. *Heterogeneous Communication* (CH3)), existing APIs require wrapper modules in order to translate and transform messages.

The presented systems add an adjustable testing and evaluation environment for Linked Data Web Services. Conforming to LDP specifications, it provides the functionality to quickly generate large-scale scenarios for distributed applications with reasonable overhead. The generic Web Services stubs allow functionality adjustments at runtime and therefore support an environment with dynamic changes under laboratory conditions. On top, the combination of a Linked Data server with RESTful Web Service capabilities disperses the separation between data and services.

A sustainable transformation strategy for digital integration of manufacturing systems needs to be aligned with well-established practices of the internet. Only the distributed, loose coupling of systems and the encapsulation of relevant information at the resource itself can create a sustainable IT landscape. The outlined framework for various IIoT protocols provides one step for a plug-and-play behavior in future data-driven manufacturing.

5.4 RESTful APIs with Linked Data: The SOLIOT Approach

Combining all previously presented segments and concepts with a clear data sovereignty approach has lead to ***SOL****id for* ***IIOT*** (SOLIOT) as the consistent model for *Interactions between IIoT Digital Twins* (CO3). The SOLIOT specification is an extension of the Solid framework towards the IIoT [23]. SOLIOT is a combination of lightweight industrial protocols with the integration and data control provided by Solid (cf. Fig. 5.16). An in-depth requirement analysis examines the potential but also current limitations of the approach. The conceptual capabilities are outlined, compared, and extended for the IIoT protocols

CoAP and MQTT. The feasibility of the approach is illustrated through an open-source implementation, which is evaluated in a virtual testbed and a detailed qualitative analysis of the containing concepts.

5.4.1 Guiding Principles

Technologies and practices from the Web increasingly find their way into industrial manufacturing processes. The originally distinctly separated Operation Technology merges with the decentralized technology stack of the Web. One reason for this is the assimilation of requirements: large-scale distributed systems must provide interoperability but also implement protection of data and systems by design. As such, the digitizing of the industrial manufacturing domain can benefit from the latest developments of the Web community.

The ongoing digitization promises a new age of interconnection and interoperability of systems, devices, and actors. Motivated by the potential, uncountable procedures and practices have been developed to connect computers and devices, share data, and enable composed processes. For several years, the various potentials of the IIoT have gained more and more popularity, aiming for integrating nearly any kind of Assets into Internet-based networks. The concept of Digital Twins follows the paradigm to merge the physical and digital representation of such Assets into one indispensable entity and thereby tracks more and more attention.

However, the created heterogeneity of protocols, interfaces, interaction patterns, data formats, or identifiers has proven themselves as challenging and reoccurring problems, especially in distributed systems and architectures. Therefore, several standardization groups and committees started efforts to establish a common ground. These initiatives originate from the Operations Technology domain but also take place in communities around Web and Cloud technologies. The target of Solid [91], for instance, is the human user and its data in the Web. The Digital Twin appears hereby as the combination of the human user with its digital data collection. The overlaps in requirements and faced concerns with IIoT settings are significant and indicate a largely shared problem understanding. However, distinct requirements (normative processes, control assurance, sustainability, resource constraints) prohibit a direct one-to-one transfer from the Solid specification to an IIoT solution.

Furthermore, the IIoT scenarios do not require the same usability degree while having similar requirements regarding interoperability, system federation, security, and data protection (privacy). In the following, a detailed analysis of the overlapping features and missing requirements of Solid and IIoT use cases is conducted, and the results are further specified into SOLIOT. A proof of concept has been developed as an open-source prototype, which is used for a qualitative and empiric evaluation of the framework.

The main focus of the proposed approach is on the distributed character of IIoT scenarios. This requires clear interaction patterns, interfaces, and a shared understanding of the meaning and implications of entities, states, and interactions. While most approaches only target the communication layer and therefore enable syntactic interoperability, the federated interaction between the entities themselves requires a mature and transparent concept of their behavior, interactions, capabilities, and authorities. REST is probably the most successful and well-known pattern in this regard, allowing clean APIs, transparent interactions, and clean expectations on consequences. Extensions such as the Linked Data Platform for data further develop these ideas. Solid transforms the patterns into personal profiles with access rules.

As for instance Pfrommer et al. [254] state, current approaches are tailored to their target scenario and then adapted afterwards. SOLIOT presents a Digital Twin concept primarily based on the standardized *Web of Things* data model, the self-descriptive interaction patterns of LDP, and the data protection mechanism as developed by the Solid approach. No other framework has yet regarded these aspects as

the cornerstones of a consistent, comprehensive solution instead of unrelated developments. SOLIOT shows how the commonly shared principles driving these technologies outline a broadly followed trend towards a distinct set of core principles and how they can be combined to implementable Digital Twins and which steps are still missing. Several works have already translated the concepts and patterns for Web-based architectures and map them to IIoT protocols. This work builds on the conducted requirement analysis of IIoT systems from Section 2.2 and the detailed analysis of their fulfillment by the current Solid stack. SOLIOT extends these activities with the following core contributions:

- The extension of the Solid interaction model towards IIoT requirements
- A mapping of the Solid and WoT data model to IIoT Digital Twins
- The introduction of Usage Control capabilities
- Presenting a Digital Twin representation defined by the above contributions, its potential, and open gaps

Solid puts the user and its personal data at the center of its considerations. So-called *pods* serve as containers for the user's data, containing all its valuable data. Therefore, the Pod's content can be regarded as the Digital Twin of the human user, especially as consuming applications make no further difference between the user itself and its representation through the Solid Pod.

In particular, SOLIOT extends the proposed principles from Section 5.2 and further develops the Integration Manager model of Section 5.3 into a completely self-descriptive hosting and management platform for Digital Twins. Similar to the Integration Manager, the SOLIOT prototype also realizes a protocol-agnostic view. However, this time different IIoT protocols have been chosen to examine the general feasibility and not only focus on OPC UA.

The mapping of Solid features to two different IIoT protocols, the Message Queuing Telemetry Transport (MQTT, [80]) and the Constraint Application Protocol (CoAP, [255]), illustrates the potential of SOLIOT. Both protocols explicitly target different communication patterns. While CoAP follows a request/response process, MQTT is based on publish/subscribe. Both CoAP and MQTT contain only a minimal set of added requirements, which makes them optimal representatives to illustrate the SOLIOT core principles. Different from OPC UA, CoAP, and MQTT do not define their own information model and require less bandwidth. While OPC UA has a rich set of standards and companion specifications, their operation semantics is not defined by any central authority and need to be settled as a community-driven process. SOLIOT contributes to this discussion by proposing an interaction model inspired by the work from Section 5.2 and 5.3 with the Solid patterns. The novelty of SOLIOT is, however, not contained by the CoAP or MQTT mapping themselves but by the compliant merging of their underlying concepts and the restriction to self-descriptive entities with an elaborated Access and Usage Control model.

The Solid reference project in NodeJS is extended with support for both protocols to provide two proof-of-concept developments and general showcases. Up to ten instances of the servers have been started in a federated testbed. Therefore, the scalability of the two approaches is compared to the plain Solid on HTTP solution. Furthermore, an analysis is applied investigating which of the identified requirements can be fulfilled by the SOLIOT approaches.

5.4.2 The IIoT from a Data Sovereignty Viewpoint II

This viewpoint of the scenario extends Section 4.4.2 and outlines the ideas and approaches for an integrated Usage Control of the Digital Twin (cf. Figure 5.17). The Operator *O* of the robot provides static master data together with sensor observations in the form of the Digital Twin. The robot communicates

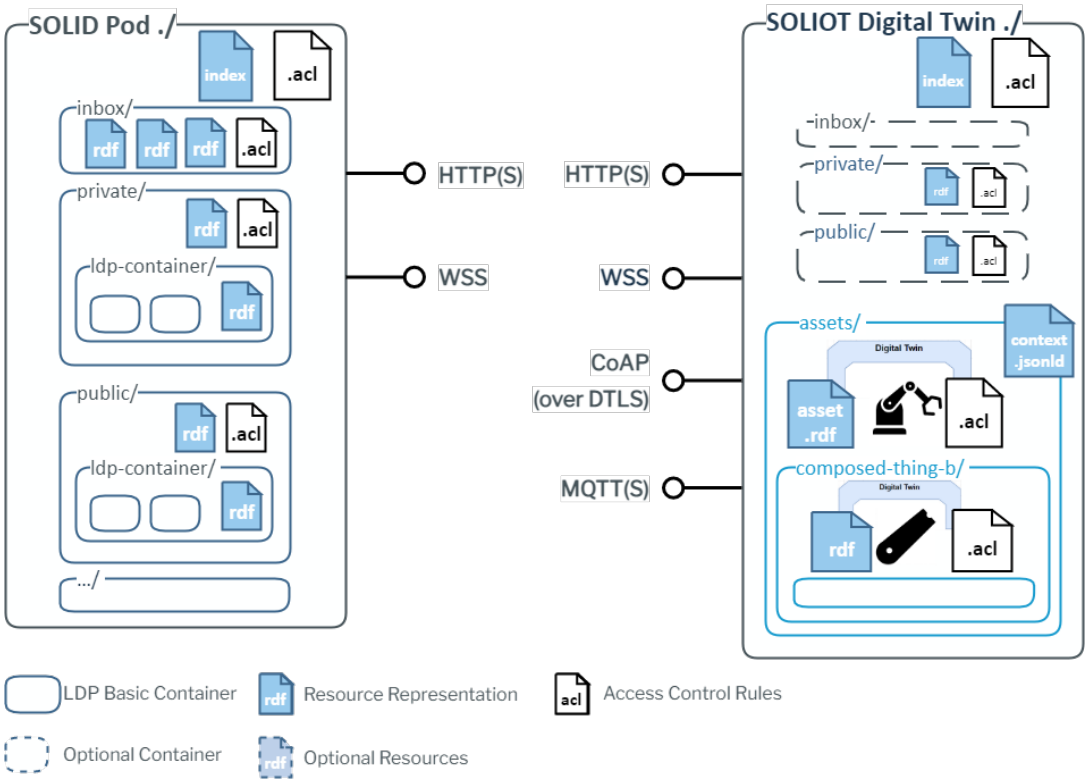


Figure 5.16: The SOLIOT concept extends the Solid approach with IIoT endpoints and reserved resource representations of WoT Things.

through common IIoT protocols and reports its observed state to a remote server. O mandates the Data Analyst D to access the created data, download a filtered representation (Digital Twin') and thus operate his failure prediction algorithm. The prediction results are directly inserted in the Data Analyst's local Digital Twin' and then updated at the original resource.

O is also willing to share the insights contained in the Data Analyst's added attributes with the manufacturer of the robot arm (M_3), as information on potential breakdowns helps them to improve their design and create better-suited solutions for future products. However, it is crucial for O to prohibit any access of other companies in the supply chain, for instance M_1, as his IT environment has not been certified yet and therefore is not regarded as trustworthy enough.

Figure 5.17 outlines the information flow in this simplified setting. The digital representation of the robot, enriched to a Digital Twin, provides static master data and dynamic sensor streams. This interaction is managed by an access control engine deployed directly at the SOLIOT server. The Data Analyst as an intermediary can request the data using IIoT endpoints of that Digital Twin. He also may copy the complete Digital Twin and move it into its own SOLIOT environment (Digital Twin').

However, as the Digital Twin' is now located at the Data Analyst, the contained information is not under the control of O anymore but still contains sensitive data. As such, the Data Analyst's IT environment needs to evaluate the justification of the usage request before giving the analytics component access to it. The required instructions and descriptions must be contained in the Digital Twin and forwarded with the download in the Digital Twin'.

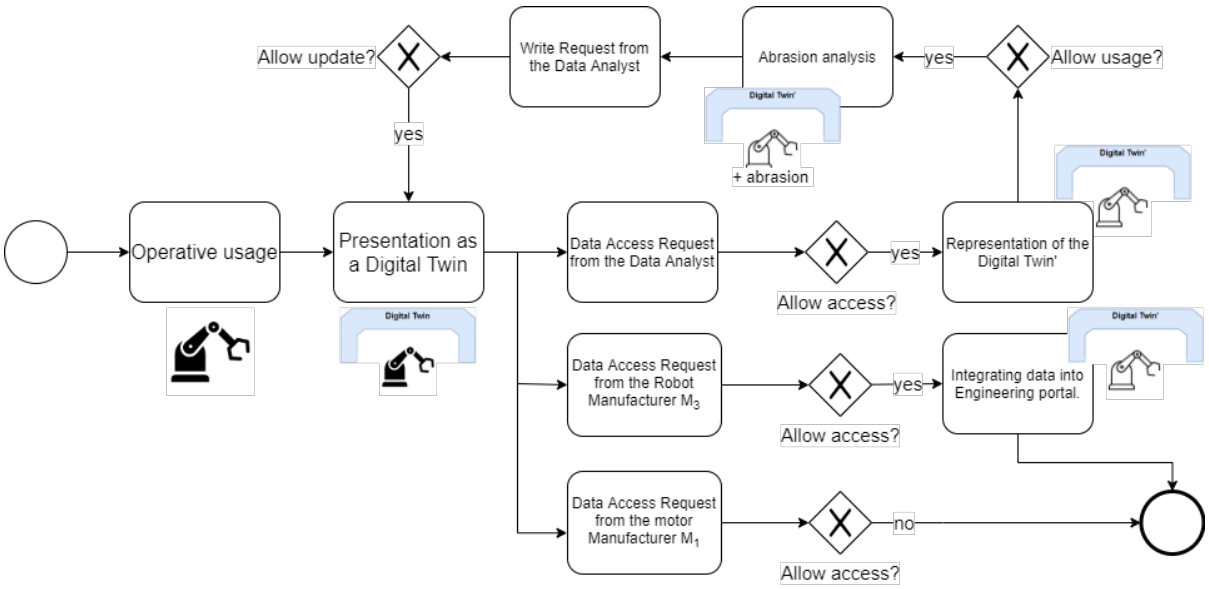


Figure 5.17: Example collaboration network. The Digital Twin information flows from left to right.

5.4.3 Mapping and Interaction Model

The target environment of Solid is the Web, where data and applications are placed in Cloud servers. Physical Assets and their digital counterparts may reside on Cloud servers but also at the Edge/Fog or directly embedded on a device. The obvious differences of these environments must be taken into consideration for a suitable IIoT mapping. While typical IIoT frameworks and Digital Twin models start with the Asset, the primary target group of the Solid specification is not the represented entity (human user) but only the digital representation (the pod). The interaction model and interfaces are also not directly applicable and require intermediary tools. Therefore, the approach aims for consumers at the application level and intends to simplify the integration at this point as much as possible. Different from the original IIoT approaches, Solid focuses on the consuming instead of the providing side. SOLIOT combines both views and thereby lowers the integration gap between the physical Assets and the consuming applications. In this section, an outline of the details of this vision, according to the outlined categories of the analysis from Section 2.2 (cf. Figure 5.2) is presented.

Functional Characteristics

At the core of all IIoT specifications is the necessity to establish interoperability between a huge number of different systems and devices which are currently in place but also will come in the future. While new protocols and interaction patterns are introduced on a regular basis, the core challenge of each new design pattern is to reach a critical distribution size. The necessity to obtain a large dissemination and adoption can hardly be underestimated as new participants in the IIoT domain tend to focus on already broadly implemented technologies. These network effects, in particular, affect the IIoT and the currently observable convergence process towards a smaller number of technology stacks. Solid builds upon the most successful and widespread interaction pattern currently known, the Web. The already proven and well-known techniques, like its server-client architecture or the connection of related information resources through hypermedia links, constitute an integral benefit for every integration challenge. SOLIOT takes advantage of the Solid interoperability (F1) approach, as a broad range of

Table 5.2: Mapping of Functional Requirements.

	Solid	**SOLIOT**
Interoperability (F1) Shared data:	LDP, RDF, SPARQL Solid Pod	Read/write RDF for IIoT Thing Descr. as the Digital Twin
resource Interactions (F2)	LDP specification, Web Socket Updates, SPARQL	CoAP CRUD operations, MQTT Updates
Manage Notifications (F3)	Linked Data Notifications	MQTT Updates
Invoke Operations (F4)	-	-
Identification (F5)	WebID	Identity and/or Security Token
Authorization (F6)	HTTP/1.1 Authentication	–
Access Policies (F7)	Web Access Control (WAC)	WAC through ACL files
Usage Control (F8)	–	–

Web-compatible applications already exists and users, developers, and administrators are familiar with its characteristics (cf. Table 5.2).

The targeted use case of Solid is an on-hop interaction between a defined Web client as the data consumer and the Solid server acting as the origin server. The Pod is hosted in a Cloud environment, where the Cloud server complies with the Solid specification but can be located anywhere in the Web. The users are either the data sovereign controlling the Pod or consuming applications authorized by the data sovereign. As Solid models the social data of the data sovereign, the Pod can be regarded as the Digital Twin of the data sovereign.

In an IIoT setting, this overlap of roles is not possible anymore. The referent of the Digital Twin is usually not a natural person but a physical object. Still, a (legal) entity must exist which has the authority over the target Asset. An additional difference is that in the basic Solid setting, all components communicating in the network have sufficient resources in terms of computing power, local storage, memory, or provided energy. One can certainly assume that a service hosting Solid Pods, like a typical state of the Art Cloud service, can scale and allocate additional resources as needed. At the same time, the user agent, usually a modern Web browser executed on a regular PC or laptop, is not challenged by the computational or network requirements of Solid (F1.4).

In an IIoT setting, however, constraint devices, battery-powered sensors, network disruptions, etc., are common challenges. Usually, a constant Internet connection cannot be maintained, neither the sending of complex data structures like full-flavored RDF or linked-data graphs. Consequently, SOLIOT-supporting devices cannot be assumed to be at the same network layer as a Solid server (F1.2). Edge or Fog Gateways can serve as suitable bridges between such restricted devices and the rich expressions of SOLIOT. Such gateways serve as lifting and lowering points (F1.3) between the proprietary device communications. They usually lack the rich expressions of the SOLIOT representation and can therefore not directly be integrated with higher-level applications (F1.1). The SOLIOT functionality extends the Edge Gateway service and links the represented resources, overcomes the heterogeneity challenges, and connects what is often known as the shop floor using Operational Technologies (OT) with the Information Technologies (IT) of the office floor.

However, when it comes to the details, several restrictions have been made. While Solid promotes LDP interactions but also enables Web Sockets updates and SPARQL queries, SOLIOT only implements the

	Solid	**SOLIOT**
Trustworthiness (N1)	Depending on the hosting provider	Depending on the hosting provider
Data Privacy (N2)	TLS encryption, Access Control Lists (ACL)	TLS recommended Access Control Lists (ACL)
Integrity (N3)	-	-
Safety (N4)	-	-
Reliability (N5)	-	-
Resilience (N6)	-	-
Scalability (N7)	Decentralized Identity Provisioning, Decoupling of client and server	Depending on the MQTT Broker Decentralized Identity Provisioning Decoupling of client and server
Security (N8)	TLS encryption Access Control Lists (ACL)	MQTT over TLS (not implemented) CoAP with DTLS (not implemented)
Non-repudiation (N10)	-	Not regarded

Table 5.3: Mapping of Non-functional Requirements.

basic CRUD operations (F2). Although these operations follow the LDP guidelines as much as possible, the whole expressiveness must be reduced. Complex queries or even operational calls (F4)—beyond CRUD—are not possible.

Non-Functional Characteristics

Many requirements from OT systems are not supported by Solid as they are not relevant for its use cases (cf. Table 5.3). For instance, code integrity (N3) of the hosting platform, reliability (N5), or resilience (N6) are not objectives for the specification and the reference implementations. Furthermore, Solid itself does not regard physical safety (N4) at all, as no Solid application is intended to physically interact with its users or any other real-world object.

The Solid use cases focus on decentralization of data storage and control. Consequently, privacy (N2) and scalability (N8) of Solid networks are distinguishing features. The intended easy transmission of pods is currently without comparison. However, the current state of the specification [9] does not yet give sufficient guidelines on how one hosting service can ship a Pod to another. Still, in a dynamic IIoT network with potentially changing settings, the ability to adjust the location of resources, and to ship all information seamlessly is an important feature.

The merging of local, decoupled networks with the global internet is one of the most impacting developments. While previously, security and integrity were ensured by restricting the access to sensitive parts of the shop floor, the rising need for interconnections undermines any firewall or otherwise intended separations. Consequently, each component needs to implement the complete security stack, as if it were placed in the open Internet. Such sometimes called ‘Zero Trust’ (N1) approaches demand the same proofs from any connecting party. Data or functionality must be prohibited without proper identity claims

[9] https://github.com/solid/solid-spec (accessed on 10 June 2020)

and whenever not backed up by the ACL rules. This transparent data protection (N2) approach enables the arriving user agents to understand potential obstacles and deficiencies.

Measures to ensure a required Integrity (N3) and Resilience (N6) level are, for instance, proposed by the IDS [36]. Hardware-based trust anchors or virtual monitoring of a server's state, together with certified software modules, can increase the trust level both of the party operating a SOLIOT instance, and the interacting user agent. These concepts are still in their maturation process and are only mentioned here. In general, a guaranteed resilience or integrity level cannot be accomplished without regarding the use case context.

This is certainly also true for Safety (N4) considerations. Neither Solid nor SOLIOT have an out-of-the-box understanding of the meaning—and potential risks—included in their data. Gradual approaches to this issue could be outlier detection or Complex Event Processing engines on top of the SOLIOT notifications. The self-descriptive nature of SOLIOT events, through their RDF-encoded content, certainly lowers the integration effort for third-party tools. However, as such concepts are very use case-specific, they shall not be regarded further here.

In contrast to that, Reliability (N5) can be supported by common methods like periodic backups and load balancing. Such measures can be applied to the SOLIOT hosting instance itself. One important consideration is the distinction between the offered state of a data resource and its internally managed history. A SOLIOT implementation should use transaction-driven data handling for each resource, allowing the seamless reverse of harmful interactions.

As the network surrounding of a SOLIOT instance is not observed by the instance and maybe not observable at all, it must be assumed as compromised by malicious third parties. Therefore, a SLOIOT system must insist on the delivery of valid authentication and authorization proofs from anyone. The recommended pattern is Bearer Tokens from a local OAuth service (N7) with a duration of less than one hour. If the requesting of such dynamic tokens overstrains the abilities of a device, longer or even stable tokens may be accepted. Nevertheless, whenever unsafe interactions are requested, a TLS channel and an additional examination of the local ACL rules are required (N8).

The blank spaces in Table 5.3 are obvious indicators for open research gaps. Even though the idea of the IIoT and the merging of production spaces with open networks is not new, the lack of commonly accepted and implementable technologies is certainly one of the most relevant obstacles for a seamlessly integrated IIoT setting in productive use. Solid and SOLIOT-inspired solutions, like any other currently available, still have significant weaknesses in the outlined areas. Traditional systems can perform better in some respects but usually are highly customized and therefore significantly harder to integrate with other applications.

MQTT and CoAP Protocol Bindings

In terms of the running example, all manufacturers M_1, M_2, and M_3 require a safe and trustworthy but also easily accessible platform for the publishing of their resources. In order to stay open for further clients and use cases, they decide against the implementation of proprietary or customized interfaces but stick to a mature and widespread standard. Let's assume that M_3 has decided to rely on CoAP for its OT communication. In contrast to M_3, the machine park of the Operator is equipped with MQTT sockets and controls its communication over an MQTT broker located in the internal company network. Both require a gateway for hosting the IIoT data as the embedded devices are proprietary and do not allow reconfiguration of their installed software.

They connect their local machines to a gateway server running SOLIOT. The observed resources, the grinding machines, are represented by SOLIOT Assets. As the whole configuration and every available information might be needed for future applications, all events and states are collected at the SOLIOT

Table 5.4: Protocol Bindings: SOLIOT in MQTT and CoAP.

	HTTP	**MQTT**	**CoAP**
Interaction Type (P1)	Synchronous, request/response	Asynchronous, publish/subscribe	Synchronous, request/response
Interaction Announcement (P2)	Allow & WAC-Allow Hdrs announce avail. operations	TD Affordance	TD Affordance, WoT CoAP Binding
Resource Discovery	Link Header, HATEOAS	Topics under soliot/#	/.well-known/soliot HATEOAS
Operations (P3)	RESTful CRUD/SPARQL	Event notifications	RESTful CRUD
Server-Client Coupling (P4)	Decoupled, opt. coupled through Web Sockets	decoupled	Central Message Broker
Efforts at device -side (P5)	High (HTTP stack)	low	low
Bandwidth (P6)	High	Low to medium	low
Energy consumption (P7)	High	Low to medium	low
Latency (P8)	Low priority	Real-time Control	near Real-time
Media Types (P9)	Turtle, JSON-LD, binary...	JSON-LD	JSON-LD

server. However, as the created dataset allows insights into their respective operational activities and even might uncover their competitive advantages and technical know-how, access has to be restricted.

The following examination about protocol bindings and data representations is highly related to the design decisions of an IIoT system. In contrast to the previous sections, only the identified challenges and issues are outlined. As Solid's deep integration with HTTP presents a severe challenge for resource-restricted environments, the main focus is applied on the concept transfer and interaction mapping to the mentioned, less-demanding protocols. Naik has already examined the resource demands of IIoT protocols [256]. The results are presented in Table 5.4. Obviously, CoAP has the lowest requirements of the selected protocols, followed by MQTT.

Lin et al. [4] argue that the single focus on core OT requirements is not sufficient anymore. The adoption of IT practices, including HTTP and its rich client landscape, can introduce a game-changer for the usage experience in IIoT applications. While embedded and restricted devices may still be limited to particular protocols, support of HTTP at the Edge or beyond may solve a huge set of interoperability problems.

A first mapping of LDP interactions to CoAP has been created by Loseto et al. [141]. Extending this mapping to Solid functionalities serves as the foundation of the SOLIOT CoAP protocol binding explained in this section. The binding itself is outlined through a comparison of interaction patterns (cf. Table 5.5), a translation of Solid header definitions (cf. Table 5.6), and a discussion on the different interpretations of the message body interpretation.

Table 5.5 distinguishes the different interaction intentions relevant to an IIoT scenario, as explained in the example. Some of the thereby included patterns (subscriptions, notifications, alarms) are not part of the core Solid use case. Still, as for instance Kovatsch et al. [62] or Bassi et al. [242] outline, a limited request/response approach is not sufficient. The design of the table reflects this insight, therefore also

Table 5.5: Interaction methods with CoAP, based on Loseto et al. [141].

Interaction	Mandatory	HTTP Method	CoAP Method	CoAP Status Code
Create	no	PUT/POST	put/post	2.01 Created
Read Resource	yes	GET	get	2.05 Content
Read Metadata	yes	HEAD	get ?ldp=head	2.03 Valid
Read Operations	yes	OPTIONS	get ?ldp=options	2.05 Content
Extend	no	PATCH	put ?ldp=patch	2.04 Changed
Overwrite	no	PUT	put	2.04 Changed
Delete	no	DELETE	delete	2.02 Deleted
Subscribe	no	–	–	–
Unsubscribe	no	–	–	–
Notify	no	polling?	–	–
Alarm	no	polling?	–	–
Invoke Functions	no	–	–	–

Table 5.6: Protocol Headers, similar to Loseto et al. [141].

Function	HTTP Header	CoAP Option
Data Serialization	Content-Type	Content-Format (ct) option
Allowed Operations	Allow	(JSON in Response Body: TD Affordance)
Server-supported Media Types	Accept-Post	(JSON in Response Body: TD Affordance)
Server-accepted Media Types	Accept-Patch	(JSON in Response Body: TD Affordance)
Proposed Resource Name	Slug	title
Resource Location	Location	location-path
Security Token	Bearer Token	(custom Security Token option)

containing white spaces. These white spaces (e.g., row ‘invoke functions’) can be relevant for further specifications and therefore have been kept intentionally in the tables.

Obviously, the amount of protocol methods is lower for CoAP. As Loseto et al. [141] suggested, the replacement of the HTTP HEAD, OPTIONS, and PATCH headers can be expressed through query parameters. While one can argue that the core CRUD operations are still natively possible through the given CoAP methods (get, put, post, delete), the mandatory requirement for HEAD and OPTION support in Solid makes a compromise necessary. As CoAP’s main intention is the reduction of resource consumption, several critical requirements of an IIoT system must be disregarded. For instance, the limitation of the protocol headers enforces the expression of several information points in the message body, increasing the integration effort at the client.

The CoAP headers are significantly different from the originally used ones through the HTTP binding. CoAP headers are encoded as numbers to reduce the message size. Important headers, for instance ‘Accept’ (‘17’) or ‘Content-Type’ (called Content-Format: ‘12’), are registered at IANA. In addition, common media types are globally defined as well (‘41’: ‘application/xml’, ‘50’: ‘application/json’). SOLIOT promotes the usage of the reserved media type code ‘432’: ‘application/td+json’ for Thing Descriptions. The resulting content is a JSON-LD serialized RDF representation of an Asset.

Different from CoAP and HTTP, MQTT is a message-based protocol and event-driven. While it is possible to realize all listed interactions through explicit descriptions in the message body—for instance,

a READ request could be encoded as an especially formatted JSON attribute—such an approach would shift the interpretation task to every receiver of a message. The additional effort to parse and interpret the complete content before realizing its meaning seems not an efficient pattern.

Therefore, a three-stage process is applied to encode messages. The first step is realized by the native MQTT method itself. PUBLISH, SUBSCRIBE, and UNSUBSCRIBE define the native separation of messages and are interpreted by the message broker without regarding the rest of the message (cf. Table 5.7). The second step is specified by the root part of the used topic. The reserved upper-level topic *soliot* tells the involved parties that the following message corresponds to the SOLIOT interaction semantics. The distinct interaction is then listed at the second position of the topic string. It is noteworthy that for the SUBSCRIBE and UNSUBSCRIBE calls, the variability with the wildcard at the second position is reflected. The meaning of the operation is already completely defined by the combination of the method and the *soliot* topic, and therefore no further information needs to be transmitted.

The data flow is also already defined at this stage. The SOLIOT servers—origin servers acting as MQTT clients—react to the unsafe interactions 'Extend' ('soliot/patch/#') and 'Overwrite' ('soliot/put/#'). These messages originate at the user agents—also MQTT clients—and must contain a valid JSON-LD object. The SOLIOT server is therefore required to also subscribe to all topics related to its own MQTT-enabled resources. These resources are placed at the third part of the topic sequence through their base64-encoded URI identifier. However, even though receiving a change request, the origin server may or may not actually change the targeted resource.

The MQTT Message Brokers as proxies are responsible for handling the core MQTT methods PUBLISH, SUBSCRIBE, and UNSUBSCRIBE. It is recommended that the message broker enforces encrypted communication (MQTT over TLS using default port 8883). Additional protection can be gained by maintaining access-control lists directly at the message broker and restrict access to known users. Using ACLs and user authentication at the MQTT Broker, however, creates redundancy with the native SOLIOT access control scheme using Web Access Control and maintaining the permissions directly at the resource.

User agents, also appearing as MQTT clients, signal resource changes to the origin server ('Extend' or 'Overwrite'). This usually applies when the clients get informed through the resource itself, for instance, a sensor sending a new observation. The user agent must regard the 'Extend' interaction as not idempotent, meaning that repeatedly submitting the same event will create an inconsistent state at the origin server. In addition, notifications and alarms are triggered by the origin server. In both cases, the subscribed MQTT clients are the information sinks. Similar to the update interactions, the server can use the topics 'soliot/patched/#' to only send the added attributes or 'soliot/putted/#' to send the complete new state.

In contrast to CoAP, the MQTT binding only regards the event-driven interactions. Create, Read, and Delete are not regarded as their context is state-based, therefore being more effectively implemented by either CoAP or HTTP. Furthermore, Tables 5.5 and 5.7 both do not specify the invocations of remote functions or services. Even though this functionality is mentioned by several requirement lists, the exact semantic of such remote service calls are highly use case-specific. For now, the given protocol bindings for their description and execution are intentionally left open for future work.

The *headers* of MQTT are not intended for the transfer of rich interaction information. For instance, a typical MQTT client receiving information cannot understand or look up the origin server's offers or capabilities. Nevertheless, the demand to receive information on the message broker's current state has led to the convention of the top-level topic '$SYS'. A similar pattern would be possible for SOLIOT look-ups. The event-based nature of MQTT, however, speaks against this approach. The continuous flooding of the same origin server descriptions, independent of any demand, usually only blocks important resources without added value. Whenever a client is interested in the origin server's state, it should use a state-driven protocol as CoAP or HTTP.

Table 5.7: Interaction methods with MQTT, based on Loseto et al. [141].

Interaction	Mandatory	MQTT Method	Status Code
Create	no	–	–
Read Resource	yes	–	–
Read Metadata	yes	–	–
Read Operations	yes	–	–
Extend	no	publish soliot/patch/{resource}	–
Overwrite	no	publish soliot/put/{resource}	–
Delete	no	–	–
Subscribe	no	subscribe soliot/+/{resource}	–
Unsubscribe	no	unsubscribe soliot/+/{resource}	–
Notify	no	publish soliot/{interaction}/{resource}	–
Alarm	no	publish soliot/alarm/{resource}	–
Invoke Functions	no	–	–

The *message body* must again contain a valid JSON-LD object with exactly one root resource. The URI of the root node must be either stated explicitly—and be identical to the decoded resource URI of the message topic. Alternatively, a relative URI allowing a more flexible scheme requires the receiver of the message to relate the content with the resource stated in the topic. Using a differing topic but specifying the actually intended resource solely in the JSON-LD payload is therefore not allowed.

Two basic architecture designs are possible. The SOLIOT server can host an MQTT message broker and expose its endpoint to subscribing clients. Alternatively, the SOLIOT application only implements an MQTT client socket and distributes the notifications using an external MQTT message broker. The obvious advantage is a reduced computing load on the SOLIOT instance and increased scalability through the usage of several Message Brokers. However, the access control—moreover, any kind of data protection—is thereby transferred to the MQTT Broker and not under the control of the provider anymore. An MQTT Broker may or may not take the respective ACL files into account, so the providing SOLIOT application cannot be certain. While this challenge certainly can be solved through additional mechanisms, for instance, exchanging encrypted payloads and at the same time handing out proper decryption keys only to authorized subscribers [257], this shall only be briefly mentioned at this position. Still, the outlined issues are highly relevant but regarded as out of scope for explaining the core of the SOLIOT approach.

The next requirement is the choice of the appropriate media type (**P9**). RDF and its serializations have been designed with a clear focus on interoperability and self-description. The schema-less characteristics of RDF objects allow its flexible usage and simple extension at runtime. As the commonly appearing data files are in the kilobyte to low megabyte range, this is usually problematic for both Web servers and clients. However, in the regarded use case, this is obviously a significant challenge. One way to solve it is to examine the different serialization formats, namely RDF/XML, NTriples, Turtle, and JSON-LD.

A differentiation must be made for JSON and its RDF variant, JSON-LD. A plain encoding of RDF with completely serialized URIs as JSON keys and values requires much space and is therefore not suitable. Namespace replacements through the '@context' element reduce the required space significantly but increase the parsing effort. Receiving parties need to look up the reference if they do not know it yet, resulting in probably even higher traffic. The current context of the 'Thing Description' alone has more than 22,000 bytes. Still, the broad usage of JSON justifies this trade-off and makes it the

Table 5.8: Data Representation Mapping.

	Solid	**SOLIOT MQTT/CoAP**
Res. Identifier (D1)	HTTP(S) URL	Topic ID, CoAP URL
Resource Locator (D2)	HTTP(S) URL	CoAP URL, TD Affordance
Information Model (D3) generic (D3.1) domain-specific (D3.2)	RDF, RDFS, LDP pim/space, posix/stat DCTERMS, FOAF VCARD	RDF, RDFS, LDP Thing Description SSN, SOSA, SAREF, (ECLASS, IOF, IEC CDD)
Self-Description (D4)	</profile/card#me> as a Person	</assets/asset#me> as a Digital Twin
Interaction Descr. (D5)	HTTP Headers (Allow, Allow-Post, Allow-Patch, WAC-Allow)	TD Affordances
Security Profiles (D6)	WAC in ACL files	WAC in ACL files
Data Schemes (D7)	–	(SHACL)

recommended format. To face the context issue, SOLIOT places a respective JSON file at the reserved path '/things/context.jsonld'.

The further identified requirements of protocol bindings (Bandwidth (P6) and Latency (P8)) are discussed in Section 5.4.7. Energy Consumption (P7) is by its nature hard to measure for a concept such as SOLIOT and was not examined separately. It is assumed that the bandwidth is a sufficient indicator for the power demand of a SOLIOT implementation.

Data Representation

Solid's data model is oriented by the human user and its attributes. IIoT devices require a different vocabulary and data scheme (cf. Table 5.8). The Thing Description (TD) ontology, Semantic Sensor Network (SSN), and Smart Appliances Reference (SAREF) ontology are only examples of the rich set of vocabularies and ontologies recently designed for the domain. ECLASS and the IEC Common Data Dictionary (IEC CDD) include rich sets of non-RDF classes and attributes. While a broad range of terms and concepts is given, the actual integration between different parties and organizations requires more restrictions and selection of supported entities.

The LDP container model allows the usage of any correctly structured RDF, whether the property or entity is known to the server or not. In an IIoT system, the host must restrict the variety to some degree. Therefore, schemes and shapes such as SHEX or SHACL can be used to validate incoming information. Alternatively, the server can provide extended mapping capabilities and ensures the provisioning of known terms in its own authority. Nevertheless, this step, if executed autonomously, is error-prone and therefore not feasible in an operating environment.

Consequently, using Solid in an IIoT environment requires additional restrictions and validation mechanisms, even though this limits the expressiveness of the provided information. Using the Thing Description vocabulary (namespace prefix 'td'[10]) as the Core Vocabulary (D3.1) in combination with the Linked Data Platform annotations, the instances of td:Thing are the top-level entities in the reserved top-level container '/assets/' (cf. Figure 5.18). The identifying URI of such an instance must be composed

[10]https://www.w3.org/2019/wot/td# (accessed on 10 June 2020)

of the authority of the server, '/assets/', and a locally unique character sequence. The Asset is treated and described as an ldp:BasicContainer, and implements the thereby required interaction behavior. Alternatively, a relative URI only using the locally unique character sequence can be applied. The full URI of the td:Thing is implicitly set regarding the location on the hosting server.

An instance of td:Thing contains at least a minimal self-description (D4) outlining its type (class relations by *rdf:type*), human-readable annotations in English (*rdfs:label*, *rdfs:comment*), and potentially more languages. It can link to further resources using three different patterns. Datatype Properties are contained directly in the document representing the instance. Object Properties referencing external resources shall also be contained in this RDF document but limiting the provided information to the Object Property and referenced object URI. The origin server shall shift the decision to the client, whether it wants more information of that object entity. If so, the client resolves the identifier in a linked-data conform way and discovers the references resource itself.

The third pattern is the reference to locally hosted resources. The instance of td:Thing acts as an ldp:BasicContainer, using relative URLs to point to the intended target. Furthermore, it outlines the existence of child resources through ldp:contains properties. It is recommended to add a property to tell the client which kind of relation the td:Thing connects to its child. Listing 5.1 contains a representation of the intended Digital Twin model. Please note that despite that JSON-LD is the recommended format, the Turtle serialization has been chosen to increase the readability of the example.

Listing 5.1: An IIoT Asset presented as a SOLIOT Digital Twin.

```
<coap://18.157.197.66:5683/assets/robot/>
   a td:Thing, ldp:BasicContainer, saref:Device ;
   rdfs:label "Robot Gripper Arm"^^xsd:string ;
   td:title "Robot Gripper Arm"^^xsd:string ;
   td:description "Moves assets from on eplace to another."@en ;
   td:security <./.acl> ;
   td:actions <./affordanceCreate>, <./affordanceDelete> ;
   td:events <./affordanceNotification> ;
   ldp:contains <./120363/>, <./affordanceCreate>, <./affordanceNotification>,
   [...]
.
```

All enabled interaction possibilities (D5) are also expressed using Thing Description Affordances. In addition to the discovery of interaction patterns through the linked-data specifications using HTTP Header, the affordances bridge the gap to the non-HTTP protocol bindings. A client, unaffected by its used protocol, can thereby find the related resources, respective endpoints with protocol specifications and look up the necessary input and provided output details. For CoAP, the CoRE Link Format further standardizes the principles of discoverable resources in RFC 6690. These attributes are used to reflect the HTTP headers wherever possible.

Domain Vocabularies (D3.2) for now are the Semantic Sensor Network (SSN/SOSA) ontology[11] and the Smart Appliances REFerence (SAREF) ontology[12]. Further descriptions of concepts and aspects originating from the description of core manufacturing, logistics, or engineering attributes shall be used from the Asset Administration Shell ontology [19] and the currently developed Industrial Ontology Foundry (IOF) [258]. Furthermore, the rich domain vocabularies maintained by ECLASS [259] and IEC CDD (IEC 61360) define lightweight concept definitions of huge term sets using IRDIs. The currently undergoing efforts to deliver ECLASS also through linked-data principles as RDF Resources and URI identifiers promise an even simpler integration.

[11] https://www.w3.org/TR/vocab-ssn/ (accessed on 10 June 2020)
[12] https://ontology.tno.nl/saref/ (accessed on 10 June 2020)

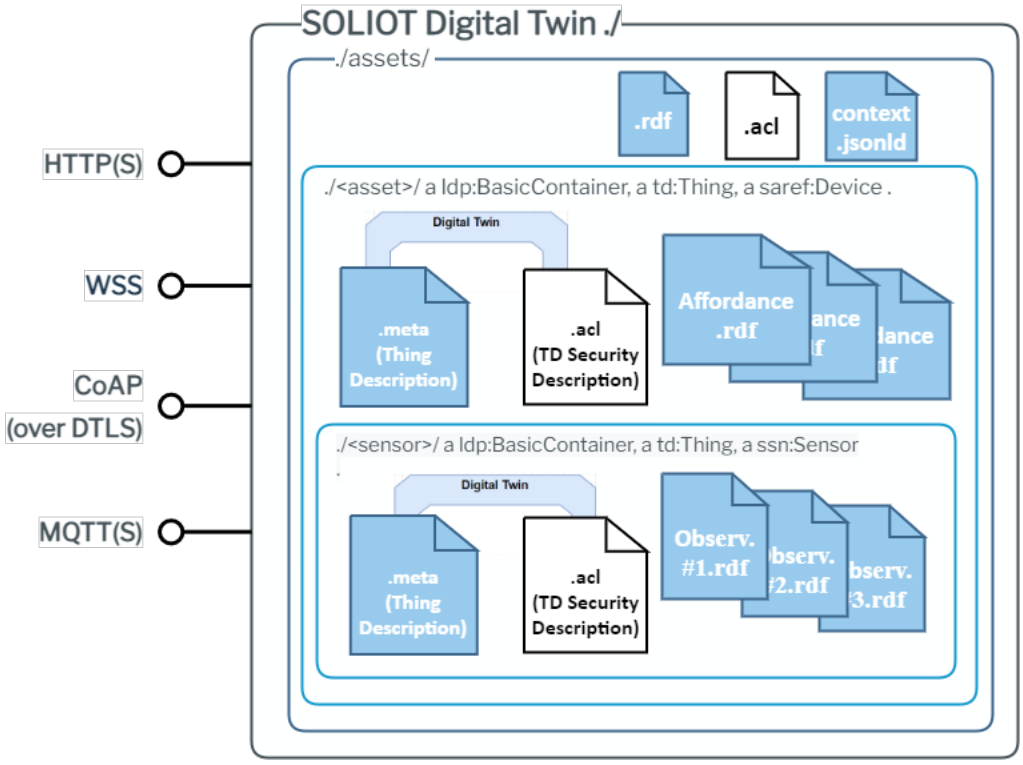


Figure 5.18: Representation of the Digital Twin through interlinked Containers, Resources, and ACL Files in a SOLIOT instance.

The security profiles (D6) are represented in the same way as proposed by Solid. ACL files containing Web Access Control statements can be positioned at the resources and containers to describe and at the same time configure a Role-based Access Control model. However, a SOLIOT instance should not rely on WebIDs, as its communication partners may not be able to prove their identity in this way. Shorter tokens, especially JSON Web Tokens with encoded claims through JSON-LD, can compose a suitable trade-off.

A further restriction of data intersections is composed through SHACL shapes. These shapes serve two purposes. They directly address the need for data validation and schema restrictions (D7) but also, as readable resources themselves, describe the expected or offered data. A sending client can thereby understand the expected data structure beforehand and even test its outgoing resources locally. A receiving client can adjust its own expectations by analyzing the shapes and derive a better forecast of the resulting resources.

The resulting virtual presentation of the Digital Twin is, therefore, as shown in Listing 5.1, formally presented through a ‘td:Thing’ as defined by the Thing Description ontology. The thereby incorporated machine-readable properties and attributes allow a client to autonomously interpret its capabilities, given that it is familiar with the vocabulary. If not, the client can also dereference the identifiers and further discover their meaning on the fly. The interaction model is further outlined through the presentation as an ‘ldp:BasicContainer’ and the ‘td:actions’ and ‘td:events’ attributes. These provide the client all information necessary to interact with the SOLIOT Digital Twin. Moreover, also every REST and HATEOAS-aware client can consume the complete representation, manipulate attributes and further discover the related sub-resources.

Usage Control Model

The Usage Control language from Section 4.4 contains the necessary concepts and attributes to describe rich usage permissions and how to enforce them. These concepts have been created within the IDS Usage Policy Language and follow the very holistic scope of an overall usage control perspective. The Solid promoted data protection solution – the *Web Access Control* language (WAC, [109]) – differs slightly. First, its scope is restricted to pure access control. The expressiveness of the protection rules, an *Access Control List* (ACL) or *acl:Authorization*, ends as soon as the data resource is requested by the consumer. Following usage activities or even the further distribution of the resource is not reflected.

Nevertheless, both approaches follow the same identification pattern. The data resources and the Authorization (Usage Contract) related to them follow the URI schema, usually as HTTP URIs. While ACLs, in general, are also used in many other environments and are open to any – also non-URI – identifier scheme, the special usage of Web Access Control language in Solid requires compliance with the LDP specification. *acl:Authorizations* determine the permitted RESTful operations on the (LDP) target resource and can also be accessed as their own LDP Resources. As such, one can even think about *acl:Authorizations* managing the access to other *acl:Authorizations*, something which is not possible for IDS Usage Contracts.

The WAC language does not allow the specification of a resource owner. This is the same approach as in the IDS model. The authorized entity that has the technical ability to control the resource itself and define its usage permissions is implicitly assumed to be the sovereign of this resource. This procedure somehow solves the challenge of ownership by relying on the technical capabilities of users or user agents. One has to note that such indirect proceedings are necessary for ownership concepts – as for physical goods – but not possible or applicable for digital data. The thereby created implication is that the Data Sovereign is the one who has the technical capability and permission to manage the access to a resource and that the access permissions and control rights are granted to the Data Sovereign. Obviously, this logical circle requires more in-depth examination.

One further challenge is the definition of user groups. The range starts with a group of the size one, which is equivalent to an individual user. In a WAC statement, this user is identified by its WebID. On the other end of the possible descriptions is the group including everyone, which is expressed by the class *foaf:Agent*. As everyone, either human or non-human, organization or individual, is by definition an instance of *foaf:Agent*, any permission assigned to this class also applies to *everyone*.

Between these somehow trivial cases are the more relevant defined groups, for instance, the members of a certain organization or company. The recommended encoding for such groups is the usage of URLs. The Web resource of the group URL shall provide a list of its members. The access management service, therefore, needs to dereference the group identifier and thereby receives all required information to accept or decline the access request. The IDS handles such groupings completely differently. Its business-to-business focus leads to the view of all users or organizations as instances of the generic *ids:Participant* class. Recursive declarations allow then the descriptions of memberships. A department is modeled as a Participant, who is part of the bigger company, also an IDS Participant.

The core challenge for both approaches is the clarification of whether or not a user is a valid member of a group or organization. In the terminology of XACML, a Policy Information Point for this type of information is needed. The WAC specification solves it by using the identifier also as the reference to the PIP interface - the group URI is also the pointer to the Web resource containing its members. That is not possible in the IDS, where membership information requires the highest protection level and must not be open to arbitrary Web requests. The respective PIPs in IDS ecosystems are therefore globally known components like the *Participant Information Service* or company-specific like an LDAP system.

Defined by the scope of their language, the Web *Access* Control language can only describe the limited

amount of access operations - read, write, append, and control. The IDS Usage Control language must also express activities after the granted request, for instance to not distribute, log, notify about usages, or even delete the resource from the receiving system. Obviously, such obligations cannot be ensured by the providing server but require a trusted system at the receiving party. While the IDS includes specifications and guarantees to achieve this, it is not in the scope of Solid or the WAC language.

An additional difference can be found in the inheritance regime of the WAC and the IDS Usage Control language. Solid ACL files are interpreted relative to their location in *LDP Containers*. That means that the same permissions apply for contained containers and resources. As the IDS does not follow a strict container model, such passing of rules is not possible. Each *ids:Resource* needs its own assigned usage policy. Nevertheless, the IDS follows a similar concept when it comes to the appearance of a data Asset. By default, policies specified for the *ids:Resource* are propagated through the related *ids:Representations* to the final *ids:Artifacts*. Solid misses this differentiation and consequently has only one way to describe the target data Asset.

Another significant difference lies in the embedding into HTTP headers for the runtime discovery of ACL files. As the IDS supports several protocol bindings, the limitations to RESTful discovery operations (GET, HEAD) is not sufficient. Therefore, potential consumers need to use infrastructure services, in particular the IDS Metadata Broker, or directly the resource self-description at the hosting connector to find the applying Usage Policies. As the Solid interactions are only enabled through the LDP operations, the discovery mechanism using *Link* headers is sufficient, and no third-party component is required.

The presented system combines the ACL approach of Solid, encoded in the Web Access Control language, with the more expressive IDS Usage Contracts. SOLIOT covers the required characteristics of Digital Twins, the Digital Twin, through self-described data representations. Furthermore, the complete interaction model is explicitly outlined, giving both a consuming application and a producing data source all necessary information. As all related actors communicate relying on the same interaction patterns—even though applying the protocol of their choice—the overall complexity of the interfaces is reduced significantly. This second requirement for Digital Twins, the interaction between the Asset and the Digital Twin, is therefore solved in the same manner as the outside communication. This increases the scalability and maintainability of a SOLIOT-based network and, at the same time, reduces the integration effort of higher-level applications.

However, the Asset itself is less regarded by the proposed model. To truly connect the physical with the virtual world, not only the virtual part of a Digital Twin needs to address the Asset but also vice versa. The common use case is the dereferencing of an Asset's physical identifier. One could expect that each Asset directly refers to its virtual counterpart. This is still not yet sufficiently examined. Challenges are, for instance, the existence of more than one digital representation for a single Asset. Furthermore, a physical identifier must stay applicable throughout the complete lifecycle of the underlying Asset, which can be decades for many production-related Assets. Neither SOLIOT nor any other Digital Twin model currently supports this kind of durability. While the Asset Administration Shell describes the first steps in this direction using nameplates [12], the overall challenge remains unsolved.

5.4.4 Architecture and Prototypical Implementation

The proposed SOLIOT approach serves as an Edge Gateway between the production networks and the open Internet. Similar to Solid, the related entity is described by its profile and presented as an LDP Resource. However, the extension of the supported protocols to CoAP and MQTT added IIoT-specific terms and concepts and designed tailored interaction patterns. Widely deployed architectures rely on a distinct integration layer or specialized gateways connecting the data-providing Assets with the data-consuming applications [8, 70, 116, 242]. Figure 5.20a shows a simplified representation of this

concept, where the Digital Twins appear at the Edge Layer (also called integration layer, IIoT platform, gateway depending on the context). The robot from the example is deployed in the shop floor network, and its data output is hosted on the gateway. The operator of this gateway is responsible for granting or denying access, lift information, and manage incoming requests. The basic architecture of SOLIOT is closer to the approach of Pfrommer et al. [254] or Jammes and Smit [190] as an equal node on the same network level as the IIoT device and the (external) consumers (cf. Figure 5.20b).

SOLIOT does not make any differences between the connected components. Theoretically, also the Data Analytics module of the Analyst could represent itself as a Digital Twin on a SOLIOT server. The distinction between data producers and consumers is therefore not specified by their role in an interaction but their applied permissions. The permissions themselves are also accessible resources (ACL files in Figure 5.20b). This follows the vision of reducing the network barriers, merging shop floor networks with office floor applications and even the public Internet. While both the Data Analyst and the robot manufacturer have entries in the ACL policies, this is not the fact for M_1. Consequently, his access requests will be denied. Nevertheless, the dissolution of network barriers also exposes previously shielded areas and devices, requiring proper security and protection mechanisms. However, following the 'zero trust' argumentation, the increasing degree of interconnections, required gateways, and introduced foreign applications make a network-based security approach more and more challenging.

Figure 5.19: Proof of concept for the SOLIOT concept[13]

SOLIOT has been prototypically implemented as an open-source project [13] under an Apache 2.0 license. The NodeJS reference implementation of Solid serves as the core server, extended with a CoAP server [14] and MQTT [15] capabilities. The SOLIOT reference server provides RESTful, LDP-conform interactions for CoAP clients and resolves event-based interactions using its MQTT adapter. Currently, an external Mosquitto MQTT message broker [16] collects and distributes MQTT messages.

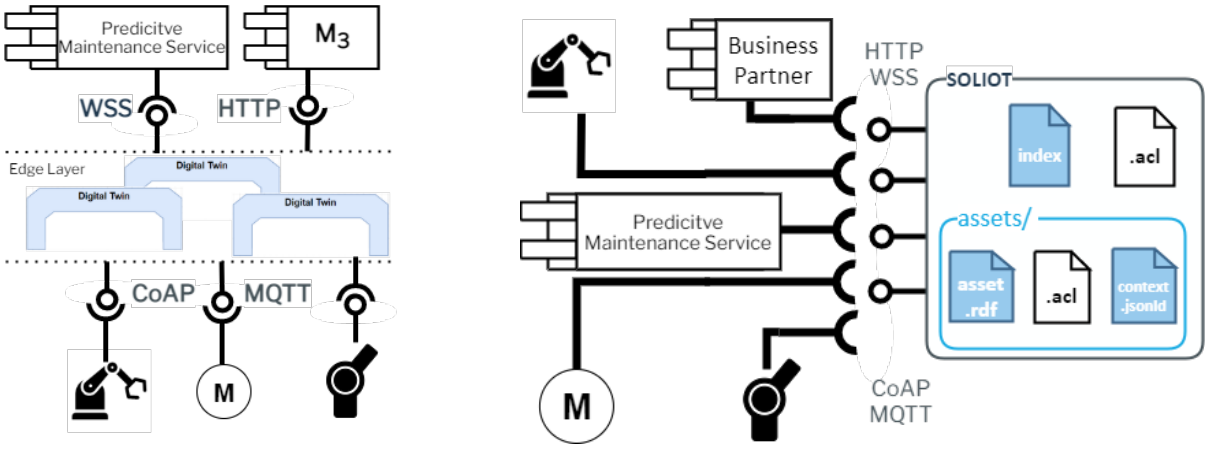
5.4.5 Interactions and Protocols

For the prototypical implementation, CoAP as a lightweight request–response protocol and MQTT as a publish-subscribe implementation have been selected. Respective endpoints for each protocol socket need to be opened, by default 1883 for MQTT and 5683 for CoAP.

At each stage of the Digital Twin container hierarchy, the possible interactions are described through TD affordances. The affordances themselves impose regular RDF resources and are treated as LDP children of their describing container. As such, the container links to them both through *td:actions*/*td:events* and *ldp:contains* attributes.

In terms of MQTT interactions, several architectures are possible. For (a) point-to-point communication, an origin server could push its state to an MQTT Broker at the user agent. The user agent also deploys a client subscribing to its own broker, thereby receiving the sent message. Alternatively, the broker connector could be placed directly at the origin server, and all user agents interested in the resource state are subscribed there (b). Any event is then published by the origin server's internal MQTT

[13] https://github.com/sebbader/SolIoT
[14] https://github.com/mcollina/node-coap (accessed on 10 June 2020)
[15] https://github.com/mqttjs/MQTT.js (accessed on 10 June 2020)
[16] https://mosquitto.org/ (accessed on 10 June 2020)



a Generalized Edge Architecture

b SOLIOT Network

Figure 5.20: Architecture Comparison.

client connector to this broker. This approach would, for instance, allow the origin server to control which user agent receives the messages.

However, both approaches inherit significant drawbacks. In case (a), the origin server must know the user agent beforehand, which is an unrealistic assumption in a dynamic IIoT network. Approach (b) requires that the device creating the data hosts an additional broker component. Embedded devices may not have the computing power or energy resources to do so. Furthermore, 1-to-m, n-to-1, or 1-to-1 interactions as imposed by both (a) and (b) do not consider the strengths of MQTT but misuse the protocol to some degree. Communication patterns such as that can be implemented easier with other protocols and less overhead.

Regarding the publish/subscribe pattern of MQTT, the assigned broker should be deployed outside of the SOLIOT instance but also of the origin server. The SOLIOT server only implements an MQTT client connector. In contrast to a state-based view of HTTP and CoAP, none of the CRUD interactions have been implemented yet. The information flow is, therefore, currently only supported from the SOLIOT origin server to the user agent. The next iterations of SOLIOT will address this issue and extend the current model unsafe interactions. This way, Assets with client sockets will be enabled to directly influence their own Digital Twin's Thing Description.

5.4.6 Use Case Scenario

The Operator and Data Analyst connect their local machines via CoAP or MQTT. The SOLIOT instances are the only providers waiting for incoming requests. These requests must follow Solid specifications and are therefore easy to implement for any connecting party.

The administrators responsible for *O*'s shop floor initialize pods for all devices, hosted sensors, machines, or even complete facilities relevant for their operation. They create according profiles, deposit the master data and configure the security regime. One part is the storage of identification information to let the SOLIOT server know the addresses of the incoming data. Furthermore, the .acl files are adjusted so that all administrators obtain full control, internal systems gain read and write permissions where needed, and external applications only can access the negotiated information. They do this through the same interaction patterns (Read: GET, Update: PUT, etc.) as the clients at runtime.

As soon as the system has been set up, the local sensors push their observations via their preferred

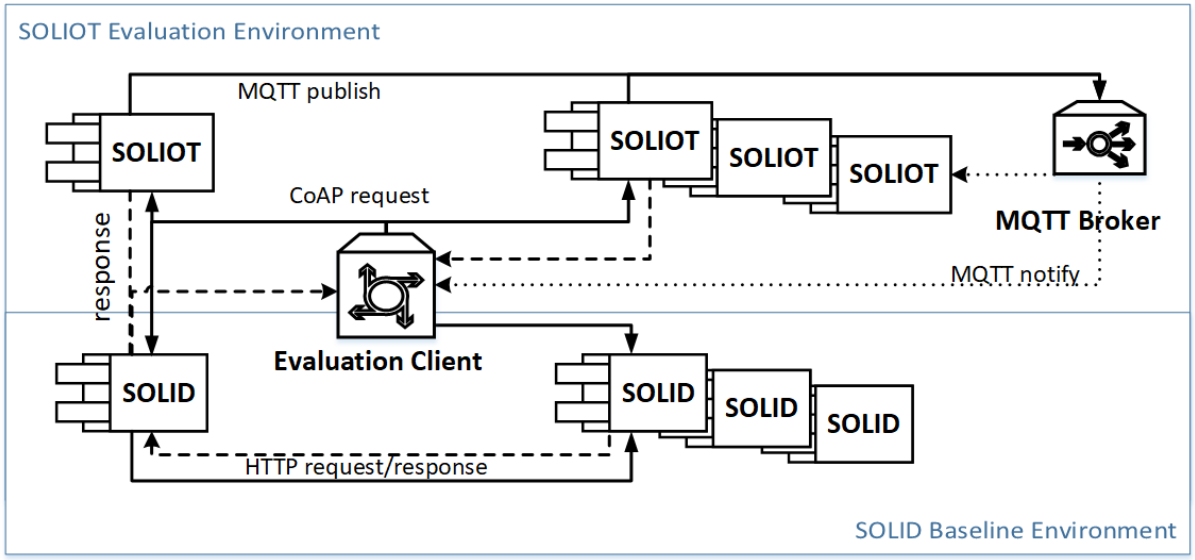


Figure 5.21: Information flow in the evaluation setting. The use case is implemented with SOLIOT (top) and equivalently with plain Solid instances (bottom).

protocol to the SOLIOT server. For supplier A, the devices send CoAP PUT messages with the measured values in JSON. The SOLIOT server validates the token and maps and validates the content. If the server can process the message, the according attribute in the sensor profile is updated.

5.4.7 Evaluation

At the core of the proposed SOLIOT approach is the combination of the uniform interfaces of the LDP specification (interoperability requirement) with the mature access control regime (data control) as defined by Solid with the restricted resources of IIoT applications (constraint devices). Evaluating these stated contributions target different dimensions and therefore requires different methods. As the interoperability features are heavily relying on the provided Solid specifications, a distinct validation of their features would only provide a minimal added value. It has been decided to supply proof of concept also as a deployed sandbox instance at (https|mqtt|coap)://18.157.197.66:(8443|8883|5683). The typical ports for HTTPS (8443), CoAP (5683), and MQTT (1883) are used. The open codebase allows the interested reader to verify the created endpoints implementation and examine the security implementation.

The evaluation setup relies on Amazon AWS EC2 instances. The prototypical SOLIOT server has been shipped to T2 Micro instances and deployed using the AWS Client API. This way, the fast deployment of many SOLIOT servers is possible. All instances have their own hostname and thereby act as distinguishable nodes in the evaluation network. Currently, all instances have been equipped with one Thing Description representing one single device. As Solid, and therefore also SOLIOT, maintains the resources as files on the local disk instead of keeping it in its local RAM, it is assumed that more resources would not significantly affect the performance behavior.

Each instance has an allocated disk space of 8 GB, which limits in no test scenario have been reached. The complete identical setting was used for Solid reference servers as baselines. Both the SOLIOT and Solid test instances asynchronously report information on incoming and outgoing messages to one central Web server. This server collects and persists the performance results, which serve as the inputs for the following assessments.

Table 5.9: Performance measures for one distinct Solid and SOLIOT instance with 1 and 10 clients requesting in parallel.

Criteria	Solid, 1 Cli.	SOLIOT, 1 Cli.	Solid, 10 Cli.	SOLIOT, 10 Cli.	Setting
Start Time[17]	1109 s	1580 s			(column denotes
Disk Size	225 MB	257 MB			requests/client)
GET @Client	224 ms	123 ms	242 ms	132 ms	1000 requests
GET @Server	10 ms	20 ms	19 ms	18 ms	(Solid: HTTP,
Bandwidth GET	184 Bytes	338 Bytes	184 Bytes	338 Bytes	SOLIOT: CoAP)
Bandwidth Resp.	2845 Bytes	1988 Bytes	2845 Bytes	1988 Bytes	
POST @Client	215 ms	89 ms	284 ms	80 ms	1000 requests
POST @Server	7 ms	11 ms	9 ms	9 ms	(Solid: HTTP,
Bandwidth POST	828 Bytes	908 Bytes	828 Bytes	908 Bytes	SOLIOT: CoAP)
Bandwidth Resp.	772 Bytes	0 Bytes	772 Bytes	0 Bytes	
PUT @Client	221 ms	72 ms	287 ms	80 ms	1000 requests
PUT @Server	2 ms	15 ms	4 ms	15 ms	(Solid: HTTP,
Bandwidth PUT	810 Bytes	980 Bytes	810 Bytes	980 Bytes	SOLIOT: CoAP)
Bandwidth Resp.	704 Bytes	0 Bytes	704 Bytes	0 Bytes	
DELETE @Client	226 ms	64 ms	283 ms	72 ms	1000 requests
DELETE @Server	4 ms	9 ms	5 ms	10 ms	(Solid: HTTP,
Bandwidth DEL.	195 Bytes	406 Bytes	195 Bytes	406 Bytes	SOLIOT: CoAP)
Bandwidth Resp.	942 Bytes	250 Bytes	942 Bytes	250 Bytes	
Publish (MQTT)					
Latency Client	–	0.11 ms	–	0.08 ms	200 events
Bandwidth Pub.	–	675 Bytes	–	675 Bytes	(SOLIOT: MQTT)

The instances without data files require around 257 MB, where most (243 MB) is used for external libraries. The exemplary Digital Twin consists of 32 Turtle files (88 KB), within a total of 376 triples. The container structure, reflecting in many different files and folders also on the disk, increases the amount of required storage space. However, the management of each RDF subject in one file simplifies the discovery look-ups and simplifies the discovery of relations to additional resources for the clients.

The available memory (0.5 and 1.0 GB) and computing power of the EC2 Micro and Nano instances are sufficient for all tests. A memory usage above 50% of the available amount was never observed. Still, the computing power of the host is challenged by the prototype. CPU usages around 80% and higher indicate a bottleneck here.

The use case scenario has been implemented through the architecture shown in Figure 5.21. Tables 5.9 and 5.10 for one SOLIOT and Solid instances being requested simultaneously by every 1 and 10 clients, and a cluster of ten instances, respectively. The latency has been measured as perceived at the client itself but also at the server endpoint to show the influence of the network. However, the whole setup does not measure the parsing to and from the network socket. This typical bottleneck is included in the network latency as it could not be influenced but also not reliably measured.

Table 5.11 provides a comparison of the used bandwidth for each request. This sequence has been 100 times by each client. One can see that typically the request size of the CoAP and HTTP messages are quite similar, while the response deviates significantly. One reason is the rich headers provided by the Solid endpoint. Losing this kind of information is a significant drawback of the current state of the IIoT

[17] Measurements on an average of 7 runs and a variance of 4.53 for one Solid client and an average of 35 runs and a variance of 0.091 for one SOLIOT client.

Table 5.10: Performance measures for 10 instances each with 1 and 10 parallel requesting clients.

Criteria	Solid, 1 Cli.	SOLIOT, 1 Cli.	Solid, 10 Cli.	SOLIOT, 10 Cli.	Setting
GET @Client	184 ms	116 ms	205 ms	132 ms	1000 requests
POST @Client	202 ms	60 ms	252 ms	81 ms	(Solid: HTTP,
PUT @Client	202 ms	62 ms	252 ms	82 ms	SOLIOT: CoAP)
DELETE @Client	202 ms	59 ms	250 ms	79 ms	per instance and cli.

Table 5.11: Interaction sequence for the evaluation setting.[18]

Resource	SOLIOT Interaction	Size (bytes)	Resp. Size	Solid Interaction	Size (bytes)	Resp. Size
/th[...]/maximumTemperature	CoAP get	398	641	HTTP GET	204	1495
/assets/[...]/currentTemperature	CoAP get	398	705	HTTP GET	204	1556
/assets/3S7PM0CP4BD/120636/	CoAP get	308	7067	HTTP GET	161	8853
/assets/	CoAP get	109	2426	HTTP GET	142	3324
/assets/[...]/testTemperature	CoAP put	980	0	HTTP PUT	810	704
/assets/[...]/testTemperature	pub: updated	675	–	HTTP PUT	810	704
/assets/[...]/testTemperature	CoAP get	406	372	HTTP GET	203	1218
/assets/[...]/testTemperature	CoAP delete	406	128	HTTP DELETE	190	627
/assets/[...]/	CoAP post	980	0	HTTP POST	828	772
/assets/[...]/	pub: created	675	–	HTTP POST	828	772
/assets/[...]/testTemperature	CoAP get	406	960	HTTP GET	200	1212
/assets/[...]/testTemperature	CoAP delete	406	128	HTTP DELETE	187	666

complete URL for the first entry:
coap://18.157.197.66:5683/assets/robot/120636/2018-10-24T01-22-30-866Z/maximumTemperature

protocol mapping. Furthermore, the MQTT publishing behavior only reflects the actual publishing party. The receiving clients or the performance of the MQTT message broker have not been regarded. Another limitation of this evaluation approach is the limited representativeness of the used demo data. The Digital Twin model has been aligned with an existing product, unrelated to the SOLIOT developments. Still, a representative Digital Twin testbed would further increase the informative value of the experiments.

Comparing the results of the SOLIOT prototype, one can state a significant reduction in necessary transferred bytes but also in execution time. That is noteworthy regarding the fact that the handling of IIoT requests takes longer than their HTTP counterparts (right side of columns in Tables 5.9 and 5.10, rows '@Server'). The implication is that the reduced data size results in a higher network speed. Furthermore, the SOLIOT prototype heavily relies on the core Solid code. It mainly wraps the IIoT endpoints but calls the underlying LDP functions in the same way as the Solid baseline server does. Having a native integration directly on the persisting functions should further increase its speed.

Figure 5.22 further supports this insight. One can easily see the speed advantage. The interested reader may note the marked outlier for one CoAP GET request. This one might be caused by the at some times unreliable behavior of the SOLIOT prototype. Another relevant insight is the difference in the behavior of GET requests for both Solid (faster than the others) and SOLIOT (slower), an effect that requires further investigations. All collected measures, together with the resources to reproduce the measurements, are provided publicly [19].

[18] Each iteration was executed 10 times (in total 100 CoAP and 20 MQTT interactions) per test instance. Presented are average measures per request.

[19] https://github.com/sebbader/soliot_evaluation

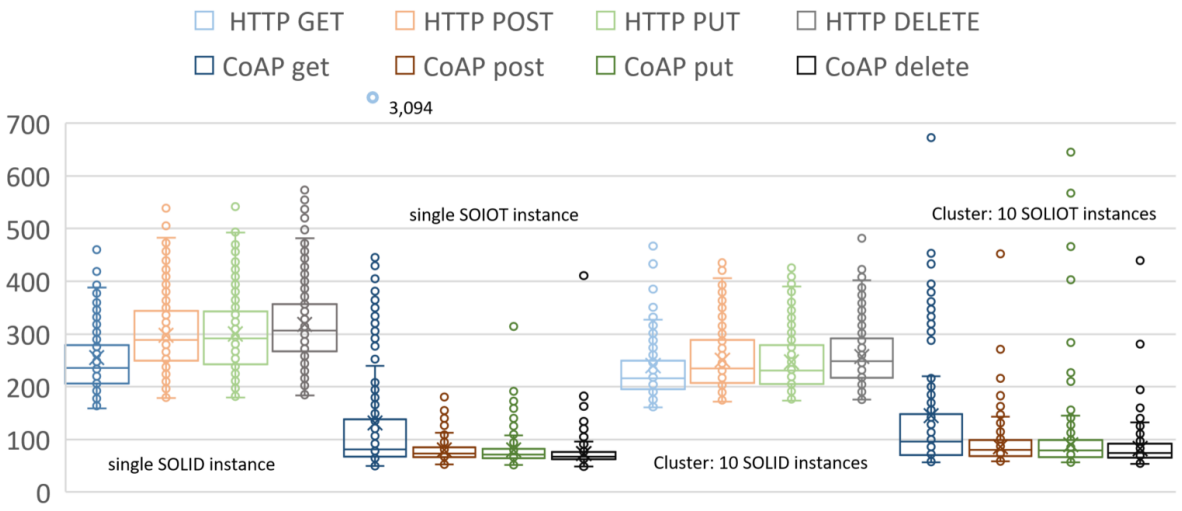


Figure 5.22: Variance of measures presented in Tables 5.9 and 5.10 (in milliseconds, 10 clients requesting each instance).

5.4.8 Summary: An IIoT Framework for Semantic Digital Twins

This section presents SOLIOT, an approach to merge concepts following Linked Data Platform and Solid patterns into a consistent model for transparent *IIoT Interactions between Digital Twins* (CO3) and aligned with *IIoT Requirements* (CO4). The respective capabilities and requirements have been discussed according to functional, non-functional, protocol- and data-related aspects. The basic functionality is shown through a prototypical server implementation, together with the provisioning of Web and IIoT protocol endpoints, discovery mechanisms, and how self-descriptive representations can model a Digital Twin.

SOLIOT outlines how the latest developments from the Web community can be transferred to IIoT challenges. The solely resource-driven view on the representation simplifies both the consumption and interaction with the thereby supported Digital Twins and allows visiting clients to directly understand the Digital Twin itself but also helps developers and system integrators to adjust their workflows and APIs. This is achieved by the reduction of interaction patterns together with the self-descriptive nature of all involved concepts, promising a decrease of the overall integration complexity and increasing maintainability. The Digital Twin outlines both its content, interaction patterns, and authentication model to the client. In addition, this is not only the case for the consuming applications, usually with higher computational power, but also for the restricted devices acting as the information sources.

The proposed approach still holds significant gaps. While the most crucial interaction patterns have been mapped, the true potential of the Solid to IIoT approach demands the complete coverage of all specified discovery mechanisms. The IIoT mappings do not yet cover all headers and response mechanisms of a full-sized Solid solution. In addition, the identity and authentication potentials of WebIDs are not sufficiently reflected at this stage. This is especially relevant, as a decentral yet reliable and, at the same time, simple to implement identity mechanism is essential for any scalable IIoT approach. The core challenge is certainly the proper treatment of a physical Asset and its Digital Twin moving across networks, organizations and lifecycles. The thereby created challenges towards interoperability, legislative obligations, ownership rights, etc., are still not sufficiently understood.

Nevertheless, the general feasibility of SOLIOT is demonstrated through a CoAP and MQTT binding.

More powerful protocol stacks have not been tested yet. In particular, the OPC UA specifications define not only a communication protocol but also an elaborated interaction and information model. Respective endpoint adapters, data lifting and lowering modules, and appropriate mappings for authorization need to be developed to examine the characteristics further.

The provided analysis showed a series of relevant but still unaddressed requirements. As directly visible through the significant number of gaps shown in the mapping tables, more work is necessary for a complete picture. Even though one can certainly question whether each IIoT concept needs to address all outlined requirements, the awareness of their relevance and how they impact the desired solution is necessary. This work should also be regarded as an inspiration for further approaches by stating the current possibilities but also its limitations.

At the core of SOLIOT is the translation of established and clean interaction patterns from the Web into the IIoT, where all necessary information is explicitly described and available for look-ups. The further decoupling of IIoT communication will introduce new flexibility and dynamics into device-to-device networks. An ongoing integration of previously separated and segregated OT networks with the global Internet network can be stated. This requires a shared understanding of how Assets need to interact and how they need to be described. SOLIOT is comprising the combined practices of the Semantic Web community and intends to outline one contribution to this challenge.

5.5 Summary

The contributions of this chapter present solutions to encounter the interoperability issues of IIoT Assets, communicating with each other as Digital Twins but also with legacy IT systems (CO3, Sections 5.2 to 5.4). This builds on the *Model of IIoT Requirements* (CO4, Sections 5.1 and 5.4). Together, both contributions together enable a seamless communication layer using the semantic data models from CO1 and CO2.

The dynamic nature of IIoT scenarios and the huge investment costs enforce continuous adoptions and updates to the Assets themselves but also to the communication network. However, anticipating the requirements of future use cases is hardly possible, mainly due to the long operating times of the facilities. Therefore, SOLIOT approach targets CH1 by promoting atomic operations, inspired by Web APIs that have a clear call semantic and can be easily combined to more complex workflows. The resource-centric view still ensures that each communication partner, either server or client, is aware of the side-effects and his respective responsibilities in the interaction.

As presented in Section 5.3, the operations are independent of the used protocol stack, and therefore the lifting of the underlying Assets to IIoT Digital Twins can also easily bridge such gaps. The mappings and transformations are handled at the Edge and mapped into standardized presentations of Digital Twins with consistent and transparent operation semantics for the downstream applications. This allows the integration of the approach also into already running shop floors and thereby coping with already existing *Brownfield Settings* (CH2). The previously applied description models give the interacting systems the definitions to also understand the provided data objects.

The *Heterogeneous Communication Patterns* (CH3) between the Assets and their control systems due to proprietary and vendor-specific protocols, data schemes, and control workflows are encapsulated through the Digital Twin and hidden from further applications. As such, also other Digital Twins do not need to care about the specific behavior of other Assets, as long as its Digital Twin responds accordingly. As shown in Section 5.2, not only the look-ups but also control operations and their own configuration can be applied using the same paradigm. This repeating treatment of all involved functions through the same patterns relieves the developers and system integrators from significant burdens and fastens the

establishment of data and control channels.

In addition, the wrapping of rich but thereby complex interaction calls into smaller, but many requests simplifies the creation of consuming systems. Assuming that the proper creation of this side of the communication systems is generally harder, the initial deployment or upgrading effort is intentionally increased for the offering APIs. This additional investment is, however, justified by the effort reduction at the consuming side as the usually appearing *Non-transparent Design Decisions* (CH4) create significant obstacles for every third-party system.

This procedure also minimizes the efforts for cumbersome synchronization tasks, for instance, the need for aligning the clocks in all network-enabled devices. The state-based nature of the integration process automatically creates time slices, which are usually sufficiently accurate. Real-time control with millisecond intervals is not covered as both the Edge gateway as the network itself may lead to associations of non-simultaneous events. For such use cases, streaming-native approaches need to be considered.

The detailed requirement analysis, together with the interaction model for SOLIOT Digital Twins, defines the interaction model for the IIoT. It transfers the scalability and decentralization characteristics of the Web into the industrial internet and thereby simplifies the integration between the Assets at the shop floor and the applications of the office floor but can also use the same operations for data exchange across organizations. That creates environments where the same principles can be applied throughout the supply chain and the complete lifecycle of the Assets, independently if they are raw material, products, or even huge production facilities composed of countless components and devices.

CO3 and CO4, together with the previous presented CO1 and CO2, therefore answer RQ2, how IIoT Assets need to be presented for the seamless integration in dynamic networks. The presentation as Digital Twins, not only by their appearance as data objects but as modifiable encapsulations of their underlying Assets, leads to reappearing entities instead of individual, heterogeneous devices. This reduction in terms of complexity also simplifies potential attacks. State-of-the-art security concepts, for instance the Zero Trust approach, already propose feasible solutions. As the future shop floor Assets will be connected to the global internet, one way or another, the sealing-off is not a sustainable approach anymore.

CHAPTER 6

Distributed Architectures and IIoT Reference Frameworks

Speaking about Digital Twins requires a shared understanding of their capabilities and characteristics. Grouping those into views allows their structured analyses by clustering related requirements together. This chapter gives an outline of commonly used categories represented through stacked layers. Based on international standards and well-accepted conventions, the outlined reference architecture arranges Digital Twins from business considerations down to the physical data transmission and explains the necessary considerations from security and governance perspectives.

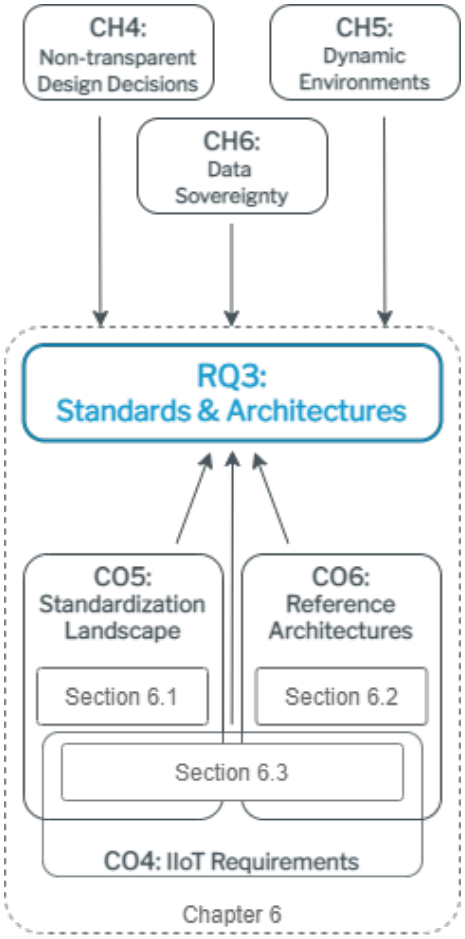


Figure 6.1: Contributions for RQ3.

Semantically described Assets, enhanced with standardized interaction capabilities, require further components to engage in applicable federated architectures. A huge amount of Reference Frameworks and Architecture Models have been developed in the meantime, each with a different scope, requirement, and strength. Implementations following the recommendations of one standardization group, for instance the IIC [4], can result in the usage of descriptions and patterns that are hard to retrace by IIoT experts following the IoT-Architecture [242] conventions. The underlying *Design Decisions* (CH4) depend on the background and experience of the application's developer and are therefore non-transparent for external parties that later need to update or modify an IIoT Asset. It is, therefore, necessary to organize the domain, select the most impacting ones, present common views, unique approaches but also blind spots that require further investigation.

The *Dynamic Environments* (CH5) of IIoT settings further require in-depth knowledge on conventions and best practices. Otherwise, the unavoidable modification and retrofitting processes for IIoT Assets become even more complex and expensive. The engineering of Assets and their integration into the physical facilities and digital networks according to defined standards further increases their reusability and supports the

interoperability of the overall system.

In addition, modern business models require collaborations between many partners, and their IT systems, along the lifecycle of the Assets and their supply chain. Protecting valuable data on an individual level is not sufficient, as the data moves between several systems at different organizations. The overall protection level is determined by the weakest link of the data processing chain. Therefore, a comprehensive view of *Data Sovereignty* (CH6) is required. All these challenges appear on the level of individual Assets, facilities, and organizations but also in the dynamic environment and flexibly reorganized interactions between varying IIoT Assets and networks. This leads to the third research question (cf. Fig. 6.1):

RQ3: Standards & Architectures – How can standardization activities ensure the compliance of independent IIoT integration efforts?

The work of this chapter builds on early work presented at ESWC2017 [33] and contributes to the stated challenges by providing a structured *database of relevant IIoT standards* (CO5) in the form of an ontology-based knowledge graph (Section 6.1, presented at ESWC2020 [29]). The gained insights of this publication lead to the proposition of a generic reference model for IIoT architectures (Section 6.2, published as a book chapter [38]). Using this terminology, a critical evaluation of the existing industry- and research-driven standardization activities for IIoT reference architectures is conducted to extract commonly accepted patterns but also to point out required fields for improvement and research needs (Section 6.3, published as an article in the MDPI journal *Future Internet* [32]). It utilizes the knowledge graph for IIoT standards together with the structured analysis of the *Requirements of IIoT Systems* (CO4, already presented in Chapter 5) and extends it with an interactive visualization tool for flexible on the fly examinations of its original content together with the manually extracted information of the research-driven standardization activities for IIoT reference architectures (presented at ETFA2019 [31]). The resulting contribution for *IIoT Reference Architectures* (CO6) contains a publicly available resource and an examination using it. This analysis results in a detailed description of the community discussion, determining the generally accepted areas but also the blind spots.

6.1 Industrial Standards and Norms for the IIoT

Industrial processes are driven by norms and standards. In comparison to other domains and communities that rely on common agreements and best practices, the specific reliability and safety requirements of industrial manufacturing demand strict and formal specifications. International institutions such as ISO, IEC, or ETSI, together with national organizations such as NIST, DIN, or ANSI, face this demand and form a network of highly recognized authorities, ensuring the quality of published standards and norms.

The rising popularity of digitizing processes, components, and complete production lines has consequently led to an increasing number of standards targeting the various related aspects. In particular, the recent developments towards an industrial internet have drawn significant attention not only inside the manufacturing companies but also in academia and government. The result is an already overwhelming but further growing amount of relevant norms, standards, and specifications. The necessary effort for both domain experts and newcomers is also increased by the lack of suitable guidance and limited metadata. The interested reader can only evaluate the significance of a specific publication after examining the complete text – a substantial challenge regarding the number of available specifications. This rising need for a structured access point with organized and preprocessed entities and explicitly outlined interlinks and attributes is still not addressed. Therefore, a structured dataset in the form of a semantically annot-

ated knowledge graph for the IIoT related standards, norms, and reference frameworks is proposed in this chapter. The graph provides a Linked Data-conform collection of annotated, classified reference guidelines supporting newcomers and experts alike in understanding how to implement IIoT systems. The suitability of the graph for various use cases is illustrated, its already existing applications, the maintenance process, and an evaluation in its completeness and quality is presented.

This knowledge graph contains the latest state of IIoT specifications with respect to standards, reference frameworks as well as key requirements (cf. Tab. 6.1). The interlinked nature of the content and its various relations to outside topics led to the design of an RDF-based graph schema that can easily be extended and interlinked with further datasets. Utilizing the information content of the proposed knowledge graph, the following types of relevant information can be retrieved:

1. Where can additional information about a certain IIoT requirement be found?
2. Which specification is most appropriate for establishing a secure data exchange between IIoT Digital Twins?
3. What are the requirements related to a specific IIoT challenge, and where can appropriate guidance be found to solving them?
4. How can the formalization of the domain support a machine learning pipeline?

A key feature of this work is the provisioning of relations to external data sources. Openly available information, for instance from DBpedia, enhances the understanding and points the user to further data sources in the Linked Open Data Cloud. The thereby accessible content makes the knowledge graph relevant for several potential consumer groups: *System architects* are interested in finding and learning about suitable design patterns, *IIoT experts* working in standardization groups need to be aware of and observe related initiatives, *component developers* require best practices for interfaces and models, *system integrators* need to understand common data models and interaction patterns, *machine manufacturers* need to ensure the sustainability of their digital interfaces, and *IIoT newcomers* want to reduce their onboarding time.

This work contributes to the outlined challenges with the following aspects: (1) to present the knowledge graph as the information backend (originally named *Industry 4.0 Knowledge Graph*, I40KG), (2) to explain its maintenance and curation processes, (3) to discuss its applicability as the basis for other resources and applications, and (4) to show the applied analysis methods and results on top of the existing data, compared to previous publications. The I40KG helps to overcome hindrances related to realizing the IIoT vision, which prerequisites not only comprehensive knowledge about distinct standards but needs to consider the semantics and relations between standards, standardization frameworks, as well as their requirements.

6.1.1 Design and Technical Quality

The I40KG design follows best practices of publishing resources as Linked Data. As stated in Tab. 6.1, the resource conforms to the FAIR principles and is created, curated, and accessible in a transparent and open manner. The required characteristics are listed in brackets using the notation of Wilkinson *et al.* [28]. The graph also reuses common RDF vocabularies wherever possible. Upper-level ontologies, such as DUL or DCTERMS, support the understanding of classes and properties. Relations to DBpedia resources help to identify the intended entity but also provide valuable directions for further look-ups.

The I40KG is designed in a modular way in order to ensure the maintainability of the sources and increase the readability for the users. As recommended by Parent and Spaccapietra [260], each partition

Table 6.1: *I40KG* details: Relevant aspects of the *I40KG* and related resources.

General	**Name**	Industry 4.0 Knowledge Graph (I40KG)
	DL Expressivity	SHOIF(D)
	License (R1.1)	Creative Commons 3
	Size	44 classes, 35 object properties, 22 data properties, 1335 individuals
	Standards and Norms (R1.2)	338 standards and standard parts, 49 ISO standards, 67 IEC standards, 11 DIN standards
	Frameworks	18 reference frameworks divided into 138 classification sections
	Concerns	160 interrelated IIoT concerns in 6 categories
	External Links (F3, I3)	286 to DBpedia resources, 271 to Wikipedia pages
	Reasoning	4.257 derived triples
	Total size	16.447 unique triples without derived ones
Reuse	**Reused Ontologies** (I2, R1)	DCTERMS, DCELEMS, PROV, DUL, FOAF, OM, etc.
	Reused ODPs	Componency ODP
Documentation	**Element description** (F2, R1)	By means of `rdfs:label`, `rdfs:comment`, `skos:prefLabel` and `rdfs:isDefinedBy`
	Ontology Documentation	http://i40.semantic-interoperability.org/sto/
Conventions	**Naming pattern**	CamelCase notation for the schema and Ada for instances
	Linked Data (R1.3)	5 Star Linked Data
Multilinguality	**English labels for all terms**	`rdfs:label` and `rdfs:comment` with the @en notation
Availability	**PersistentURI** (F1)	https://w3id.org/i40/sto
	Serializations (I1)	Turtle, RDF/XML
	GitHub (A1)	https://github.com/i40-Tools/I40KG/
	LOV (F4)	http://lov.okfn.org/dataset/lov/vocabs/sto
	OntoPortal (A2)	http://iofportal.ncor.buffalo.edu/ontologies/STO
	License	Creative Commons 3.0
	VoCol Instance (A2)	http://vocol.iais.fraunhofer.de/sto/

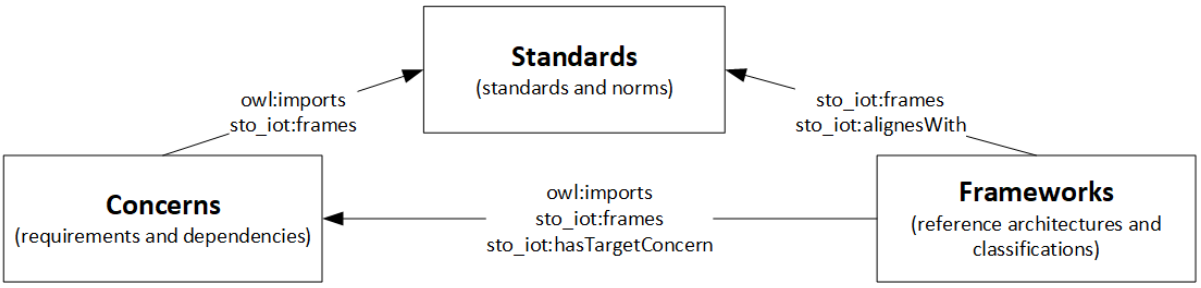


Figure 6.2: The three partitions of the I40KG as ontology modules.

focuses on one of the mentioned sub-domains – *Standards*, *Concerns*, and *Reference Frameworks* (cf. Fig. 6.2), published in separate Turtle files. The partitions themselves depend on each other utilizing *owl:imports* statements.

The original standards ontology has been extended but still serves as the foundation for the other modules. It is focused on the description of a *standard* as a logical concept, defines attributes and relations, and contains all standard instances. Concerns, as defined in ISO 42010 [218], can be understood as domain requirements, motives or issues, which a stakeholder can have about an IT system in general. To increase readability, the terms 'concern' and 'requirement' are used synonymously, even though the definitions in ISO 42010 slightly differ.

While ISO 42010 defines the terminology of a concern itself, it lacks an approach to supply a set of usable instances. The I40KG, therefore, contains a taxonomy for I40-related concerns, which is intended as a first outline undergoing further refinements. Starting with six top-level concerns (*Data Sovereignty*, *(Industrial) Internet of Things*, *Trustworthiness*, *Data Analytics*, *Interoperability*, *Business Context*), cycle-free dependencies of sub-concerns are formed. Further details about the concerns themselves have also been presented by Bader *et al.* [31].

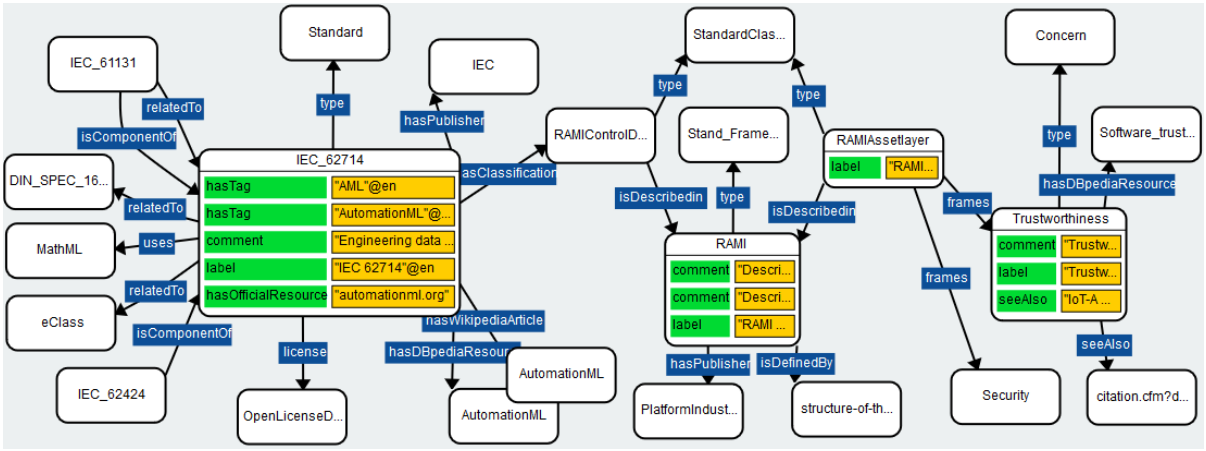


Figure 6.3: Contained entities: Standards (IEC 62714) link to standard classifications (RAMI Asset Layer) with frameworks (RAMI) and requirements (Trustworthiness).

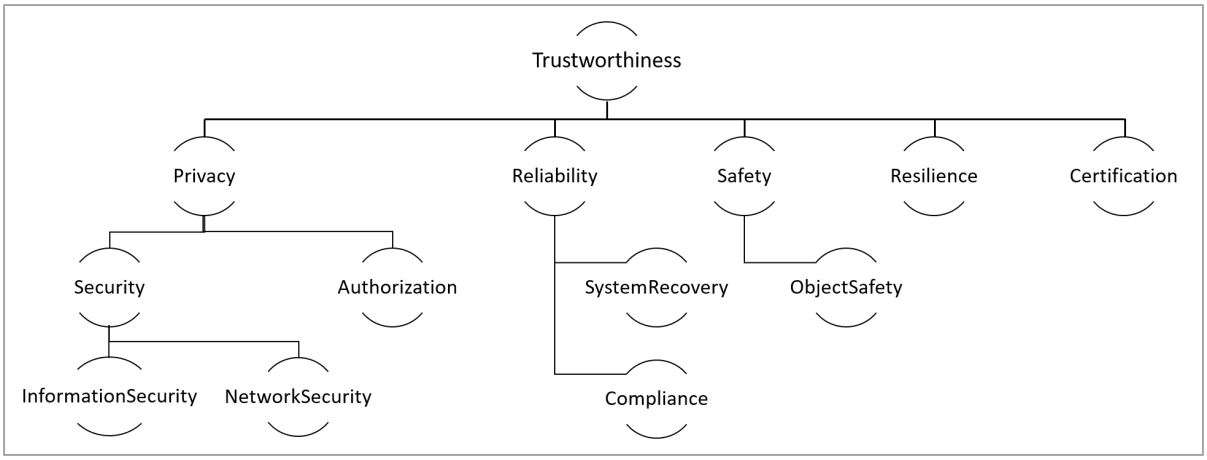


Figure 6.4: Concern hierarchy. Illustration for the "Trustworthiness" of a system, the created structure of concerns allows the reasoning and aggregation of characteristics regarding their targeted requirement and the underlying concern taxonomy.

Figure 6.3 shows a set of I40KG instances. The *IEC 62714* about AutomationML has various links (*sto:uses*, *sto:isComponentOf*, *sto:relatedTo*) to other standards. In addition, annotations (green) and values (yellow) explain the entity itself, containing – among others – the official location of the source document. For IEC standards, this is usually the IEC webstore site of the respective standard. Nevertheless, in case a distinct website of the standard exists, the more specific one is listed. More relations to external resources are also supplied, mainly to Wikipedia/DBpedia.

As depicted in Figure 6.3, IEC 62714 is classified as relevant for the *RAMI Control Device*, a *Standard Classification* scheme related to the *RAMI4.0 Standardization Framework*. A user can traverse these links and discover another *Standard Classification* instance of RAMI4.0 frames *Trustworthiness*, which is the *Concern* also presented in Figure 6.4. In this way, further information can be accessed, and the user is able to further explore the I40KG.

The knowledge graph is maintained following three different insertion processes. As depicted in Figure 6.5, one process for the selection, examination, and annotation for standards (top) and reference frameworks (bottom) have been established. Details about the selection criteria have already been

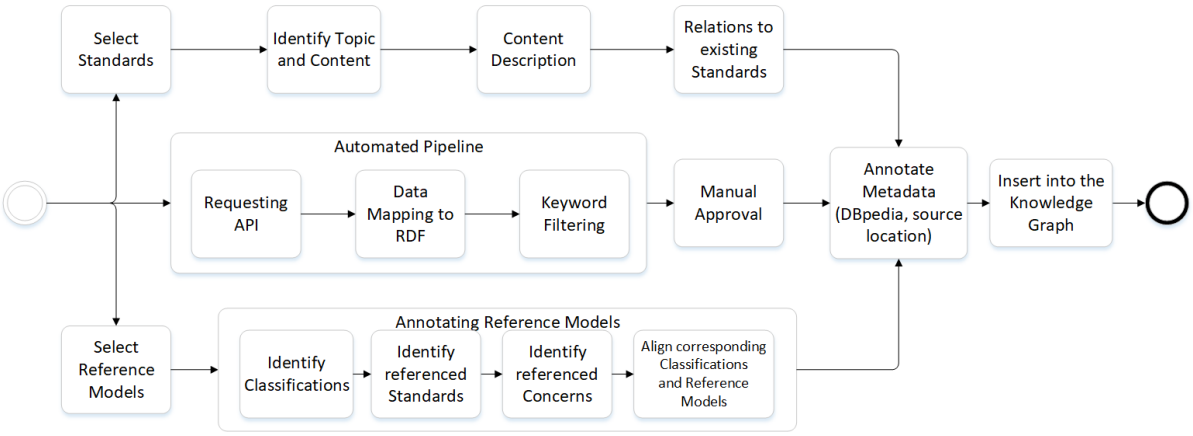


Figure 6.5: Insertion Process: Three different sub-processes to create the I40KG content.

explained [215] [32] and are therefore omitted here. Both approaches are transparently executed using the GitHub repository and its commit history.

In addition to the manual extensions, an automated update process has been introduced (cf. Fig. 6.5). As the frequency of new standards and updates of already published ones is too high, a bot searches for such events, maps the metadata to RDF, filters relevant standards and norms, and proposes the resulting entities for insertion into the I40KG. Currently, only IEC standards are monitored, but a further generalization is intended. The automated proposals require manual approval, usually together with additional annotations to external resources, for instance to DBpedia resources.

6.1.2 Knowledge Graph for the IIoT

The presented list of reference models is only an extract of the available guidelines. The created knowledge graph includes several more specifications, for instance guidelines form FIWARE[1], Edgecross [261], Industrial Value Chain Initiative [262], X-Road[2], and more.

Reference models like RAMI4.0 or the IIRA consist of several layers, views, perspectives, and other selection categories to better depict the topic of interest. For instance, RAMI4.0 is organized into six layers, seven hierarchy levels, and four basic lifecycle and value stream phases. The various dimensions are collected as entities of the classification class and interlinked with the entities representing their frameworks but also the standards and publications which they refer to (cf. Fig. 6.6).

The classifications are also the entities that link the reference frameworks with the concerns and requirements they target. Concerns represent issues or challenges which are relevant for a respective scenario. For instance, the connectivity and interoperability of devices are central to the Industrial Internet Consortium, therefore, the IIRA classifications discuss related topics in detail. Data sovereignty and data description are more in the focus of the IDS, resulting in more linked concerns of these categories. In general, the classifications play a major role in the data model of the knowledge graph and interlink and group the entities of the other classes.

The IIoT relies on shared technologies and the seamless exchange of data. Integration of systems and the connectivity of facilities requires clear technical specifications and supporting manuals. Established standardization organizations meet this demand by developing standardization documents and technical specifications. Among the ones with the highest reputation are the International Organization for

[1] https://www.fiware.org/

[2] https://www.ria.ee/en/state-information-system/x-tee.html

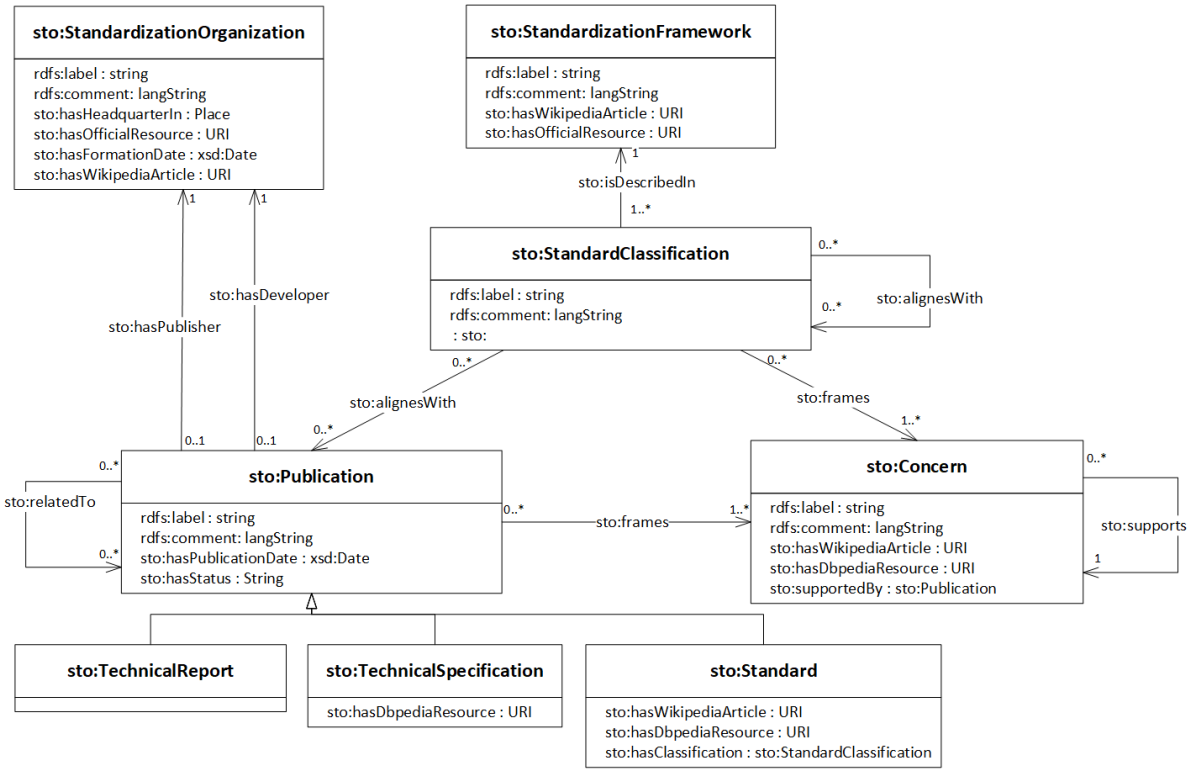


Figure 6.6: IIoT Reference frameworks are described by publications and concerns. Reference frameworks describe the IIoT topics through several classification categories. Technical publications, like standards, explain the implementation details.

Standardization (ISO), the International Electrotechnical Commission (IEC), the American National Institute of Standards and Technology (NIST), and the German Institute for Standardization (DIN).

The ontology is designed according to classes of standards, reference frameworks, and their promoted classification models. Each standard is annotated with a rich set of attributes, for instance the corresponding creator and publisher, its publishing date, and a short description. Overlaps and similar scopes with other standards are explicitly presented through *sto:relatedTo* predicates. Even further information is accessible by following outgoing links to the original publication but also to respective resources in the Linked Open Data Cloud. Thereby, the interested user can directly access additional information from the whole Web and benefit from the constantly growing volume of available knowledge.

Similar to the modeling of standards, the frameworks and reference models are directly annotated with relevant context information and also link to additional resources in the Web. Furthermore, each reference framework is divided into classifications based on the structure of its reference architecture. The combined and interlinked data is hard to analyze, especially for users who are not familiar with knowledge graphs. For the outlined use cases, several types of illustrations are necessary to optimally target the respective information needs. Relying on the knowledge graph, several Web-based views have been developed. Each one uses a different visualization pattern in order to present the information in the most intuitive manner.

As mentioned, this work relies on the work published in [215] and therefore follows a similar data extraction and integration process (cf Fig. 6.5). The manual collection and filtering of relevant reference models are followed by the iterative identification of referred standards and concerns. Furthermore,

Table 6.2: Logical axioms as SWRL rules.

	Rule	Description
1	*sto:relatedTo(?s1, ?s2) -> sto:relatedTo(?s2, ?s1)*	Symmetry of the sto:relatedTo property
2	*sto:relatedTo(?s1, ?s2) ∧ sto:relatedTo(?s2, ?s3) -> sto:relatedTo(?s1, ?s3)*	Transitivity of the sto:relatedTo property
3	*rdfs:subClassOf(?c1, ?c2) ∧ rdf:type(?x, ?c1) -> rdf:type(?x, ?c2)*	Class inheritence of entities
4	*sto:frames(?class, ?con) ∧ sto:isDescribedin(?class, ?fram) -> sto:hasTargetConcern(?fram, ?con)*	Reference frameworks refer to the concerns of their classifications
5	*sto:supports(?con1, ?con2) ∧ sto:frames(?class, ?con1) -> sto:frames(?class, ?con2)*	Transitive nature of framed concerns

similarities and matches in the structure of the identified reference models and standards are encoded through alignment relations between the respective graph nodes. Additional metadata like links to external information sources and annotations – including title, short descriptions, and references to the original documents – is applied before the data objects are inserted into the knowledge graph.

The formalized structure of the graph and its defined semantics allow a further population by explicitly stating implicitly encoded facts. For instance, the information that the Common Data Dictionary (IEC 61360) is referenced by AutomationML (IEC 62714) and the fact that AutomationML is also related to ECLASS indicates that also the Common Data Dictionary is related to ECLASS. A semantic reasoner can automatically discover such connections and add them to the knowledge graph as long as the respective knowledge is supplied in the form of axioms. Axioms are rules which encode the relationships as logical formulas (cf. Tab. 6.2).

Regarding the example of the Common Data Dictionary (IEC 61360), 65 atomic statements are present in the original graph. After the reasoning process, 53 more explicit facts could be added. The reasoning itself was conducted with the Linked Data-Fu streaming engine [100] before the data was loaded into the endpoint. The execution time for the whole process is in all cases lower than two seconds, which is especially remarkable regarding the high number of inferred facts (cf. Tab. 6.1). A noteworthy fact is the explicit decision to *not* apply the full expressiveness of possible reasoning rules. For instance, the well-known rule sets OWL Full and OWL DL could further increase the resulting dataset. The decision against OWL reasoning reflects the trade-off between its significantly longer computation time while gaining mostly syntactically and not semantically relevant facts.

6.1.3 Availability of I40KG

The I40KG is documented following the established best practices for ontologies and Linked Data resources. It supplies a human-readable documentation page for all classes, properties, and instances[3]. Furthermore, several serializations, e.g., RDF-XML, Turtle, N-Triples, etc. are provided, where the Turtle-files act as the single source of truth. Redirects and content negotiation is in place to supply each client with the most appropriate serialization.

The I40KG and its entities are defined in the STO namespace, using W3IDs for long-term accessibility. STO was the original acronym for “standards ontology” and is retained for sustainability reasons. The knowledge graph is available under the **Creative Commons 3.0 license** and can be reused by anyone and for any purpose. Extensions to the original graph in terms of A- and T-Box are possible but require

[3] http://i40.semantic-interoperability.org/sto/

approval of the graph creators in order to ensure the consistency and quality of the content. Change requests can be placed at its official location, a publicly available **GitHub repository** (cf. Tab. 6.1).

The maintenance and further development of the knowledge graph are organized in the mentioned GitHub repository, in particular through GitHub issues. The issue system is also the main communication channel in order to propose changes, document errors, and outline extensions. The complete sources are accessible, and all changes and updates are executed in a publicly visible and transparent manner. Following best practices of the Semantic Web, each entity is annotated with well-known annotation properties, i.e., `rdfs:label`, `rdfs:comment`, and is linked to DBpedia resources, wherever a suitable entry exists.

6.1.4 Reusability of the Graph Content

The described knowledge graph is used in several projects. In the context of the International Data Spaces (IDS)[4], it is used in its data model but also as a reference resource for the IIoT domain in general and the most up-to-date reference frameworks and architectures.

The knowledge graph is already utilized for learning embeddings to exploit the meaning of the relationships between standards[5]. Furthermore, Grangel et al. have already employed unsupervised Machine Learning methods, e.g., Clustering, to unveil existing relations of standards in the I40KG [263]. Furthermore, a public SPARQL endpoint[6] provides the latest version of I40KG, also hosted at a VoCol instance[7] [206] for additional documentation purposes.

All commonly used RDF tools can work with the I40KG and its source files. Its core classes are, wherever suitable, linked to upper-level ontologies. In particular, the linking to commonly-known DBpedia resources allows its direct integration with other knowledge graphs and especially the Linked Open Data Cloud. However, the I40KG does not intend to fully cover the domain nor represent or judge the internal quality of the referred standards, norms, and frameworks. It is – and always has to be – in the responsibility of the user to finally decide on the suitability of a certain standard or norm regarding the specific context or use case. The I40KG can support the user to effectively gain an overview and discover unknown resources. Also, in conformance with the open-world assumption, the absence of an attribute or relation does not allow to expect its negation. Still, sufficient completeness of the domain is necessary and has been examined using the criteria of Chapter 6.3.4. The presented selection shows how academic and industry impact has been examined to optimally discover and filter the I40KG entities.

The proposed I40KG shall satisfy the various information needs around industrial standards and frameworks. As the regarded area is huge and characterized by fast changes, reaching complete coverage is not realistic. Nevertheless, a comprehensive overview with as much content as possible is desirable. The supplied content must comply with best practices and meet the expectations of potential users in order to provide value. The knowledge graph is therefore evaluated using two different approaches.

The outlined information content is without comparison regarding its relations to the Linked Open Data Cloud resources and the amount of described technical standards and architectural propositions. The knowledge graph itself can be used in many different ways, as shown in the and user stories of Section 6.3.2.

[4] https://www.internationaldataspaces.org/

[5] https://github.com/i40-Tools/I40KG-Embeddings/

[6] https://dydra.com/mtasnim/stoviz/

[7] https://vocol.iais.fraunhofer.de/sto/

6.1.5 Technical Evaluation

The syntactic quality has been checked by commonly used tools such as the Ontology Pitfall Checker[8] and RDF-TripleChecker[9]. These tools indicate that the I40KG is consistent and correct in terms of common RDF and ontology pitfalls. Wherever the mentioned tools indicated potentials for improvement, the respective sections have undergone an intense manual evaluation. The reports are also hosted in the GitHub repository.

The reports, for instance, mention two issues. Several properties miss domain and/or range attributes, and sometimes, the disjointness of classes is not sufficiently declared. However, it has been explicitly decided not to set the range and domain of all properties, as their implications for reasoning on the I40KG can easily result in inconsistencies. Complete disjointness statements, on the other hand, are rather uncommon, adding only limited added value to the graph itself but requiring extensive maintenance.

Furthermore, the quality of the I40KG is evaluated using the metrics proposed by Färber *et al.* [264]. Table 6.3 contains all metrics grouped by these categories in order to provide as much information as possible. Nevertheless, the expressiveness of several of the suggested criteria is certainly limited. One reason is that the I40KG covers a new domain for structured or open data. Therefore, no gold standard exists (cf. *Completeness*). In addition, it has been explicitly decided to avoid certain statements and relations. For instance, the validity time of standards is not determined by the publishers, making any inserted information wrong by default (cf. *Validity period*). Regarding the suggestions for interlinking resources, *owl:sameAs* would result in wrong inferences, leading to the introduction of, for instance, *sto:hasDBpediaResource* and *sto:hasWikipediaResource* properties.

6.1.6 Summary: The Landscape of Industrial Standards

This section presents the Industry 4.0 Knowledge Graph, a component of CO5, that contains the latest status of IIoT standards, reference frameworks, and concerns in a structured manner. The graph describes, connects, and outlines the most relevant information sources. The I40KG has been created following best practices, conforms by design to the Linked Data principles, and is enhanced with a set of supporting tools, documentation, and hosting services. It is transparently maintained and open to the community.

The cumbersome search and structuring of the information resources for each involved participant are some of the most crucial obstacles for efficiently realizing IIoT use cases. The presented approach addresses precisely this challenge. The benefits of the Semantic Web technology stack can support the industrial community and furthermore reach new application areas. The I40KG can solve some of these issues and create added value for various target groups. Using the I40KG is then extended to the proposed *Standardization Landscape* (CO5) in Section 6.3 using also the architecture model for IIoT Digital Twins, which is introduced in the next section.

6.2 Generalized Architecture Model for IIoT Digital Twins

Several works aim to create a framework for software architecture descriptions. [218] proposes architecture descriptions derived from a list of so-called *concerns*, being addressed by several *architecture views*. An architecture view is a projection and, therefore, a simplification of the abstract architecture in order to describe specific topics. For instance, many reference architectures cover both interoperability and security-related aspects. Though there are many inter-dependencies, describing both concerns in one view decreases readability and significantly increases complexity.

[8] http://oops.linkeddata.es/
[9] http://graphite.ecs.soton.ac.uk/checker/

Table 6.3: *I40KG* evaluation results.

Metric	Result	Explanation
Accuracy		
Synt. validity of RDF doc.	m_{synRDF}(I40KG) = 1	RDF documents are syntactically valid.
Synt. validity of literals	m_{synLit}(I40KG) = 1	Literals conform to their datatype.
Semant. validity of triples	m_{sem}(I40KG) = 1	No gold standard available. References to original information sources applied.
Trustworthiness		
KG level	m_{graph}(I40KG) ≥ 0.5	Manual data curation and automated process
Statement level	m_{fact}(I40KG) = 0.5	Provenance provided on resource level.
Unknown/empty values	m_{NoVal}(I40KG) = 0	Unknown values are not indicated
Consistency		
Schema restr. at insertion	$m_{checkRestr}$(I40KG) = 1	Schema restrictions are (partly) checked.
Class constraints	$m_{conClass}$(I40KG) = 1	Empty set of class constraints.
Relation constraints	$m_{conRelat}$(I40KG) = 1	Domain and range are consistent.
Relevancy		
Ranking of statements	$m_{Ranking}$(I40KG) = 0	Ranking of statements is not feasible.
Completeness	–	No gold standard available
Timeliness		
Frequency of the KG	m_{Freq}(I40KG) = 0.5	Discrete periodic updates, also through the automated pipeline.
Validity period of stmts.	$m_{Validity}$(I40KG) = 0	Validity statements are not intended.
Modification date of stmts.	m_{Change}(I40KG) = 0	Modification dates are only supplied on knowledge graph level.
Ease to understand		
Description of resources	m_{Descr}(I40KG) = 1	All resources have a label and comment.
Labels in multiple lang.	m_{Lang}(I40KG) = 0	Only some resources have multi-language annotations
RDF serialization	m_{uSer}(I40KG) = 1	Serializations in Turtle and RDF/XML
Self-describing URIs	m_{uURI}(I40KG) = 1	Self-describing URIs are always used.
Interoperability		
Blank nodes & RDF reification	m_{Reif}(I40KG) = 1	No blank nodes or RDF reification.
Serialization formats	$m_{iSerial}$(I40KG) = 1	RDF/XML and Turtle are supplied when dereferencing URIs.
Using external vocabulary	m_{extVoc}(I40KG) = 0.65	Ratio of external properties
Used proprietary vocab.	$m_{propVoc}$(I40KG) = 0.63	34 classes and 23 proprietary properties without relations to external definitions out of 66 classes and 88 properties
Accessibility	m_{access}(I40KG) = 1	cf. Tab. 6.1
License	$m_{macLicense}$(I40KG) = 1	Machine-readable licensing available
Interlinking		
Interlinking via owl:sameAs	m_{Inst}(I40KG) = 0	owl:sameAs not appropriate for external links. *sto:hasDBpediaResource* used wherever possible.
Validity of external URIs	m_{URIs}(I40KG) = 1	External URIs are resolvable.

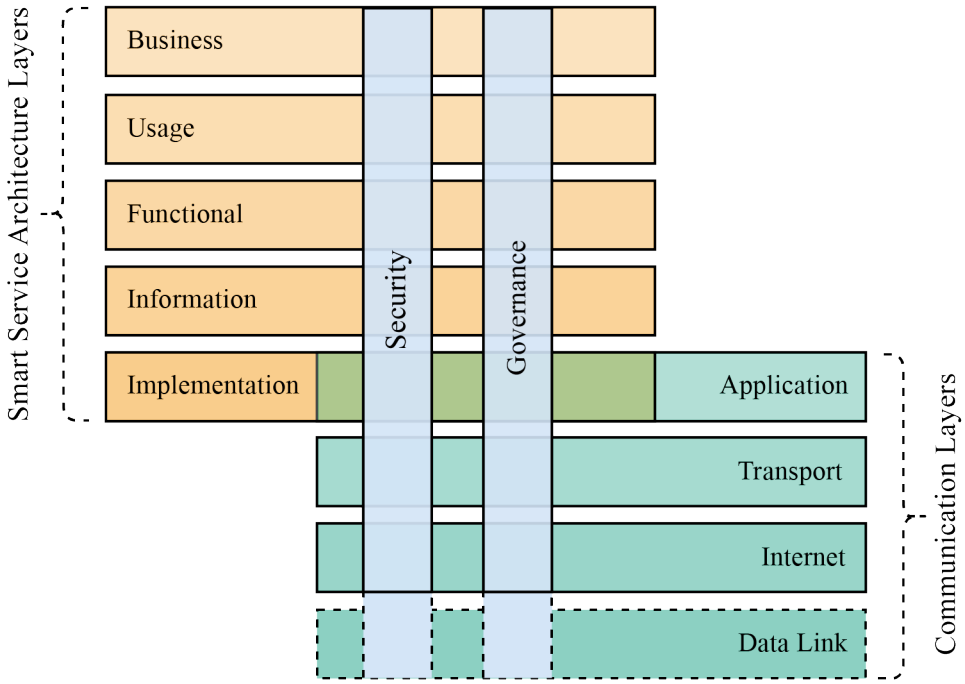


Figure 6.7: Layers (horizontal) and perspectives (vertical) of Digital Twins. Architecture Layers describe higher-level concerns and interactions, while the Communication Layers contain the message exchange functionalities.

Despite the variety of models and frameworks, a number of best practices and similarities can be identified. One recurring pattern is the structuring of the architecture views in hierarchical layers, organizing components, requirements, and functionalities in separated categories. Usually, the dependencies between different layers are restricted to the direct neighbors – a pattern simplifying the relations and reducing the respective complexity.

Most modern reference architectures start with a **Business Layer**. This layer contains definitions about the independent stakeholder roles of a system or network, which might have economic or otherwise originating interests of the described systems. System administrators, developers, service suppliers, or customers are typical entities of a Business Layer. Their intentions, conditions, and workflows need to be specified in order to depict the requirements for the following layers. Diagrams using Business Process Modeling Notation (BPMN) or similar notation patterns may support generally understandable provisioning of information.

The task of the **Usage Layer** is to illustrate the interactions of the previously mentioned Business Layer. Workflows are separated into interaction sequences and processes. UML-based sequence diagrams are commonly used to indicate information flows. The purpose of this layer is the definition and specifying of the basic interactions and information flow between the system's components, depicting a common set of exchange patterns.

Depending on the selected framework, the **Functional Layer** might even occur before the Usage Layer. It explains the mandatory and optional features of each component, documents inputs and outputs, and states side effects. The Functional Layer should not contain API documentation or specification, as those depend on implementation-specific decisions like used protocols or data formats. Still, it needs to further outline each functionality mentioned in the Usage Layer and define its characteristics as far as the other relevant components are affected.

The **Information Layer** – or sometimes called Data Layer – specifies the meaning of entities and

processes. While, on the one hand, the description of data objects is supported by the provisioning of annotations and attributes through a shared terminology, the thereby defined common understanding also eases the communication between the stakeholders themselves. The resolution of synonyms and homonyms is essential to collectively reach a common goal and to prevent misunderstandings. That means that in addition to the definition of data objects, their syntax and meaning, the Information Layer also aligns the different terminologies of the various stakeholders through a lingua franca.

Depending on the specifications of the Information Layer, the **Implementation Layer** targets the actually used technology stacks, created software artifacts, and APIs. The above-mentioned interactions and functionalities need to be backed up with applicable solutions and reflected in code. The Implementation Layer is the first one containing executable code and realizes the above-outlined characteristics with current systems.

While the Implementation Layer represents the lowest section of most reference architectures, an IIoT ecosystem fundamentally relies also on network and communication functionalities. The Internet Protocol Suite [265] depicts a standardization framework for structuring the underlying communication stack. Developers and users of IIoT Digital Twins need to understand their relevance. For instance, security-critical services need to ensure that proper encryption mechanisms are applied and that also no unintentional uncovered metadata can compromise their usage. That implies that a requirement for protected communication at the Functional Layer can easily be prevented through insufficient configurations at the Transport Layer. Routing information or interaction frequencies can be observed even if the message content itself is properly encrypted. Observing such meta-information may be valuable already to identify communication partners or guess their intentions.

Different from the other layers, the variety of patterns and used technologies generally decreases towards the Transport and Internet Layers. One can identify an informal agreement that the Internet Protocol (IP) more and more acts as a common denominator not only for the IIoT but also for digital communication in general. The Data Link Layer and any lower functionalities are therefore less relevant for this chapter and are only briefly reflected in the following.

As already mentioned, considerations about the **Security** of an IIoT ecosystem itself but also how it affects the whole network architecture is an indispensable necessity. Different from the previously mentioned features, security must be regarded across all layers and functions (cf. Fig. 6.7) and affects distinct entities, mostly Digital Twins as far as this thesis is concerned, and IIoT networks at the same time. Although sometimes treated as a functional characteristic in order to reflect its importance, the implementation of security must not be seen as a present characteristic but as an ascending vector and a dynamically evolving goal. A sufficient security level needs to be defined individually, taking the intended environment, the purpose, and the costs of the implementation into account.

Digital Twins in the IIoT interact and impact their surrounding in many ways. Business processes and message exchange as regarded in the respective layers only represent a subset of relations and dependencies. The **Governance Perspective** covers the concerns of less obvious stakeholders, for instance governments, external market players, or socio-political organizations. Closely related is the concept of **Compliance**. Both concepts intend to achieve the main goals in a way that all applicable laws and regulations are met and at the same time adhering to their own standards and values. Such values can contain the desire to create eco-friendly and sustainable solutions but also to prohibit discrimination of certain user groups. The distinct derivation of values is certainly a sophisticated challenge, especially as social norms change over time. The target of the Governance or Compliance Perspective is, however, to ensure that the legitimate rights of each stakeholder are respected and the long-term success of the application is ensured.

The general reference architecture outlines a structure to organize the requirements and features at each layer and perspective. The specific concerns and regarded topics are presented, starting with the Business

Layer. The selected sequence reflects the belief that IIoT applications are created to generate a certain value of any kind, not necessarily only reflecting in monetary effects. Independent of this consideration, a discussion on specific design decisions can start at any point of the outlined reference architecture and derive implications to the neighboring layers and perspectives.

6.2.1 Business Aspects of IIoT Digital Twins

To ensure that Digital Twins can be successfully created and offered, various factors must be fulfilled for all data ecosystem participants as well as for the overall system. The driving mechanism behind ecosystems is the value co-creation of services with data as the key resource. Consequently, data must be considered as an Asset with an economic value behind it. In particular, from a business perspective, it must be clear what the data can be used for and how it supports decision-making, especially for the data provider.

As a starting point for the use of data as a resource, it is necessary to establish a strategic framework (cf. Governance Perspective) for efficient data management. The definition of roles and assignment of responsibilities for data handling is of high importance to establish a sense of responsibility for data and to enforce corresponding guidelines.

Challenges are especially to convince customers to pass on their data and to motivate them to exchange it. To do this, it is elementary that the security of the data is guaranteed and that the use of the data does not extend the agreed purposes. The decisive factor here is that the customer (data sovereign) agrees to the appropriate use of his data. Furthermore, the rights of use or ownership of data must be explicitly clarified. For this purpose, individual contractual regulations must be created to eliminate ambiguities. The question of data rights is essential for the sale of data and subsequent data trading, explicitly regulating who may do what with which data objects. If several customers and partners join the data ecosystem, it is also possible to achieve network effects. Data from several sources and ideally across different stages of the value chain lead to more detailed statements about the behavior of machines and plants under different environmental conditions. Thus, even more dense knowledge can be created, and services can be continuously improved so that, in the end, the customers benefit once again from their involvement in the overall system. Further information about the challenges and potential of data-driven networks can be found, for instance, in the BDVA Strategic Research and Innovation Agenda [266].

6.2.2 Usage Aspects of IIoT Digital Twins

After identifying the stakeholders and necessary capabilities of the business parties, the interactions and roles need to be defined. The Usage Layer serves as a bridge between actors and technical roles. Clients and suppliers are mapped to system operators, users of Digital Twins, or providers of data. While the former is determined by its interests, the latter contains attributes, capabilities, and permissions.

Business interactions driven by the exchange of resources, money, and value are translated into interaction workflows and processes. Relations between organizations and legal entities need to be represented through information flows between systems and networks. Business process models are replaced by sequence diagrams and vice versa.

The Usage Layer is, therefore, an indispensable bridge between the general idea about how to create value and which components and interactions are required to realize that. Detailed workflows and sequence processes depict the various involved components, both Digital Twins and human actors, and specify the mutual dependencies. Thereby, the Usage Layer clarifies the responsibilities and enables the derivation of technical requirements.

A simple example of analyses performed at this layer is the onboarding of a new organization in an IIoT network. The new entity might be registered, provided with a digital identity, and announced to the existing members. Already in this simple example, an identity provider, an exchangeable identity proof, and a notification mechanism can be derived, which then need to be reflected through functional definitions. The information flow needs to be specified and declared to all involved parties. More information about usage-related aspects can be found in the Usage View of Asset Administration Shell [267].

6.2.3 Functional Aspects

The core capabilities of Digital Twins, in particular their APIs and interfaces, the behavior, delivered quality, and other characteristics, are affecting their perceived value. At the core are the technical preconditions required to implement Digital Twins. These *functional* features include the necessary inputs, input formats, the activation sequence, and other dependencies required to interact with the Digital Twin. Furthermore, the capabilities to discover, compose, orchestrate, and deploy a Digital Twin instance are depicted.

In addition to the preparation steps, the functional aspects of a Digital Twin also include documentation of the executed transformation and its output in terms of interactions with the environment and the produced data. The capabilities to interact with other parts of the network need to be defined and the life-cycle stages specified.

However, even though the description of the core functional features allows the understanding of how to interact with a Digital Twin but does not explain whether or not the offered functionality is actually appropriate for the regarded situation. *Non-functional* requirements are crucial information to enable well-founded decisions in this regard. Most importantly, the Quality of Service and the price per interaction determine the suitability of a distinct offer to actually solve a task. Other categories of non-functional requirements include the maintainability or usability of systems but also the reliability or availability of Digital Twins. As those aspects are also framed by the Security and Governance Perspectives, it is obvious that these sections do actively affect each other and cannot be treated independently.

Another category of relevant features is depicted by the *behavioral* characteristics of Digital Twins. This category discusses the operations and states which can be reached. Especially as the Digital Twin comprises a certain autonomy and the ability to independently adjust their decision-making, their behavior is not strictly determined anymore. In particular, this implies that definite predictions of their behavior are not possible. Still, in order to ensure safe executions, the boundaries of their operation space need to be described. This information allows other components to prepare for effects in these given limits. For instance, a control unit for an electric power grid might dynamically adjust its voltage output autonomously and self-controlled but needs to respect the safety norms for electronic devices. The IoT-Architecture description [242] offers more information about the Functional Layers.

All these features and capabilities need to be described in ways that both humans and other Digital Twins can read and process. The unambiguous and logically consistent provisioning of these descriptions requires extensive data models and vocabularies. While these definitions are crucial for the composition, orchestration, and therefore operation of IIoT networks, they are usually treated in the next category, the Information Layer.

6.2.4 Data Formats and Semantics

Digital Twins are the actuators in an IIoT environment defined by digital information. The ability to request, transform, analyze and forward data objects is at their core. The challenges at this point can

mainly be grouped into the two categories of *syntactic* and *semantic* heterogeneity. While humans are usually capable of guessing the meaning of unknown data based on the delivered context or their personal experience, Digital Twins require formal methods and explicit world models. Both are hard to accomplish and require elaborate techniques and significant additional efforts.

Syntactic heterogeneity frames the variety of data formats and provisioning methods. Information is stored in closed silos and proprietary formats exchanged through customized interfaces and implicit, non-transparent dependencies. Tackling this issue, an informal movement towards JSON-based data exchange can be recognized, which to some degree reduces this hurdle. While other data formats (for instance XML or relational datasets) certainly have their justification and will not be abolished, the adoption of JSON imposes reduced efforts and requirements for developers, technology stacks, and maintainers.

Recognizing and communicating the semantic meaning of data entities is another significant challenge. Digital Twins provide their full potential in areas where previously unexpected information appears and needs to be processed. Controlled vocabularies are a lightweight approach to accomplish a shared understanding, usually applied in distinct domains or industries. Taxonomies add further hierarchical structures in the form of part-of or subclass relations. Successful examples are ECLASS[10] and the IEC Common Data Dictionary [268] for domain-specific identifiers.

Even more explicit formalizations and machine-interpretable logic is provided by ontologies and logically grounded vocabularies. The Semantic Web Stack, based on the graph-based Resource Description Format (RDF), proposes a formal ecosystem of technologies, standards, and tools to represent information in both human and machine-readable manners. Linked Data extends this further and allows the distributed provisioning of RDF data using the established practices of the Web [269]. This enables the direct linking to a publicly available, distributed knowledge graph using Web standards, called the Linked Open Data Cloud. Another benefit of the formal definitions of RDF and Linked Data is the usage of upper-level ontologies. These well-known and widespread dictionaries serve as logical anchors for references and more fine-grained derivations. Digital Twins can use the thereby supplied terms and attributes to directly express meaning. For instance, an attribute referring to the Dublin Core property 'creator'[11] unambiguously tells the data consumer who the originator of a certain data resource is.

A huge amount of semantic description languages for interfaces and federated systems has been proposed already. The SOAP technology stack and its service description language WSDL, for instance, have been extended with the WSMO and WSMO-Light ontologies [137]. OWL-S is a similar OWL-based ontology for semantic descriptions of interfaces and especially for Digital Twins. Furthermore, description languages for REST-APIs have recently gained popularity, most prominently OpenAPI, previously named 'Swagger'[12]. Similar, API Blueprint[13], and RAML[14] but also the RDF-based Hydra [98] vocabulary illustrate the great need to unambiguously describe remote interfaces in both human and machine-readable manners.

6.2.5 Implementation Aspects

The Implementation Layer contains the technology-specific considerations of a Digital Twin. The accessible remote interfaces and APIs are specified and their concrete instances designed. This layer serves as the bridge between the upper-level sections used for describing the Digital Twin in his IIoT

[10] https://www.eclass.eu
[11] see http://purl.org/dc/elements/1.1/creator
[12] https://swagger.io/docs/specification/about/
[13] https://apiblueprint.org
[14] https://raml.org/

networks and the communication-specific characteristics. According to Figure 6.7, this layer fairly corresponds to the Application Layer of the Internet Protocol Suite. The interactions with resources, operations, and interactions are translated into executable code, relying on the respective functionalities provided by the network layers below.

The Implementation Layer thereby abstracts the actual implementation logic for the description and presentations at the Information Layer. Relevant interaction patterns in an IIoT network include request/response and publish/subscribe. While the former is suitable for point-to-point connections where a distinct client initiates the communication and asks for data, the latter pattern has its strengths in one-to-many communications. In a publish/subscribe scenario, both the provider and consumer of data interact with an internet server, which receives and then further publishes them based on prior subscriptions.

Both patterns support a loose coupling of components over internet networks. The thereby defined distinct roles and responsibilities ease the integration of new components and actors and allow the network to grow dynamically. Especially for the request/response pattern, the further restrictions imposed by the Representational State Transfer (REST) are currently considered as the de facto standard for resource-oriented systems on the Web. The clear interaction model and the deep integration with HTTP reduce complexity both for the client and the server.

In addition, also bidirectional point-to-point interaction is necessary for certain scenarios. WebSockets create a virtual tunnel on top of HTTP, enabling the exchange of data from both ends. Relying on HTTP offers easier integration with current networks and firewalls as those are typically well-configured for HTTP connections. Still, with the upcoming of more and more resource-restricted devices equipped with internet connections, a significant demand for lightweight and resource-conserving protocols appeared. While modern Web browsers can cope with the requirements of HTTP-based communication, machine-to-machine – or Digital Twin to Digital Twin – scenarios often do not rely on the thereby created overhead. Driven also through the development of the IIoT, the Constraint Application Protocol (CoAP) presents a downsized alternative for request/response interactions. The MQTT protocol can similarly serve for publishing and subscribing to messages with only a limited network bandwidth.

In contrast to these relatively simple protocols, OPC UA outlines a complete technology stack highly integrated for industrial manufacturing facilities. The goal of OPC UA is not to replace existing protocols but rather to support the transmission of information for new applications in the IIoT. OPC UA is a client/server protocol based on TCP/IP that defines operation calls for the interaction with a server-side information model over the network. Recently, the OPC UA specification has been extended for communication based on the publish/subscribe communication paradigm. Two options are supported: (1) broker-based message distribution according to the IEC standards AMQP and MQTT and (2) a custom UDP-based distribution protocol, called UADP, based on the multicast mechanisms of the IP standards. Open62541[15] is the world's first open-source implementation of OPC UA publish/subscribe and demonstrated its real-time capability in combination with Time-Sensitive Networking (TSN).

A deep discussion on the Implementation Layer and its effects and requirements is outlined in the Industrial Internet Connectivity Guide [84].

6.2.6 Transport and Internet Layer

One reason for the success of the internet as the de-facto standard for today's digital communication is certainly the acceptance and adaption of a compact set of transport protocols. TCP and UDP constitute the common denominator for all distributed architectures, in particular for the IIoT. The remarkable

[15] http://open62541.org

success of these two protocols consolidates the approaches of the previously mentioned protocols and establishes the common infrastructure.

That infrastructure is based on the IP network protocol. Currently, the address restrictions of IPv4 led to the introduction of IPv6. Even though the thereby defined address space is drastically larger than for IPv4, the transition happens significantly slower than expected. Network operators and other infrastructure provider have not updated their routers at the originally predicted speed. In addition, a high number of legacy systems are still in place and can only be replaced gradually. Both developments slow down the conversion to IPv6, requiring a set of intermediate approaches.

Underlying technologies ensuring the physical transportation of signals are usually not in the focus anymore. Ethernet or WiFi are examples of mature and proven technologies providing the foundation for the whole digital communication. Their capabilities are well understood, and their implementation straightforward.

However, that does not imply that developments at those points do not impact IIoT applications. For instance, the transition from 4G to 5G wireless communication techniques is widely regarded as a breakthrough affecting all layers above. The latency, bandwidth, and transportation speed promises disrupt current application scenarios and thereby create completely new business models. Still, while such developments affect the perceived value for IIoT users, the functional interactions are expected to stay nearly the same. The transition costs and efforts for providers and consumers are therefore manageable due to the integrative nature of the Transport and Network Layers.

6.2.7 Security and Trustworthiness

Security is often treated as a necessary but tedious task, an additional feature with significant effort but limited added value. This results in a set of uncoordinated approaches instead of a comprehensive security strategy. While Security by Design has drawn significant attention in the recent discussions, the lack of even basic mechanisms in current implementations emphasizes the need for further progress here. State-of-the-art encryption or reliable update functionalities are mandatory for each and every IIoT application and must be part of the basic toolbox of every developer and architect.

One reason for the unsatisfactory state of current implementations is also the complexity of the topic. In the following, six categories are distinguished to better structure discussions, where each category has its own focus and requirements:
Privacy frames all topics intended to conceal and protect data and to prohibit its unintended distribution. The *Reliability* of IIoT Digital Twins explains which measures are in place in order to achieve its general functionality over time.
Resilience characterizes the ability of single Digital Twins – but also of the complete IIoT environment – to withstand attacks and to prohibit penetrations.
Security, as used in the following, is related to IT and communication protection and includes, for instance, encryption techniques.

Lower-level priority or even ignored are often the implications of Digital Twins into the physical world. As the adoption of IIoT applications becomes more and more widespread, the boundaries between the digital and physical world decline further.
Safety – or the ability of an Asset to prevent harming humans, equipment, or any other entity in its environment – becomes a relevant topic. Digital Twins need to understand the context they are deployed in and which consequences their actions may have. Without that, their ability to autonomously decide and act involves an unacceptable risk for their surrounding. While the legal responsibilities have not yet been sufficiently understood or determined, all involved stakeholders of IIoT applications already need to prepare themselves for the disruptive nature of such developments.

While most other functionalities and features can be implemented and verified in objective ways, this is fundamentally different for security and related aspects. A certain message format can be recognized or not, a payment process can be executed or not, or a Digital Twin can deliver its data or not. In contrast to that, a designer of a Digital Twin can never know in advance if the amount of protection is sufficient. The only reliable information flow appears after an intrusion, or an otherwise corrupted system, was identified. In any other situation, the obvious uncertainty of whether or not a certain security level is sufficient has to be managed.

Therefore, it is impossible to determine whether the system's security mechanisms are sufficient. Furthermore, security cannot be measured objectively but has to be treated as a moving target. Depending on the expected environment, the trade-off between protection, cost, and required user convenience has to be solved. A perfectly secured application is easily achieved by blocking any interaction. The most convenient option does not restrict any communication with its users. And the cheapest solution in terms of implementation and interaction costs disregards security at all. Obviously, neither of these alternatives is a reasonable solution.

The provider or operator of a Digital Twin should therefore try to prioritize security aspects by answering the question: "Given that a certain incident appeared, how much will this affect the perceived value of my target group?" If a simple restart can already lead to a not corrupted state, the protection against intrusion might be less important. If, however, personal or customer data can be leaked, the implemented protection features must be significantly higher.

Independent from the context of a certain Digital Twin, a basic set of guidelines can be formulated:

- State of the art encryption and authentication mechanism
- Constant and promptly updates of all externally accessible components
- Provide recovering strategies for corrupted states or intrusions
- Revocation mechanisms for lost identities
- Notification strategies for affected Data Owners

Using up-to-date communication encryption is certainly without question. In the case of long-living systems, however, the rapidly changing capabilities and computing powers can significantly impact the appearing protection level. The only reliable strategy to sufficiently secure interconnected Digital Twins, or any connected service in general, is through regularly supplied updates. Especially functionalities located close to or directly interacting with the open networks must be treated with high priority.

As already stated, security is not absolute but always determined by the trade-off between potential risks, costs, and user restrictions. Consequently, no system is completely invulnerable, arising the need for adjusted response strategies. In case of a detected intrusion or otherwise misbehaving Digital Twin, a further intrusion must be prohibited and ongoing damage limited. A reset to a non-corrupted backup might be already sufficient for state-less Digital Twins. State-full Digital Twins require more elaborate strategies as even revocation of most parts might leave malicious code untouched. Appropriate activities highly depend on the network architecture and the used technologies. Therefore, a generally valid process is not possible.

In any case, the Digital Twin operator must follow a transparent notification strategy towards its stakeholders, especially the ones whose data has been affected. The revocation of credentials, tokens, or any kind of digital identity has the highest priority to prevent further distribution. In addition, the fast provisioning of information to affected parties is the only way to regain trust and to enable them to limit the entailed consequences. The inevitable loss of reputation can only be encountered with quick

reactions and a well-organized communication strategy. The details of the reaction strategy are depicted in the Governance Perspective.

Further starting points for security guidance are provided in NIST Special Publication 800-82, [271] and [272].

6.2.8 Governance and Compliance

Digital Twins operate and interact with their environment, in particular in the IIoT. Therefore, their activities have to be evaluated regarding the surrounding context. While the main considerations at creating Digital Twins usually focus on technical features and the attempt to provide technical solutions to the occurring requirements, Governance demands a wider perspective. Indirectly involved stakeholders, for instance, the government with legislative regulations or society as a whole, do impose relevant factors.

Especially in the data economy, major questions are still unanswered. The most obvious difference to traditional business interactions is certainly the ownership of digital information and products. Copyright and licenses do support the enforcement of financial interests for certain scenarios. Still, the immanent nature of the data economy is depicted by the fact that copying and sharing of information are cheap while their creation is expensive. The production cost might be determined by the invested money but also by the number of other resources needed to create a piece of information. The perceived value, and thereby the price, is usually determined by totally different factors and therefore only indirectly affected by the original production costs.

This results in a demand for new concepts and principles on how digital applications and information need to be treated. For person-related data, the European General Data Protection Regulation (GDPR) imposes one framework to define and control the usage and dissemination of data. However, GDPR regulations are, in general, not applicable for other types of sensitive data in the context of business-to-business interactions. One approach targeting this challenge is the introduction of *Data Sovereignty*, declaring that one party in an ecosystem has a principal interest and right on a certain data resource. The so-called *Data Sovereign* is the only entity allowed to formulate usage restrictions and thereby control the dissemination of its resource. Examples for such restrictions might be the access restriction to third parties, the implementation of certain data protection mechanisms, or also reimbursing the Data Sovereign for using the resource.

The collaborating parties in such ecosystems might acknowledge the Data Sovereign's right, adjust their processes accordingly and provide trustworthy proofs to the Data Sovereign. This process affects all architecture layers from top to bottom. While the general agreement and its details must be formulated as textual contracts – only thereby become legally binding – and approved in the Business Layer, the respective negotiations (Usage Layer), capabilities (Functional Layer), and the shared understanding of the included restrictions (Information Layer) needs to be transferred and imported (Implementation Layer). In the context of the IIoT, these contracts need to be machine-readable and, in cases where autonomous decision-making is effected, also be enforceable. These requirements add significantly more logical refinements to usage restrictions, as the necessary formalization needs to be evaluated in a setting where information is generally uncertain, incomplete, and not necessarily reliable. While human actors have proven their ability to manage such obstacles all the time, the capabilities of AI systems have not yet reached this level.

In addition to the mentioned challenge of enforcing interest in the usage of a digital resource, the potential of moving decision-making capabilities, and thereby responsibilities, from human participants to Digital Twins contains a huge potential for conflicts. While the vision of self-aware acting Assets is certainly still more a topic for the future, the liability issue directly affects all involved parties in the IIoT context. The developers, operators, and consumers of IIoT systems need to realize and then

agree on the effects their Digital Twin might have. It is noteworthy that these questions are still not answered sufficiently and require significant progress in the current state of both legal, ethical, and social discussions but nevertheless need to be regarded for a successful system already.

6.2.9 Summary: A Layered Architecture for the IIoT

The described layered *Architecture for IIoT Systems* (CO6) of this section is a condensed model representing the reoccurring categories of the currently published works. It comprises a structure to locate topics and align design approaches with a shared understanding of the respective dependencies. The usage of layers, each encapsulating its requirements and suitable solutions from the other layers, helps to simplify the search for applicable solutions.

Obviously, countless further models and frameworks have been proposed, for the IIoT, for Digital Twins, and the combination of both. Some of those are rather similar to the one of this section, for instance, RAMI4.0 [74] or the IDS Reference Architecture Model [35], while others differ significantly, for instance, the IoT-Architecture Model [242] or the Edgecross Framework [261]. That fact underlines the argument that no single model can contain all relevant aspects and views. Still, the necessity for an alignment approach exists, as the various stakeholders of IIoT applications need to converge their designs to increase the interoperability of the resulting systems. This demand is targeted in the next section.

6.3 Visual Analytics of Standards and Frameworks

[31] The industry and research efforts to standardize IIoT related developments have merged into an unmanageable amount of reference models, architectures and specification activities. As these efforts have only been roughly coordinated, an incomprehensible and confusing landscape occurred. These developments contradict the initial need for more clarity and structure, especially as many different aspects are framed under the same terminology.

Especially the manufacturing industry is actively involved in numerous activities related to this topic. Organizing this area and enabling effective discussions and design decisions are the targets of several standardization efforts. Many of them provide reference frameworks and architecture models. Reference frameworks in this context provide the necessary structure to transform the combined experiences and best practices, the opportunities of available technologies, and the expected implications into understandable guidance for the involved stakeholders [273].

As a common understanding has not yet been reached, the current situation is characterized by the variety of proposed models and frameworks created by groups of experts from different countries and domains. Whereas the goal of each approach is to overcome the current confusion, the huge number of published models is again becoming a source of heterogeneity and misunderstanding. Newcomers and non-experts are overwhelmed by the amount of published recommendations and suggestions, contradicting terminology, inconsistent structuring, and proposed best practices. The uncountable efforts intended to structure the domain have by now created another dimension of complexity. The thereby created barriers aggravate the adaption of crucial developments and decelerate further progress. Moreover, the rising difficulty to find and classify relevant information undermines the further propagation of the core principles.

The presented knowledge graph is the basis for an in-depth analysis of core features and components, concerns, and requirements. These insights have been provisioned in interactive visualizations for standards, concerns, and reference frameworks [31, 32].

The contribution explained in this section is an interactive overview of the current state of IIoT

standardization. A graphical landscape of IIoT specifications and standards based on the integrated knowledge graph from Chapter 6.1 contains various, adaptable views with different illustrations adapted to the different information needs. Explicitly stated relations between IIoT frameworks and technical standards enable the flexible discovery of related information. In addition, the use of machine inference techniques adds new links and further extends the displayed content.

In particular, this contribution faces the depicted challenges by:

1. Providing a methodology to structure, align and compare the various reference frameworks;
2. Presenting a collection of relevant concerns, their hierarchical structure and relationships;
3. Providing configurable visual views of the characteristics and relations between the concerns and the reference **frameworks**;
4. Offering an analysis of the thereby gained insights, for example, frequently covered areas or inconsistencies, which need further attention from the community.

The ongoing IIoT discussions can be categorized by the groups of stakeholders. Experts have organized themselves in various consortia and standardization organizations in order to create reference frameworks and technical standards. Experienced developers and system architects are familiar with parts of the related technologies but search for more detailed information for their specific challenges. Newcomers require a structured introduction to the field in order to recognize key players and to effectively gain a first overview.

All involved stakeholders, however, face the challenge that a comprehensive search through the already existing literature requires too much effort. Traditional mechanisms, for instance based on personal communication, media articles, or professional training, cannot face the speed and complexity of the developments. Furthermore, an individual examination is very time-consuming and bears the risk of missing important aspects as of unknown terminology or biased search strategies. While specific problems require in-depth knowledge, only a certain degree of familiarization with the IIoT topics allows the contextualization of information, bridging the gap from knowing facts to being able for reasonable decision-making.

Collecting the state of the art of IIoT reference frameworks is crucial to prevent the further fragmentation of the field. As the value of new applications highly depends on their interoperability with other systems, following best practices and relying on widespread patterns is essential. In contrast to that, the widespread interest in the topic led to an overwhelming variety of usages, interpreted in different ways and regarded from different perspectives. This heterogeneity needs to be structured in order to first identify challenges, then discover relevant information material, and finally best practices and common methods. An approach targeting this challenge needs to consider the variety of interest groups. For instance, decision-makers require detailed knowledge of objectives and risks. System architects need a structured reference point to distill guidelines relevant to their use cases. Developers have to be able to identify suitable software artifacts and their capabilities.

In contrast to the number of publications reviewing IIoT topics, as well as other digitization initiatives and data exchange frameworks, significantly less work can be found on approaches that actually introduce the concepts into usable guidelines for the industry. While the existing literature repetitively outlines the key technologies and abstract processes, the efforts to transform those ideas to appropriate guidelines and specifications are hardly investigated.

Three directions are proposed to sort the landscape: First, the IIoT standards are sorted by the IIoT reference frameworks, which explicitly promote their usage. Readers are thereby enabled to start with the frameworks they heard about and discover the technical specifications behind them. Another set of views

examines dependencies and connects related standards. This approach is inspired by the already informed expert who is aware of a number of relevant specifications. Finally, a list of potential requirements for an IIoT system is created. The interested user is directly guided to the relevant sections of the IIoT reference frameworks but also the respective standards.

6.3.1 Interactive Presentation and Information Selection

```
PREFIX sto: <https://w3id.org/i40/sto#>
SELECT ?class ?relatedClass
WHERE{
   # $classType = sto:Standard, sto:Concern
   ?class a $classstype .
   OPTIONAL{
   # $relation = sto:relatedTo, sto:frames
     ?class $relation ?relatedClass .
   }
}
```

Listing 6.1: Configurable SPARQL Query. Parameters are provided by a set of configurable queries.

A public SPARQL endpoint[16] provides a Web interface to the IIoT knowledge graph. In addition to the pre-processed reasoning with the axioms of Tab. 6.2, another reasoning mechanism comes into place through the SPARQL queries. Fig. 6.1 shows one of the applied query templates. The query here looks for relations between classes. Parameter *$classType* can have values `sto:Standard` or `sto:Concern` while *$relation* can have values `sto:relatedTo` or `sto:frames`. The respective information is not necessarily stated in the knowledge graph and can be discovered during the query execution phase. As most users are not familiar with this query language, nor with analyzing the raw knowledge graph in plain RDF, a set of web views has been created[17]. Each view generates unique projections of the knowledge graph by converting user inputs to customized SPARQL templates. The respective query results are rendered using the D3 JavaScript library[18].

Figure 6.8: Web service to analyze and discover IIoT Standards and Reference Frameworks[17] .

The hierarchical relations between the IIoT reference models, their classifications, and the related standards are depicted by zoomable circle diagrams (cf. Fig. 6.9). Thereby, a fast and easy-to-grasp discovery of each technical reference is achieved. The zooming intuitively orders the entities for the user without further explanations. Concepts of the same class are positioned on the same level and ordered by their upper-level affiliation. For instance, the RAMI4.0 layers and dimensions are presented next to each other, containing the respective standards and technical specifications.

However, a plain, unfiltered view of the whole knowledge graph has only minor usage possibilities. Still, the traversal of connections between a filtered set of nodes might uncover unknown relations. Instances of one or at maximum two classes can still be visualized properly even though a rising number of connections impacts the readability. Especially for classes with a limited number of relations per instance, for instance standards and classifications, the plane graph visualization can still be feasible.

[16] https://vocol.iais.fraunhofer.de/sto/querying
[17] https://i40-tools.github.io/StandardOntologyVisualization/
[18] https://github.com/d3/d3

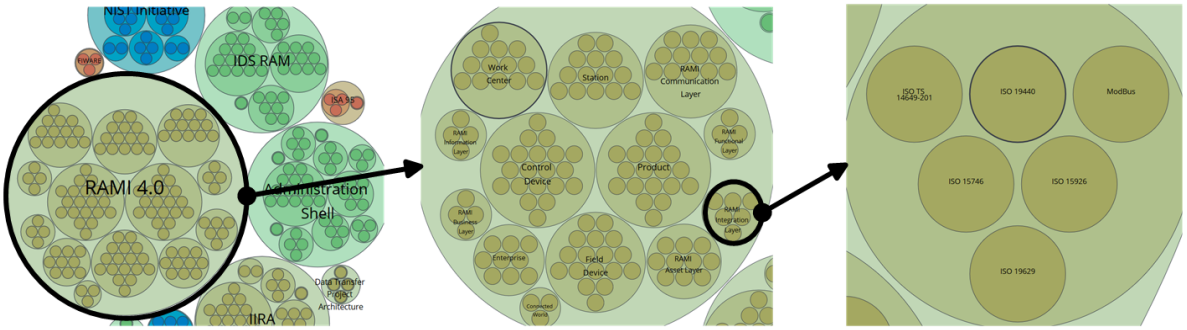


Figure 6.9: Visualizing reference models and their associated standards. The three zooming stages visualize the IIoT reference models, their related classification categories, and the referred standards.

Set-related coherences are illustrated in the form of Venn diagrams (cf. Fig. 6.11). While the hierarchical representation of Fig. 6.9 can put several levels of relations into context (reference frameworks to classifications to standards), Venn diagrams are limited to two dimensions. However, the representation of overlaps and disjunctions is directly presented by the latter. One has to notice that due to the chosen rendering algorithm, only up to three circular sets can be drawn without a potential loss of information. A fourth set, having intersections with all previously drawn areas, will result in areas that can not be positioned in the two-dimensional view anymore. While this is still theoretically possible, no currently available rendering engine solves this issue.

Similar to Venn diagrams, co-occurrence matrices compare the relations between two classes of entities (cf. Fig. 6.12). Co-occurrence matrices are better suited when high numbers of relations appear in the hierarchy of the entities, or directions of the relations are less relevant. Therefore, co-occurrence matrices are selected to illustrate the relations of the concerned class with the other classes. Utilizing this table-like structure enables the user to interpret the view in two ways, either comparing the identified concerns with their related reference frameworks or vice versa.

6.3.2 The IIoT Landscape from a Standardization Viewpoint

In the following, three usage scenarios are outlined. Alice, Bob, and Charlie represent three typical users, each with a different background and information need in the context of IIoT.

Alice, who is just starting with IIoT applications, needs to quickly gain an overview of the most influential reference frameworks. She has to communicate with consultants and potential suppliers using the correct terms in order to effectively manage her resources. Alice looks through the hierarchy view (Fig. 6.9), learning which frameworks contain which categories and standards. Focusing on any of the entities allows her to learn more about it in the sidebar. She makes notes about them, as well as a quick overview of which standards are the most prevalent in almost all the frameworks. Next, she finds the relations between the classification categories and follows the links to relevant standards or other publications but also to other, less well-known reference frameworks. This view gives a general impression of the structure of the reference frameworks. The categories of each framework are interactively displayed, allowing to discover standards at the relevant position.

Bob, a senior system architect, is aware of the concerns and requirements of the system he needs to implement. With the aim to ensure the effectiveness and reliability of his architecture, he searches for best practices for implementing upcoming technologies and to check the suitability of the latest trends (cf. Fig. 6.10).

The co-occurrence matrix (cf. Fig. 6.12) depicts which reference frameworks and which respective

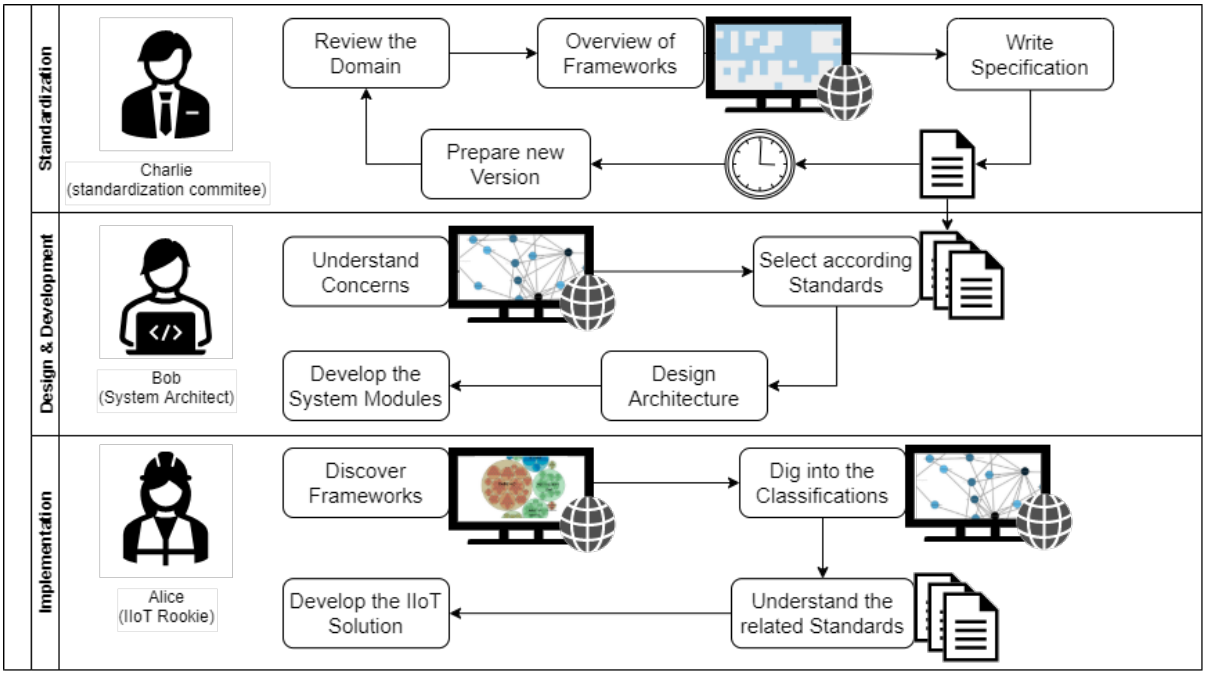


Figure 6.10: Workflows of the IIoT Landscape for the different roles as depicted in the use cases.

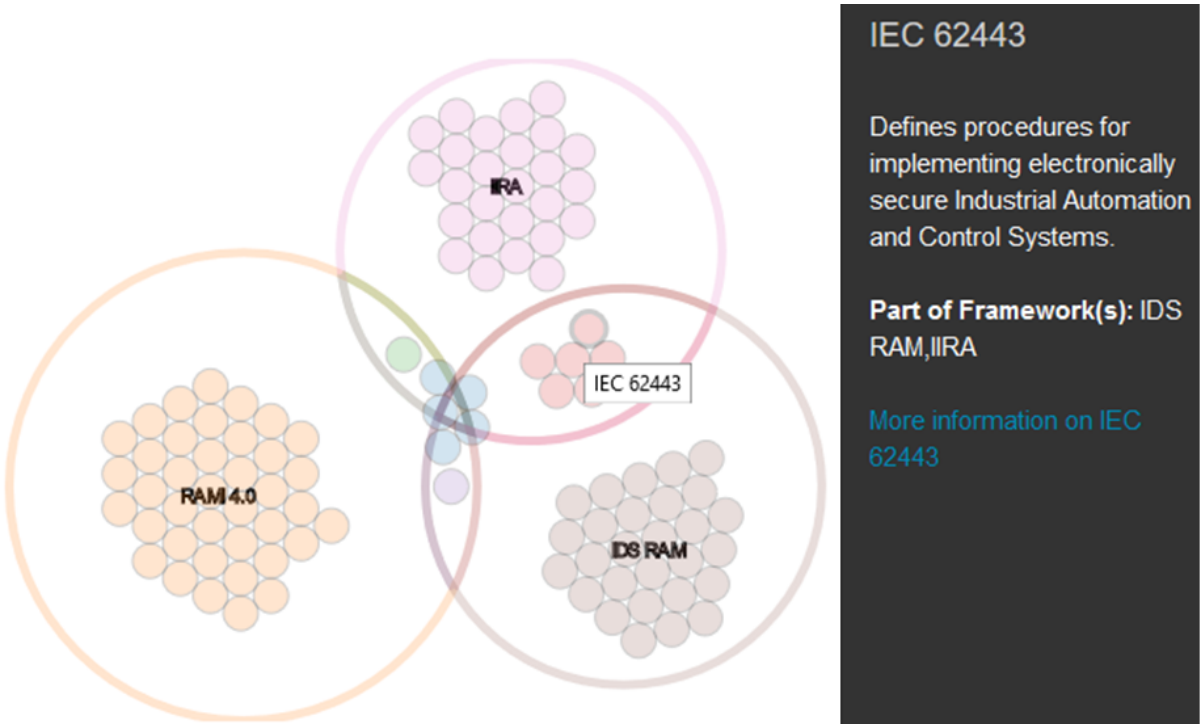


Figure 6.11: Venn diagrams for reference frameworks and standards. The Venn diagrams localize the standards in regard to the reference frameworks. That way, a user can instantly recognize the overlaps and unique areas.

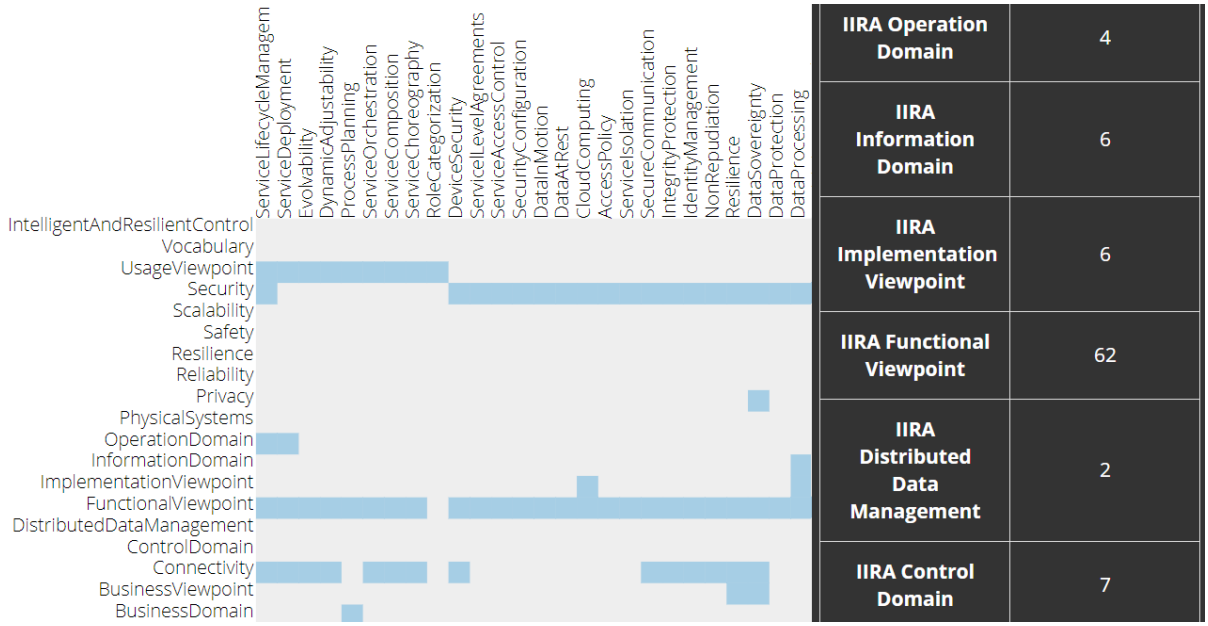


Figure 6.12: Co-occurrence matrix between concerns and classification categories. The co-occurrence matrix enables insights into which concerns are targeted by which classification categories of the presented frameworks.

classifications frame Bob's concerns. He is directed to the most suitable models and can find the necessary guidelines to cope with his concerns. In addition, he is able to analyze whether a framework of his preference misses any important aspects and which alternative models might have the required specifications.

The third stakeholder using the application is Charlie, a collaborator of a standardization framework. He knows his own reference framework in detail and is informed on the implicit and explicit concerns, which led to proposed solutions of his framework. Now, Charlie wants to know what the other frameworks are proposing and whether there are similar views in order to find existing and/or superior recommendations. Furthermore, Charlie searches for good ideas for his own standardization work.

Charlie takes a look at the framework and standards overlap visualization, notices his framework has many standards in common with another framework (cf. Fig. 6.12). This leads to an investigation of the other standardization framework and the collaboration possibilities or coming up with new ideas for his own work.

The ongoing digital transformation has the potential to revolutionize nearly all industrial manufacturing processes. However, its concrete requirements and implications are still not sufficiently investigated. In order to establish a common understanding, a multitude of initiatives have published guidelines, reference frameworks, and specifications, all intending to promote their particular interpretation of the Industrial Internet of Things. As a result of the inconsistent use of terminology, heterogeneous structures, and proposed processes, an opaque landscape has been created. The consequence is that both new users and experienced experts can hardly manage to get an overview of the amount of information and publications and make decisions on what is best to use and to adopt. The contributions of this chapter extend the state of the art by providing a structured analysis of existing reference frameworks, their classifications, and the concerns they target. The alignments of shared concepts identify gaps and give a structured mapping of regarded concerns at each part of the respective reference architectures. Furthermore, the linking of relevant industry standards and technologies to the architectures allows a more effective search for

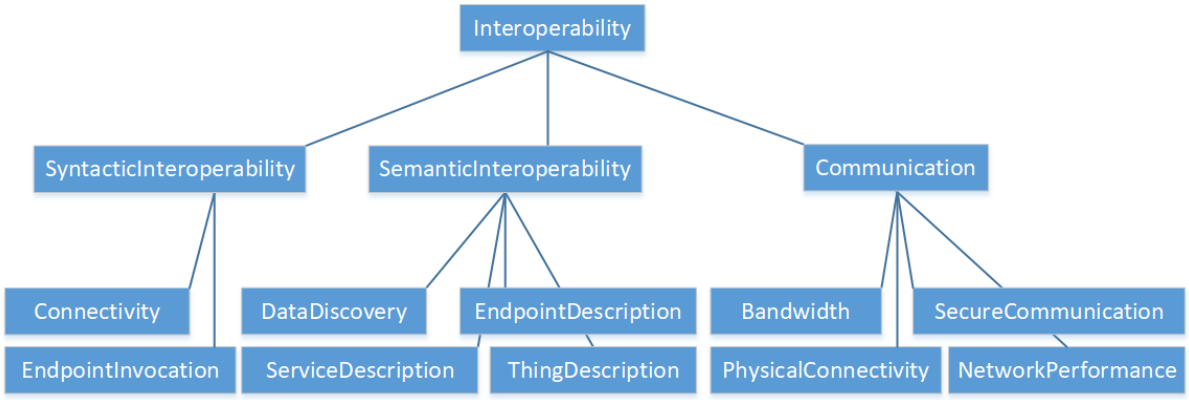


Figure 6.13: Interoperability-related concerns. Lower level concerns influence the fulfillment degree of higher ones.

specifications and guidelines and supports direct technology adoption.

6.3.3 Considered Concerns

This section briefly outlines the subset of collected concerns based on Yaqoob et al. [274] and the results of the IoT-Architectures research project [247], categorized in a hierarchy with interoperability, trustworthiness, and business aspects as the top-level entities. Following ISO/IEC 42010, concerns are the aspects or challenges of an IIoT system that are covered by a certain section of a reference framework. A comprehensive list of relevant concerns allows the identification of blind spaces but also to outline different definitions and scopes of reference frameworks. The hierarchy of concerns creates a taxonomy-like structure. Lower level concerns influence a higher-level one in terms of supporting its satisfiability. For instance, having a system with secure communication, this fact indicates a higher level of security and, therefore, the trustworthiness of the system. Modeling such relations explicitly outlines previously intangible connections. Furthermore, formalizing information in the graph enables automated reasoning processes to further populate facts and extend the contained facts. The thereby created network supplies a defined vocabulary, which is used to match the distinct terminology of the various frameworks and to map their assertions. In addition, the formalized relations between the concerns themselves enable the precise assignment of contributions of very specific, lower-level concerns to more general, higher-level ones. No current work contains a comparable structure of IIoT concerns and their respective interconnections or applies its structuring of the domain based on formalized dimensions like the one presented in this chapter.

In the following, the most central concerns with a selection of their direct sub-concerns are introduced. *Interoperability* or connectivity is the fundamental building block for enabling IIoT ecosystems (cf. Figure 6.13). Two IIoT components are interoperable when they can work together without any restrictions or additional adjustments. In this sense, interoperability contains integration aspects as endpoint descriptions and communication patterns. Therefore, both syntactic and semantic interoperability has to be established. *Syntactic interoperability* characterizes all aspects in order to exchange data, for example, network protocols and data formats. *Semantic interoperability* targets a shared understanding of the meaning of the data, for example, by information models.

Trustworthiness and the highly related *security* of data exchange ecosystems determine the ability to prevent unintended or unauthorized access, change or destruction and therefore behave as expected. *Access control*, *provenance tracking*, *identity management*, and *authorization* but also *reliability*, *availab-*

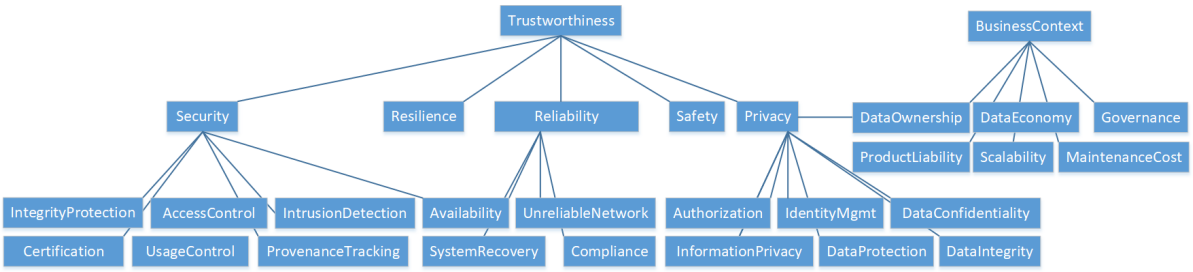


Figure 6.14: Selected extract of trust- and business-related concerns. Only the first two hierarchy levels are presented.

ility, and *resilience* influence security and are required for a trustworthy system (cf. Figure 6.14). Access control frames all methods necessary to ensure data exchange only for entitled parties. Assigning the rights is part of an *authorization* process by defining roles or policies. Provenance tracking denotes any mechanism to document the source of a data object and any kind of transformation or modification. Identity management faces the challenge of supplying unambiguous identifiers and mechanisms for third parties to verify an identity claim (authentication). As IIoT also interacts with the physical world, functional *safety*, as a system's ability to prevent physical damage, is an important requirement. Any productive IIoT system must have security concerns at its core. Therefore, security-related concerns are judged more and more as non-optional but fundamental for any ecosystem.

Business-related concerns outline, for instance, requirements around *data economics* and *governance* and value-adding features. As the IIoT reference frameworks face significant influence from the network and communication communities, technical concerns generally surpass economic considerations in quantity and level of detail. Nevertheless, reliable business models facing actual needs and future economic opportunities are essential building blocks for any reference architecture.

This list gives only a rough impression. In total, 186 concerns have been defined and grouped into a cycle-free, typed hierarchy as an integral axis of the knowledge graph. The complete set of concerns with explanations, their sources, and further links to additional web resources is provided as instances of the class *Concern* (cf. Figure 6.6).

6.3.4 Selecting the Reference Architectures

The selection of noteworthy reference frameworks needs to be based on the interest of the target groups from the industry and research. One way to quantify this interest is based on the number of searches done via internet search engines, mainly Google. The following assumptions are made for the process of determining this interest: (a) Only broadly known reference frameworks are able to create the necessary impact, and (b) the interest in the respective framework is more accurately indicated by the name of the publishing organization (for instance 'Industrial Internet Consortium') rather than the name of the reference architecture ('Industrial Internet Reference Architecture'). In particular, the second assumption is based on a previously conducted literature search and results in higher precision. 'Plattform Industrie 4.0', 'Industrial Internet Consortium', 'Industrial Data Space' and 'Industrial Value Chain Initiative' were the seed searches. The other approaches were iteratively added during the literature research and returned by the named search engines.

Table 6.4 gives an estimation of the impact on the research community and the overall interest. Google Scholar statistics give a rough impression of the influence based on mentions in research publications. The collected dimensions are reflecting the mentioning of the respective terms in the whole indexed

Table 6.4: Ranking of selected IIoT reference frameworks ranked by Google Scholar (research relevance) and Google Trends (public interest) appearance.

All Time Framework Name	All Time References	Rank by Ref.	References since 2018	Rank by Ref. since 2018	Main Publication	Citations	Rank by Cit.	Research Rank	Google Trends Rank	Overall Rank
OpenFog	558,000	(5)	17,600	(5)	[275]	23	(6)	3	2	1
IIC	181,000	(7)	12,300	(7)	[4]	74	(2)	1	7	2
Plattform Industrie 4.0	17,000	(10)	4340	(9)	[74]	57	(3)	5	6	3
IDS	4,480,000	(2)	101,000	(1)	[35]	14	(8)	2	10	4
x-Road	3340	(14)	288	(15)	[276]	46	(4)	10	3	5
FIWARE	2740	(15)	566	(13)	[277]	18	(7)	12	1	5
Industrial Value Chain	4,590,000	(1)	22,100	(3)	[262]	3	(12)	8	8	7
Industrie du Futur	125,000	(8)	5130	(8)	[278]	0	(13)	13	4	8
IoT-Architecture	6680	(11)	2540	(11)	[242]	239	(1)	4	13	8
Open Connect. Found.	1,330,000	(4)	19,200	(4)	[279]	9	(9)	6	11	8
BDVA	4,150,000	(3)	32,300	(2)	[266]	4	(11)	7	12	11
Edgecross	5950	(12)	330	(14)	[261]	0	(13)	15	5	12
Arrowhead Framework	37,900	(9)	3490	(10)	[174]	36	(5)	9	13	13
Piano Industria 4.0	5570	(13)	691	(12)	[280]	7	(10)	14	9	14
Alliance of Ind. Internet	291,000	(6)	16,500	(6)	[77]	0	(13)	11	13	15

literature and in the papers published until April 2018. In addition, the total amount of citations of the main reference architecture publication have been counted wherever possible. The aggregated rank (Table 6.4) equally reflects the academic relevance through references and citations, on the one hand, and the popularity in the Google Trends list on the other. One can easily recognize that the more industry-driven architectures have significantly fewer citations but tend to be more popular in terms of web searches.

The combination of the outlined measures indicates the impact and importance of the respective frameworks. The noted overall rank, therefore, supplies a rough impression of the significance of a framework in relation to the other frameworks. The interested user can better estimate the value of the provided specifications in terms of its innovative potential but also the possible community impact. This is especially important in the IIoT domain, where only the early detection of new best practices guarantees future-proven and, therefore, sustainable solutions.

Reference Architecture Alignment Process

The most popular and impacting IIoT reference architectures are aligned regarding their overlaps and unique features (cf. Figure 6.5). A qualitative comparison complements the analysis with configurable visualizations, extending the work presented in Chapter 6.3. System architects can use the visualizations to find according proposals related to their problems. Creators of architectures can find corresponding specifications from different initiatives. Decision-makers can use the knowledge graph to get an overview of covered domains.

The different reference *frameworks* are compared through their related concerns of their *classifications* (cf. Figures 6.17–6.19) and the relations (alignments) between them. Therefore, a comprehensive set of relevant and suited concerns needs to be formulated. The list of concerns, and their relations to the frameworks, are extracted manually, searching for explicitly stated topics and challenges in the respective publications. The core set of concerns is additionally based on the Unified Requirements list collected by the IoT-A project [242], extended by the analysis of the various architecture descriptions and other publications of the IIoT domain in general.

While some concerns are essential for most architecture descriptions, like describing interoperability between Assets, Digital Twins, and applications, others are only relevant in a specific context, for example, the ability to have clear data provenance across a system architecture or only specified by a single source, for example, the ability to control the transmitting power in wireless networks. A mentioned concern is added to the list if either more than one architecture description specifies its relevance or if it is proposed as a crucial concern by at least one prominent publication. A complete spreadsheet with all concerns, explanations, and references is publicly available[19].

Each selected IIoT reference framework was analyzed according to its coverage of each concern. Their mentioned categories are stored as entities of the classification class and annotated with additional meta information. For instance, short textual descriptions, links to their official resources, and - if existing - links pointing to DBpedia are added. DBpedia is the open knowledge graph representing the structured information in Wikipedia and plays a central role for the Linked Open Data Cloud. Therefore, additional look-ups and the discovery for further information for all entities are directly provided (cf. Figure 6.6).

Figure 6.15: Combined requirements and concerns for IIoT architectures[19].

A classification or framework is linked to a concern if there is at least some coverage of the related aspects of the respective concern (lowest common denominator approach). The alignment enables the traversal of links in the graph. Starting at one view, related specifications and definitions can be discovered by following the links to other frameworks with related foci. Classifications with nearly no alignment relations may present a unique perspective on the IIoT, whereas highly connected nodes most probably address fundamental and widely regarded topics.

The explained relations between the frameworks are represented as formal links between the respective classification nodes in the knowledge graph. Classifications with high degrees of incoming or outgoing links indicate a broader coverage, for instance, the IIRA Information domain (center) and the RAMI Functional Layer (bottom). More focused and target-oriented classifications, like the IVI Activity View (bottom left) are reflected by fewer connections. While the interested reader will find more overview information in the higher connected sections, the less-connected entities usually provide more in-depth discussions and guidelines.

[19] https://github.com/sebbader/architecturecomparison/blob/master/IoT-A%20Requirements.xlsx

The IDS as a Use Case Model

The IDS focuses on secure and trustworthy data exchange patterns in the manufacturing domain. The IDS Reference Architecture Model [35] (IDS-RAM) consists of five layers to establish interoperability and three cross-cutting perspectives for reaching its main target, namely to ensure end-to-end data sovereignty of the data owner. The syntactic interoperability is accomplished by the IDS Connectors with their standardized interfaces and exchange protocols.

The Viewpoints of the IIRA map only to a limited extent to the IDS Layer model. The scope of the IDS leads to a stronger focus on configuration, modeling, and integration aspects mainly regarded from a system integration point of view. The IIRA scope includes more stakeholders resulting in several 'Viewpoints'. In general, the aspects of IDS Reference Architecture are mentioned in the IIRA's Functional Viewpoint. The layers and perspectives of the IDS-RAM can – to some extent – be mapped to the IIRA domains of the mentioned Functional Viewpoint.

RAMI4.0 outlines a comprehensive view of manufacturing-related aspects in an IoT landscape. Its focus is on the Asset and its digital representation, the Administration Shell, as its first-class citizen. In contrast to that, IDS focuses on the data and data exchange while RAMI4.0 mainly specifies the integration of shop floor and office-related components. Therefore, Each architecture targets different challenges of industrial IoT. This observation is also presented in Figure 6.16, where the purple-colored concerns depict the issues of physical objects and the description of data entities. Concerns of security, authentication but also deployment of digital components (mainly found in the blue-colored intersection) are less covered. One can also notice that interoperability-related concerns (green) are relevant for the RAMI4.0, the IIRA, and the IDS Reference Architecture Model. In addition, RAMI4.0 has a stronger view on physical objects (part of purple) and IDS-RAM on data and usage control (part of brown).

The obvious similarities are the Business and Functional Layers in both reference models. Disregarding the same terminology, IDS and RAMI interpret the meaning differently. In the context of RAMI4.0, the Business Layer contains organizational and economic aspects as monetary conditions and legal regulations. On the other hand, IDS-RAM considers business-related aspects, as for example, the role of a participant in the data exchange network.

The Functional Layer in terms of the IDS comprises its core functional requirements and considerations, mainly data sovereignty, trustworthy interactions, and supported data exchange. In contrast, RAMI defines its Functional Layer as a view regarding and describing abstract capabilities and *functions* of Assets and data workflows.

Network and integration-related topics as defined by the RAMI Layers Communication and Integration are combined in the IDS System Layer. The different focus of the IDS considers a suitable communication level through internet standards as a precondition. RAMI, targeting also non-IP-capable devices, defines fine-grained views on connection patterns but also on representing and identifying real-world objects as digital entities. The Asset Layer further strengthens the viewpoint by defining and providing the physical entity. In contrast, a physical object is only regarded by its digital representation in the context of the IDS.

The major differences between both models is the emphasis on the cross-cutting perspectives (IDS) and lifecycle and hierarchy dimensions (RAMI). In order to enable cross-organizational data exchange, security, certification, and governance are the key consideration at all levels for the IDS. RAMI's focus on the integration of physical objects and manufacturing plants takes these aspects for granted. The IDS instead lacks specific lifecycle and organizational structures.

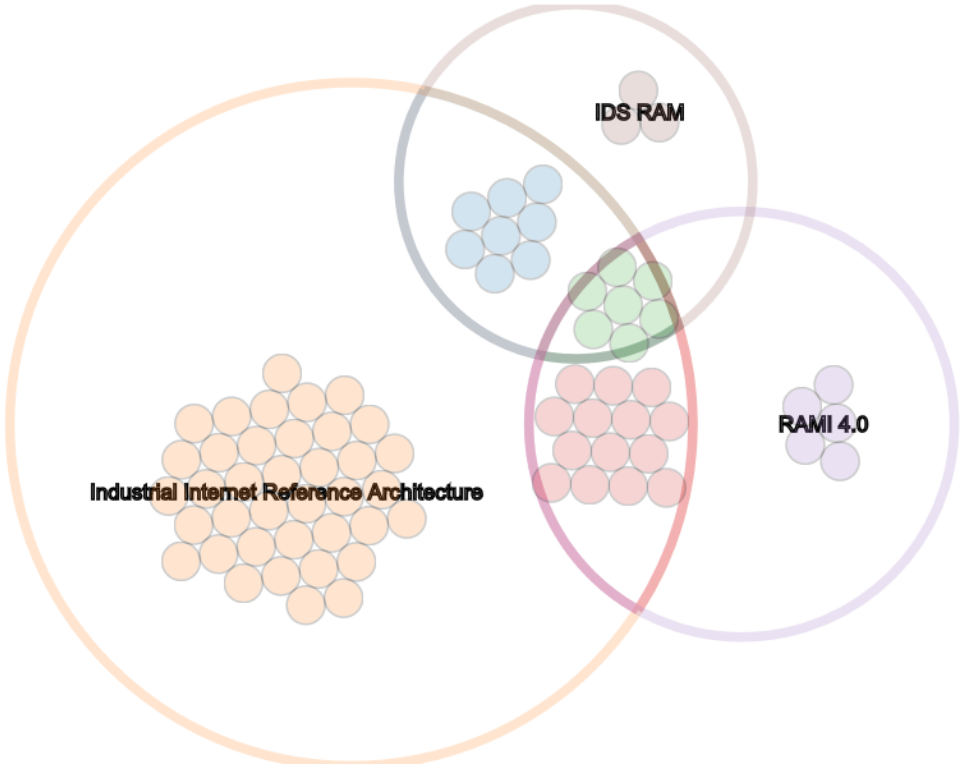


Figure 6.16: Comparison of RAMI4.0, IDS-RAM, and IIRA by considered concerns.

6.3.5 Limitations

The modeling of relations between reference frameworks, concerns, technologies, and standards in the knowledge graph provides a flexible format for the modeling of the various relevant dimensions and dependencies. Nevertheless, the selected model has several disadvantages. First of all, the presented graph is by its nature incomplete. The number of publications in the IIoT domain, considering its velocity and variety, does not allow a complete coverage. Therefore, the non-existence of an entity or a link between two entities must not be interpreted as though the two nodes are unrelated. For instance, not having a *sto:aligendWith* relation between classification A and B does not imply that A and B are not aligned. Nevertheless, the application of the open-world assumption inherent to ontology-based reasoning allows the application of machine-supported knowledge graph completion. The number of explicitly stated facts can thereby automatically increased from more than 15,000 to up to 24,000.

Second, the detailed characteristics of a single relation cannot be presented, which means that there is either a relation or not instead of having, for instance, indications for stronger or weaker links. For example, a reference architecture such as the IDS views usage control as its main scope, whereas the IIRA merely mentions it. As a matter of readability and simplicity, and restricted by the expressiveness of the RDF model, the gradations of this relation and other potentially relevant features are not part of the graph. The knowledge graph model solely allows qualitative statements of relations (relation exists or not). Consequently, a link between two entities states that a relationship between these entities exists but says nothing about the nature or degree of this relation. Applying logical reasoning, this restriction makes it hard to depict whether one statement ‘is stronger’ or ‘has more support’ than another or which degree

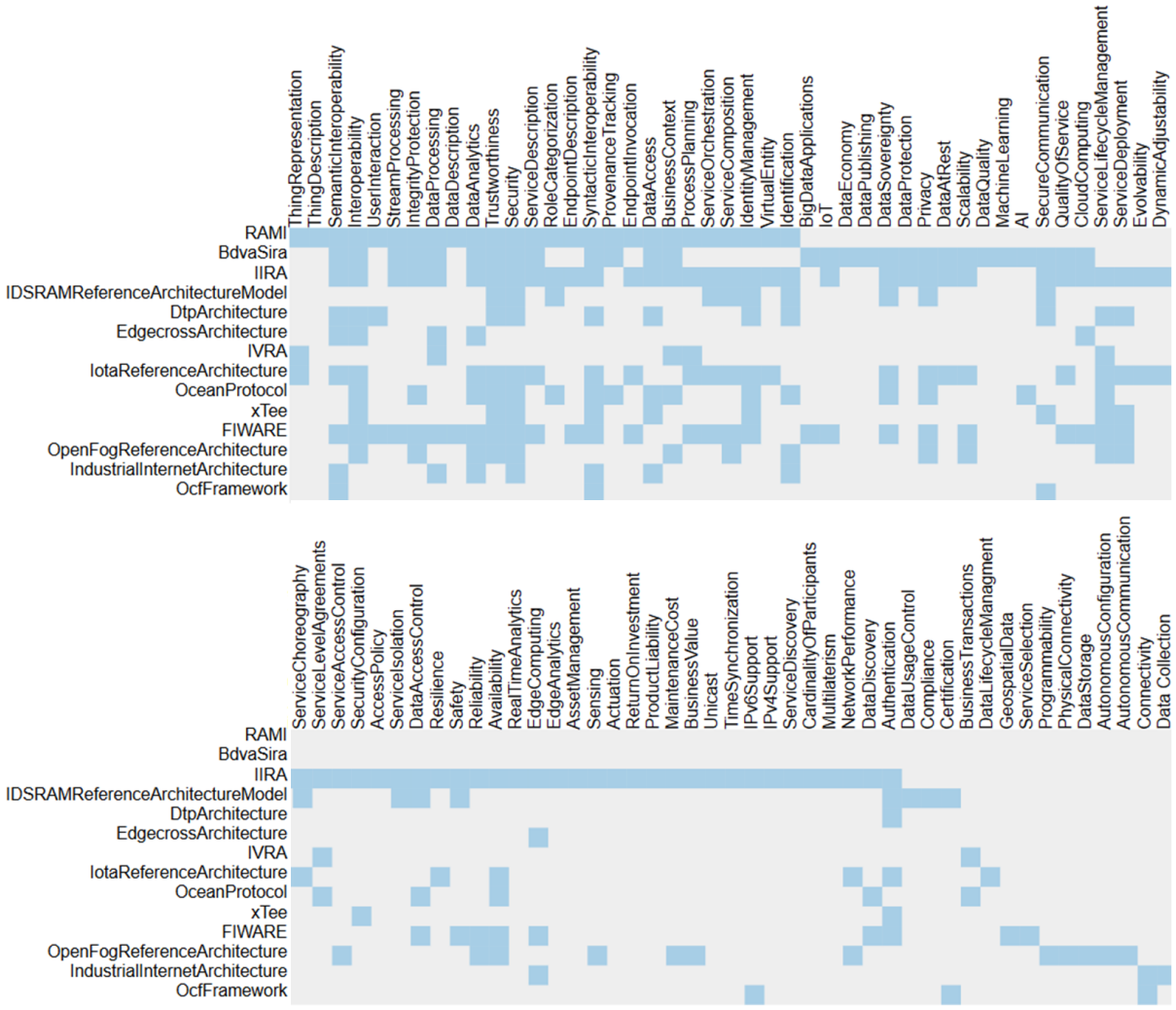


Figure 6.17: Concerns addressed by the frameworks as a co-occurrence matrix[19].

of uncertainty is included. Other approaches, in particular property graphs, have aligning capabilities but, in general, lack the web-based nature of the RDF graph.

Furthermore, the chosen graphical representations are limited to two-dimensional relations. While the knowledge graph itself is a highly interconnected and non-planar structure, the visualizations in the form of Venn diagrams and co-occurrence matrices limit the comparisons to a maximum of two dimensions. For instance, frameworks can be plotted in relation to concerns, their classifications, or their alignments to other frameworks but not in the same view. Extending the existing plotting options for more-dimensional presentations, including more kinds of attributes, will increase the amount of information presented to the user.

6.3.6 Findings and Best Practices

Two major trends can be identified throughout most of the considered reference architectures. First, nearly all promote cyber-security as one of the key IIoT requirements as the IIoT will disable the isolation of critical infrastructure and devices behind restrictive firewalls or even in separated networks. However, the scope is usually limited to overview descriptions while concrete specifications are missing. For instance, the Security Framework provided by the IIC [85] adopts the categories confidentiality, integrity, and availability for a more fine-grained definition of cyber-security. Still, the level of coverage is generally not further examined. Furthermore, additional and more detailed security-affecting concerns have been identified. Amongst the previously mentioned, these are certification processes, provenance tracking, network security, and process isolation, which have to be regarded at all levels of the architecture. Respective measures need to be outlined, starting with embedded devices, over edge/fog/cloud computing, to huge distributed systems. While the reference architectures claim to guide the interested reader, implementation details about using which method or technology at which position of the architecture are generally missing. For instance, security concerns and requirements of IIoT systems regarding a network-wide detection of intrusion attempts as outlined by Sadeghi, Wachsman, and Waidner [281]. The unifying approach behind the presented architecture should further promote such best practices in order to reach common patterns and broader adoption.

Especially regarding security concerns, a considerable demand and effort by the IIoT community can be noted. This is reflected in the remarkable amount of security- and trust-related concerns referenced by nearly all frameworks. In addition, the congruent usage of the same terms indicates an already reached common understanding. However, this is contradicted by the low level of detail provided by the considered frameworks. While certain practices have gained wide acceptance in the web-based communities and also new, innovative approaches – like flexible intrusion detection systems for IIoT networks [225] or data usage control in dynamic settings [282] – are available, the majority of IIoT frameworks does not explain which specific recommendations are appropriate in the industrial IoT domain.

The second relevant trend is the repeated reference to a set of core technologies, which have the potential to shape the IIoT. While at the beginning of the IIoT discussions, mainly RFID and real-time analytics dominated (for instance in [77, 222, 242]), later 5G and AI are more prominently presented [84, 223, 266] as the major developments for IIoT. While the mentioning of these technologies is certainly justified, their actual disruptive impact can only be estimated. Still, the seamless substitution of technology terms indicates an unclear vision of the real requirements and objectives to be met. In particular, it seems that a generally shared understanding of the medium layers of the architectures has been established, while the resulting business workflows and the, therefore, necessary applications –

[19] http://i40.semantic-interoperability.org/sto-visualization/views/matrix_fw_concern.html

being artificial intelligence, big data analytics, or even Blockchains – are taken for granted. Moreover, the upper-level layers related to business and usage aspects are less elaborate in order to enable real end-to-end IIoT application networks.

In this direction are also the frequently mentioned Service Level Agreements (SLA) and Quality of Service (QoS) measurements. Cao and Chen [283] argue that a Quality of Experience (QoE) approach is, however, better suited. Ad-hoc vehicular clouds can provide exemplary scenarios for mobile edge nodes with on-the-fly connections and outline routing algorithms with high-velocity [283]. Quality of Experience has been discussed in various IoT scenarios, mainly for smart cities by References [283–285]. New implications regarding the upcoming IoT technologies in smart cities are also reviewed by Lemayian and Al-Turjman [286]. The work outlines the consequences of the emerging IoT and discusses the requirements and implications for the truly connected city.

In addition to the already provided visualized relations between the reference frameworks, several outstanding commonalities can be identified. On the technical implementation level, most reference architectures promote the usage of container-based – mainly Docker – deployment methods. The advantages of having unified modules in the regarded distributed and heterogeneous environments have reached a common agreement. Moreover, the provisioning of APIs following the REST paradigm can be seen as a default baseline for endpoint interaction. While a significant ratio of use cases requires different interaction methods, for instance in event- and streaming-based interactions, RESTful APIs are a central building block of nearly every proposed reference architecture. In particular for automated deployment and configuration tasks of middleware applications and the fast connectivity of external components.

Another example for a commonly shared agreement in the used solutions is supported by the repetitive mentioning of certain technologies. In particular, OAuth for authorization, HTTP for basic data exchange, and MQTT for publish/subscribe applications present widespread building blocks. For the industrial internet, OPC-UA becomes one of the de facto standards for machine-to-machine communication.

Furthermore, certain areas of interest can be identified. While the IIoT-related architectures extensively describe technical and syntactical integration techniques, the higher-level processes and especially the consuming applications are only roughly described (cf. Figure 6.17). Since naturally the focus of IIoT is on the handling of the Asset, the full application stack is able to exploit the potential. Commonly referred analytical services working with artificial intelligence are usually not further enriched with best practices and remain mostly a black box.

One can also notice a difference in the prioritization between the different domains and communities. While the rather IIoT-focused approaches present in detail data security mechanisms, the data exchange approaches provide more details on data privacy and usage control. Aspects concerning connectivity are widely covered and are discussed by nearly every reference architecture. More visualizations supporting this finding are also available on the website. However, the blind spots, for instance the unambiguous description of Assets, devices, and endpoints, are only slightly discussed. While not one single framework needs to cover all of this, references to certain well-defined specifications between the frameworks can express a better overview. Stronger and explicitly stated differentiation to other works will speed up the progress in the field and prevent inefficiencies.

Furthermore, the focus on connectivity and security does not reflect the overall potential of the IIoT. While the industrial community currently examines the interlinking and communication between IoT devices, their implications are only roughly realized. Self-controlled devices, in combination with flexible edge and cloud computing architectures [227, 286], need to adjust and act autonomously in order to achieve the necessary scalability and robustness. However, the main target is still the achievement of interoperability between heterogeneous systems rather than a real new design of future IIoT networks. The already existing complexity in terms of device variety, velocity, and volume will further increase with the ongoing deployment of more and more connected IIoT systems. Therefore, simply following

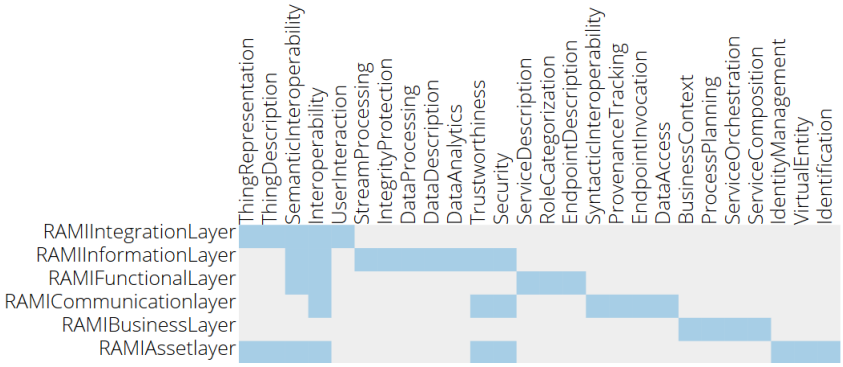


Figure 6.18: RAMI4.0 Layers with related concerns.

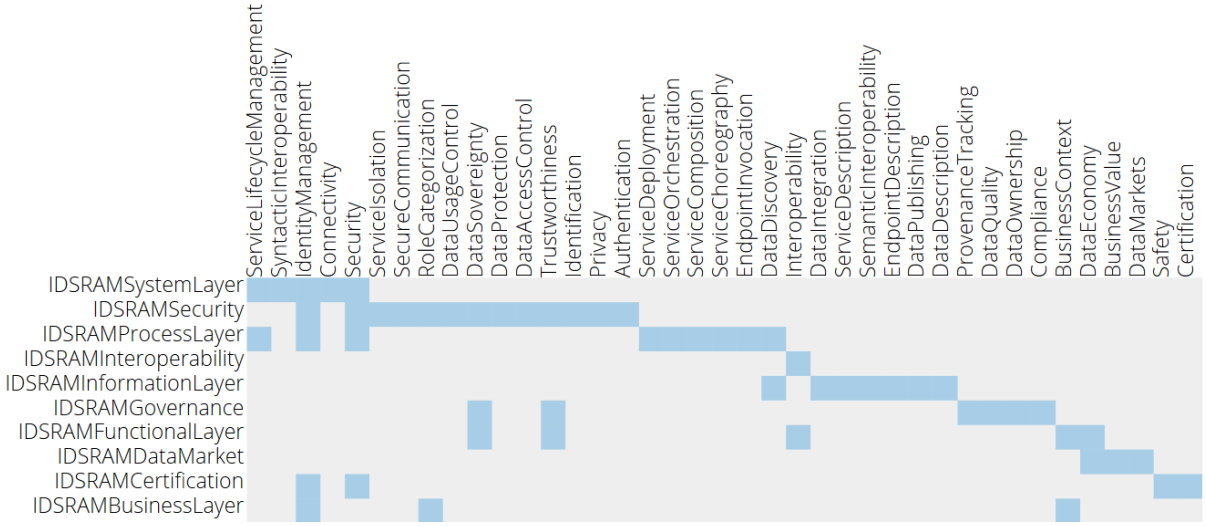


Figure 6.19: IDS Classifications with related concerns.

existing coordination and orchestration patterns for the quickly growing number of digital components will not be sufficient.

An overview of the different scopes and priorities is given in Figure 6.17. The co-occurrence matrix shows that the more generic concerns, for instance interoperability, security, or data analytics, are covered by many frameworks while more specific challenges, like data access and the certification of IIoT systems. Figures 6.18–6.20 show the same content on the level of classifications. This allows the interested reader to quickly gain insights into the available material and to effectively discover and select the most suited guidelines. All presented views are available on the website and can be configured interactively. The provided graphs are always created at request time and illustrate the latest state of the knowledge graph and its content.

A major drawback of nearly all outlined specifications is the lack of negative examples in terms of specifications, in which scenarios and settings are disregarded. Valuable knowledge can be provided by outlining why certain commonly used paradigms or techniques are perceived as bad practices and help to gain better insights. Making such intangible assumptions explicit helps the IIoT implementer to better understand the respective scope.

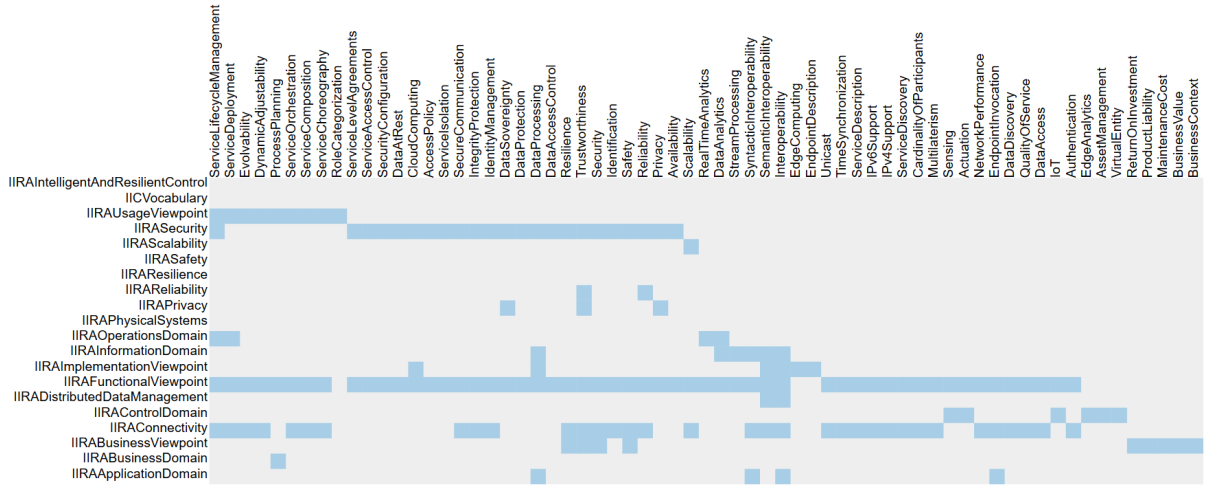


Figure 6.20: Concerns as referenced by the Industrial Internet Reference Architecture (IIRA)

6.3.7 Lessons Learned and Outlook

Despite the multitude of works on the topic, there is still no widely adopted scheme for creating and illustrating IIoT reference frameworks. Although the referenced norms deliver guidelines, the heterogeneity of published architectures both in content but also format is significant. The different presentation patterns, arrangements, and terminology hide similar requirements, assumptions, and considerations and therefore make it very difficult to grasp a common ground. This work explicitly does not intend to provide yet another (meta-) IIoT framework, but instead, it aims to increase the applicability of the existing ones.

The currently missing transparency of the various frameworks and standardization activities leads to several severe drawbacks. The interested reader is hampered in the attempt to understand the respective contributions as the heterogeneous terminology, structures, and conventions add unnecessary complexity. Consequently, the introduction of sophisticated techniques and patterns is slowed down. On the other hand, the developer of reference architectures themselves can benefit from a structured and coherent overview of already existing approaches to avoid repeating – and therefore ineffective – efforts.

The explained contributions of this chapter outline a common ground in the form of a hierarchy of IIoT concerns in order to describe, index, and interlink the various frameworks. As a matter of fact, the list is and will always be incomplete and not being able to cover all use cases. Nevertheless, explicitly mentioning a regarded subset of concerns at all architecture sections significantly increases transparency and comparability and therefore creates the foundation for achieving further progress. The interested reader is invited to use the interactive web views for his investigation. The provided knowledge graph gives an overview of the current scope of the most popular proposals, enabling the various stakeholders to discover and select reference points more efficiently. Automated reasoning further extends the number of manually inserted facts by nearly 50%.

6.3.8 Summary: A new Way to Analyse the IIoT

The contributions of this chapter, the *Standardization Landscape* (CO5), in combination with the *alignment model for IIoT Reference Architectures* (CO6), presents several views and applications on top of an integrated knowledge graph, the included categories as well as the related technical standards. The formalized semantics of the knowledge graph and further extensions of its model support the

representation of the interlinked information of the domain. The different graphical presentation methods enable intuitive examinations of the encoded facts without insights into the underlying technology stack.

The outline usage scenarios of the developed views explain the intended usage to reduce one of the most crucial obstacles hampering further dissemination of IIoT concepts: The complicated and cumbersome search for information and establishing of a mental structure of the landscape. The graphical approach based on a web application supports the necessary scalability in order to reach the required target group.

6.4 Summary

This section presented the integrated *Standardization Landscape* (CO5) for technical recommendations and normative specifications and its extension, the examination of *IIoT Reference Architectures* (CO6). The publicly available datasets in RDF are annotated according to the FAIR principles and interlinked with the LOD Cloud. In addition, the dataset is organized as a modularized knowledge graph, which sections can easily be extended in further activities.

The additionally provided visualization tools are also open source and can be used through every common web browser. They present a valuable tool for understanding the current state of the IIoT standardization efforts and are also a flexible instrument to examine those developments and their interrelations. The thereby carried out analysis revealed several blind spots, which require further attention from the research community.

The first ontology of IIoT Concerns proposes an unambiguous vocabulary to refer to the requirements for the intended Assets, composed systems, and complete environments. Their interrelated character and the supplied hierarchy of the concepts further allow to derive coverage analysis but also to formulate requirement lists. This is a first approach to tackle the challenge of *Non-transparent Design Decisions* (CH4).

The described contributions are also valuable for the design, the later description, and even the reorganization of IIoT settings. The comprehensive visualization of the landscape contains the opportunities to quickly create suitable analysis views by choosing from a set of supplied templates and filter options. *Dynamic changes in the IIoT Environment* (CH5) can therefore be resolved with fast and adequate interaction features. This way, updates and unplanned modifications can be conducted after only very short look-ups and information searches.

In particular, the non-functional aspects of security and *Data Sovereignty* (CH6) require a comprehensive view beyond individual systems and towards the complete ecosystem. While security draws already a significant amount of attention in the IIoT community and is regarded by most models and reference frameworks, this is slightly different for the latter. The earlier mentioned ontology of Concerns puts Data Sovereignty already in the context of other IIoT requirements and thereby supports its understanding and how it affects the other characteristics of an ecosystem. The I40KG outlines the various recommendations on how to reach Data Sovereignty from the selected reference frameworks and directly connects them to the normative standard documents. Thereby an effective access and use case-specific access to the available documents is possible.

Putting all these aspects together, the contributions of this section provide the ability to easily and quickly discover and compare the combined landscape of IIoT specifications and help the different stakeholders to fulfill their different information needs. Therefore, It presents a suitable answer to RQ3, how the current standardization activities can be composed to increase the chances of finding the appropriate specifications and to support the compliance of the implemented solutions in the huge and confusing IIoT domain.

CHAPTER 7

Conclusion

This thesis investigates six current challenges in the digitization of the manufacturing industry. It analyses how the IIoT can be further developed from an abstract vision to concrete, value-adding environments by using semantically described Digital Twins. This combination of the Assets with their virtual representations encapsulates the inherent complexity and significantly simplifies the integration and modification of IIoT environments.

As shown in Figure 7.1, the characteristics of IIoT Digital Twins have been extended towards standardized and implementable representations in unambiguously described ecosystems. Using these representations, a resource-oriented interaction model promotes the use of atomic operation calls that can easily be combined to complex workflows and control functions. The state-less nature of each operation ensures the necessary flexibility to cope with new requirements and modifications in the network and to integrate new Assets and services on the fly.

The overall architectures of IIoT environments and the current state of the standardization process have also been examined, formally processed, and prepared for the community in the form of open resources

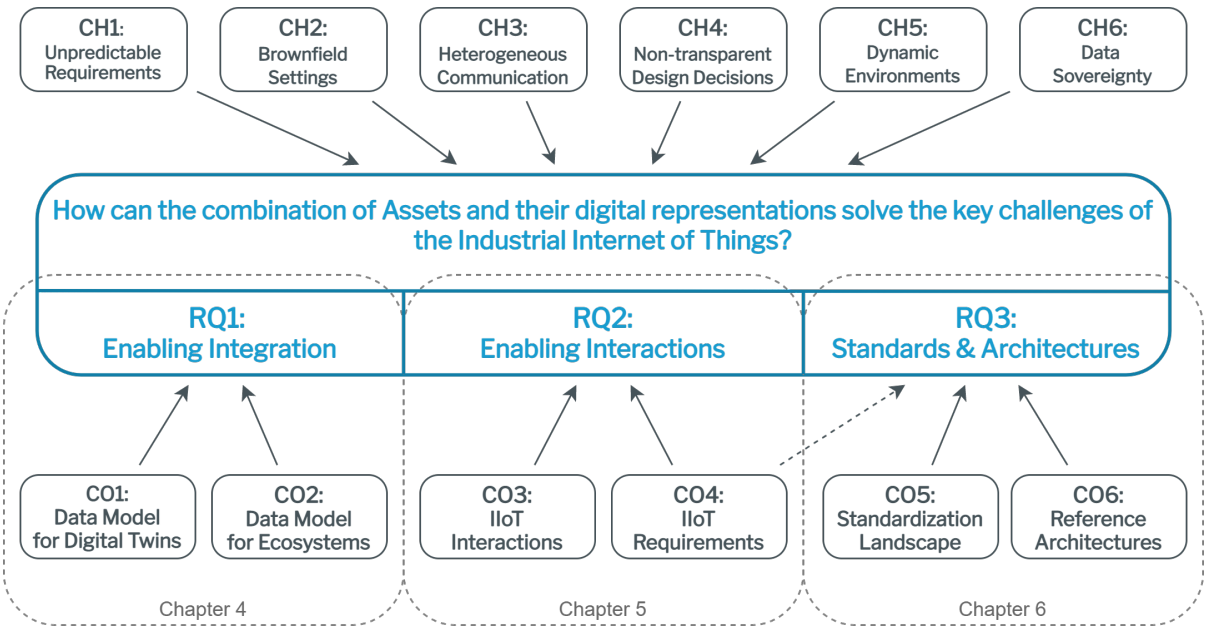


Figure 7.1: Repetition of the regarded challenges, contributions, and research questions as shown in Fig. 1.5.

and publicly available analytic tools. Standardization processes have further also been targeted for the dissemination of the developed contributions of this thesis. The data models, interaction sequences, and the gained insights on the comprehensive environments have been incorporated in several research- and industry-oriented standardization bodies, in particular the International Data Spaces and the Plattform Industrie 4.0 but also already into the establishment of the GAIA-X activities.

This chapter briefly repeats the contributions of this thesis and explains how they answer the initial research questions. Section 7.1 presents how the data models for the IIoT Digital Twins and their ecosystem, CO1 and CO2, enable the seamless integration in IIoT settings. Section 7.2 recapitulates the interaction model on top of these data models and outlines how CO3 and CO4 ensure a suitable data and control flow in dynamic IIoT networks. Section 7.3 further extends the previous propositions with the examinations of the current standardization state of the IIoT and how it fosters the interoperability between independent IIoT systems through the support of the contributions CO5 and CO6. All these aspects are brought together in Section 7.4, which finally outlines how the concept of Semantic Digital Twins can solve the six identified key challenges of the IIoT.

7.1 Standardized Data Models for IIoT Ecosystems and Digital Twins

As explained in Chapter 4, the Asset's attributes, characteristics, and capabilities need to be presented in unambiguous, self-descriptive, and formalized methods. An overwhelming amount of proposals for such data models has been published. Still, they are not compliant with each other and therefore even increase the interoperability challenges. This is the focus of the first research question:

RQ1: Enabling Integration – How should IIoT Assets be represented in order to enable their seamless integration in IIoT settings?

The dynamic nature of IIoT applications makes it impossible to create the precisely fitting models for later usages of Assets already at their creation time. The only way to cope with inherent uncertainty is by decoupling the underlying characteristics and digital representations through a standardized, generic model. The Digital Twin concept is especially promising for IIoT use cases, as their considerations are significantly affected by the physical appearance of the Assets.

CO1: A Data Model for Digital Twins
The first contribution targets this observation and derives a generic and reusable data model for Digital Twins based on the Asset Administration Shell. Its enhancement with semantic technologies leads to the *Semantic Asset Administration Shell*, a self-descriptive representation of each Asset, its characteristics, and relations. The extension of the data model into RDF goes simultaneously with the additional introduction of semantic models in relevant industrial activities, for instance, the ECLASS catalog and the W3C Web of Things. The thereby reached standard information layer builds on a consistent technology stack, significantly reducing the transformation and integration needs.

The maintenance of CO1 is ensured by the activities and processes of the Plattform Industrie 4.0, an industry-driven standardization body. The close relationship with the demands and use cases of the affected companies further guarantees the model's adoption in commercial products. That way, the standardization needs of the industry are met, while at the same time, the combination with other semantic resources, for instance, the Linked Open Data Cloud, is enabled. While such resources have already been successfully integrated into many web-based applications, the manufacturing domain has mostly avoided their integration due to unclear reliability and liability concerns. In this context, the Semantic

Asset Administration Shell presents a significant step towards the further integration of the OT world with the IT and Cloud applications of the open Web.

CO2: A Data Model for IIoT Ecosystems
The second contribution of this thesis continues the model from CO1 with the specification of an IIoT infrastructure as an open, web-inspired environment. By means of the *International Data Spaces Information Model*, CO2 describes a vocabulary to express the components, messages, and intentions in a data-centric ecosystem. Relying on the same technologies and conventions as CO1, the information model is easily combined with the Semantic Asset Administration Shells to a comprehensive IIoT environment. In particular, the combination of both specifications contains the capabilities to provide IIoT Assets in unambiguous manners but also to state machine-readable restrictions on their usage. CO2, however, contains not only the data model itself but also the related aspects of the ecosystem, particularly the architecture components and their cooperation and the Usage Control specifications. In total, the contribution comprises a general approach to represent and exchange data in an IIoT environment, which can go beyond the borders of facilities, companies, and even whole industries.

Like CO1, the further maintenance and adoption of CO2 are ensured through a responsible support organization, the International Data Spaces Association. The information model is embedded in a holistic approach to define data exchange between arbitrary participants. The combination of the Semantic Asset Administration Shell as the Digital Twin for Assets and the feasibility of the IDS as the surrounding environment has been shown in several prototypical implementations. In particular, the Usage Control Language - an integral part of CO2 - serves as the bridge between the Asset-oriented view from the Semantic Asset Administration Shells and the sovereignty ensuring capabilities of the IDS.

Both contributions together, therefore, present one answer to RQ1. The virtual representations of IIoT Assets, their Digital Twins, need to be modeled according to standardized, machine-readable, and extensible definitions. CO1 and CO2 introduce these features, and in addition, provide the capabilities to merge the IIoT with the IT applications located in the world wide web.

7.2 Modular Communication Patterns for IIoT Digital Twins

After determining how to model the Assets of IIoT environments for RQ1, Chapter 5 examines how they exchange data and control commands in dynamically changing networks. As this needs to reflect the long operation periods of IIoT Assets, several update and modification phases will appear during their lifecycles. Therefore, it is nearly impossible to consider the later appearing requirements at the design time of the Assets, allowing only the usage of generic and adaptable interfaces. The second research question consequently examines suitable communication patterns and how a generic paradigm for fast and simple modifications in the data exchange processes can be achieved:

RQ2: Enabling Interaction – How do IIoT Assets and their Digital Twins interact with each other in dynamic and unpredictable environments?

Answering this question requires two different aspects, a scalable and flexible communication scheme that provides the building blocks for any possibly needed operation workflow and a deep understanding of the appearing requirements and their relations. The former is reached through the third contribution and the latter through a detailed taxonomy of IIoT concerns in the fourth contribution.

CO3: An Interaction Paradigm for Digital Twins
The third contribution of this thesis provides the combination of the read/write operations on Linked Data with the specific demands of the IIoT. Using the Semantic Asset Asset Administration Shell concepts as

the IIoT Digital Twin model and the expressivity of the IDS for usage restrictions, the so-called SOLIOT framework incorporates RESTful interactions with semantic data descriptions and the unambiguous statement of access and usage policies. Unlike similar approaches in the literature, this contribution is not limited to the HTTP protocol but brings the operation semantics of the approach to widely used IIoT protocols, particularly CoAP and MQTT.

CO3 contributes to meaningful activities to combine the strengths and unique capabilities from previously separated domains into joint approaches. Proven techniques from the shop floors, such as the reliability and safety assurance from OT applications, are merged with the scalability and self-descriptive features of web services. This comprehensive view on digital environments is reflected in SOLIOT where, for instance, the meaning of an operation can be expressed equivalently in any used protocol and interaction scheme.

CO4: A Framework of IIoT Requirements
The design of IIoT solutions affects many different perspectives and targets. Security, interoperability, or reliability are only a few examples. Still, none of the currently dominant IIoT standardization bodies is able to deliver a sound and comprehensive overview of topics of stakeholder concerns. Therefore, it is the task of the individual developer, system architect, or API designer to predict the requirements his product will face in the future. The fourth contribution targets this gap by extracting a sound set of topics in most IIoT scenarios.

CO4 cannot be the final requirement list for a specific IIoT use case. It serves as an entry point and a first structure to align discussions and to sort design decisions. Like the procedure conducted for CO3, IIoT solution designs can be evaluated according to the requirement framework. The implementers can prioritize the single concerns according to their needs but also actively decide to disregard some. Nevertheless, the active consideration already increases the maturity of the final result significantly.

This way, CO3 and CO4 present this thesis' proposition for RQ2. The extensive model of relevant IIoT concerns has already impacted the approach for the SOLIOT framework and thereby ensured the sufficient coverage of the important IIoT requirements. The partitioning of interactions into small, resource-centric requests increases the expected number of unique interactions and, in general, also the network traffic. However, its generic nature and the thereby reached scalability and flexibility justify this additional effort as soon as anything needs to be changed at the Asset itself or in its surrounding IIoT environment. Therefore, the provided interaction pattern of CO3 building on the requirement model of CO4 is the proposed solution for the second research question.

7.3 Standardization and Reference Architectures for the IIoT

The third part of this thesis examines how the different standardization activities can be aligned and sufficiently supplied for the IIoT stakeholders. Technical standards are essential for this industry, as nearly all aspects of modern manufacturing processes are regulated through technical specifications. While other digitization domains, for instance, web development or machine learning, are mainly driven by best practices or conventions, the manufacturing industry's safety and quality assurance requirements affect every aspect from engineering, operation, and maintenance to the final disposal of Assets.

Standards from well-known bodies like ISO, IEC, or DIN and reference models from industry consortia are the backbone of the normative landscape. The recent popularity of IIoT topics and around the concept of Digital Twins has led to a huge variety of proposals, unrelated and sometimes even conflicting specifications, and uncoordinated guidelines. The third research question raises the critical issue, how the current variety can be tackled to ensure the adoption of the best-suited specification and how to manage the complexity created through the variety of standardization activities:

RQ3: Standards & Architectures – How can standardization activities ensure the compliance of independent IIoT integration efforts?

Solving this question requires several aspects. One is already presented by CO4, the outlined framework of IIoT requirements. This is further integrated into a landscape of technical documents and normative guidelines and examines the latest reference frameworks and architecture models.

CO5: Landscape of Standards and Norms
The core content of the fifth contribution, the integrated knowledge graph of the IIoT (Industry 4.0) standards, I40KG, serves as the basis for further activities on top of it. Its three main modules, the annotation of technical standards and the issuing bodies, the reference frameworks and their corresponding support organizations, and the taxonomy of IIoT concerns, contain the information processed and analyzed with the downstream applications.

One of these applications to examine and present the state of industrial standardization is the Landscape Visualization Service. This publicly available tool delivers the guidelines for the different stakeholders of IIoT scenarios to generate the views according to their needs and thereby discovering and structuring the available documents and resources. The curated dataset of CO5 is also interlinked with additional open and machine-readable data. Hence it can serve as an entry point for following look-ups and enriching information from other sources.

CO6: IIoT Reference Architecture Analysis
The sixth contribution focuses on examining and aligning the currently published reference architectures and frameworks for IIoT applications. Relying on CO5, it presents the influential propositions, structures their content, and extracts the recent state of the community understanding. This review is expanded to identify blind spots and the suggestion of a further research agenda for the standardization groups. Using the visualization techniques of CO5, this contribution provides visual interaction methods for IIoT users to make it available for stakeholders outside the research domain.

CO5 and CO6, together with the preparations through CO4, provide the necessary capabilities to investigate the state of the IIoT standardization processes. They enable the different stakeholders to quickly find the required resources and then continue their research at the appropriate places. The contributions thereby reduce the search overhead and support the implementation of the best-fitted specification. This increases the maturity and usability of IIoT systems. Combined, CO5 and CO6 present a suitable answer for the third research question. They align and structure the various activities, help to understand the different kinds of relevant resources, and put them into a comprehensive view of the complete landscape. As such, they increase the effectiveness of researchers, implementers, and members of standardization groups.

7.4 The Impact of Semantic Digital Twins for the Industrial Internet of Things

As shown in Figure 7.1, this thesis tackles the six identified challenges by answering the research questions RQ1, RQ2, and RQ3. Appropriate contributions address each question. The combination of all these activities addresses the common goal that is reflected by the underlying research gap:

How can the combination of Assets and their digital representations solve the key challenges of the Industrial Internet of Things?

This thesis promotes the concept of Digital Twins, extended with semantic technologies to encode the format and their meaning in machine-interpretable manners, as the unique entities for the digitization of the manufacturing domain. Their encapsulation of data provisioning and operation logic makes them suitable candidates to form the interoperability layer in decentralized IIoT environments. Unlike settings where all integration steps are performed in central cloud platforms, this approach keeps the control of the Assets and their data at their owners.

Therefore, these Semantic Digital Twins present the entities needed to form scalable and flexible networks that are easy to update, reconfigured, and modified whenever necessary. The clear separation of concerns and responsibilities simplifies their maintenance, enabling a seamless transfer of the Assets and their Digital Twins throughout their complete lifecycle and between the involved companies. Consequently, they constitute a sufficient solution for the overall research gap and can shape the next generation of the IIoT.

The standardization efforts affecting the contributions of this thesis are also influenced by the hereby gained insights and the conducted activities. This ensures the further adoption of the propositions and also indicates the adequate quality of their content. The insights have been published in peer-reviewed conferences and journal papers and implemented in several industry projects. All resources, final results as well as preliminary material are published according to open source standards and available for the community.

The research presented in this thesis adds to the state of the art. Nevertheless, many challenges still remain before the full opportunities of the IIoT can be realized. While the transformation of the manufacturing industry is never finished, several topics require further investigation. A selection is presented in the next chapter to outline extension points from this thesis and to suggest future research activities.

CHAPTER 8

Future Work

While this work outlines solutions to six critical IIoT challenges, further research is required to target also issues not contained by those six. This chapter outlines several obvious and not-so-obvious-areas, which require further investigation and are relevant for further adopting the IIoT vision. In particular, this thesis focuses on the technical and representational aspects of IIoT Digital Twins.

However, successful IIoT networks need to address the economic, social, legislative, and many more challenges. Consequently, the topics described in this work cover only one area, and many more research questions are still waiting for answers. This chapter intends to give inspirations for relevant challenges and tries to pose them for each presented domain.

At the core, an *Industrial* Internet of Things implementation is motivated by economic motives, either by reducing costs, decreasing the duration of processes, or increasing the value for customers. Participating in the IIoT requires initial investments, which - like any other business decision - need to deliver a return value as soon as possible. Comparable to the internet in the years between 2000 and 2010, business models are still unclear and need to be proven in large-scale applications. However, compared to the development of business models, the IIoT tends to be driven by economies of scale and thereby may lead to the dominance of a few central and powerful players.

Anticipating these potential future dependencies prevents companies today form upgrading their IT landscape to global IIoT networks. Hence, they will be left behind at some point and miss the competitive advantages and productivity increases contained by the new technologies. Solving the conflict between the risks - both for established business models and the protection of competitive advantages - and the new opportunities is therefore critical. The presented data protection mechanism in the form of a Usage Policy Language (Chapter 4.4) is only one building block and not sufficient for a consistent and convincing solution.

On the other hand, protection and security measures have a price, which may appear in direct investment costs for security software but as well come in the form of longer reaction times, higher maintenance and organizational efforts, or missed business opportunities. It is not possible to answer it in a general manner. Instead, answers have to be found for each individual use case. Still, general best practices and success stories are needed to better and faster balance the conflicting factors. This consideration leads to the following question, which needs to be answered:

What is a reasonable trade-off between protecting current competencies and the collaboration in an unrestricted IIoT?

In the following sections, additional topics are raised that impact the success of the IIoT. Each explanation is concluded with an explicit question to inspire further investigations. To solve them, the sole focus on the computer science field is not sufficient. Instead, interdisciplinary approaches are

necessary.

8.1 Ownership and Control of digital Information

Different from the Assets themselves, the ownership of Digital Twins is not yet grounded on a legal basis. Since the concept of ownership is not defined for digital information, other measures have to be applied to stay in charge of the virtual representation and the contained data. Possible concepts like copyright or privacy laws only provide inadequate protection as usually neither the creative aspect for the former nor the close relation to personal data for the latter is applicable for industrial data. This is also in the scope of the Gaia-X initiative that currently draws significant attention from media, academia and industry.

Nevertheless, licenses or specifically agreed contracts must be seen as workarounds at the moment.The combination of received information contains a significant part of the value proposition (for instance, collections of Digital Twin instances) and processing it to something different (for instance an AI-trained model). The ownership of the model is not dependent on the originating Digital Twins anymore. Nevertheless, the processed entity may still contain critical information of its sources. This significant distinction between physical goods and digital Assets leads to the following question:

Who can benefit from data in the IIoT, and how is the gained margin distributed between the value-adding parties?

8.2 Human and Machine Interaction

In addition to technical or process-oriented solutions, many challenges arising from the further digitization of the manufacturing domain can only be solved by society as a whole. The overall availability of data will change the way humans perceive and interact with industrial processes. Human-Machine-Interfaces (HMI) will become more relevant, and the effective integration of human intelligence with automated processing of huge data streams will become a critical success factor.

The contributions of this thesis appear “under the hood” of the developed applications. The regular user will not recognize that the underlying communication is based on Semantic Web or IIoT principles. More affected stakeholders, like system architects and API developers, will perceive the difference. Most importantly, the self-descriptive and Web-centric paradigm behind the presented contributions enables them to understand the faced situation using the most basic Web access tools - Web Browsers and even plain HTTP clients on the console.

Apart from the self-descriptive nature of IIoT Digital Twins, for instance, their geometry, activity logs, or safety requirements need more advanced presentation techniques. Virtual and Augmented Reality are potential technologies to bring digital representations into the physical, human-perceivable world. These aspects of Digital Twins are still underrepresented but critical for their success.

How can digital information be brought back into the physical layer?

8.3 Uncertainty and incomplete Information

The view of a human, an analytics application, or a Digital Twin is always restricted to a constrained part of the world. Consequently, the available information is uncertain, incomplete, might be true or not, and can be misleading. While humans can manage uncertainty successfully in their daily lives and professional environments, machines and automated data processing applications have significant difficulties operating on uncertain or incomplete data.

As one core application of IIoT Digital Twins is the automation of manufacturing facilities, and in the future also their independent orchestration, their correct behavior needs to be ensured at all time. This is especially relevant if safety-critical processes are involved with possibly high kinetic forces or electric energies - with potentially fatal consequences. The intended modifications in the control flow as envisioned in this thesis affect such processes, and the thereby introduced uncertainty imposes significant risks that need to be managed and controlled:

How can uncertain, unreliable, or incomplete information be integrated into IIoT Digital Twins?

8.4 Semantifying Industrial Catalogues

While the logical constructs of ontologies, particularly with OWL DL, enable the derivation of new axioms through inferencing techniques, these capabilities are rarely required. The broad amount of use cases can be met with controlled vocabularies or taxonomies. As a consequence, many of such vocabularies have been formed through industrial consortia. Prominent examples are, as named previously, the ECLASS catalog or the IEC Common Data Dictionary. Other vocabularies are, for instance, the iiRDS maintained by tekom[1], the VDI 2770 for documentation or the various information models proposed by the OPC Foundation.

Web-oriented vocabularies like schema.org do not play any significant role in the IIoT. The most obvious reason is the different scope of the schema.org terms that are mainly used to describe e-commerce offers and websites. Still, industry and IIoT-focused ontologies as collected by the LOV4IoT[140] catalog are hardly used. Most likely, this is because a commonly accepted, reliable, and trustworthy supporting organization is missing.

Consequently, this gap needs to be solved from both directions. There are noteworthy developments inside the established catalog providers, in particular ECLASS, to lift the rich vocabulary content and its scheme into RDF representations. Influenced by W3C activities, these approaches more and more move from a label and full-text search-based access method through content management platforms to machine-readable resolvement patterns. Soon, catalogs will provide their information and annotation through on the fly accessible APIs, as presented in Chapter 5.

These are not limited to read-only interactions but also contain the proposal and processing of new terms. While a human curation will still be necessary, more and more automated pipelines will appear to address the need for quick evolvements and updates efficiently. Static catalogs with static content in long-term release cycles will be replaced through adaptive self-explaining definitions with machine-readable status annotations.

This, however, creates a series of new challenges, which need to be investigated. The quality of the content and the absence of conflicts must be ensured, as well as the liability and accountability need to be clarified. Furthermore, the consuming applications need to become flexible so that they can process data based on live look-ups of their meaning. The Linked Data principles may guide the way. Nevertheless, significant research is still necessary. For instance, fallback mechanisms for unreliable network connections, potentially corrupted data definitions and business models for catalog providers and users are missing. Activities in these directions are already executed at the time of finalizing this thesis.

How can the established processes of industrial catalogs be extended to provide fast update mechanisms and on-the-fly look-ups while ensuring their quality?

[1] https://iirds.org/

8.5 Real-time Control and Orchestration of Production Facilities

This thesis focuses on the deeper integration of the shop floor (operation technology) and the office floor with the open communication flow of the Web (information technology). However, the seamless and on-the-fly integration of components at the shop floor can also benefit from the outlined technologies and approaches. For instance, the unambiguous and machine-readable presentation of control interfaces can simplify the orchestration of the different signal and control devices. The often discussed *batch size of one* requires the flexibility to reorganize the production systems dynamically and without predefined patterns.

The thereby introduced complex signal streaming and event management challenges have not been regarded in this thesis. Technologies like Semantic Stream Processing or Complex Event Processing make promising progress at the moment. Also, original shop floor technologies like the OPC UA stack evolve more and more into semantically defined integration approaches and federated networks across sites, organizations, and domains. One of the key differences to the focus of this thesis is the necessary event-driven perspective - while the contributions of this thesis mainly follow a state-based view.

Therefore, completely different challenges appear. Time and clock synchronization are still hard to achieve, especially in real-time environments distributed over cross-company supply chains. Furthermore, as not only the analysis of data streams but also the (remote) manipulation of machine behavior is enabled, safety and reliability become critical. Algorithms and remote interactions must not put the physical integrity of their environment and - even more important - the health of surrounding workers at risk. These viewpoints are explicitly not regarded in this thesis, but present critical obstacles on the path to the envisioned IIoT.

How can the presented data models and interaction patterns be extended to event-based approaches?

8.6 Upcoming Megatrends

Kagermann and Wahlster, two of the founding fathers of Industry 4.0, outline their vision on the future driving factors and challenges according to six megatrends [287]. According to them, *Industrial AI*, *Edge Computing*, *5G*, *Human-Robot-Collaboration*, *Autonomous Logistic Systems*, and *Trustworthy Data Infrastructures* are the ones with the most potential. For the context of this thesis, processing data at the Edge, Autonomous Systems and Trustworthy Data Infrastructures are the most relevant. Kagermann and Wahlster argue that standardization and certification processes are the backbones for these developments.

Still, there are significant efforts necessary to bring evaluation and certification processes for digital services, particularly Digital Twins, to an applicable level. Different from traditional safety-oriented tests, the quality assurance for software has not reached the required maturity. The number of capable evaluation bodies is still relatively limited in the market. Furthermore, the transition from technical standards to verifiable test catalogs presents significant difficulties that slow down the process and lead to delays in the time to market.

How is it possible to certify continuously modified Digital Twin applications and the standard conformance be tested?

Kagermann and Wahlster also mention the Gaia-X initiative as an example of Trustworthy Data Infrastructures. Such infrastructures incorporate cryptographic measures and certification processes to ensure reliability and prevent harmful behavior through technical measures. The overall trustworthiness of interactions is also dependent on organizational, social, and legal circumstances and can not wholly be covered by the software itself. The according awareness for these non-technical characteristics is still not

sufficiently developed. Nevertheless, the growing autonomy of IIoT processes demands for a suitable management of trust.

Consequently, the perceived trust in an application or business partner needs to be calculated somehow, and different levels of trust need to lead to different decisions. For instance, an operator of a production facility could ask for predictive maintenance analytics at several different data analysts. Dependent on their (digital) reputation and provided certification claims, different volumes and granularities of the sensor data are forwarded. At the moment, such scenarios are not possible, and the risk assessments depend on objective opinions.

How can the trustworthiness of applications, organizations, and whole IIoT ecosystems be measured, and which consequences does it have for autonomous business processes?

The raised topics for future activities give an impression of the additional research needs in the domain. The required effort is still remarkable, and huge efforts from the computer science domain and legal, social, engineering, and many other scientific domains are necessary. Nevertheless, the opportunities of the IIoT and the application of Digital Twins therein are without question, and the already gained progress in recent years promises exciting future developments.

Bibliography

[1] S. Sarma, D. L. Brock and K. Ashton, *The networked physical world*, Auto-ID Center White Paper MIT-AUTOID-WH-001 (2000) (cit. on pp. 1, 2, 22, 23).

[2] D. Infso, *Networked Enterprise & RFID INFSO G. 2 Micro & Nanosystems, in co-operation with the Working Group RFID of the ETP EPOSS, Internet of Things in 2020, Roadmap for the Future [R]*, Information Society and Media, Tech. Rep **10** (2008) (cit. on p. 1).

[3] Gartner, *Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016*, available online: https://www.gartner.com/newsroom/id/3598917 (accessed 19.08.2020), 2017 (cit. on p. 1).

[4] S.-W. Lin, B. Miller, J. Durand, R. Joshi, P. Didier, A. Chigani, R. Torenbeek, D. Duggal, R. Martin, G. Bleakley et al., *Industrial Internet Reference Architecture*, version 1.80, Industrial Internet Consortium (IIC), Tech. Rep (2017), https://www.iiconsortium.org/IIRA.htm, accessed 14.12.2018 (cit. on pp. 2, 22, 23, 25, 26, 98, 101–103, 126, 143, 171).

[5] W. Dorst, *Umsetzungsstrategie Industrie 4.0: Ergebnisbericht der Plattform Industrie 4.0*, Bitkom Research GmbH, 2015 (cit. on p. 2).

[6] Object Management Group, *Business Process Model and Notation (BPMN)*, version 2.0.2, OMG, 2013, URL: http://www.omg.org/spec/BPMN (cit. on p. 3).

[7] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang and F. Sui, *Digital Twin-driven Product Design, Manufacturing and Service with Big Data*, The International Journal of Advanced Manufacturing Technology (2017) 1 (cit. on pp. 3, 35).

[8] M. Mrissa, L. Médini, J.-P. Jamont, N. Le Sommer and J. Laplace, *An avatar architecture for the web of things*, IEEE Internet Computing **19** (2015) 30 (cit. on pp. 3, 40, 134).

[9] S. Malakuti, P. van Schalkwyk, B. Boss, C. Sastry, V. Runkana and other, *Digital Twins for Industrial Applications*, IIC White Paper, 2020, URL: https://www.iiconsortium.org/white-papers.htm (cit. on pp. 3, 31, 32, 58).

[10] E. H. Glaessgen and D. Stargel, "The Digital Twin Paradigm for future NASA and US Air Force Vehicles", *53rd Struct. Dyn. Mater. Conf. Special Session: Digital Twin, Honolulu, HI, US*, 2012 1 (cit. on pp. 3, 35).

[11] R. Stark and T. Damerau, "Digital Twin", *CIRP Encyclopedia of Production Engineering*, ed. by S. Chatti and T. Tolio, Springer Berlin Heidelberg, 2019 1 (cit. on p. 3).

[12] B. Boss, M. Hoffmeister, T. Deppe, F. Pethig, S. Bader, E. Barnstedt et al.,
Details of the Asset Administration Shell Part 1, The exchange of information between partners in the value chain of Industrie 4.0, tech. rep., version 2.0, Plattform Industrie 4.0, ZVEI, 2019, URL: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/Details-of-the-Asset-Administration-Shell-Part1.html (cit. on pp. 3, 10, 12, 25, 31, 32, 38, 58, 64, 69, 73, 134).

[13] A. Halbardier, D. Waltermire and M. Johnson, "Specification for the Asset Reporting Format 1.", *NIST Interagency Report 7694*, National Institute of Standards and Technology (NIST), 2011 (cit. on p. 3).

[14] bitkom, *Industrie 4.0 - so digital sind Deutschlands Fabriken*,
available online: https://de.statista.com/statistik/daten/studie/830813/umfrage/hemmnisse-beim-einsatz-von-industrie-40-anwendungen-in-deutschland/ (accessed 13.11.2020), 2018 (cit. on p. 6).

[15] H. Kagermann, "Change Through Digitization — Value Creation in the Age of Industry 4.0", *Management of Permanent Change*, Springer, 2015 23 (cit. on p. 6).

[16] G. Hohpe and B. Woolf,
Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions, Addison-Wesley Professional, 2004 (cit. on pp. 7, 45).

[17] B. Otto, *Interview with Reinhold Achatz on "Data Sovereignty and Data Ecosystems"*, Business & Information Systems Engineering **61** (2019) 635 (cit. on pp. 7, 48).

[18] R. Camilli and A. Duisberg, *Big Data and the automotive industry*,
(2013), accessed on 15.11.2020, URL: https://www.twobirds.com/~/media/pdfs/video-transcripts/big-data-and-the-automotive-industry-roberto-camilli-and-alexander-duisberg.pdf (cit. on pp. 7, 49).

[19] S. R. Bader and M. Maleshkova, "The Semantic Asset Administration Shell",
International Conference on Semantic Systems, Springer, 2019 159
(cit. on pp. 10, 58, 64, 73, 131).

[20] S. Bader, J. Pullmann, C. Mader, S. Tramp, C. Quix, A. Mueller, H. Akyürek, M. Böckmann, B. Imbusch, J. Lipp, S. Geisler and C. Lange,
"The International Data Spaces Information Model", *International Semantic Web Conference*, accepted for publication, Springer, 2020 (cit. on pp. 10, 58).

[21] S. R. Bader and M. Maleshkova, *Towards Enforceable Usage Policies for Industry 4.0*, Joint Proceedings of the 1st International Workshop on Knowledge Graph Building and LASCAR, coolocated with ESWC2019 (2019), http://ceur-ws.org/Vol-2489/, accessed 25.11.2019 (cit. on pp. 10, 50, 58, 83, 84, 86, 98).

[22] S. R. Bader and M. Maleshkova,
Towards Integrated Data Control for Digital Twins in Industry 4.0,
(2020), Proceedings of the International Workshop on Semantic Digital Twins
(cit. on pp. 10, 58, 84).

[23] S. R. Bader and M. Maleshkova,
SOLIOT — Decentralized Data Control and Interactions for IoT, Future Internet **12** (2020) 105
(cit. on pp. 11, 98, 118).

[24] S. R. Bader, C. Wolff, M. Vössing and J.-P. Schmidt,
"Towards Enabling Cyber-Physical Systems in Brownfield Environments",
International Conference on Exploring Service Science, Springer, 2018 165 (cit. on p. 11).

[25] S. R. Bader and M. Maleshkova, "Virtual representations for an iterative IoT deployment",
Companion Proceedings of the The Web Conference 2018, 2018 1887
(cit. on pp. 11, 35, 40, 98, 105, 114, 115).

[26] T. Käfer, S. R. Bader, L. Heling, R. Manke and A. Harth,
"Exposing Internet of Things devices via REST and Linked Data Interfaces",
Proc. 2nd workshop semantic web technol. Internet Things, 2017 1 (cit. on pp. 11, 98).

[27] S. R. Bader, M. Vössing, C. Wolff, J. Walk and M. Maleshkova,
"Supporting the Dispatching Process for Maintenance Technicians in Industry 4.0.", *WM*, 2017
131 (cit. on pp. 11, 112).

[28] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak,
N. Blomberg et al.,
The FAIR Guiding Principles for Scientific Data Management and Stewardship,
Scientific Data **3** (2016) (cit. on pp. 11, 81, 145).

[29] S. R. Bader, I. Grangel-Gonzalez, P. Nanjappa, M.-E. Vidal and M. Maleshkova,
"A Knowledge Graph for Industry 4.0", *European Semantic Web Conference*, Springer, 2020 465
(cit. on pp. 11, 144).

[30] S. Bader and J. Oevermann, "Semantic Annotation of Heterogeneous Data Sources: Towards an
Integrated Information Framework for Service Technicians",
Proceedings of the 13th International Conference on Semantic Systems, 2017 73
(cit. on pp. 11, 58).

[31] S. R. Bader, I. Grangel-González, M. Tasnim and S. Lohmann,
"Structuring the Industry 4.0 Landscape", *2019 24th IEEE International Conference on
Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2019 224
(cit. on pp. 12, 144, 146, 163).

[32] S. R. Bader, M. Maleshkova and S. Lohmann,
"Structuring Reference Architectures for the Industrial Internet of Things", vol. 11, 7,
Multidisciplinary Digital Publishing Institute, 2019 151 (cit. on pp. 12, 144, 148, 163).

[33] S. R. Bader, "Automating the Dynamic Interactions of Self-governed Components in Distributed
Architectures", *European Semantic Web Conference*, Springer, 2017 173 (cit. on pp. 12, 144).

[34] S. R. Bader, A. Wolf and F. L. Keppmann,
"Evaluation Environment for Linked Data Web Services.",
Joint Proceedings of SEMANTiCS 2017 Workshops,
available online: http://ceur-ws.org/Vol-2063/salad-paper1.pdf, 2017 (cit. on pp. 12, 98).

[35] B. Otto, S. Lohmann, S. Auer, G. Brost, J. Cirullies, A. Eitel, T. Ernst, C. Haas, M. Huber,
C. Jung et al., *Reference Architecture Model*, version 3.0, Fraunhofer-Gesellschaft, Munich
(2019), accessible at https://www.internationaldataspaces.org/wp-content/uploads/2019/03/IDS-
Reference-Architecture-Model-3.0.pdf
(cit. on pp. 12, 50, 58, 77, 78, 81, 84, 101, 103, 104, 163, 171, 173).

[36] A. Eitel, C. Jung, C. Kühnle, F. Bruckner, G. Brost, P. Birnstill, R. Nagel and S. Bader, *Usage Control in the International Data Spaces, Position Paper*, tech. rep., version 2.0, International Data Spaces Association, 2019, URL: https://www.internationaldataspaces.org/wp-content/uploads/2020/06/IDSA-Position-Paper-Usage-Control-in-IDS-2.0.pdf (cit. on pp. 12, 58, 125).

[37] S. R. Bader, C. Azkan and L. Stojanovic, "Introduction to Smart Service Architectures", *Smart Service Management - Design Guidelines and Best Practices*, ed. by M. Maleshkova, N. Kühl and P. Jussen, to be published, Springer Nature, 2020, chap. 3.1 (cit. on p. 12).

[38] S. R. Bader, C. Azkan and L. Stojanovic, "Reference Architecture Models for Smart Services", *Smart Service Management - Design Guidelines and Best Practices*, ed. by M. Maleshkova, N. Kühl and P. Jussen, to be published, Springer Nature, 2020, chap. 3.2 (cit. on pp. 12, 144).

[39] S. R. Bader and L. Stojanovic, "Reference Architecture Models for Smart Service Networks", *Smart Service Management - Design Guidelines and Best Practices*, ed. by M. Maleshkova, N. Kühl and P. Jussen, to be published, Springer Nature, 2020, chap. 3.3 (cit. on p. 12).

[40] L. Stojanovic and S. R. Bader, "Smart Services in the Physical World: Digital Twins", *Smart Service Management - Design Guidelines and Best Practices*, ed. by M. Maleshkova, N. Kühl and P. Jussen, to be published, Springer Nature, 2020, chap. 3.4 (cit. on pp. 12, 58).

[41] B. Boss, S. Bader, A. Orzelski and M. Hoffmeister, "Verwaltungsschale", *Handbuch Industrie 4.0: Produktion, Automatisierung und Logistik*, ed. by M. ten Hompel, B. Vogel-Heuser and T. Bauernhansl, Springer Berlin Heidelberg, 2019 1, URL: https://doi.org/10.1007/978-3-662-45537-1_139-1 (cit. on pp. 12, 58, 64).

[42] G. Eggers, B. Fondermann, B. Maier, K. Ottradovetz, J. Pfrommer, R. Reinhardt, H. Rollin, A. Schmieg, S. Steinbuß, P. Trinius, A. Weiss, C. Weiss and S. Wilfling, *GAIA-X: Technical Architecture*, 2020, URL: https://www.bmwi.de/Redaktion/EN/Publikationen/gaia-x-technical-architecture.html (cit. on pp. 12, 27).

[43] Gaia-X European Association for Data and Cloud AISBL, *Software Requirements Specification for Gaia-X Federation Services - Federated Catalogue Core Catalogue Features*, available online: https://www.gxfs.de/public-tender/ (accessed on 11.07.2021), 2021 (cit. on p. 12).

[44] S. Demeyer, "Research methods in computer science", *ICSM*, 2011 600 (cit. on p. 12).

[45] S. Feldmann, *Trending Topics in the Consulting Industry*, available online: https://www.statista.com/chart/16882/trending-topics-consulting/ (accessed 13.11.2020), 2019 (cit. on p. 14).

[46] bitkom, *Welches sind die wichtigsten IT-Trends des Jahres 2018?*, available online: https://de.statista.com/statistik/studie/id/6287/dokument/itk-branche-deutschland-statista-dossier/ (accessed 13.11.2020), 2018 (cit. on p. 14).

[47] bitkom, *Welches sind die wichtigsten IT-Trends des Jahres 2017?*, available online: https://de.statista.com/statistik/studie/id/6470/dokument/it-branche-deutschland-statista-dossier/ (accessed 13.11.2020), 2017 (cit. on p. 14).

[48] 451 Research,
Internet of Things (IoT) merger and acquisition (M&A) spending worldwide from 2017 to 2019,
available online: https://www.statista.com/study/27915/internet-of-things-iot-statista-dossier/
(accessed 13.11.2020), 2019 (cit. on p. 14).

[49] Markets and Markets and Statista,
Industrial Internet of Things market size worldwide from 2017 to 2025,
available online: https://www.statista.com/study/27915/internet-of-things-iot-statista-dossier/
(accessed 13.11.2020), 2020 (cit. on p. 14).

[50] Experton Group, *Investition in Industrie 4.0 in Deutschland in den Jahren 2013 bis 2020*,
available online: https://de.statista.com/statistik/daten/studie/372846/umfrage/investition-in-industrie-40-in-deutschland/ (accessed 13.11.2020), 2014 (cit. on p. 15).

[51] Valid Research, *Industrie 4.0 im deutschen Mittelstand*,
available online: https://de.statista.com/statistik/daten/studie/831146/umfrage/umfrage-zur-nutzung-digitaler-technologien-im-deutschen-mittelstand/ (accessed 13.11.2020), 2018
(cit. on p. 15).

[52] VDE, *VDE Tec Report 2019*,
available online: https://de.statista.com/statistik/daten/studie/1013642/umfrage/umfrage-zum-entwicklungsstand-der-industrie-4-0-in-deutschland/ (accessed 13.11.2020), 2019
(cit. on p. 15).

[53] Staufen, *Wie weit ist Ihr Unternehmen auf dem Weg zur "Smart Factory"?*,
available online: https://de.statista.com/statistik/daten/studie/1078438/umfrage/smart-factory-umsetzung-in-deutschen-unternehmen/ (accessed 25.12.2020), 2019 (cit. on pp. 15, 59).

[54] M. Krötzsch, "OWL 2 Profiles: An Introduction to Lightweight Ontology Languages",
Reasoning Web International Summer School, Springer, 2012 112 (cit. on p. 17).

[55] T. Heath and C. Bizer, *Linked data: Evolving the web into a global data space*,
Synthesis lectures on the semantic web: theory and technology **1** (2011) 1.

[56] S. Das, S. Sundara and R. Cyganiak,
R2RML: RDB to RDF Mapping Language, W3C Recommendation,
Cambridge, MA: World Wide Web Consortium (W3C)(www.w3.org/TR/r2rml) (2012)
(cit. on p. 21).

[57] W3C, *JSON-LD 1.0 – A JSON-based Serialization for Linked Data – W3C Recommendation 16 January 2014*, 2014, URL: https://www.w3.org/TR/2014/REC-json-ld-20140116/.

[58] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens and R. Van de Walle,
"RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data", *LDOW*,
2014 (cit. on p. 21).

[59] P. J. Hayes and P. F. Patel-Schneider, *RDF 1.1 Semantics*, W3C Recommendation (2014),
URL: https://www.w3.org/TR/rdf11-mt/ (cit. on pp. 18, 72, 74).

[60] S. Speicher, J. Arwe and A. Malhotra, *Linked Data Platform 1.0*, https://www.w3.org/TR/ldp/,
W3C Recommendation, accessed 30.06.2016, 2015 (cit. on p. 21).

[61] H. Knublauch and D. Kontokostas, *Shapes Constraint Language (SHACL)*,
http://www.w3.org/TR/shacl/, W3C Recommendation, accessed 27. September 2020, 2017
(cit. on pp. 21, 72).

[62] M. Kovatsch, R. Matsukura, M. Lagally, T. Kawaguchi, K. Toumura and K. Kajimoto, *Web of Things (WoT) Architecture*, Candidate Recommendation, available at https://www.w3.org/TR/wot-architecture/, accessed 28.09.2019, 2019 (cit. on pp. 23, 24, 40, 98, 102–104, 126).

[63] C. K. Ogden and I. A. Richards, *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*, vol. 29, K. Paul, Trench, Trubner & Company, Limited, 1923 (cit. on pp. 19, 20).

[64] T. Gruber, *Toward Principles for the Design of Ontologies used for Knowledge Sharing*, International Journal Human-Computer Studies **43** (1995) 907 (cit. on p. 20).

[65] D. U. Board, *DCMI Metadata Terms*, Dublin Core Metadata Initiative (2020), also reflected by ISO 15836-1:2017, URL: http://dublincore.org/specifications/dublin-core/dcmi-terms/2020-01-20/ (cit. on p. 20).

[66] A. Miles, B. Matthews, M. Wilson and D. Brickley, "SKOS Core: Simple Knowledge Organisation for the Web", *International Conference on Dublin Core and Metadata Applications*, 2005 3 (cit. on p. 20).

[67] F. Maali, J. Erickson and P. Archer, *Data Catalog Vocabulary (DCAT)*, tech. rep., W3C, 2014, URL: https://www.w3.org/TR/vocab-dcat/ (cit. on p. 20).

[68] T. Berners-Lee, *Design Issues – Linked Data*, 2009, URL: http://www.w3.org/DesignIssues/ (visited on 16/09/2016) (cit. on p. 21).

[69] S. Staab, J. Lehmann and R. Verborgh, "Structured Knowledge on the Web 7.0", *Companion Proceedings of the The Web Conference 2018*, 2018 885 (cit. on p. 21).

[70] H. Boyes, B. Hallaq, J. Cunningham and T. Watson, *The Industrial Internet of Things (IIoT): An analysis framework*, en, Computers in Industry **101** (2018) 1 (cit. on pp. 22, 53, 134).

[71] M. Hermann, T. Pentek and B. Otto, "Design Principles for Industrie 4.0 Scenarios", *Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, USA, 05.01.2016*, IEEE Computer Society Washington, DC, USA, 2016 3928 (cit. on p. 22).

[72] R. Baheti and H. Gill, *Cyber-physical Systems*, In The Impact of Control Technology **12** (2011) 161, Available online: http://www.gaudisite.nl/ReferenceArchitecturePrimerSlides.pdf (accessed on 08.10.2018) (cit. on p. 22).

[73] *ISO/IEC 18000: Information technology — Radio frequency identification for item management*, Technical Report (2008), available online: https://www.iso.org/standard/46145.html (accessed on 12.04.2021) (cit. on p. 22).

[74] P. Adolphs, S. Berlik, W. Dorst, J. Friedrich, C. Gericke, M. Hankel, R. Heidel, M. Hoffmeister, C. Mosch, R. Pichler, U. Rauschecker, T. Schulz, K. Schweichhart, E. J. Steffens, M. Taube, I. Weber, M. Wollschlaeger and S. Mätzler, *DIN SPEC 91345: Reference Architecture Model Industrie 4.0*, DIN SPEC **4** (2016), Available online: https://webstore.ansi.org/Standards/DIN/DINSPEC913452016 (accessed on 13.11.2018) (cit. on pp. 23, 98, 101, 103, 104, 163, 171).

[75] OPC Foundation, *OPC Unified Architecture - Part 1: Concepts*, version 1.0, OPC Specification, 2006 (cit. on p. 23).

[76] *IEC 62714: Engineering data exchange format for use in industrial automation systems engineering - Automation markup language - Part 1: Architecture and general requirements*, Technical Report (2014), updated by IEC 62714-1:2018, available online: https://webstore.iec.ch/publication/7388 (accessed on 26.10.2018).

[77] Alliance of Industrial Internet, *Industrial Internet Architecture*, White Paper, version 1.0, White Paper, Available online: http://en.aii-alliance.org/uploadfile/2017/0307/Industrial.pdf (accessed on 29.03.2019), 2016 (cit. on pp. 26, 171, 176).

[78] B. Otto, S. Lohmann, S. Auer, G. Brost, J. Cirullies, A. Eitel, T. Ernst, C. Haas, M. Huber, C. Jung et al., *Reference Architecture Model for the Industrial Data Space*, tech. rep., Fraunhofer-Gesellschaft, Munich, 2017.

[79] B. Boss, M. Hoffmeister, T. Deppe, F. Pethig, E. Barnstedt et al., *Details of the Asset Administration Shell Part 1, The exchange of information between partners in the value chain of Industrie 4.0*, tech. rep., version 1.0, Plattform Industrie 4.0, ZVEI, 2018, URL: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/Details-of-the-Asset-Administration-Shell-Part1.html.

[80] B. Raymor, R. Coppen, A. Banks, E. Briggs, K. Borgendale and R. Gupta, *MQTT Version 5.0*, OASIS Committee Specification 02, 2018 (cit. on pp. 31, 120).

[81] IEC TR 62541-1:2016, *OPC Unified Architecture - Part 1: Overview and concepts*, 2016 (cit. on p. 23).

[82] F. Volz, L. Stojanovic and R. Lamberti, "An Industrial Marketplace - the Smart Factory Web Approach and Integration of the International Data Space", *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, 2019 714 (cit. on p. 23).

[83] S. Malakuti, J. Schmitt, M. Platenius-Mohr, S. Grüner, R. Gitzel and P. Bihani, "A Four-Layer Architecture Pattern for Constructing and Managing Digital Twins", *European Conference on Software Architecture*, Springer, 2019 231 (cit. on p. 24).

[84] R. Joshi, P. Didier, J. Jimenez and T. Carey, *The Industrial Internet of Things Volume G5: Connectivity Framework*, ed. by R. Joshi, S. Mellor and P. Didier, Technical Manual, Available online: https://www.iiconsortium.org/IICF.htm (accessed on 02.08.2018), Industrial Internet Consortium (IIC), 2018 (cit. on pp. 25, 27, 102, 159, 176).

[85] S. Schrecker, H. Soroush, J. Molina, J. LeBlanc, F. Hirsch, M. Buchheit, A. Ginter, R. Martin, H. Banavara, S. Eswarahally, K. Raman, A. King, Q. Zhan, P. MacKay and B. Witten, *Industrial Internet of Things Volume G4: Security Framework*, ed. by S. Mellor, M. Buchheit, J. LeBlanc, S. Schrecker, H. Soroush, J. Molina, R. Martin, F. Hirsch, K. Raman, J. Caldwell, D. Meltzer and J. Lund, Technical Manual, Available online: https://www.iiconsortium.org/IISF.htm (accessed on 02.08.2018), Industrial Internet Consortium (IIC), 2016 (cit. on pp. 25, 176).

[86] bitkom, *Industrie 4.0 - so digital sind Deutschlands Fabriken*, available online: https://de.statista.com/statistik/daten/studie/830903/umfrage/fuehrende-nationen-beim-thema-industrie-40-in-deutschland/ (accessed 13.11.2020), 2018 (cit. on p. 25).

[87] J. Fonseca, P. Guillemin, M. Bauer, L. Frost, G. Privat, A. Abbas, W. Li, D. Fernández, M. Fisher, A. Wright and S. M. Jeong, *Context Information Management (CIM); NGSI-LD API*, ETSI, 2018, URL: https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?wki_id=54473 (cit. on p. 26).

[88] A. Greenberg, J. Hamilton, D. A. Maltz and P. Patel, *The Cost of a Cloud: Research Problems in Data Center Networks*, SIGCOMM Computer Communication Review **39** (2009).

[89] M. Satyanarayanan, P. Bahl, R. Caceres and N. Davies, *The Case for VM-Based Cloudlets in Mobile Computing*, IEEE pervasive Computing **8** (2009) 14.

[90] D. Riemer, F. Kaulfersch, R. Hutmacher and L. Stojanovic, "StreamPipes: solving the challenge with semantic stream processing pipelines", en, ACM Press, 2015 330, ISBN: 978-1-4503-3286-6 (cit. on p. 31).

[91] A. V. Sambra, E. Mansour, S. Hawke, M. Zereba, N. Greco, A. Ghanem, D. Zagidulin, A. Aboulnaga and T. Berners-Lee, *Solid: A Platform for Decentralized Social Applications Based on Linked Data*, 2016 (cit. on p. 119).

[92] M. Maleshkova, P. Philipp, Y. Sure-Vetter and R. Studer, *Smart Web Services (SmartWS)–The Future of Services on the Web*, IPSI Transactions on Advanced Research **12** (2016) (cit. on p. 43).

[93] N. Mitra, Y. Lafon et al., *Soap version 1.2 part 0: Primer*, W3C recommendation **24** (2003) 12 (cit. on p. 28).

[94] D. Booth and C. K. Liu, *Web services description language (WSDL) version 2.0 part 0: Primer*, W3C Recommendation **26** (2007) (cit. on p. 28).

[95] R. T. Fielding, *Architectural styles and the design of network-based software architectures*, Dissertation, University of California, Irvine, 2000 (cit. on pp. 30, 111).

[96] T. Berners-Lee, R. Fielding and L. Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, RFC 3986 (INTERNET STANDARD), Updated by RFCs 6874, 7320, Internet Engineering Task Force, 2005, URL: http://www.ietf.org/rfc/rfc3986.txt (cit. on p. 30).

[97] J. Gregorio, R. Fielding, M. Hadley, M. Nottingham and D. Orchard, *URI Template*, RFC 6570 (Proposed Standard), Internet Engineering Task Force, 2012, URL: http://www.ietf.org/rfc/rfc6570.txt (cit. on p. 30).

[98] M. Lanthaler and C. Gütl, *Hydra: A Vocabulary for Hypermedia-Driven Web APIs.*, LDOW **996** (2013) (cit. on pp. 30, 40, 158).

[99] R. Verborgh, T. Steiner, D. Van Deursen, S. Coppens, E. Mannens, R. Van de Walle and J. G. Vallés, "RESTdesc - A Functionality-centered Approach to Semantic Service Description and Composition", *Proceedings of the 9th Extended Semantic Web Conference, Crete, Greece*, 2012 27 (cit. on pp. 30, 39).

[100] S. Stadtmüller, S. Speiser, A. Harth and R. Studer, "Data-Fu: a language and an interpreter for interaction with read/write linked data", *WWW2013*, 2013 1225 (cit. on pp. 31, 72, 74, 150).

[101] S. Kaebisch, T. Kamiya, M. McCool, V. Charpenay and M. Kovatsch,
Web of Things (WoT) Thing Description, W3C Proposed Recommendation, 2020,
URL: https://www.w3.org/TR/2020/PR-wot-thing-description-20200130/ (cit. on pp. 31, 104).

[102] H. Bedenbender, A. Bentkus, U. Epple, T. Hadlich, R. Heidel, O. Hillermeier, M. Hoffmeister, H. Huhle, Markus Kiele-Dunsche, H. Koziolek, S. Lohmann, M. Mendes, J. Neidig, F. Palm, S. Pollmeier, B. Rauscher, F. Schewe, M. Wollschlaeger, I. Weber and B. Waser,
Industrie 4.0 Plug-and-Produce for Adaptable Factories: Example Use Case Definition, Models, and Implementation, http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/Industrie-40-%20Plug-and-Produce, 2017 (cit. on p. 31).

[103] E. Bournival, E. Simmon, M. Buchheit, C. Baudoin, F. Hirsch, B. Boss et al.,
The Industrial Internet of Things Vocabulary Technical Report, Technical Manual, Available online: https://hub.iiconsortium.org/vocabulary (accessed on 28.11.2019), IIC, IIC, 2019 (cit. on pp. 33, 235).

[104] T. Moses, A. Anderson, A. Nadalin et al.,
eXtensible Access Control Markup Language (XACML), version 2.0, 2004,
URL: http://docs.oasis-open.org/xacml/access_control-xacml-2_0-core-spec-cd-04.pdf (cit. on p. 50).

[105] J. Park and R. Sandhu, *The UCON ABC usage control model*,
ACM Transactions on Information and System Security (TISSEC) **7** (2004) 128 (cit. on p. 50).

[106] B. Parducci, H. Lockhart, E. Rissanen et al.,
extensible access control markup language (xacml) version 3.0, version 3.0, 2013,
URL: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf (cit. on p. 49).

[107] C. Jung, A. Eitel and R. Schwarz,
"Enhancing Cloud Security with Context-aware Usage Control Policies", *GI-Jahrestagung*, 2014 211 (cit. on p. 50).

[108] R. Ianella and S. Villata, *ODRL Information Model 2.2*, tech. rep.,
https://www.w3.org/TR/odrl-model/: W3C ODRL Community Group, 2018 (cit. on pp. 49, 82, 84).

[109] Solid Community, *Web Access Control (WAC*,
Version 0.5.0, available online: https://solid.github.io/web-access-control-spec/ (accessed on 04.02.2021: https://github.com/solid/web-access-control-spec/tree/58eae0f548a11c02c30bf2f4a1d620f5ed147490), 2021 (cit. on p. 133).

[110] C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos,
Context Aware Computing for The Internet of Things: A Survey,
IEEE Communications Surveys & Tutorials **16** (2014) 414,
URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6512846 (cit. on pp. 35–37).

[111] K. A. Hribernik, L. Rabe, K.-D. Thoben and J. Schumacher,
The product avatar as a product-instance-centric information management concept,
International Journal of Product Lifecycle Management **1** (2006) 367 (cit. on pp. 35, 36).

[112] J. Hendler, *Where are all the intelligent agents?*, IEEE Intelligent systems **11** (2007) 2 (cit. on p. 36).

[113] J. Lee, B. Bagheri and H.-A. Kao,
A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems, en,
Manufacturing Letters **3** (2015) 18 (cit. on p. 36).

[114] E. Negri, L. Fumagalli and M. Macchi,
A Review of the roles of Digital Twin in CPS-based production systems,
Procedia Manufacturing **11** (2017) 939 (cit. on p. 36).

[115] M. Jacoby and T. Usländer,
Digital Twin and Internet of Things — Current Standards Landscape,
Applied Sciences **10** (2020) 6519 (cit. on p. 36).

[116] V. Souza, R. Cruz, W. Silva, S. Lins and V. Lucena,
"A Digital Twin Architecture Based on the Industrial Internet of Things Technologies",
2019 IEEE International Conference on Consumer Electronics (ICCE), IEEE, 2019 1
(cit. on pp. 36, 134).

[117] Object Management Group, *Meta Object Facility (MOF) 2.0 Core Specification*, (2020)
(cit. on p. 36).

[118] M. Sjarov, T. Lechler, J. Fuchs, M. Brossog, A. Selmaier, F. Faltus, T. Donhauser and J. Franke,
"The Digital Twin Concept in Industry–A Review and Systematization", *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1,
IEEE, 2020 1789 (cit. on p. 37).

[119] D. F. Blumberg, *Strategies for Improving Field Service Operations Productivity and Quality*, en,
The Service Industries Journal **14** (1994) 262 (cit. on p. 37).

[120] Y. Yamauchi, J. Whalen and D. G. Bobrow,
"Information Use of Service Technicians in Difficult Cases",
Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '03,
ACM, 2003 81 (cit. on p. 37).

[121] E. Schweitzer and J. Aurich, *Continuous improvement of industrial product-service systems*,
CIRP Journal of Manufacturing Science and Technology **3** (2010) 158 (cit. on p. 37).

[122] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargasvera, E. Motta and F. Ciravegna, *Semantic annotation for knowledge management: Requirements and a survey of the state of the art*, en,
Web Semantics: Science, Services and Agents on the World Wide Web **4** (2006) 14,
ISSN: 15708268 (cit. on p. 37).

[123] M. Maleshkova, C. Pedrinaci and J. Domingue,
"Investigating Web APIs on the World Wide Web",
2010 IEEE 8th European Conference on Web Services (ECOWS), IEEE, 2010 107 (cit. on p. 38).

[124] F. Bülthoff and M. Maleshkova, "RESTful or RESTless–Current state of today's top Web APIs",
European Semantic Web Conference, Springer, 2014 64 (cit. on p. 38).

[125] I. Grangel-González, L. Halilaj, G. Coskun, S. Auer, D. Collarana and M. Hoffmeister,
"Towards a Semantic Administrative Shell for Industry 4.0 Components",
International Conference on Semantic Computing (ICSC), 2016 230 (cit. on pp. 38, 69).

[126] I. Grangel-González, L. Halilaj, S. Auer, S. Lohmann, C. Lange and D. Collarana, "An
RDF-based approach for implementing Industry 4.0 Components with Administration Shells",
21st International Conference on Emerging Technologies and Factory Automation (ETFA),
IEEE, 2016 1 (cit. on pp. 38, 53).

[127] E. Tantik and R. Anderl,
Integrated Data Model and Structure for the Asset Administration Shell in Industrie 4.0, en,
Procedia CIRP **60** (2017) 86 (cit. on p. 38).

[128] B. Katti, C. Plociennik, M. Ruskowski and M. Schweitzer,
"SA-OPC-UA: Introducing Semantics to OPC-UA Application Methods",
14th International Conference on Automation Science and Engineering (CASE), IEEE, 2018
1189 (cit. on p. 38).

[129] C. Diedrich, A. Belyaev, T. Schroder, J. Vialkowitsch, A. Willmann, T. Uslander, H. Koziolek,
J. Wende, F. Pethig and O. Niggemann,
"Semantic interoperability for asset communication within smart factories",
22nd International Conference on Emerging Technologies and Factory Automation (ETFA),
IEEE, 2017 1 (cit. on p. 39).

[130] D. Roman, U. Keller, H. Lausen, J. De Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier,
C. Bussler and D. Fensel, *Web Service Modeling Ontology (WSMO), W3C Member Submission*,
2006, URL: http://www.wsmo.org/TR/d2/v1.3/ (cit. on p. 39).

[131] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan,
M. Paolucci, B. Parsia, T. Payne et al., *OWL-S: Semantic markup for web services*,
W3C member submission **22** (2004) (cit. on p. 39).

[132] C. Pedrinaci, J. Cardoso and T. Leidig,
"Linked USDL: a vocabulary for web-scale service trading",
European Semantic Web Conference, Springer, 2014 68 (cit. on pp. 39, 48).

[133] A. Dimou, R. Verborgh, M. V. Sande, E. Mannens and R. Van de Walle, "Machine-interpretable
dataset and service descriptions for heterogeneous data access and retrieval", en,
ACM Press, 2015 145 (cit. on p. 39).

[134] R. Verborgh, A. Harth, M. Maleshkova, S. Stadtmüller, T. Steiner, M. Taheriyan and
R. Van de Walle, "Survey of semantic description of REST APIs",
rest: Advanced Research Topics and Practical Applications, Springer, 2014 69 (cit. on p. 39).

[135] M. Taheriyan, C. A. Knoblock, P. Szekely and J. L. Ambite,
"Rapidly integrating services into the linked data cloud", *ISWC*, Springer, 2012 559
(cit. on p. 39).

[136] M. Vander Sande, R. Verborgh, A. Dimou, P. Colpaert and E. Mannens,
Hypermedia-Based Discovery for Source Selection Using Low-Cost Linked Data Interfaces:
IJSWIS **12** (2016) 79 (cit. on p. 40).

[137] J. Domingue, D. Roman and M. Stollberg,
Web service modeling ontology (WSMO)-An ontology for semantic web services, 2005
(cit. on pp. 40, 158).

[138] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi and K. Taylor,
"IoT-Lite: a lightweight semantic model for the Internet of Things",
*Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing,
Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People,
and Smart World Congress*, IEEE, 2016 90 (cit. on p. 41).

[139] R. Agarwal, D. G. Fernandez, T. Elsaleh, A. Gyrard, J. Lanza, L. Sanchez, N. Georgantas and V. Issarny, "Unified IoT ontology to enable interoperability and federation of testbeds", *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, IEEE, 2016 70 (cit. on p. 41).

[140] A. Gyrard, G. Atemezing, C. Bonnet, K. Boudaoud and M. Serrano, "Reusing and unifying background knowledge for internet of things with LOV4IoT", *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, IEEE, 2016 262 (cit. on pp. 41, 189).

[141] G. Loseto, S. Ieva, F. Gramegna, M. Ruta, F. Scioscia and E. Di Sciascio, *Linking the web of things: LDP-CoAP mapping*, Procedia Computer Science **83** (2016) 1182 (cit. on pp. 41, 126, 127, 129).

[142] T. Binz, G. Breiter, F. Leyman and T. Spatzier, *Portable cloud services using tosca*, IEEE Internet Computing **16** (2012) 80 (cit. on p. 41).

[143] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, *Edge Computing: Vision and Challenges*, en, IEEE Internet of Things Journal **3** (2016) 637 (cit. on p. 41).

[144] V. R. Sampath Kumar, A. Khamis, S. Fiorini, J. L. Carbonera, A. Olivares Alarcos, M. Habib, P. Goncalves, H. Li and J. I. Olszewska, *Ontologies for Industry 4.0*, The Knowledge Engineering Review **34** (2019) (cit. on p. 41).

[145] A. Whitmore, A. Agarwal and L. D. Xu, *The Internet of Things—A survey of topics and trends*, en, Information Systems Frontiers **17** (2015) 261, ISSN: 1387-3326, 1572-9419 (cit. on p. 41).

[146] A. Dohr, R. Modre-Opsrian, M. Drobics, D. Hayn and G. Schreier, "The internet of things for ambient assisted living", *Seventh International Conference on Information Technology: New Generations (ITNG)*, IEEE, 2010 804 (cit. on p. 42).

[147] A. J. Jara, F. J. Belchi, A. F. Alcolea, J. Santa, M. A. Zamora-Izquierdo and A. F. Gómez-Skarmeta, "A Pharmaceutical Intelligent Information System to detect allergies and Adverse Drugs Reactions based on internet of things", *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, IEEE, 2010 809 (cit. on p. 42).

[148] K. Aberer, M. Hauswirth and A. Salehi, "A middleware for fast and flexible sensor network deployment", *Proceedings of the 32nd international conference on Very large data bases*, VLDB Endowment, 2006 1199 (cit. on p. 42).

[149] A. Gómez-Goiri and D. López-de-Ipiña, *A triple space-based semantic distributed middleware for internet of things*, Current Trends in Web Engineering (2010) 447 (cit. on p. 42).

[150] Y. Huang and G. Li, "A semantic analysis for internet of things", *Intelligent computation technology and automation (icicta), 2010 international conference on*, vol. 1, IEEE, 2010 336 (cit. on p. 42).

[151] Z. Song, A. A. Cárdenas and R. Masuoka, "Semantic middleware for the internet of things", *Internet of Things (IOT), 2010*, IEEE, 2010 1 (cit. on p. 42).

[152] T. Hubauer, S. Lamparter, P. Haase and D. M. Herzig,
"Use Cases of the Industrial Knowledge Graph at Siemens",
International Semantic Web Conference (P&D/Industry/BlueSky), 2018 (cit. on p. 42).

[153] E. Kharlamov, D. Hovland, M. G. Skjæveland, D. Bilidas, E. Jiménez-Ruiz, G. Xiao, A. Soylu, D. Lanti, M. Rezk, D. Zheleznyakov et al., *Ontology Based Data Access in Statoil*,
Journal of Web Semantics **44** (2017) 3 (cit. on p. 42).

[154] M. N. Mami, D. Graux, S. Scerri, H. Jabeen, S. Auer and J. Lehmann,
"Squerall: Virtual Ontology-Based Access to Heterogeneous and Large Data Sources",
International Semantic Web Conference, Springer, 2019 229 (cit. on p. 42).

[155] M. N. Mami, D. Graux, H. Thakkar, S. Scerri, S. Auer and J. Lehmann,
The Query Translation Landscape: a Survey, arXiv preprint arXiv:1910.03118 (2019)
(cit. on p. 42).

[156] J. Lehmann, G. Sejdiu and H. Jabeen, *Distributed Knowledge Graph Processing in SANSA*,
HPI Future SOC Lab: Proceedings 2017 **130** (2019) 21 (cit. on p. 42).

[157] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kröller, M. Pagel, M. Hauswirth et al., *SPITFIRE: Towards a Semantic Web of Things*,
IEEE Communications Magazine **49** (2011) 40 (cit. on p. 42).

[158] F. L. Keppmann and M. Maleshkova,
Smart Components for Enabling Intelligent Web of Things Applications,
INTELLI 2016 (2016) 128 (cit. on pp. 43, 45).

[159] N. Sahlab, N. Jazdi and M. Weyrich,
"Dynamic Context Modeling for Cyber-Physical Systems Applied to a Pill Dispenser", en, *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*,
IEEE, 2020 1435 (cit. on p. 43).

[160] O. Hartig, P.-A. Champin, D. Arndt, J. Broekstra, B. DuCharme, G. Kellogg, P. Patel-Schneider, E. Prud'hommeaux, A. Seaborne, T. Thibodeau and B. Thompson, *RDF-star and SPARQL-star*,
https://w3c.github.io/rdf-star/cg-spec/2021-02-18.html,
W3C Draft Community Report, accessed 24.03.2021, 2021 (cit. on p. 44).

[161] A. Novotny and C. Bauer, "What do we really talk about when we talk about context in pervasive computing: a review and exploratory analysis", *Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services*, 2017 301
(cit. on p. 44).

[162] W. Li, G. Privat, J. M. Cantera, M. Bauer and F. Le Gall,
"Graph-based Semantic Evolution for Context Information Management Platforms",
2018 Global Internet of Things Summit (GIoTS), IEEE, 2018 1 (cit. on p. 44).

[163] B. Caesar, A. Hänel, E. Wenkler, C. Corinth, S. Ihlenfeldt and A. Fay,
"Information Model of a Digital Process Twin for Machining Processes", *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1,
IEEE, 2020 1765 (cit. on p. 44).

[164] T. Stock and G. Seliger, *Opportunities of Sustainable Manufacturing in Industry 4.0*, en,
Procedia CIRP **40** (2016) 536, ISSN: 22128271 (cit. on p. 44).

[165] J. Davis, T. Edgar, J. Porter, J. Bernaden and M. Sarli,
Smart manufacturing, manufacturing intelligence and demand-dynamic performance, en,
Computers & Chemical Engineering **47** (2012) 145, ISSN: 00981354 (cit. on p. 44).

[166] J. D. Hedengren and A. N. Eaton,
Overview of estimation methods for industrial dynamic systems,
Optimization and Engineering **18** (2017) 155 (cit. on p. 44).

[167] B. Stein and A. Morrison, *The enterprise data lake: Better integration and deeper analytics*,
PwC Technology Forecast: Rethinking integration **1** (2014) 1 (cit. on p. 45).

[168] P. Tanuska, L. Spendla and M. Kebisek,
"Data integration for incidents analysis in manufacturing infrastructure",
Computing Conference, 2017, IEEE, 2017 340 (cit. on p. 45).

[169] T. Lechler, J. Fuchs, M. Sjarov, M. Brossog, A. Selmaier, F. Faltus, T. Donhauser and J. Franke,
"Introduction of a comprehensive Structure Model for the Digital Twin in Manufacturing", *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*,
vol. 1, IEEE, 2020 1773 (cit. on p. 45).

[170] C. Pautasso, A. Ivanchikj and S. Schreier, "A Pattern Language for RESTful Conversations",
Proceedings of the 21st European Conference on Pattern Languages of Programs, 2016 1
(cit. on pp. 45, 46).

[171] A. Harth, C. A. Knoblock, S. Stadtmüller, R. Studer and P. Szekely,
"On-the-fly integration of static and dynamic linked data",
Proceedings of the Fourth International Conference on Consuming Linked Data-Volume 1034,
2013 1 (cit. on pp. 45, 108, 116, 118).

[172] T. C. Käfer, *Behaviour on Linked Data-Specification, Monitoring, and Execution*,
(2019), Ph.D. Thesis (cit. on p. 45).

[173] L. D. Xu, E. L. Xu and L. Li, *Industry 4.0: state of the art and future trends*,
International Journal of Production Research **56** (2018) 2941 (cit. on pp. 46, 52).

[174] J. Delsing, *IoT Automation: Arrowhead Framework*, CRC Press, 2017 (cit. on pp. 47, 171).

[175] M. Romanato, D. Drozdov, S. Patil, J. Delsing and V. Vyatkin,
"Arrowhead Datamanager integration with Eclipse 4DIAC environment", en, *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*,
IEEE, 2020 1377 (cit. on p. 47).

[176] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher and V. Young,
Mobile edge computing—A key technology towards 5G, ETSI White Paper **11** (2015) 1
(cit. on p. 47).

[177] F. Bonomi, R. Milito, J. Zhu and S. Addepalli,
"Fog computing and its role in the internet of things",
Proceedings of the first edition of the MCC workshop on Mobile cloud computing, ACM, 2012
13 (cit. on p. 47).

[178] M. Grieves and J. Vickers,
"Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems",
Transdisciplinary perspectives on complex systems, Springer, 2017 85 (cit. on p. 47).

[179] M. Gundall, C. Glas and H. D. Schotten, “Introduction of an Architecture for Flexible Future Process Control Systems as Enabler for Industry 4.0”, *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, IEEE, 2020 1047 (cit. on p. 47).

[180] A. Beygelzimer, A. Riabov, D. Sow, D. S. Turaga and O. Udrea, “Big Data Exploration Via Automated Orchestration of Analytic Workflows”, *ICAC 13*, 2013 153 (cit. on p. 47).

[181] E. Sirin, B. Parsia, D. Wu, J. Hendler and D. Nau, *HTN planning for Web Service composition using SHOP2*, en, Web Semantics **1** (2004) 377 (cit. on p. 47).

[182] E. Wittern and R. Fischer, “A Life-Cycle Model for Software Service Engineering”, *ESOCC*, Springer, 2013 164 (cit. on p. 47).

[183] S. Mayer, R. Verborgh, M. Kovatsch and F. Mattern, *Smart Configuration of Smart Environments*, IEEE Transactions on Automation Science and Engineering **13** (2016) 1247, ISSN: 1545-5955, 1558-3783 (cit. on p. 47).

[184] M. B. Alaya, S. Medjiah, T. Monteil and K. Drira, *Toward semantic interoperability in oneM2M architecture*, IEEE Communications Magazine **53** (2015) 35 (cit. on p. 47).

[185] S. Speiser, “Semantic annotations for WS-Policy”, *ICWS*, IEEE, 2010 449 (cit. on p. 47).

[186] M. Palmonari, M. Comerio and F. De Paoli, “Effective and flexible nfp-based ranking of web services”, *Service-Oriented Computing*, Springer, 2009 546 (cit. on p. 47).

[187] G. La Torre, S. Monteleone, M. Cavallo, V. D’Amico and V. Catania, “A Context-Aware Solution to Improve Web Service Discovery and User-Service Interaction”, IEEE, 2016 180 (cit. on p. 47).

[188] A. Perzylo, S. Profanter, M. Rickert and A. Knoll, “OPC UA NodeSet Ontologies as a Pillar of Representing Semantic Digital Twins of Manufacturing Resources”, *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019 1085 (cit. on p. 48).

[189] R. Schiekofer, S. Grimm, M. M. Brandt and M. Weyrich, “A formal mapping between OPC UA and the Semantic Web”, en, *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, IEEE, 2019 33 (cit. on p. 48).

[190] F. Jammes and H. Smit, *Service-Oriented Paradigms in Industrial Automation*, en, IEEE Transactions on Industrial Informatics **1** (2005) 62 (cit. on pp. 48, 135).

[191] S. Weyer, T. Meyer, M. Ohmer, D. Gorecky and D. Zühlke, *Future Modeling and Simulation of CPS-based Factories: an Example from the Automotive Industry*, IFAC-PapersOnLine **49** (2016) 97 (cit. on p. 48).

[192] D. Oberle, A. Barros, U. Kylau and S. Heinzl, *A unified description language for human to automated services*, Information Systems **38** (2013) 155 (cit. on p. 48).

[193] H. Labbaci, N. Cheniki, Y. Sam, N. Messai, B. Medjahed and Y. Aklouf, "A Linked Open Data Approach for Web Service Evolution", *OTM Confederated International Conferences: On the Move to Meaningful Internet Systems*, Springer, 2019 265 (cit. on p. 48).

[194] H. Perera, W. Hussain, D. Mougouei, R. A. Shams, A. Nurwidyantoro and J. Whittle, "Towards Integrating Human Values into Software: Mapping Principles and Rights of GDPR to Values", *2019 IEEE 27th International Requirements Engineering Conference (RE)*, IEEE, 2019 404 (cit. on p. 48).

[195] Google, *Evaluation of Cohort Algorithms for the FLoC API*, White Paper, accessed on 15.11.2020: Google Research, 2020, URL: https://github.com/google/ads-privacy/tree/master/proposals/FLoC (cit. on p. 49).

[196] G. Flouris, T. Patkos, I. Chrysakis, I. Konstantinou, N. Nikolov, P. Papadakos, J. Pitt, D. Roman, A. Stan and C. Zeginis, "Towards a Collective Awareness Platform for Privacy Concerns and Expectations", *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, Springer, 2018 135 (cit. on p. 49).

[197] J. Wagner, *China's Cybersecurity Law: What You Need to Know*, Newspaper Article, accessed 22.11.2020, 2017, URL: https://thediplomat.com/2017/06/chinas-cybersecurity-law-what-you-need-to-know/ (cit. on p. 49).

[198] R. Brandom and C. Lecher, *House passes controversial legislation giving the US more access to overseas data*, Newspaper Article, accessed 22.11.2020, 2018, URL: https://www.theverge.com/2018/3/22/17131004/cloud-act-congress-omnibus-passed-mlat (cit. on p. 49).

[199] S. Yu, C. Wang, K. Ren and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing", *2010 Proceedings IEEE INFOCOM*, Ieee, 2010 1 (cit. on p. 49).

[200] P. Mazzoleni, B. Crispo, S. Sivasubramanian and E. Bertino, *XACML Policy Integration Algorithms*, Transactions on Information and System Security **11** (2008) (cit. on p. 50).

[201] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege and D. Spence, *RFC 2904: AAA Authorization Framework*, Request For Comment, Network Working Group (2000) (cit. on p. 50).

[202] F. Loukil, C. Ghedira-Guegan, K. Boukadi and A. N. Benharkat, "LIoPY: A legal compliant ontology to preserve privacy for the Internet of Things", *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, IEEE, 2018 701 (cit. on p. 50).

[203] F. Loukil, C. Ghedira-Guegan, K. Boukadi and A. N. Benharkat, "Semantic IoT Gateway: Towards Automated Generation of Privacy-Preserving Smart Contracts in the Internet of Things", *OTM Confederated International Conferences*, Springer, 2018 207 (cit. on p. 50).

[204] V. Nagendra, A. Bhattacharya, V. Yegneswaran, A. Rahmati and S. Das, "An Intent-Based Automation Framework for Securing Dynamic Consumer IoT Infrastructures", WWW '20, Association for Computing Machinery, 2020 1625 (cit. on p. 50).

[205] M. A. Musen et al., *The Protégé Project: a look back and a look forward*, AI matters **1** (2015) 4 (cit. on p. 51).

[206] L. Halilaj, N. Petersen, I. Grangel-González, C. Lange, S. Auer, G. Coskun and S. Lohmann, "VoCol: An Integrated Environment to Support Version-Controlled Vocabulary Development", *EKAW 2016*, vol. 10024, Springer, 2016 303 (cit. on pp. 51, 151).

[207] S. Lohmann, V. Link, E. Marbach and S. Negru, "WebVOWL: Web-based Visualization of Ontologies", *EKAW 2014 Satellite Events. Revised Selected Papers.* Vol. 8982, LNCS, Springer, 2014 154 (cit. on p. 51).

[208] P. McBrien and A. Poulovassilis, "A Conceptual Modelling Approach to Visualising Linked Data", *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, Springer, 2019 227 (cit. on p. 51).

[209] S. Auer, S. Dietzold and T. Riechert, "OntoWiki – A Tool for Social, Semantic Collaboration", *5th International Semantic Web Conference, ISWC 2006*, vol. 4273, LNCS, Springer, 2006 736 (cit. on p. 51).

[210] M. Krötzsch, D. Vrandecic and M. Völkel, "Semantic MediaWiki", *5th International Semantic Web Conference, ISWC 2006*, vol. 4273, LNCS, Springer, 2006 935 (cit. on p. 51).

[211] S. Lafia, A. Turner and W. Kuhn, "Improving Discovery of Open Civic Data", *10th International Conference on Geographic Information Science, GIScience*, vol. 114, LIPIcs, Schloss Dagstuhl, 2018 9:1 (cit. on p. 52).

[212] C. Xiong, R. Power and J. Callan, "Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding", *26th International Conference on World Wide Web 2017*, ACM, 2017 1271 (cit. on p. 52).

[213] Y. Lu, K. C. Morris and S. Frechette, *Current Standards Landscape for Smart Manufacturing Systems*, National Institute of Standards and Technology **8107** (2016) 39 (cit. on p. 52).

[214] S. Andreev, O. Galinina, A. Pyattaev, M. Gerasimenko, T. Tirronen, J. Torsner et al., *Understanding the IoT Connectivity Landscape: A Contemporary M2M Radio Technology Roadmap*, IEEE Communications Magazine **53** (2015) 32 (cit. on p. 52).

[215] I. Grangel-Gonzalez, P. Baptista, L. Halilaj, S. Lohmann, M.-E. Vidal, C. Mader and S. Auer, "The Industry 4.0 Standards Landscape from a semantic Integration Perspective", en, *ETFA*, IEEE, 2017 1 (cit. on pp. 52, 53, 148, 149).

[216] J. A. Saucedo-Martínez, M. Pérez-Lara, J. A. Marmolejo-Saucedo, T. E. Salais-Fierro and P. Vasant, *Industry 4.0 framework for management and operations: a review*, Journal of Ambient Intelligence and Humanized Computing **9** (2018) 789 (cit. on p. 52).

[217] N. F. Noy, D. L. McGuinness et al., *Ontology Development 101: A Guide to Creating Your First Ontology*, 2001 (cit. on p. 53).

[218] R. Hilliard et al.,
ISO/IEC/IEEE 42010: Systems and Software Engineering - Architecture Description, 2011 (cit. on pp. 53, 146, 152).

[219] R. L. Nord, P. C. Clements, D. Emery and R. Hilliard,
"Reviewing architecture documents using question sets",
Proceedings of the 2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture, Cambridge, United Kingdom, IEEE, 2009 325 (cit. on p. 53).

[220] M. Weyrich and C. Ebert, *Reference Architectures for the Internet of Things*,
IEEE Software **33** (2016) 112 (cit. on p. 54).

[221] P. Sethi and S. R. Sarangi, *Internet of Things: Architectures, Protocols, and Applications*,
Journal of Electrical and Computer Engineering **2017** (2017) 1, Available online:
https://www.hindawi.com/journals/jece/2017/9324035/ (accessed on 01.10.2018) (cit. on p. 54).

[222] R. Y. Zhong, X. Xu, E. Klotz and S. T. Newman,
Intelligent Manufacturing in the Context of Industry 4.0: A Review, en, Engineering **3** (2017) 616 (cit. on pp. 54, 176).

[223] K.-D. Thoben, S. Wiesner and T. Wuest, *"Industrie 4.0" and Smart Manufacturing – A Review of Research Issues and Application Examples*, en,
International Journal of Automation Technology **11** (2017) 4 (cit. on pp. 54, 176).

[224] R. Strange and A. Zucchella, *Industry 4.0, global value chains and international business*,
Multinational Business Review **25** (2017) 174 (cit. on p. 54).

[225] M. Aloqaily, S. Otoum, I. Al Ridhawi and Y. Jararweh,
An intrusion detection system for connected vehicles in smart cities, Ad Hoc Networks (2019) (cit. on pp. 54, 176).

[226] S. Otoum, B. Kantarci and H. T. Mouftah,
On the Feasibility of Deep Learning in Sensor Network Intrusion Detection,
IEEE Networking Letters (2019) (cit. on p. 54).

[227] Y. Kotb, I. Al Ridhawi, M. Aloqaily, T. Baker, Y. Jararweh and H. Tawfik,
Cloud-Based Multi-Agent Cooperation for IoT Devices Using Workflow-Nets,
Journal of Grid Computing (2019) 1 (cit. on pp. 54, 177).

[228] M. Al-khafajiy, T. Baker, H. Al-Libawy, Z. Maamar, M. Aloqaily and Y. Jararweh,
Improving Fog Computing Performance via Fog-2-Fog Collaboration,
Future Generation Computer Systems **100** (2019) 266 (cit. on p. 54).

[229] Staufen, *Was sind aus Sicht Ihres Unternehmens die wichtigsten Vorteile von Künstlicher Intelligenz im Kontext von Industrie 4.0?*,
available online: http://de.statista.com/statistik/daten/studie/990505/umfrage/umfrage-zu-vorteilen-kuenstlicher-intelligenz-in-deutschen-industrieunternehmen (accessed 25.12.2020),
2020 (cit. on p. 60).

[230] R. Finn, K. Wadhwa, S. Grumbach and A. Fensel, *Byte Final Report and Guidelines*,
tech. rep. D7.3, BYTE Project, 2017,
URL: http://new.byte-project.eu/wp-content/uploads/2014/02/D7.3-Final-report-FINAL.pdf (cit. on p. 76).

[231] S. Grumbach, *Intermediation Platforms, an Economic Revolution*, ERCIM News **2014** (2014) (cit. on p. 76).

[232] B. Otto and M. Jarke,
"Designing a multi-sided data platform: findings from the International Data Spaces case",
Electronic Markets, vol. 29, Springer, 2019 (cit. on pp. 77, 78, 81).

[233] J. Attard, F. Orlandi and S. Auer, "Data value networks: Enabling a new data ecosystem",
International Conference on Web Intelligence (WI), IEEE, 2016 (cit. on p. 80).

[234] V. Presutti, E. Daga, A. Gangemi and E. Blomqvist,
"eXtreme Design with Content Ontology Design Patterns",
Proc. Workshop on Ontology Patterns, 2009 (cit. on p. 81).

[235] P. Hitzler, A. Gangemi and K. Janowicz,
Ontology engineering with ontology design patterns: foundations and applications, vol. 25,
IOS Press, 2016 (cit. on p. 81).

[236] J. Pullmann, N. Petersen, C. Mader, S. Lohmann and Z. Kemeny,
"Ontology-based Information Modelling in the Industrial Data Space", *ETFA*, IEEE, 2017
(cit. on p. 82).

[237] R. Dadwal, D. Graux, G. Sejdiu, H. Jabeen and J. Lehmann,
"Clustering Pipelines of Large RDF POI Data", *European Semantic Web Conference*,
Springer, 2019 24 (cit. on p. 91).

[238] W3C, *The Organization Ontology*, ed. by D. Raynolds, W3C Recommendation, 2014,
URL: http://www.w3.org/TR/2014/REC-vocab-org-20140116/ (cit. on p. 91).

[239] M. T. Frank, S. Bader, V. Simko and S. Zander,
"LSane: Collaborative Validation and Enrichment of Heterogeneous Observation Streams",
Proceedings of the 14th International Conference on Semantic Systems, vol. 137, Elsevier, 2018
235 (cit. on p. 98).

[240] R. Mahmud, S. N. Srirama, K. Ramamohanarao and R. Buyya,
Quality of Experience (QoE)-aware placement of applications in Fog computing environments,
Journal of Parallel and Distributed Computing **132** (2019) 190 (cit. on p. 98).

[241] M. T. Lazarescu,
Design of a WSN platform for long-term environmental monitoring for IoT applications,
IEEE Journal on emerging and selected topics in circuits and systems **3** (2013) 45 (cit. on p. 98).

[242] A. Bassi, M. Bauer, M. Fiedler, T. Kramp, R. v. Kranenburg, S. Lange and S. Meissner,
Enabling Things to Talk: Designing IoT solutions with the IoT Architectural Reference Model, en,
Springer, 2013 (cit. on pp. 98, 99, 101, 103, 104, 126, 134, 143, 157, 163, 171, 172, 176).

[243] T. Kindberg and S. Hawke, *The 'tag' URI Scheme*, RFC 4151,
Internet Engineering Task Force, 2005, URL: http://www.ietf.org/rfc/rfc4151.txt (cit. on p. 100).

[244] P. Leach, M. Mealling and R. Salz, *A Universally Unique IDentifier (UUID) URN Namespace*,
RFC 4122, Internet Engineering Task Force, 2005, URL: http://www.ietf.org/rfc/rfc4122
(cit. on p. 100).

[245] D. Reed, M. Sporny, D. Longley, C. Allen, R. Grant, M. Sabadello and J. Holt,
Decentralized Identifiers (DIDs), ed. by D. Reed, M. Sporny and M. Sabadello,
version 2.0, W3C Working Draft, 2020, URL: https://www.w3.org/TR/did-core/ (cit. on p. 100).

[246] S. Kavianpour, B. Shanmugam, S. Azam, M. Zamani, G. Narayana Samy and F. De Boer, *A Systematic Literature Review of Authentication in Internet of Things for Heterogeneous Devices*, en, Journal of Computer Networks and Communications **2019** (2019) 1, ISSN: 2090-7141, 2090-715X (cit. on pp. 101, 102).

[247] IoT-A, *IoT-A Unified Requirements List*, Available online: https://web.archive.org/web/20160322053934/http://www.iot-a.eu/public/requirements (accessed on 25.09.2018), Internet of Things - Architecture, 2016 (cit. on pp. 102, 104, 169).

[248] RFC 6347, *Datagram Transport Layer Security Version 1.2*, https://tools.ietf.org/html/rfc6347, accessed 29.04.2020, 2012 (cit. on p. 103).

[249] C. Pedrinaci, D. Lambert, M. Maleshkova, D. Liu, J. Domingue and R. Krummenacher, "Adaptive service binding with lightweight semantic web services", *Service engineering*, Springer, 2011 233 (cit. on p. 105).

[250] F. L. Keppmann, M. Maleshkova and A. Harth, "Semantic Technologies for Realising Decentralised Applications for the Web of Things", *ICECCS*, 2016 71 (cit. on pp. 107, 114).

[251] H. Bauer, C. Baur, D. Mohr, A. Tschiesner, T. Weskamp, K. Alicke, R. Mathis, O. Noterdaeme, A. Behrendt, R. Kelly, D. Wee, M. Breunig, S. Narayanan, M. Roggendorf, U. Huber and V. von der Tann, *Industry 4.0 after the initial hype–Where manufacturers are finding value and how they can best capture it*, McKinsey Digital (2016) (cit. on p. 113).

[252] H. Rauen, C. Mosch, O. Niggemann and J. Jasperneite, *Industrie 4.0 Kommunikation mit OPC UA*, Leitfaden zur Einführung in den Mittelstand. Hg. v. VDMA und Fraunhofer-Anwendungszentrum Industrial Automation. Frankfurt am Main (978-3-8163-0709-9) (2017) (cit. on p. 115).

[253] *Apache Marmotta*, accessed on 03.03.2017, 2017, URL: http://marmotta.apache.org/index.html (cit. on p. 116).

[254] J. Pfrommer, S. Grüner, T. Goldschmidt and D. Schulz, *A common core for information modeling in the Industrial Internet of Things*, en, at - Automatisierungstechnik **64** (2016) (cit. on pp. 119, 135).

[255] Z. Shelby, K. Hartke and C. Bormann, *The Constrained Application Protocol (CoAP)*, RFC 7252, https://tools.ietf.org/html/rfc6347, accessed 30.04.2020, 2014 (cit. on p. 120).

[256] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP", *2017 IEEE international systems engineering symposium (ISSE)*, IEEE, 2017 1 (cit. on p. 126).

[257] Q. Han, Y. Zhang and H. Li, *Efficient and robust attribute-based encryption supporting access policy hiding in Internet of Things*, Future Generation Computer Systems **83** (2018) 269 (cit. on p. 129).

[258] E. Wallace, D. Kiritsis, B. Smith, C. Will et al., "The Industrial Ontologies Foundry proof-of-concept project", *IFIP International Conference on Advances in Production Management Systems*, Springer, 2018 402 (cit. on p. 131).

[259] eCl@ss, *eCl@ss Major Release 11.0 - Classification and Product Description*, GitHub project, https://www.eclass-cdp.com/portal/info.seam (accessed on 10 June 2020), 2019 (cit. on p. 131).

[260] C. Parent and S. Spaccapietra, “An Overview of Modularity”, *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, Springer, 2009 5 (cit. on p. 145).

[261] Edgecross Consortium, *Edgecross Consortium to Address Edge Integration in IIoT-enabled Architectures*, ARC WHITE PAPER, Available online: https://www.edgecross.org/ja/data-download/pdf/Edgecross_Consortium_WP(E).pdf (accessed on 29.09.2018), ARC Advisory Group, 2018 (cit. on pp. 148, 163, 171).

[262] Industrial Value Chain Initiative, *Industrial Value Chain Reference Architecture (IVRA)*, Available online: https://iv-i.org/wp-test/wp-content/uploads/2017/09/doc_161208_Industrial_Value_Chain_Reference_Architecture.pdf (accessed on 08.10.2018), 2016 (cit. on pp. 148, 171).

[263] A. Rivas, I. Grangel-González, D. Collarana, J. Lehmann and M.-E. Vidal, “Unveiling Relations in the Industry 4.0 Standards Landscape based on Knowledge Graph Embeddings”, *International Conference on Database and Expert Systems Applications*, Springer, 2020 179 (cit. on p. 151).

[264] M. Färber, F. Bartscherer, C. Menne and A. Rettinger, *Linked Data Quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO*, Semantic Web **9** (2018) 77 (cit. on p. 152).

[265] B. Leiner, R. Cole, J. Postel and D. Mills, *The DARPA Internet Protocol Suite*, IEEE Communications Magazine **23** (1985) 29 (cit. on p. 155).

[266] Big Data Value Association, *European Big Data Value Strategic Research and Innovation Agenda*, version 4.0, Available online: http://www.bdva.eu/SRIA (accessed 23.09.2018), 2017 (cit. on pp. 156, 171, 176).

[267] A. Braune, C. Diedrich, S. Grüner, G. Hüttemann, M. Klein, C. Legat, M. Lieske, U. Löwen, M. Thron and T. Usländer, *Usage View of the Asset Administration Shell*, Discussion Paper, accessible at https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/2019-usage-view-asset-administration-shell.html, 2019 (cit. on p. 157).

[268] IEC 61360, *Standard data element types with associated classification scheme*, ISA/IEC Standard, International Electrotechnical Commission, https://webstore.iec.ch/publication/28560, 2017 (cit. on p. 158).

[269] C. Bizer, T. Heath and T. Berners-Lee, “Linked Data: The Story so far”, *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, IGI Global, 2011 205 (cit. on p. 158).

[270] K. Stouffer, J. Falco and K. Scarfone, *Guide to Industrial Control Systems (ICS) Security*, NIST special publication **800** (2011) 16.

[271] ISO/IEC 27001, *Information technology – Security techniques – Information security management systems – Requirements*, ISO/IEC Standard, http://www.iso.org/iso/home/standards/management-standards/iso27001.htm, 2013 (cit. on p. 162).

[272] ISA/IEC 62443, *Security for industrial automation and control systems*, IEC Standard, https://webstore.iec.ch/publication/7031, 2019 (cit. on p. 162).

[273] G. Muller, *A Reference Architecture Primer*, White paper, Available online: http://www.gaudisite.nl/ReferenceArchitecturePrimerSlides.pdf (accessed on 08.10.2018), 2008 (cit. on p. 163).

[274] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran and M. Guizani, *Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges*, en, IEEE Wireless Communications **24** (2017) 10, ISSN: 1536-1284 (cit. on p. 169).

[275] OpenFog Consortium Architecture Working Group, *Openfog Architecture Overview*, White Paper OPFWP001 **216** (2016) 35, Available online: https://www.openfogconsortium.org/ra/ (accessed on 05.12.2018) (cit. on p. 171).

[276] M. Freudenthal, V. Hanson, I. Nõgisto, I. Kromonov and S. Annuk, *X-Road Architecture, Technical Specification*, version 1.2, Available online: https://www.ria.ee/riigiarhitektuur/wiki/lib/exe/fetch.php?media=an:x-tee_kohtumised:arc-g_x-road_arhitecture_1.2_y-879-3.docx&usg=AOvVaw3fhwi5k_UEfnpBMsjJov5V (accesed on 05.10.2018), Cybernetica, 2015 (cit. on p. 171).

[277] A. Glikson, "Fi-ware: Core Platform for Future Internet Applications", *Proceedings of the 4th Annual International Conference on Systems and Storage*, ed. by P. Ta-Shma, ACM, 2011 (cit. on p. 171).

[278] P. Faure and P. Darmayan, *Le plan français Industrie du futur*, FR, Annales des Mines - Réalités industrielles **vembre 2016** (2016) 57, URL: https://www.cairn.info/revue-realites-industrielles-2016-4-page-57.htm (cit. on p. 171).

[279] S. Park, "OCF: A new open IoT consortium", *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, IEEE, 2017 356 (cit. on p. 171).

[280] M. Tiraboschi and F. Seghezzi, *Il Piano nazionale Industria 4.0: una lettura lavoristica*, Labour & Law Issues **2** (2016) 1 (cit. on p. 171).

[281] A.-R. Sadeghi, C. Wachsmann and M. Waidner, "Security and privacy challenges in industrial internet of things", en, *Proceedings of the 52nd Annual Design Automation Conference on - DAC '15*, ACM Press, 2015 1 (cit. on p. 176).

[282] J. Schütte and G. S. Brost, "A data usage control system using dynamic taint tracking", *30th International Conference on Advanced Information Networking and Applications (AINA)*, IEEE, 2016 909 (cit. on p. 176).

[283] Y. Cao and Y. Chen, "QoE-based node selection strategy for edge computing enabled Internet-of-Vehicles (EC-IoV)", *Visual Communications and Image Processing (VCIP)*, IEEE, 2017 1 (cit. on p. 177).

[284] S. Otoum, B. Kantarci and H. T. Mouftah, "Mitigating False Negative intruder decisions in WSN-based Smart Grid monitoring", *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, 2017 153 (cit. on p. 177).

[285] X. Huang, K. Xie, S. Leng, T. Yuan and M. Ma, *Improving Quality of Experience in multimedia Internet of Things leveraging machine learning on big data*, en, Future Generation Computer Systems **86** (2018) 1413 (cit. on p. 177).

[286] J. P. Lemayian and F. Al-Turjman,
"Intelligent IoT Communication in Smart Environments: An Overview",
Artificial Intelligence in IoT, Springer, 2019 207 (cit. on p. 177).

[287] H. Kagermann and W. Wahlster, *Zehn Jahre Industrie 4.0*,
Newspaper article, accessed on 10.04.2021, 2021,
URL: https://www.faz.net/aktuell/wirtschaft/digitec/digitale-technik/digitalisierung-der-produktion-zehn-jahre-industrie-4-0-17267696.html (cit. on p. 190).

Appendix

APPENDIX A

Publications

Several publications have been published during the work on this thesis. The following list presents the papers and articles grouped by their publication type.

Journal Articles:

1. **S. R. Bader** and M. Maleshkova: *SOLIOT — Decentralized Data Control and Interactions for IoT* In Future Internet, 12, 105, 2020. https://doi.org/10.3390/fi12060105

2. **S. R. Bader**, M. Maleshkova and S. Lohmann. *Structuring Reference Architectures for the Industrial Internet of Things*, Future Internet, 11, 2019. https://doi.org/10.3390/fi11070151

Conference Proceedings:

3. **S. R. Bader**, J. Pullmann, C. Mader, S. Tramp, C. Quix, A. Mueller, H. Akyürek, M. Böckmann,B. Imbusch, J. Lipp, S. Geisler and C. Lange: *The International Data Spaces Information Model.* In Proceedings of the International Semantic Web Conference, Springer, 2020. https://doi.org/10.1007/978-3-030-62466-8_12

4. **S. R. Bader**, I. Grangel-Gonzalez, P. Nanjappa, M.-E. Vidal and M. Maleshkova: *A Knowledge Graph for Industry 4.0* In Proceedings of the European Semantic Web Conference, Springer, 2020. https://doi.org/10.1007/978-3-030-49461-2_27

5. **S. R. Bader**: *Automating the dynamic interactions of self-governed components in distributed architectures* In European Semantic Web Conference, Springer, 2017. https://doi.org/10.1007/978-3-319-58451-5_12

6. **S. R. Bader** and M. Maleshkova: *The Semantic Asset Administration Shell.* In International Conference on Semantic Systems, Springer, 2019. https://doi.org/10.1007/978-3-030-33220-4_12

7. M. T. Frank, **S. Bader**, V. Simko and S. Zander: *LSane: Collaborative Validation and Enrichment of Heterogeneous Observation Streams.* In Proceedings of the 14th International Conference on Semantic Systems, vol. 137, Elsevier, 2018. https://doi.org/10.1016/j.procs.2018.09.022

8. **S. R. Bader** and J. Oevermann: *Semantic Annotation of Heterogeneous Data Sources: Towards anIntegrated Information Framework for Service Technicians.* In Proceedings of the 13th International Conference on Semantic Systems, 2017. https://doi.org/10.1145/3132218.3132221

9. S. R. Bader, I. Grangel-González, M. Tasnim and S. Lohmann: *Structuring the Industry 4.0 Landscape.* In Proceedings of the 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2019. https://doi.org/10.1109/ETFA.2019.8869268

10. **S. R. Bader**, M. Vössing, C. Wolff, J. Walk and M. Maleshkova: *Supporting the Dispatching Process for Maintenance Technicians in Industry 4.0.* Wissensmanangement2017, 2017. http://ceur-ws.org/Vol-1821/W4_paper6.pdf

Workshop papers:

11. **S. R. Bader** and M. Maleshkova: *Virtual Representations for an iterative IoT Deployment.* In Companion Proceedings of the The Web Conference 2018, 2018. https://doi.org/10.1145/3184558.3191657

12. T. Käfer, **S. R. Bader**, L. Heling, R. Manke and A. Harth: *Exposing Internet of Things devices via REST and Linked Data Interfaces.* In Proceedings of the 2nd Workshop on Semantic Web Technologies and Internet of Things, 2017. http://ceur-ws.org/Vol-1930/paper-6.pdf

13. **S. R. Bader**, C. Wolff, M. Vössing and J.-P. Schmidt: *Towards Enabling Cyber-Physical Systems in Brownfield Environments.* In Proceedings of the International Conference on Exploring Service Science, Springer, 2018. https://doi.org/10.1007/978-3-030-00713-3_13

14. **S. R. Bader** and M. Maleshkova: *Towards Enforceable Usage Policies for Industry 4.0* In Joint Proceedings of the 1st International Workshop on Knowledge Graph Building and LASCAR, co-located with ESWC2019, 2019. http://ceur-ws.org/Vol-2489/paper8.pdf

15. **S. R. Bader** and M. Maleshkova: *Towards Integrated Data Control for Digital Twins in Industry 4.0* In Proceedings of the International Workshop on Semantic Digital Twins, 2020. http://ceur-ws.org/Vol-2615/paper1.pdf

16. S. R. Bader, A. Wolf and F. L. Keppmann: *Evaluation Environment for Linked Data Web Services.* In Joint Proceedings of SEMANTiCS 2017 Workshops, co-located with ESWC2017. http://ceur-ws.org/Vol-2063/salad-paper1.pdf

Book Chapters:

17. **S. R. Bader**, C. Azkan and L. Stojanovic: *Introduction to Smart Service Architectures.* In Smart Service Management - Design Guidelines and Best Practices, ed. by M. Maleshkova, N. Kühl and P. Jussen, Springer Nature, 2020. https://doi.org/10.1007/978-3-030-58182-4_9

18. **S. R. Bader**, C. Azkan and L. Stojanovic: *Reference Architecture Models for Smart Services.* In Smart Service Management - Design Guidelines and Best Practices, ed. by M. Maleshkova, N. Kühl and P. Jussen, Springer Nature, 2020. https://doi.org/10.1007/978-3-030-58182-4_10

19. **S. R. Bader** and L. Stojanovic: *Reference Architecture Models for Smart Service Networks.* In Smart Service Management - Design Guidelines and Best Practices, ed. by M. Maleshkova, N. Kühl and P. Jussen, Springer Nature, 2020. https://doi.org/10.1007/978-3-030-58182-4_10

20. L. Stojanovic and **S. R. Bader**: *Smart Services in the Physical World: Digital Twin.* In Smart Service Management - Design Guidelines and Best Practices, ed. by M. Maleshkova, N. Kühl and P. Jussen, Springer Nature, 2020. https://doi.org/10.1007/978-3-030-58182-4_12

21. B. Boss, **S. Bader**, A. Orzelski and M. Hoffmeister: *Verwaltungsschale.* In Handbuch Industrie 4.0: Produktion, Automatisierung und Logistik, ed. by M. ten Hompel, B. Vogel-Heuser and T. Bauernhansl, Springer Berlin Heidelberg, 2019. https://doi.org/10.1007/978-3-662-45537-1_139-1

Industry specifications and reference works:

22. B. Otto, S. Lohmann, S. Auer, G. Brost, J. Cirullies, A. Eitel, T. Ernst, C. Haas, M. Huber, C. Jung et al.: *IDS Reference Architecture Model.* version 3.0, Fraunhofer-Gesellschaft, Munich, 2019. https://www.fraunhofer.de/content/dam/zv/en/fields-of-research/industrial-data-space/IDS-Reference-Architecture-Model.pdf

23. B. Boss, M. Hoffmeister, T. Deppe, F. Pethig, **S. Bader**, E. Barnstedt et al.: *Details of the Asset Administration Shell Part 1, The exchange of information between partners in the value chain of Industrie 4.0* version 2.0, Plattform Industrie 4.0, ZVEI, 2019. https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html

24. A. Eitel, C. Jung, C. Kühnle, F. Bruckner, G. Brost, P. Birnstill, R. Nagel and **S. Bader**: *Usage Control in the International Data Spaces.* Position Paper, Version 2.0, International Data Spaces Association, 2019. https://internationaldataspaces.org/download/21053/

APPENDIX B

Scenario: Viewpoints of the IIoT

B.1 Business Viewpoint

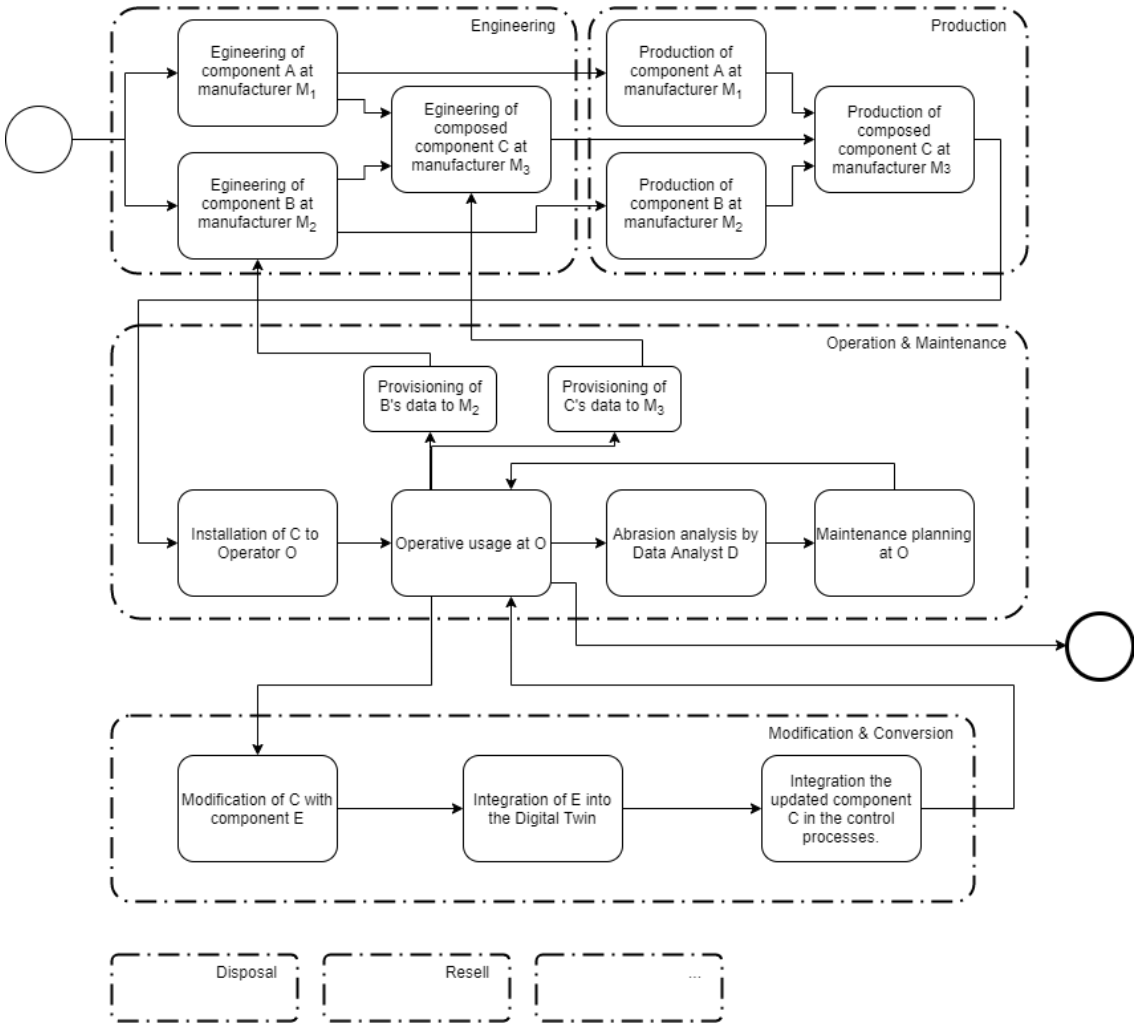


Figure B.1: Complete process of the Business Viewpoint.

B.2 Asset Viewpoint

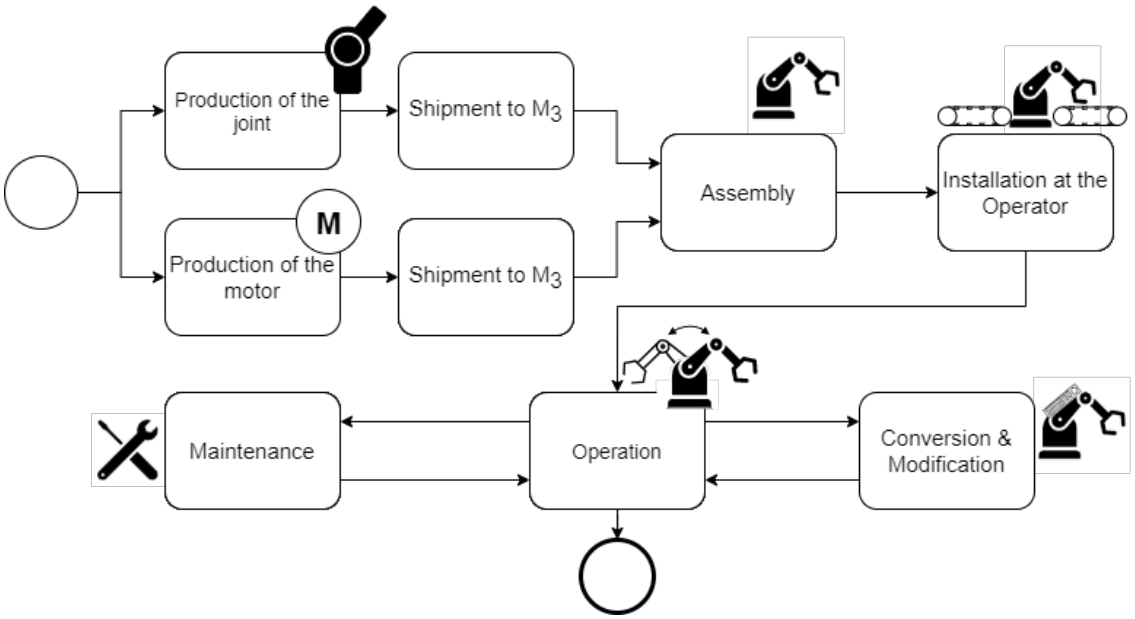


Figure B.2: Complete process of the Asset Viewpoint.

B.3 Sovereignty Viewpoint

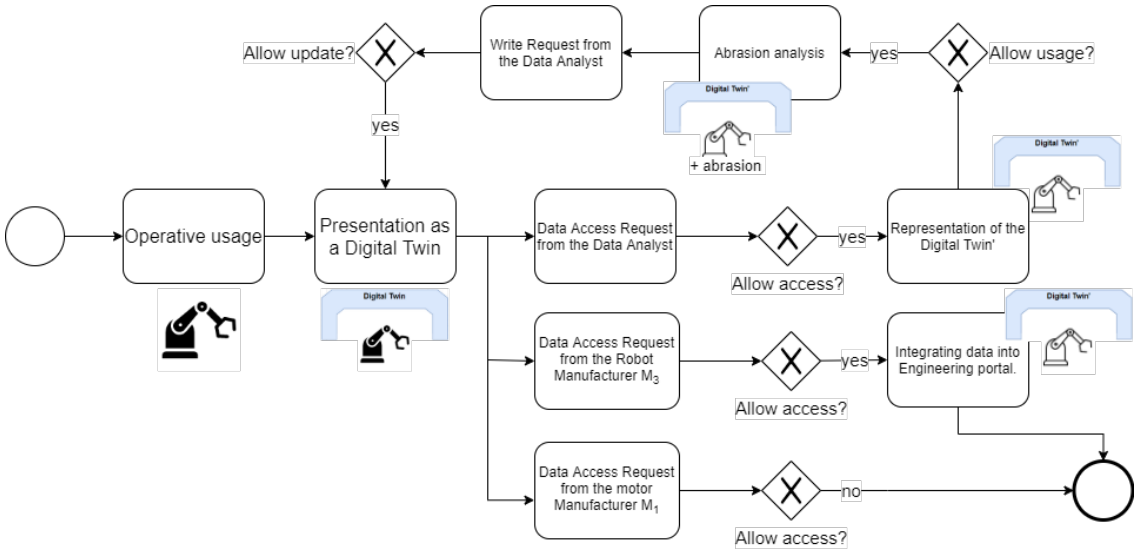


Figure B.3: The Sovereignty Viewpoint controls the interactions both at the access and usage phases.

B.4 Standardization Viewpoint

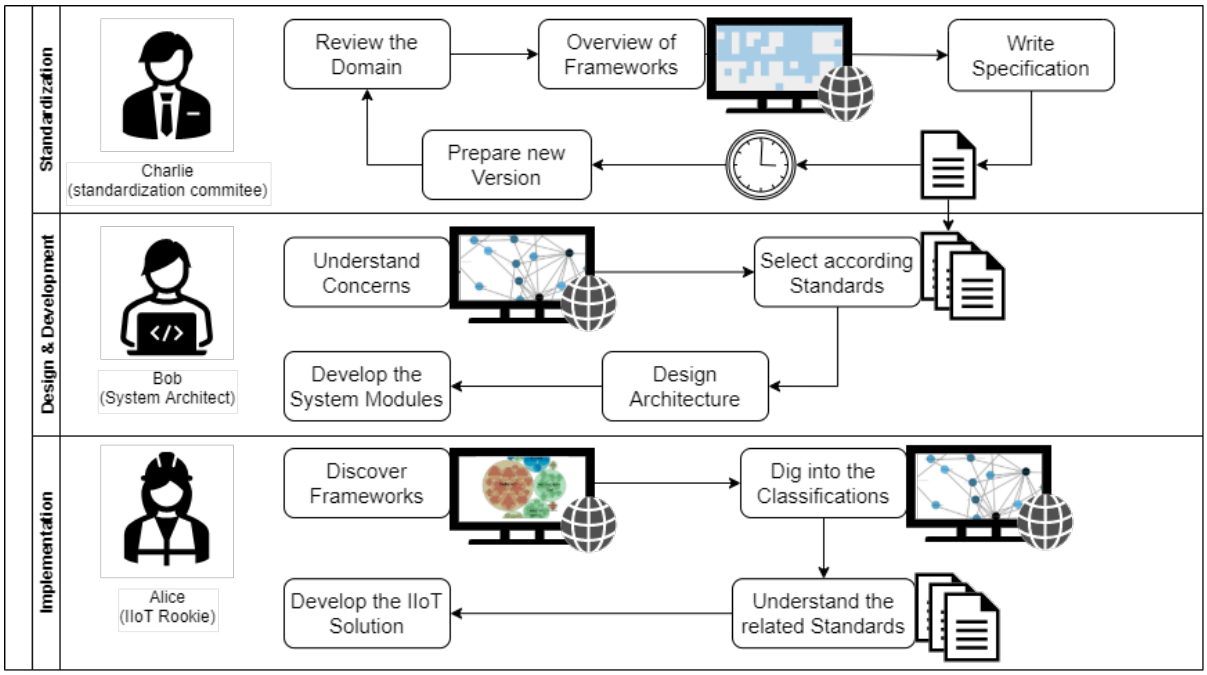


Figure B.4: The different stakeholders apply their own processes to discover the standardization landscape.

B.5 Interaction Viewpoint

```
@prefix http: <http://www.w3.org/2011/http#> .
@prefix httpm: <http://www.w3.org/2011/http-methods#> .
@prefix math: <http://www.w3.org/2000/10/swap/math#> .
@prefix ex: <http://example.org/> .

# request necessary information
{
    [] http:mthd httpm:GET ;
       http:requestURI <https://18.157.197.66:8443/assets/robot/motor/> .
}
{
    [] http:mthd httpm:GET ;
       http:requestURI <https://18.157.197.66:8443/assets/robot/joint/> .
}

# calculate abrasion state
{
    ?motor ex:numberOfRequests ?r .
    ?r math:notGreaterThan "10" .
} => {
  <#shaft> ex:hasAbrasion "0" .
} .
{
  ?motor ex:numberOfRequests ?r .
  ?r math:greaterThan "10" .
  ?r math:notGreaterThan "20" .

  (?r "10") math:difference ?diff .
  (?diff "3") math:exponentiation ?exp .
  (?exp "0.001") math:product ?abrasion .
} => {
  <#shaft> ex:hasAbrasion ?abrasion .
} .

{
  ?motor ex:numberOfRequests ?r .
  ?r math:greaterThan  "20" .
} => {
  <#shaft> ex:hasAbrasion "1" .
} .

# calculate current job
{
  ?x ex:hasStatistics ?job .
  ?job math:greaterThan "1000" .
} => {
  <#shaft> saref:accomplishes ?job .
} .

```

Figure B.5: Integrating data from external sources and deriving new information (complete)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX saref: <https://w3id.org/saref#>
PREFIX td: <https://www.w3.org/2019/wot/td#>
PREFIX ex: <http://example.org/>

CONSTRUCT {
  <https://18.157.197.66:8443/assets/robot/shaft/> rdf:type td:Thing .

  <https://18.157.197.66:8443/assets/robot/shaft/> :hasProgram
                    <https://18.157.197.66:8443/assets/robot/shaft/program> .

  <https://18.157.197.66:8443/assets/robot/shaft/> :hasQuery
                    <https://18.157.197.66:8443/assets/robot/shaft/query> .

  <https://18.157.197.66:8443/assets/robot/shaft/> saref:hasState
                  <https://18.157.197.66:8443/assets/robot/shaft/abrasion> .

  <https://18.157.197.66:8443/assets/robot/shaft/> saref:accomplishes ?job .

  <https://18.157.197.66:8443/assets/robot/shaft/abrasion> rdf:value ?abrasion .

  <https://18.157.197.66:8443/assets/robot/shaft/abrasion> rdf:type saref:State.

} WHERE {
  ?s ex:hasAbrasion> ?abrasion .
  ?t saref:accomplishes ?job .
}
```

Figure B.6: Creating the Digital Twin through a SPARQL Construct query (complete).

List of Figures

1.1 Engineering and production phases of the robot gripper arm and its components. 4
1.2 Integration of the robot arm and predictive maintenance based on data analytics. 4
1.3 The new sensor module is integrated into the physical Asset as well as into the Digital Twin. 5
1.4 The physical Assets are assembled, shipped, installed and reconfigured involving different stakeholders and owners. 5
1.5 Overview of the regarded challenges, contributions, and research questions of this thesis. 10
1.6 The research process has been partitioned into the depicted modules that impact and influence each other. 13
1.7 The IIoT is ranked as the fourth most relevant trending topic for consultants in 2018 [45]. 14

2.1 Timeline of selected Semantic Web developments from 2009 on. 18
2.2 Visual representation of the example from Listing 2.1 19
2.3 Information representation (inspired by Ogden and Richards [63], own illustration) . . 20
2.4 Popularity of terms as measured by Google Trends 22
2.5 Timeline of selected IIoT developments from 2006 on. 23
2.6 Reference Model as proposed by the WoT Community Group. 24
2.7 Layer structure of the Reference Architecture Model for the Industry 4.0 26
2.8 Industrial Internet Communication Framework 27
2.9 Popularity of interaction paradigms as measured by Google Trends. 28
2.10 Timeline of selected communication technology developments from 2006 on. 28
2.11 Popularity of structured data formats as measured by Google Trends. 29
2.12 Timeline of selected concepts of Digital Twins from 2011 on. 32
2.13 Timeline of selected developments for Usage and Access Control policies from 2004 on. 33

3.1 Core classes of the Asset Administration Shell Data Model 38
3.2 Graphical notation of a conversation between a client and a server to create exactly one resource 46

4.1 Contributions for RQ1. 57
4.2 Prototypical implementation of an IIoT-compliant dispatching module 61
4.3 HTTP wrapper component, presenting an Asset into the IIoT 61
4.4 Process steps through the provided modules. 66
4.5 Overview of the most important classes and properties of the SAAS 69
4.6 The Web resource containing the mappings. 70
4.7 Mapping times for the three Asset Administration Shells. 74
4.8 SAAS Reasoning duration. 75
4.9 Schema validation performance. 75
4.10 Partitions of the ontology by concern (pointing to standards reused). 77

4.11 IDS Reference Architecture with its main roles and interactions . . . 78
4.12 IDS core classes and their instances in the running example. . . . 80
4.13 Data Access and Usage decisions from the operator perspective. . . . 85
4.14 Stairway of Usage Policy Categories by required formalization. . . . 86
4.15 Basic model of a Digital Twin with Usage Policies according to the SAAS and IDS specifications. . . . 87

5.1 Contributions for RQ2. . . . 97
5.2 Simplified IIoT Layer Architecture . . . 99
5.3 Overview of Functional Requirements positioned in the IIoT layer model. . . . 101
5.4 Overview of Non-Functional Requirements by their location in the network architecture. 102
5.5 Overview of Protocol (left) and Data (right) Requirements, positioned in the communication stack. . . . 104
5.6 Online resource for the use cases. . . . 106
5.7 Illustration of the robot components . . . 107
5.8 Source code of the prototype implementation. . . . 107
5.9 Integrating data from external sources and deriving new information (simplified) . . . 108
5.10 Creating the Digital Twin through a SPARQL Construct query (simplified). . . . 109
5.11 Interaction model for Digital Twins . . . 110
5.12 Basic classes and relations to create a Digital Twin. . . . 111
5.13 The integration layer contains both network-enabled Assets and unconnected objects as Digital Twins. . . . 113
5.14 Integration Manager at the Edge for Digital Twins. . . . 115
5.15 Deployment time of n Docker container in seconds. . . . 117
5.16 The SOLIOT concept extends the Solid approach with IIoT endpoints and reserved resource representations of WoT Things. . . . 121
5.17 Example collaboration network. . . . 122
5.18 Representation of the Digital Twin through interlinked Containers, Resources, and ACL Files in a SOLIOT instance. . . . 132
5.19 Proof of concept for the SOLIOT concept . . . 135
5.20 Architecture Comparison. . . . 136
5.21 Information flow in the evaluation setting. . . . 137
5.22 Variance of measures presented in Tables 5.9 and 5.10 (in milliseconds, 10 clients requesting each instance). . . . 140

6.1 Contributions for RQ3. . . . 143
6.2 The three partitions of the I40KG as ontology modules. . . . 146
6.3 Contained entities: Standards (IEC 62714) link to standard classifications . . . 147
6.4 Concern hierarchy. . . . 147
6.5 Insertion Process: Three different sub-processes to create the I40KG content. . . . 148
6.6 IIoT Reference frameworks are described by publications and concerns. . . . 149
6.7 Layers (horizontal) and perspectives (vertical) of Digital Twins. . . . 154
6.8 Web service to analyze and discover IIoT Standards and Reference Frameworks. . . . 165
6.9 Visualizing reference models and their associated standards. . . . 166
6.10 Workflows of the IIoT Landscape for the different roles as depicted in the use cases. . . 167
6.11 Venn diagrams for reference frameworks and standards. . . . 167
6.12 Co-occurrence matrix between concerns and classification categories. . . . 168

6.13 Interoperability-related concerns. 169
6.14 Selected extract of trust- and business-related concerns. 170
6.15 Combined requirements and concerns for IIoT architectures. 172
6.16 Comparison of RAMI4.0, IDS-RAM, and IIRA by considered concerns. 174
6.17 Concerns addressed by the frameworks as a co-occurrence matrix. 175
6.18 RAMI4.0 Layers with related concerns. 178
6.19 IDS Classifications with related concerns. 178
6.20 Concerns as referenced by the Industrial Internet Reference Architecture (IIRA) 179

7.1 Repetition of the regarded challenges, contributions, and research questions as shown in Fig. 1.5. 181

B.1 Complete process of the Business Viewpoint. 223
B.2 Complete process of the Asset Viewpoint. 224
B.3 The Sovereignty Viewpoint controls the interactions both at the access and usage phases. 225
B.4 The different stakeholders apply their own processes to discover the standardization landscape. 226
B.5 Integrating data from external sources and deriving new information (complete) 227
B.6 Creating the Digital Twin through a SPARQL Construct query (complete). 228

List of Tables

4.1 Results of the SAAS mapping and RDF serialization. 73
4.2 Added triples by the different rule sets. 75
4.3 Key facts about the IDS Information Model and related resources. 79
4.4 Building blocks for Feature Dimensions (incomplete list). 91
4.5 Binary operators for Usage Control Constraints. Operators represent URIs with the *idsc:* namespace prefix. 93

5.1 Elementary resources of the proposed Web API. 117
5.2 Mapping of Functional Requirements. 123
5.3 Mapping of Non-functional Requirements. 124
5.4 Protocol Bindings: SOLIOT in MQTT and CoAP. 126
5.5 Interaction methods with CoAP 127
5.6 Protocol Headers 127
5.7 Interaction methods with MQTT 129
5.8 Data Representation Mapping. 130
5.9 Performance measures for one distinct Solid and SOLIOT instance with 1 and 10 clients requesting in parallel. 138
5.10 Performance measures for 10 instances each with 1 and 10 parallel requesting clients. . 139
5.11 Interaction sequence for the evaluation setting. 139

6.1 *I40KG* details: Relevant aspects of the *I40KG* and related resources. 146
6.2 Logical axioms as SWRL rules. 150
6.3 *I40KG* evaluation results. 153
6.4 Ranking of selected IIoT reference frameworks 171

Glossary

AAS Asset Administration Shell: The concept for a Digital Twin of the Plattform Industrie 4.0. 38, 58

ABox Assertion Box, the partition of an RDF ontology containing information about instances. 20, 51

Asset Anything tangible or intangible that has a value to an organization.. 19, 35, 44, 113

Automation ML Automation Markup Language: A XML-based data exchange format for engineering data. 6

Blockchain technology to store data in a decentralized manner, in so-called *blocks*, which reference earlier blocks using hash values and therefore provide a tamper-proof dataset with no central node. 50

BPMN Business Process Model and Notation, a modeling language to visualize business processes and aspects and to build the foundation for their translation into IT systems.. 3, 45

Brownfield A Brownfield is an ”existing industrial system targeted for new functionality without operational disruptions´´ [103]. 6

Class Set of Instances that share common characteristics. 19, 236

Cloud Computing concept to separate the application logic from the computing hardware. The Cloud appears as a virtual execution environment where services and users communicate over the internet. 27

CoAP Constraint Application Protocol: A lightweight IoT protocol for request/response patterns. 6, 135, 184

dc Dublin Core Elements, a vocabulary to express metadata and standardized by ISO 15836-1:2017. 20

dcterms Dublin Core Terms, a vocabulary to express metadata and standardized by ISO 15836-2:2019. 20

Digital Twin The combination of (mainly physical) Things with a virtual counterpart. 19, 31, 35, 38, 39, 45, 51, 58, 62, 64, 105, 112, 119, 141, 181, 187

General Data Protection Regulation the EU legal framework to regulate data privacy for person-related data. 48

graph Data structure of nodes and edges. Refers to a information model of URIs and typed links in this work. 20

Greenfield Terminology for a clean environment without requirements from legacy systems and proprietary interfaces. 6

HTTP Hypertext Transport Protocol, application-layer protocol for the World Wide Web. 6, 62

identifier Symbol or character sequence to name a thing. 19, 21

IDS International Data Spaces: Standardization initiative to provide self-sovereign data exchange. 40, 44

IIoT Industrial Internet of Things, the combination of IoT with the Industrial Internet. 2, 22, 35, 181

Instance Realization of a Classes. 19, 235, 236

IoT Internet of Things: Development and vision to equip any kind of Thing with internet adapters and to represent them in the internet. 1, 22

IRDI International Registration Data Identifier, idescheme ntifier as defined by ISO 29002. 39, 103

IRI Internationalized Resource Identifier: Resource Identifier as defined by RFC 3987. 19

JSON JavaScript Object Notation, data format to define the structure of machine-readable documents. 6, 18

JSON-LD RDF serialization of JSON. 18

Knowledge Graph machine-readable encoding of linked information in form of nodes and edges. See also Graph and ontology. 43

LDP Linked Data Platform, a W3C Recommendation for RESTful interactions with Linked Data (prefix: ldp, namespace: http://www.w3.org/TR/ldp/). 21

MQTT Message Queuing Telemetry Transport: A lightweight IoT protocol for publish/subscribe patterns. 6, 135, 159, 184

ontology Formal representation of a shared conceptualization. Used for RDF graphs in this work. 20, 43

OPC UA OPC Unified Architecture: A framework and datamodel for the machine-to-machine exchange of data. 6

OWL Web Ontology Language. 17, 20, 24

Plattform Industrie 4.0 A standardization initiative to enable interoperability in Industry 4.0. 44, 235

Policy Information Point XACML definition of a component which provides external information in a access or usage control framework. 49

Property Attribute of a Instance or a Class in RDF. 19

R2RML RDB to RDF Mapping Language, an RDF vocabulary to express liftings of non-RDF data to RDF graphs. 38, 237

RAMI4.0 Reference Architecture for Industrie 4.0, overview model proposed to align the Industrie 4.0 standardization activities.. 38

RDF The Resource Description Framework is a way to model resources in the Semantic Web. 17, 72, 236, 237

rdf Namespace for the RDF vocabulary (http://www.w3.org/1999/02/22-rdf-syntax-ns#). 20

RDF/XML RDF serialization in XML. 18

RDFS Namespace for the RDF Schema vocabulary (http://www.w3.org/2000/01/rdf-schema#). 20, 24, 72

Representation A document describing a Resource. 19

representation A (digital) document containing information about a resource. 48

Resource Anything that can be addressed with an identifier, mostly used for digital Things. 19, 35, 45, 237

RML RDF Mapping Language, an RDF vocabulary to express mappings of non-RDF data to RDF graphs, extending R2RML. 39

Semantic Web Research field for self-descriptive and machine-readable information modeling and processing in the Web. 1, 17, 42, 237

Semantic Web Stack Technology buidling blocks of the Semantic Web. 17

Serialization Series of bytes or any other exchangeable form of information.. 19

service An independent functionality, which encapsulates certain logic behind an API. 27

SHACL Shapes Constraint Language: A W3C Recommendation to model schema information in RDF. 17, 21

SKOS Simple Knowledge Organization System Namespace, a vocabulary to describe information concepts with RDF. 20

SOA Service-oriented Architectures, a design paradigm for IT applications in several, loosely-coupled services. 27, 47

SOAP protocol for the communication between Web Services. 28

Solid A Linked Data-based social network for self-controlled information provisioning. 118

TBox Terminology Box, the partition of an RDF ontology containing information about classes and properties. 20, 51

Thing Anything that can be addressed with an identifier. 235

UML Unified Modeling Language, a modeling language to visualize characteristics, relations, and activities of software modules or composed systems.. 3

URI Uniform Resource Identifier: Resource Identifier as defined by RFC 3986. 62, 103, 133

Usage Control mechanism to decide whether or not a user is allowed to use a remote resource after already receiving it. 183

USDL Unified Service Description Language, an extension to WSDL to describe business and legal aspects of Web Services. 48

UUID Universally Unique IDentifier: 128-bit long identifier format that is specified by RFC 4122. UUIDs are unique with a sufficiently high probability.. 100, 103

W3C World Wide Web Consortium: https://www.w3.org/. 17, 21, 31, 38, 238

W3C Recommendation Proposed standards published by the W3C. 17, 18, 237

Web World Wide Web, the application layer on top of the internet for hypermedia-driven documents (websites) and operated through HTTP interactions. 1, 18

XML Extensible Markup Language, a data format to define the structure of machine-readable documents. 6, 18