2019 International Conference on Networked Systems (NetSys)

Automated Processing of Security Requirements and Controls for a common Industrie 4.0 Use Case

Marco Ehrlich[1], Martin Gergeleit[2], Kostyantyn Simkin[2], Henning Trsek[3]

[1]inIT - Institute Industrial IT, OWL University of Applied Sciences, 32657 Lemgo, marco.ehrlich@hs-owl.de
[2]Hochschule RheinMain, 65195 Wiesbaden, martin.gergeleit@hs-rm.de, kostyantyn.simkin@student.hs-rm.de
[3]rt-solutions.de GmbH, 50968 Köln, trsek@rt-solutions.de

***Abstract*—Due to the dynamic nature of the Industrie 4.0, future production systems will be reconfigured frequently and as a part of the engineering process, new system configurations will be deployed automatically. In order to keep pace with this development, it will be required to achieve the needed security levels in an automated way and to reduce the current static procedures and manual efforts as much as possible. Therefore, the development and modelling of all cyber security related functionalities is needed. This paper describes an approach for such a modelling based on security requirements and levels of the international standard IEC 62443 part 3-3 and a system description based on OASIS TOSCA for the deployment. The approach is applied to an Industrie 4.0 use case scenario based on edge cloud computing and an evaluation is performed to demonstrate its feasibility.**

I. Introduction

The emerging Industrial Internet offers a great potential in assuring and extending Germany's position as a powerful location for production technology and innovations in the area of industrial automation. Future intelligent networks and innovative services are the basis to be able to link virtual and physical processes as a fundamental concept of Industry 4.0 (I4.0). Techniques from the Information Technology (IT) domain like (Edge) Cloud Computing and Software-Defined Networking (SDN) are becoming increasingly important also in Industrial Automation and Control Systems (IACS). Their introduction allows a fast and automated deployment of logically defined, virtualised architectures onto physical substrate hardware, which is an important factor for the promised adaptable manufacturing within the overall proclaimed I4.0 visions.

For the definition of these logical structures, such as computing nodes, network links, operating systems, and services, as well as their configuration parameters a number of concepts exist, like e.g. OpenFlow [1] and NETCONF [2] for SDNs or the Topology and Orchestration Specification for Cloud Applications (TOSCA) proposed by the Organization for the Advancement of Structured Information Standards (OASIS) [3] for cloud deployment and orchestration. However, these specifications do not explicitly address another intrinsically linked factor for the success of I4.0 systems: Security. Currently, each specified system configuration has to be checked manually by experts whether it fulfills the demanded security requirements of the given production environment. This imposes a high effort that leads to a trade-off between the dynamics of the adaption of production processes and their security. To avoid this in the future it will be required to achieve the needed security functionalities in an automated way and to reduce the nowadays static procedures and manual efforts as much as possible [4]. Therefore, the modelling of requirements and capabilities for all security related aspects is needed.

978-1-7281-0568-0/19/$31.00 ©2019 IEEE

The International Electrotechnical Commission (IEC) defines the security standard IEC 62443 containing procedures for implementing secure IACSs. It specifies four Security Levels (SLs) that are related to the skills, the resources, and the motivation of a possible adversary, ranging from an accidental intrusion (SL=1) up to a secret service style attack (SL=4). Especially IEC 62443 part 3-3 defines the technical requirements that have to be met by a complete system to conform to one of the SLs [5]. It distinguishes seven Foundational Requirements (FRs) covering the most important security objectives: Availability, integrity, and confidentiality. Each of the FRs is enhanced by more detailed technical System Requirements (SRs).

The methodology of the IEC 62443 suggests that a supplier of an industrial IT component specifies which FRs and SRs at which SL are covered by the security controls of the component under evaluation. Often a component has the ability to fulfill requirements at different SLs, even the higher ones, but there is a trade-off in terms of performance, resource consumption, or often usability and productivity. When a system integrator builds a system architecture out of these components, the integrator has to select and parametrise the components. Furthermore, the subsystems (zones) and connections (conduits) must be analyzed to assess whether they meet as a whole the FRs and SRs of the desired SL. However, with virtualisation techniques and automated deployment of components in a highly flexible production environment this process becomes very dynamic. Manual validation of conformance with security requirements will either become a time-consuming bottleneck or tends to be sloppy and introduces new security risks. A formal description of the security functionalities in terms of the IEC 62443 abstractions and an according description of the security controls of the deployed components will be beneficial here. This allows at least for basic sanity checks and can give hints on mismatches. Moreover, it allows to select the appropriate security configu-

Table I
COMPARISON OF PROPOSED INDUSTRIE 4.0 CONTENTS

Publisher & Source	Industrie 4.0 Proposal	Requirements related to Edge Cloud Computing
Research Survey [6]	4 Design Principles	Cloud computing as a top ten topic for I4.0 Urgent need for decentralized decision making processes
Plattform Industrie 4.0 [7]	9 Application Scenarios	Physical proximity to the shop floor using cloud-based deployments Support for „Plug And Produce for Field Devices" use case
German 5G ACIA [8]	3 Service Types 13 Use Cases	Ultra-Reliable and Low Latency Communication (URLLC) Massive Machine Type Communication (mMTC) Seamless communication from the field level to the edge cloud
EFFRA [9]	5 Key Priorities	Smart combination of cloud-, fog- and edge computing Ensurance of deployment scalability, flexibility, and resilience
VDI & VDE [10]	7 Use Cases	Cloud computing and big data analytics as main innovation drivers
Bitkom e.V. [11]	5 Application Scenarios	Information are locally available due to real-time computations
IIC SF TaskGroup [12]	9 Business Values	Need for storage and processing capabilities to act on data Analysis of manufacturing processes to improve operational efficiency
IEC & IEEE [13]	33 Use Cases	Redundant backbone networks resulting in a higher availability Minimize downtimes by analyzing monitoring data locally
IC4F Research Project [14]	4 Use Case Clusters	Hierarchical cloud infrastructures support distributed applications Safe, secure, and reliable processes need to be dynamically deployed
FIND Research Project [15]	4 Use Case Classes	Throughput and latency constraints have to be taken into account Allocation of sufficient resources to the needed application components

ration parameters of components for deployment and check for validation of complete systems.

The approach proposing a formal notation in accordance to the well-accepted international security standard IEC 62443 was already evaluated in former contributions and publications regarding various technologies (OPC UA, ZigBee, and LTE) and use cases (Manufacturing Execution System (MES) & Augmented Reality (AR) Worker) [4], [16]. With the further development of the model checking concept, a more general use case is required in order to check the usability and applicability of the whole approach. Therefore, this paper assesses the available and already defined I4.0 use cases to check for common requirements regarding edge cloud computing, which is generally seen as fundamental basis for the needed agile and automated deployment in future I4.0 systems [17].

Table I shows the result of the I4.0 use case elaboration and their evaluation regarding common edge cloud requirements. The assessed use case definitions from various institutions speak a common language: The local processing of production data is essential for future industrial applications in order to meet the increasing requirements regarding latency, throughput, and general decision speed. Decentralized computation algorithms directly on the shop floor to increase the operational efficiency of machines show the urgent need for secure and dynamic edge cloud deployment processes. Nevertheless, this demands currently do not fit to the security concepts available for the industrial domain. Therefore, the proposed approach will be evaluated by using the common requirements regarding edge cloud utilization to state it's general usability and applicability.

The reminder of the paper is organized as follows: First the specification of IEC 62443 parameters is explained and an introduction to TOSCA is given. Then the approach for checking such models against given security requirements is elaborated. Finally, the automated tooling process of the approach is evaluated regarding the specified edge cloud use case.

II. ESSENTIAL BACKGROUND

A. IEC 62443 Security Standard

In general, several approaches from various domains, such as telecommunication, home computing, multimedia, the industrial automation, or research, are available for the modelling of security functionalities [4], [18]. Nevertheless, the furthest developed approach is the IEC 62443 standard, which is originally suited for the security of IACSs inside the industrial automation domain. This standard gained a lot more attention during the past years and was developed into the most important one covering the fundamental issues of industrial communication networks and their intrinsically linked topic of security. The standard classifies threats or attacks regarding information security in a four stage scaling system in which each stage is stated as a Security Level (SL). The SLs are designed with a focus on the attacker's motivation, skills, and resources. SL 0 means no security requirements at all. The various security goals are

expressed in seven FRs [5]. These FRs cover the commonly identified dimensions in security, which is shown in Table II.

Table II
FOUNDATIONAL REQUIREMENTS (FRS) IN THE IEC 62443

FR No.	Topic
FR 1	Identification and Authentication Control (IAC)
FR 2	Use Control (UC)
FR 3	System Integrity (SI)
FR 4	Data Confidentiality (DC)
FR 5	Restricted Data Flow (RDF)
FR 6	Timely Response to Events (TRE)
FR 7	Resource Availability (RA)

As mentioned inside the introduction inside the IEC 62443 there are four SLs available, which are summarised in Table III. **SL 1** delivers protection against casual or coincidental violation, **SL 2** provides protection against intentional violation using simple means, **SL 3** gives protection against intentional violation using sophisticated means, and finally **SL 4** is described by protection against intentional violation using sophisticated means with extended resources.

Table III
SECURITY LEVELS (SLS) IN THE IEC 62443

SL	motivation	means	resources
1	casual/coincidental	no special	standard
2	intentional	simple	standard
3	intentional	sophisticated	high
4	intentional	sophisticated	extended

Further the IEC 62443 standard defines for each FR within so-called System Requirements (SRs), which specific technical requirements have to be fulfilled to achieve a certain SL. Generally there is a differentiation between three characteristics of SL [5]:

- **Target Security Level (SL-T):** Desired level of security for a particular system during conception phase
- **Achieved Security Level (SL-A):** Actual level of security for a particular system after finished setup
- **Capability Security Level (SL-C):** Security level that the chosen components in a setup can provide

The basis of the given procedure is always a risk analysis. The goal is to identify risks and their impact based on a segmentation of the system into cells (zones) and communication channels (conduits). The subdivision of the network is very useful to limit possible damages to a certain cell. The protection objectives of the various cells can be quite different. The result of this exercise is an architecture divided into zones and conduits and the definition of the SL-T vector for each of these units. In response to the above, the system integrator or the plant engineer configures an automation solution based on available components or systems, which inherit their own single SL-Cs. It is tried to achieve the SL-T of the zones and conduits as far as possible. If this is not sufficient, it takes additional measures, so called compensating countermeasures, which increase the protection level. If the accumulated SL-A protection level cannot meet the requirements from the SL-T vector, the operator must accept the remaining risks or compensate through further measures within his area of responsibility.

B. Resource Deployment with TOSCA

TOSCA [19], [20] is an OASIS standard language that has been developed to simplify the definition and deployment of services in a cloud environment. It allows to describe the topology of cloud based services, their hardware and software components, and the processes that manage them. TOSCA uses an object-oriented approach to model "topologies" consisting of "nodes" (all kind of components: computing nodes, networks, and also software services), their attributes, their relationships (like "runs on" or "linked to"), their capabilities, and requirements. The TOSCA standard also contains a set of basic types that are usually used to compose cloud services [20]. The classic way of defining TOSCA models is to write a declaration in a YAML language dialect given by the OASIS standard.

While not initially targeted towards IACS, TOSCA is generic enough to describe any kind of topologies and the language is also open for the definition of new types – either derived from the standard types or in a separate type hierarchy. TOSCA, in contrast to other modelling languages (like e.g. UML), has the advantage, that it has been developed with explicitly cloud computing in mind. Thus, there are also tools that can automatically deploy and maintain TOSCA models in a cloud environment. As IACS are also moving towards cloud infrastructures, even if it is "only" a local edge cloud, it becomes obvious, that TOSCA is also a candidate for modelling these kind of automation systems.

However, the current TOSCA standard has no means to handle security issues explicitly. In [16] the basic ideas of how to enhance a TOSCA specification in order to become security aware has been presented. The approach uses the FR and SL abstractions of the IEC 62443 standard. The TOSCA language capabilities are used to specify the supported SL-C vectors of the components. A model checker can now verify whether an actual system topology meets the globally defined security requirements – provided in an SL-T vector – by comparing the all the SL-Cs. In this paper, we will extend these ideas by introducing different zones with distinct security requirement vectors and by introducing a new security-related relationship that will also allow for dynamic selection of nodes, the so-called node filtering in TOSCA. The paper will also focus on a new implementation of the checking mechanisms based on the "puccini" TOSCA compiler (https://github.com/tliron/puccini).

III. TOSCA SECURITY MODELLING

In [16] we presented an approach of modelling IEC 62443 security vectors in TOSCA as capabilities of nodes and global requirements. As one global set of requirements is quite inflexible for larger applications, we are now introducing multiple security domains representing areas with different security requirements

within one topology. Also, due to the inability of the formerly used TOSCA compiler to perform checks and to do an automatic matching of capabilities and requirements based on numeric values (a missing feature in the implementation, not in the standard), we propose a slightly modified approach here with a different compiler, the "puccini" TOSCA tool. This compiler environment is still being actively developed and provides some additional features, including the ability to define JavaScript functions within TOSCA YAML-files. This has been introduced to to provide better orchestration integration, e.g. for generating deployment code from the YAML source.

In our approach a function is used for checking during the compilation, whether a component can fulfill all security requirements. However, the approach is still based on the same fundamental concepts as in [16] and still fully compatible to the TOSCA standard:

- Standard TOSCA node types representing the components of an application are extended to become "secure" node types. Each "secure" node type includes an additional vector with seven dimensions representing the security capabilities of a component according to the FRs of the ICE 62443 security standard. For technical reasons (it enables automated checking of numerical values) these security vectors are now defined as node properties instead of TOSCA capabilities.
- Each dimension has a value from 0 to 4 representing the SLs from the given IEC 62443 standard
- A system designer has to evaluate the security controls a node implements and has to assign the appropriate SL-C values to the nodes of an actual application topology.

These basic ideas have been now extended by three new concepts to enhance the proposed approach:

- A "SecurityDomain" node specifies an area of equal security requirements. Each security domain defines a vector with the seven dimensions representing the security requirements of this area as a result of a risk analysis. Such a node is abstract in a sense that there is no component deployed in running system, it is used during modelling and checking only. In a simple case the complete application topology belongs to one security domain. In a larger setup an application might span over more than one security domain, e.g. corresponding to different zones and conduits possibly belonging to different organizational units.
- A new "BelongsTo" relationship type that specifies that a certain "secure" node belongs to a "SecurityDomain" node. This is used to check, whether this node can fulfill the SLs required in that domain. A node may belong to more than one security domain, e.g. when a gateway node connects two network segments in two different security domains it typically has to fulfill the requirements of both domains.
- A "validate_req" function written in JavaScript, that does the checking. It is used for checking during the compilation whether a component has a valid "BelongsTo" relationship to a "SecurityDomain" node. As the code for applying this function is somewhat lengthy but also completely generic, a macro named "*CHECK" is used within the node declarations.

An example of a "SecureCompute" type is shown in Listing 1. It is derived from the standard TOSCA "Compute" type and extends it by two elements: a "sec_vector" property representing the 7-dimensional security capabilities of this network node and the requirement of at least one relationship named "belongs" of type "BelongsTo", i.e. a link to a security domain.

Listing 1. Declaration of a "SecureCompute" type

```
SecureCompute:
  derived_from: tosca.nodes.Compute
  properties:
    sec_vector:
      type: list
      entry_schema:
        description: Security Vector
        type: integer
        constraints:
          - in_range: [0,4]
      constraints:
        - min_length: 7
        - max_length: 7
  requirements:
    - belongs:
        capability: iec_62443
        occurrences: [1,UNBOUNDED]
        relationship:
          type: BelongsTo
```

In Listing 2 the declaration of the "BelongsTo" relationship type is shown. Its "satifies" property vector is set by the "validate_req" function as the last step of the compilation and it indicates, whether a linked node can fulfill the requirements of the linked security domain in each of the 7 dimensions.

Listing 2. Declaration of the "BelongsTo" relationship

```
BelongsTo:
  properties:
    satisfies:
      type: list
      entry_schema:
        type: boolean
      constraints:
        - min_length: 7
        - max_length: 7
      default: [true, true, true,
                true, true, true, true]
```

These declarations are written once and are provided in an include-file for usage in an actual topology description. This leads to a fairly simple notation in an actual topology definition as depicted below in Listing 3 for a sample topology.

IV. Evaluation

A. Use Case Specification

The use case analysis and elaboration of common requirements regarding edge cloud computing, which was conducted

in Section I, can be used for the evaluation of the proposed approach. Therefore, Figure 1 shows the unified use case, which is considered as the basis for the dynamic deployment of industrial applications using the proposed TOSCA approach in combination with the modelling of security following the IEC 62443. For the purpose of this paper we have chosen a simplified scenario with a limited set of components, where the reader can verify the constraints with limited effort. The more components and domains are involved the more constraints have to be checked each time after updating any part of the system configuration. Thus, an integrated and automated checking system can exploit its strengths even more when the architecture becomes larger, especially as the inherent complexity of the checking problem itself scales linearly. In [16] e.g. we have started modelling a more complex use case from the IC4F[14] project with 8 network segments and 7 services.

The sample scenario in this paper describes a high-level I4.0 architecture consisting of IT and production environments interconnected from one common factory. The industrial network having increased requirements regarding real-time capabilities, timing, and determinism of events is supported with a locally connected edge cloud for lower latency, higher performance, and availability of data. A typical application deployed there would be a soft Programmable Logic Controller (PLC) as a control component implemented as software. Modern frameworks like Docker enable a container-based deployment of these applications to ensure usability and independence from installed resources. In contrast, the office network contains workstations for the operator and a private cloud implementation hosting applications, such as monitoring, diagnostics, engineering tools, or user interfaces, with lower requirements, which can be deployed on the servers inside the IT environment. In the simplified example a data base is deployed on a private cloud server. The IT and the production environments are not only two different network segments but also two different security domains with slightly different SL-Ts, shown as grey vectors in the left upper corner.

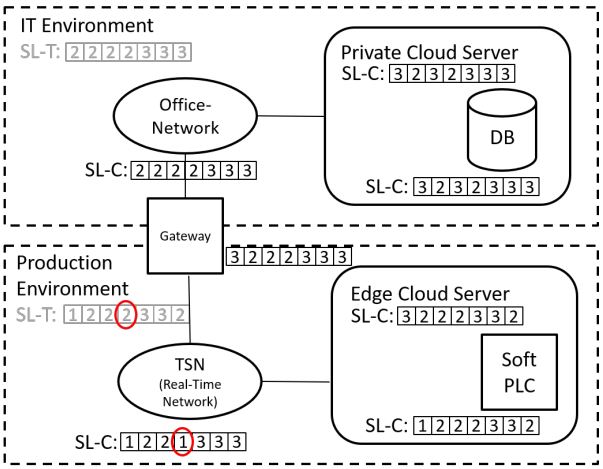

Figure 1. Use Case Scenario for automated Process Evaluation

The components are placed in the domain they belong to. The gateway router as a link between the two segments belongs to both domains. The component's SL-Cs are shown below each node. These values are given as samples here. In a real application they are resulting from a detailed evaluation of the security capabilities of the used components and their implementations. This evaluation is a non-trivial task, but it has to be done only once and can be re-used in other topologies. Discussing all values would be far beyond the scope of this paper. However, the capabilities of the Time-Sensitive Networking (TSN) real-time network [21] are described as an example. They are summarized in Table IV. It briefly describes the security mechanism, which led to the corresponding capability level. For instance, the data confidentiality (DC) of TSN is 1, because the protocol does not provide any encryption in its standard version.

Table IV
SECURITY VECTOR EXAMPLE: TSN [21]

FR	Name	SL	TSN Security Mechanisms
1	IAC	1	No authentication
2	UC	2	Network access control according to 802.1x
3	SI	2	Ingress filtering and policing
4	DC	1	No encryption
5	RDF	3	Logical separation of traffic flows
6	TRE	3	Protection of clock synchronization
7	RA	3	Protocols for high-availability

B. Checking of Security Modelling

The specification of the sample topology with all security vectors attributed results in a structured TOSCA YAML file of about 275 line of code. Listing 3 shows a snippet of this description file. It contains the definition of the security domain "Production_Environment" with its SL-T vector and two nodes with their SLCs belonging to that domain: the "Edge_Cloud_Server" and the "Real_Time_Network_TSN". The "*CHECK" macro in the two nodes is used to the validate the conformance of the vectors. It must be added for each "belongs" relationship.

Listing 3. Parts of the definition of the sample topology

```
Production_Environment:
  type: SecurityDomain
  properties:
    req_sec_vector: [1,2,2,2,3,3,2]

Real_Time_Network_TSN:
  type: SecureNetwork
  properties:
    sec_vector: [1,2,2,1,3,3,3]
  requirements:
    - belongs:
        node: Production_Environment
        relationship: *CHECK
...
```

It can be translated using the puccini-TOSCA compiler into an intermediate format that contains all the relevant information for deploying the model and for further processing e.g. for

giving a graphical overview over the model's structure. During the compilation the check against the SL-Ts of the application's security domain are performed, i.e., it is checked for all components (compute, network, and service nodes), whether their SL-C vectors are greater or equal in all dimensions than the maximum of the required SL-T values. The check of the sample scenario in Figure 1 reveals and the TOSCA compiler warns that there is one mismatch between the requirements and the capabilities of a component: namely the required SL2 for FR4 (Data Confidentiality) in the production environment cannot be provided by the real-time network, depicted by the highlighted vector components in Figure 1. In such a case either additional security controls have to be implemented, e.g. enabling general frame encryption on this network. Alternatively it can be stated that the potential risk is acceptable. For instance, if the physical protection of the environment is strong enough to ensure confidentiality.

V. Conclusion

Adaptable manufacturing systems belong to the core concepts of Industrie 4.0. In order to achieve the required level of flexibility for them, a fast and automated deployment of logically defined, virtualized and networked architectures will be an important factor. In this context, security aspects are of utmost importance. However, security is usually handled in a very static way, which contradicts to the needed flexibility and leads to the demand for more flexible approaches for security.

Hence, the paper introduces a simple modelling language that is based on the TOSCA Simple Profile in YAML and extends it towards security controls. The extension consists of security capabilities as additional attributes of components. The idea of this paper is the approach, that these security capabilities not only describe security configuration options and additional security-related functionalities, but also their mapping onto IEC 62443 FRs/SRs and the provided SLs.

After the modelling approach is introduced, the paper describes an architecture of an edge cloud use case scenario and the corresponding vectors of security requirements are modeled. The performed evaluation based on the new implementation demonstrates that the presented approach is suitable for an automated deployment of innovative industrial architectures with regard to security.

Future work will investigate if the current level of granularity of the modelled requirements and capabilities is sufficient in all cases. An enhancement by the proper usage of the FR7 "Resource Availability" of the IEC 62443 standard could be beneficial. Also, it will be investigated, how the described methodology can be integrated into a general architectural model like the "Industrial Reference Architecture (IRefA)" developed in the IC4F project. If a set of predefined building blocks including their security classifications were available, this would simplify the process of security modelling a lot.

Acknowledgments

This work was funded by the German Federal Ministry for Economic Affairs and Energy with the research project "Industrial Communication for Factories" (IC4F) and the German Federal Ministry of Education and Research with the research project "Future Industrial Network Architecture" (FIND).

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks", 2008, SIGCOMM Comput. Commun. Rev.

[2] M. Enns, R. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network Configuration Protocol (NETCONF) - RFC6241", 2011, Online: Internet Engineering Task Force (IETF).

[3] Organization for the Advancement of Structured Information Standards, "OASIS Topology and Orchestration Specification for Cloud Applications - ver.1.0", 2013.

[4] M. Ehrlich, L. Wisniewski, H. Trsek, D. Mahrenholz, and J. Jasperneite, "Automatic mapping of cyber security requirements to support network slicing in software-defined networks", 2017, 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus.

[5] International Electrotechnical Comission, "Automatic mapping of cyber security requirements to support network slicing in software-defined networks", 2015, IEC 62443-3-3: System security requirements and security levels.

[6] M. Hermann, T. Pentek, and B. Otto, "Design Principles for Industrie 4.0 Scenarios", 2016, 49th Hawaii International Conference on System Sciences (HICSS), Koloa, USA.

[7] Plattform Industrie 4.0, "Industrie 4.0 Plug-and-Produce for Adaptable Factories: Example Use Case Definition, Models, and Implementation", 2017.

[8] 5G Alliance for Connected Industries and Automation, "Whtie Paper: 5G for Connected Industries and Automation", 2018.

[9] European Factories of the Future Research Association, "Factories 4.0 and Beyond", 2016.

[10] VDI/VDE-Gesellschaft: Mess- und Automatisierungstechnik, "Arbeitswelt Industrie 4.0", 2016.

[11] S. Zehl, "Anwendungsszenarien für Industrie 4.0", 2018.

[12] Industrial Internet Consortium - Smart Factory Task Group, "Smart Factory Applications in Discrete Manufacturing", 2017.

[13] R. Belliardi, J. Dorr, T. Enzinger, J. Farkas, M. Hantel, M. Riegel, M.-P. Stanica, G. Steindl, R. Wamßer, and S. Zuponcic, "Use Cases IEC/IEEE 60802", 2018.

[14] Industrial Communication for Factories Consortium, "Building Blocks for a Secure Real-Time Communication and Computing Infrastructure for Industry 4.0", 2018.

[15] Future Industrial Network Architecture Consortium, "Use Cases and Requirements Specification", 2017.

[16] M. Gergeleit, H. Trsek, T. Eisert, and M. Ehrlich, "Modeling Security Requirements and Controls for an Automated Deployment of Industrial IT Systems", 2018, 9. Jahreskolloquium Kommunikation in der Automation (KommA), Lemgo, Germany.

[17] Omid Givehchi, Jahanzaib Imtiaz, Henning Trsek, and Jürgen Jasperneite, "Control-as-a-Service from the Cloud: A Case Study for using Virtualized PLCs", 2014, 10th IEEE Workshop on Factory Communication Systems (WFCS), Toulouse, France.

[18] M. Ehrlich, L. Wisniewski, H. Trsek, and J. Jasperneite, "Modelling and automatic mapping of cyber security requirements for industrial applications: survey, problem exposition, and research focus", 2017, 14th IEEE International Workshop on Factory Communication Systems (WFCS), Imperia, Italy.

[19] T. Binz, G. Breiter, F. Leyman, and T. Spatzier, "Portable Cloud Services Using TOSCA", 2012, IEEE Internet Computing, Volume: 16, Issue:3.

[20] Organization for the Advancement of Structured Information Standards, "OASIS TOSCA Simple Profile in YAML Version 1.1", 2018, Online: http://docs.oasis-open.org/tosca/TOSCA/v1.0/TOSCA-v1.0.pdf.

[21] W. Steiner, P. G. Peón, M. Gutiérrez, A. Mehmed, G. Rodriguez-Navas, E. Lisova, and F. Pozo, "Next Generation Real-Time Networks Based on IT Technologies", 2016, IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, Germany.