IoT4CPS – Trustworthy IoT for CPS

FFG - ICT of the Future
Project No. 863129

Deliverable D2.1
Consolidated state-of-the-art report

The IoT4CPS Consortium:
AIT – Austrian Institute of Technology GmbH
AVL – AVL List GmbH
DUK – Donau-Universität Krems
IFAT – Infineon Technologies Austria AG
JKU – JK Universität Linz / Institute for Pervasive Computing
JR – Joanneum Research Forschungsgesellschaft mbH
NOKIA – Nokia Solutions and Networks Österreich GmbH
NXP – NXP Semiconductors Austria GmbH
SBA – SBA Research GmbH
SRFG – Salzburg Research Forschungsgesellschaft
SCCH – Software Competence Center Hagenberg GmbH
SAGÖ – Siemens AG Österreich
TTTech – TTTech Computertechnik AG
IAIK – TU Graz / Institute for Applied Information Processing and Communications
ITI – TU Graz / Institute for Technical Informatics
TUW – TU Wien / Institute of Computer Engineering
XNET – X-Net Services GmbH

For more information on this document or the IoT4CPS project, please contact:
Mario Drobics, AIT Austrian Institute of Technology, mario.drobics@ait.ac.at

Document Control

Title: Consolidated state-of-the-art report
Type: public
Editor(s): Edin Arnautovic (TTTech)
E-mail: edin.arnautovic@tttech.com
Author(s): Denise Ratasich (TUW), Christoph Schmittner (AIT), Mario Drobics (AIT), Michael Jerne (NXP), Edin Arnautovic (TTTech), Violeta Damjanovic-Behrendt (SRFG), Omar Veledar (AVL)
Doc ID: IoT4CPS-D2.1

Amendment History

Version	Date	Author	Description/Comments
Version	Date	Author	Description/Comments
V0.1	03.09.2018		Initial version prepared
V0.2	10.10.2018	Edin Arnautovic	Integration of contributions from TUW, NXP, AIT, SRFG,
V0.3	11.10.2018	Edin Arnautovic	TTTech contribution (SoA Automated driving)
V0.4	11.10.2018	Denise Ratasich	Intro, references and summary for safety and security methods
V0.5	12.10.2018	Edin Arnautovic	Editing
V0.6	14.10.2018	Christoph Schmittner	Security Methods
V.07	17.10.2018	Denise Ratasich	Safety and security methods – Refinement
V0.8	23.10.2018	Edin Arnautovic	Final editing for review
V0.9	29.10.2018	Omar Veledar, Violeta Damjanovic-Behrendt	Review contributions
V1.0	30.10.2018	Edin Arnautovic	Final
V1.1	07.12.2018	Edin Arnautovic	Restructuring after additional feedback
V1.2	07.01.2019	Andreas Martin, Mario Drobics	Final editing

The IoT4CPS project is partially funded by the "ICT of the Future" Program of the FFG and the BMVIT.

Federal Ministry
Republic of Austria
Transport, Innovation
and Technology

FFG
Forschung wirkt.

Content

Abbreviations

AI	Artificial Intelligence
ASIL	Automotive Safety Integrity Level
CIM	Computer Integrated Manufacturing
CPSs	Cyber Physical Systems
CRM	Customer Relationship Management
DoS	Denial of service
DTE	Digital Twin Environment
DTI	Digital Twin Instance
DTP	Digital Twin Prototype
ECU	Electronic Control Unit
ERP	Enterprise Resource Planning
eTVRA	ETSI Threat Vulnerability and Risk Analysis
FMEA	Failure mode and Effect Analysis
HARA	Hazard and risk analysis
HMI	Human Machine Interface
ICS	Industrial Control System
IIoT	Industrial Internet of Things
IIRA	Industrial Internet Reference Architecture
IoT	Internet of Things
ISO	International Organization for Standardization
KNN	K-Nearest Neighbour
KPI	Key Performance Indicators
M2M	Machine to Machine
ML	Machine Learning
NIS	Network and Information Systems
PLC	Programmable Logic Control
PLM	Product Lifecycle Management
PSL	Process Specification Language
RAMI 4.0	Reference Architecture Model Industrie 4.0
RFID	Radio Frequency Identification
RMS	Reconfigurable Manufacturing Systems
SCADA	Supervisory Control And Data Acquisition
STRIDE	Spoofing of identity, Tampering, Repudiation, Information disclosure, D.o.S and Elevation of privilege
SVN	Support Vector Machine
WSN	Wireless Sensor Networks

1. Introduction

Within the complexity of nowadays' digital environments, both ICT and – more recently – the Internet of Things (IoT) have a significant impact on almost every societal area. "IoT" is a general term describing a system which incorporates inter-connected devices with sensors and actuators, reachable almost instantaneously through the internet from any location and any device worldwide. Nevertheless, this unrestricted reachability as well as the inter-connectedness in such smart infrastructures are accompanied by new IoT security-related threats such as the misuse of information. Since society is increasingly reliant on smart infrastructures in private as well as in professional life (e.g. smart-homes, -cities or -mobility), security, privacy and safety of IoT-based systems are crucial factors to be addressed.

Modern cyber-physical systems (CPS) [LS10], [RLSS10], [Raj12], [CBF+ 16] like (semi-) autonomous cars, wireless sensor networks or medical devices which monitor and control the physical world are increasingly being connected via the IoT. Therefore, safety is becoming deeply intertwined with security ("if it's not secure it's not safe" [BNS13]) such that security vulnerabilities provide attackers means to manipulate the CPS or cause fatal accidents. Moreover, faults and vulnerabilities are more probable due to the heterogeneity, elasticity or dynamicity, openness (internet access) and size of the network.

The resilience of the system – that is the ability that the service delivery (or functionality) that can justifiably be trusted should persists when facing changes [Lap08] – shall therefore be ensured throughout its life-cycle. To meet the related requirements of system-resilience, IoT solutions need to be based on reliable and robust technology. Furthermore, the establishment of IoT-specific standards to overcome certain challenges concerning security, privacy and safety play a key role.

This deliverable is structured as follows. Section 2 deals with general technological aspects of safe and secure IoT. Firstly, it provides the state of the art related to safety and security design and methods including detection and diagnosis, as well as recovery and mitigation. Secondly, Section 2 gives an overview of Security Verification and Analysis. Analysis Methods and Frameworks are significant parts thereof. Third part of Section 2 offers a background on Life-cycle data management for IIoT systems based on Reference Architectures such as RAMI 4.0. Section 3 focuses on industrial applications: automated driving and Industry 4.0 and presents the state of the art and practice in these domains. Section 4 concludes the deliverable.

2. Overview of the Industrial State of Practice

The evolution of the traditional industrial systems towards Industry 4.0 and Smart Manufacturing [KLCK16] makes manufacturing systems more adaptive and adds flexible decision-making mechanisms, self-awareness and self-optimization features to their core components and services [LFKF14][MOEL16]. The idea of Smart Manufacturing evolved from (i) Computer Integrated Manufacturing (CIM) in the 1980's, (ii) Reconfigurable Manufacturing Systems (RMS) [KHJM99][KOSH10], (iii) the Smart Factory initiative based on the IoT and embedded intelligence [ZUEH10], and (iv) the Ubiquitous Factory concept and its reference architecture encompassing the following four layers: the shop floor, the application system, the information infrastructure, and the lifecycle layer [YOSS12].

Nowadays, research in Industry 4.0 and Smart Manufacturing discusses technologies such as CPSs, IoT, Web of Things (WoT), Big Data and Analytics, edge computing, cloud computing, smart sensors, Digital Twins, and Artificial Intelligence (AI). For example, the role of IoT in digital manufacturing is to create and collect real-time sensor data that can be exchanged through the Web [SABA00]. As such, IoT can be used to help remotely control devices equipped with sensors, across network infrastructures, which results in higher efficiency and accuracy of industrial systems. Edge and cloud computing technologies enable the analysis and correlation of data; AI technologies enable data mining and the creation of added value through knowledge discovery, while Big Data technologies provide systematic analyses of a variety of data generated along the entire product lifecycle, supporting a rapid decision making and improving productivity of manufacturing systems [DEPB12][LEKY14].

CPSs. The first Industry 4.0 reference model [KAWJ13], introduces the CPS as the key technology that adds intelligence to traditional production processes [LEE08][LEBK15][JAZD14]. CPSs integrate computational paradigms with the physical processes [LEE08] and create capabilities of the intelligent manufacturing systems, e.g. reliability, interoperability, predictability and tracking [MONO14]. Monostori defined the concept of Cyber Physical Production System (CPPS) that is a group of "autonomous and cooperative elements and subsystems that are getting into connection with each other in situation dependent ways, on and across all levels of production, from processes through machines up to production and logistics networks" [MONO14]. The concept of CPPS is today used as a synonym for the Smart Factory, emphasizing scalable and modular structure of Smart Manufacturing [WMOG16]. The CPS concept map by the Berkley University [BERK-CPS], defines a CPS as a sensing platform [TYKH10], a reactive platform ("the system receives a stimulus and it reacts") with predictive features ("the system reacts to a future stimulus in order to avoid, modify or cause it"), which also requires Big Data technologies to analyse all data available (from the past and present) and predict the future.

A comprehensive review of existing middleware solutions for integrating heterogeneous computing and communication devices and supporting interoperability within the diverse applications and services is given in [RMJP16]. The review addresses middleware for Wireless Sensor Networks (WSN), RF identification (RFID), Machine-to-Machine (M2M) communication, and Supervisory Control And Data Acquisition (SCADA).

Smart CPSs. Smart CPSs are complex engineering systems, enabling the integration of heterogeneous hardware, software and cyberware technologies through intelligent analytics and decision-making mechanisms. The authors in [TAHO18] analyze the current understanding of Smart CPSs and recognize the following four levels in the advancement of the Smart CPSs design: (1) the CPS has conventional control mechanisms and can regulate parameters to a known degree, (2) the CPS is designed for alternative known modes of control and selection of the optimal mode of control during run-time, (3) the self-learning CPS with the ability to adapt predefined control algorithms during the exploitation period and (4) the CPS with largely unknown changes.

2.1 Industry 4.0 Reference Architectures

There exist two reference models for supporting interoperation and standardization in Industry 4.0 and Smart Manufacturing: The Reference Architecture Model for Industry 4.0 (RAMI 4.0) and the Industrial Internet Reference Architecture (IIRA).

Figure 1 illustrates the RAMI 4.0 architecture that is defined in a three-dimensional space [RAMI4.0]. The first horizontal axis of the RAMI 4.0 architecture represents the value chain and the lifecycle, the second horizontal axis represents the different hierarchies of a production system (i.e. products, field devices, control devices, station, work centers, enterprise, connected world), and the vertical axis contains the following six layers: (1) physical world (asset), (2) integration of software and hardware components, (3) communication capabilities, (4) information creation through data, (5) functional properties and (6) business processes.

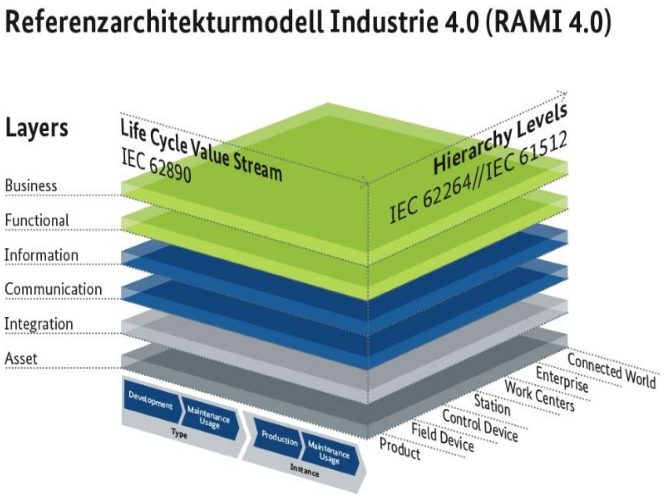

Figure 1: RAMI 4.0 Reference Architecture [RAMI4.0].

The Industrial Internet Consortium created the IIRA architecture model, that is based on ISO/IEC/IEEE 42010:2011 standard [IIRA17]. The IIRA has a focus on various perspectives (business, usage, functional and implementation viewpoints) of stakeholders in the system, i.e. users, operators, owners, vendors, developers and the technician who maintain the system. Figure 2 illustrates the five functional domains defined in IIRA, including control, operation, information, application and business domains, which are compared against system characteristics (e.g. safety, security, privacy, resilience, scalability, reliability) and cross-cutting functions (e.g. connectivity, distributed data management, industrial analytics, intelligent and resilient control).

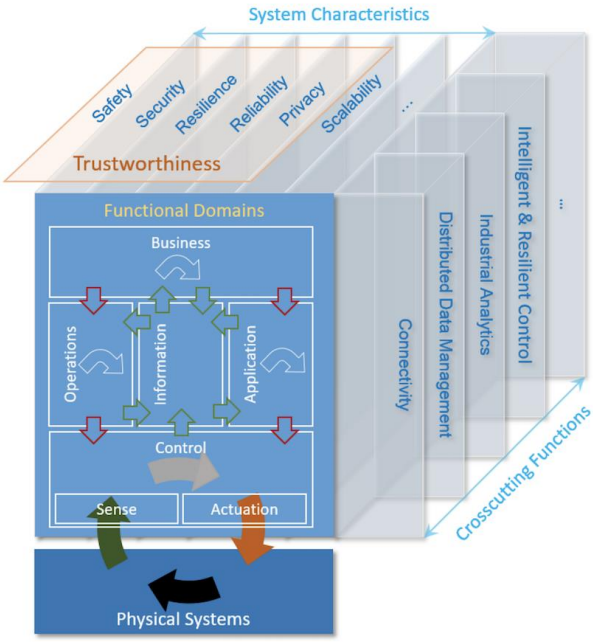

Figure 2: IIRA - Functional domains, crosscutting functions and system characteristics [IIRA17].

3. Technological Aspects of Safe and Secure IoT

3.1 Safety & Security Design and Methods

The IoT is a ubiquitous, heterogeneous, complex and dynamic/elastic/evolving system-of-systems. The elasticity requires the dependability (including safety) and security established during design time to scale up, i.e., it shall be resilient: the service delivery (or functionality) that can justifiably be trusted shall persists, when facing changes [Lap08]. The IoT shall remain dependable and secure in case of faults and threats not even considered in the design of the system [BNS13], [Lap08].

Notably, Avizienis et al. [ALRL04] published a detailed classification of fault-tolerant techniques and defined the attributes of resilience. In addition, resilience has been studied in the area of fault-tolerance in [Lap08], [C+09], [PD11], [Kop11], [PE16], [Wey17], [KLB+17].

Traditional dependability or security techniques do not handle elastic systems. Self-healing [GSRRU07], [PD11], for instance, is an approach based on self-adaptation and related to self-awareness. Self-aware systems learn and update models of the system to reason and act (e.g., self-heal) in accordance to higher-level goals (e.g., safety) [KLB+17]. The key feature of self-* techniques is their ability to learn and to evolve their models during runtime, e.g., to achieve resilience.

The software engineering community provides two main roadmaps on self-adaptation [C+09], [L+13], discussing different aspects of self-adaptation and its research challenges, e.g., requirements engineering, design, models or life-cycles. Weyns [Wey17] guides the reader through the evolution of self-adaptation. He gives an overview to architectures, runtime models and basic approaches of self-adaptive systems, including adaptation considering goals, requirements and uncertainties.

The state-of-the-art on dependability and security can be split into (1) detection and diagnosis (including fault/threat/anomaly detection) and into (2) recovery or mitigation [RKG+18] (see also the summary in Table 1). Below we provide references to surveys of resilience techniques. A more detailed classification and description of the methods can be found in [RKG+18].

3.1.1 Detection and Diagnosis

Chandola et al. [CBK09] provide an essential survey on anomaly detection including the types of anomalies, as well as methods and techniques for anomaly detection. Isermann [Ise06] surveys fault detection (Figure 3) and fault-tolerance methods from control theory (e.g., parameter estimation of process and signal models) to detect faults in CPS. Beside encompassing anomaly detection, surveys on intrusion detection [BMS14], [MC14] also contain signature-based (specification) detection.

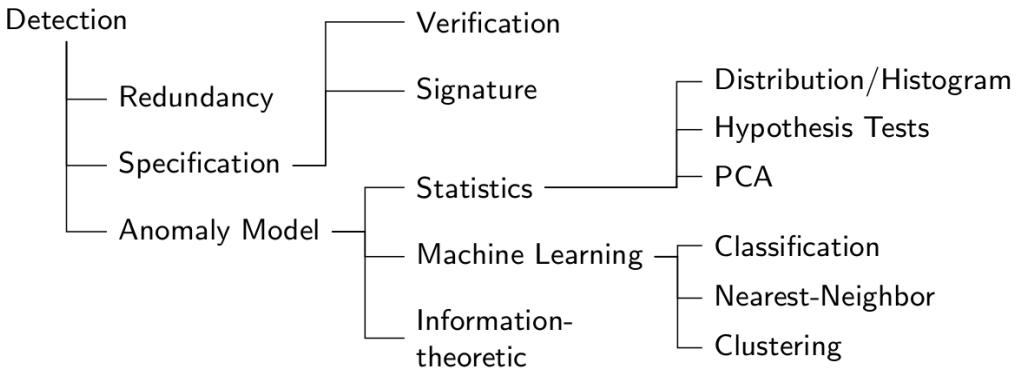

Figure 3: Overview of techniques for fault detection [RKG+18].

The surveys on self-healing features, e.g. [GSRRU07], [PD11] also include an overview and techniques to fault detection and diagnosis. Other methods to reason about failures are: runtime verification [LS09], [BF18] and fault localization [WGL+16], [BFMN18].

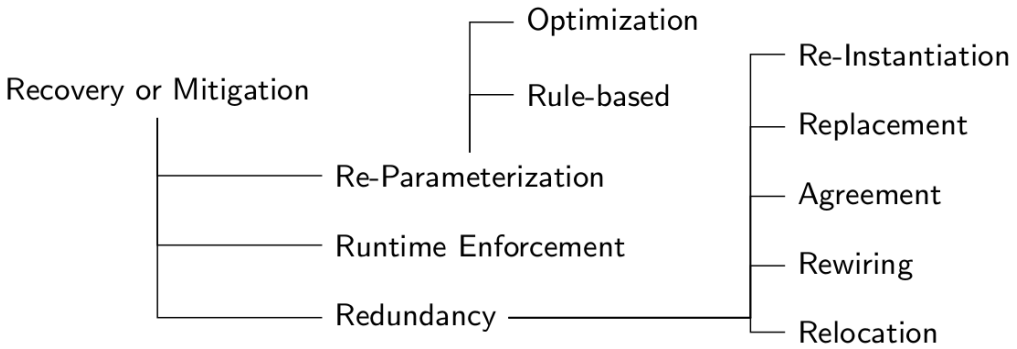

Figure 4: Overview of techniques for recovery or mitigation of faults [RKG+18].

3.1.2 Recovery and Mitigation

The authors in [PD11], [GSRRU07] provide a thorough survey on self-healing systems and classification of the techniques for fault recovery and mitigation (Figure 4). Papp and Exarchakos [PE16] focus on the design and testing for reconfigurable networked embedded systems; however, they include an overview of methods and types of runtime network reconfiguration. Ghosh et al. [GSRRU07] provide a broad overview about fault-tolerance, self-healing and health maintenance (fault-prevention). The authors additionally include health maintenance beside detection and recovery. The authors in [SB10] describes simple fault-tolerance methods (e.g., check-pointing, process migration or restart/replication) for the grid that can be applied to CPS too.

References	Techniques				
	Dependability			Security	
	Detection	Diagnosis	Recovery	Detection	Mitigation
Runtime verification [BF18], [LS09]	√				
Anomaly detection [CBK09]	√			√	

Intrusion detection [BMS14], [MC14], [BG16]				√	
Fault localization [WGL+16]		√			
Fault tolerance methods [C+09], [SB10], [L+13], [PE16], [Wey17], [KLB+17]			√		√
Fault-diagnosis [Ise06], [GSRRU07], [PD11], [EP18]	√	√	√		

Table 1: Surveys on techniques to achieve dependability and/or security [RKG+18].

3.2 Security Verification and Analysis

3.2.1 Analysis Methods

Threat Modelling

Threat modelling is the activity of defining a model of potential threats and potentially vulnerable applications. The better the assumptions, the closer is the theoretical model to the practical implementation to capture the significant attack vectors. A well-known example is the STRIDE methodology which was designed by Microsoft to be used as basis for their threat modelling approach. STRIDE stands for Spoofing of identity, Tampering, Repudiation, Information disclosure (privacy breach or data leak), Denial of service (DoS) and Elevation of privilege. When used in conjunction with a model of the target system, e.g. a Dataflow Model, STRIDE enables the identification of threats to a system. The basic STRIDE threat model pairs threats with elements to which they could be applicable (Table 2).

Element	S	T	R	I	D	E
External Entity	x		x			
Process (Sensor, ECU)	x	x	x	x	x	x
Data Store (Data store in an ECU or map store)		x		x	x	
Dataflow (communication between elements)		x		x	x	

Table 2: STRIDE Methodology.

On a high-level, threat modelling can be divided in three steps: (1) the Item is divided in functions or elements, including consideration of external interactions, (2) applicable threats are identified as subjects to a threat model, and (3) identified threats are documented and rated.

Depending on the phase in which threat modelling is applied the model can be refined to include further information, e.g. which security measures are already in place or for a communication which physical layer is used. This can lead to refined threats and automotive specific threat models.

Threat modelling can be applied throughout the complete development lifecycle. The existing examples include an integration into a security-aware hazard and risk analysis (HARA) method for the concept phase and into Failure mode and Effect Analysis (FMEA) for the system design phase. It

can even be used for the production and operation phase to support testing activities and the threat model can be extended to include known vulnerabilities.

Attack Trees

Attack trees represent multiple sequences of actions an attacker could take to reach a certain goal. The root node represents the goal of the attacker while the leaf nodes represent actions and attack steps. Leaves can be combined with "AND" and "OR". "AND" represents multiple actions which are required in combination to reach the top node. "OR" represents multiple actions where one of them is required to reach the top node. Systems can have multiple goals which can be identified by a previous activity. On a high-level, attack tree generation can be divided into three steps: (1) identification of attack goals, (2) identify potential attack sequences and (3) rate nodes.

For the generation of an Attack Tree, partial trees from previous analyses can be reused and an attack tree can also include countermeasures.

Although Attack Trees appear to be similar to Fault Trees from the perspective of safety, they are difficult to combine due to different level of values which can be assigned to leaves. In a Fault Tree, a hardware failure represented in a leaf node can have a well-known and experience-based failure probability, something which is difficult to enumerate for a security event. In security, assigned values can be costs, complexity or required time for an attack, which can be used to prune the tree by defining thresholds, but not for direct calculation. There are also some approaches to combine fault trees and attack trees to consider complex scenarios, e.g. [FOMA09], [STLI13], [BRPA03].

3.2.2 Frameworks

ETSI Threat Vulnerability and Risk Analysis (eTVRA)

eTVRA is a generic approach to Threat, Vulnerability and Risk analysis. It was developed for the telecommunication sector and later applied to Intelligent Transportation Systems. The goal of eTRVA is a systematic identification and mitigation of unwanted incidents.

eTRVA starts with an identification of the assets followed by identifying of vulnerabilities, threats which can exploit these vulnerabilities and potentially, following system level impacts. The quantification of the threats is based on ISO/IEC 15408. Based on impact and quantification, risks are ranked. The method proposes a template to be used for recording threats, threat agents, weaknesses and vulnerabilities. The steps itself are:

1. Identification of security objectives
2. Identification of the requirements derived from the objectives from step 1.
3. Inventory of the assets.
4. Identification and classification of vulnerabilities, threats and unwanted incidents
5. Quantifying the occurrence likelihood and impact of the threats.
6. Establishment of the risks.
7. Identification of countermeasures

EVITA Method

The EVITA project aimed at securing vehicular On-Board systems and besides security solutions, it developed a methodology for threat and risk analysis. EVITA methodology rates risks based on severity and attack potential. While EVITA defines a security engineering lifecycle, the focus here is on the steps for threat identification.

1. Develop view on system
2. Describe relevant use cases
3. Identify assets to be protected within the use cases
4. Identify threats to the assets
5. Evaluate and rank risks
6. Identify security requirements for the threats based on risk analysis

For the identification of threats (step 4) “dark-side scenarios” are used. This approach aims at identifying potential attacker motivation and capabilities and, based on this information to model the attacks. Based on attack goals that satisfy the motivation of the attacker, the Attack Trees are developed to identify scenarios how an attack could be conducted. EVITA structures the Attack Trees in three major level. Level 0 is the high-level goal of the attacker. Level 1 contains multiple objectives, e.g. how an attacker could achieve the goal and have a negative impact on stakeholder. Level 2 and below model attack methods which can consist of multiple intermediate steps that are connected with “AND” and “OR”.

The risk analysis (step 5) is based on high-level security objectives (operational, safety, privacy and financial) where the severity of a threat is rated and the rating of attack potential of the identified scenarios.

HEAVENS Method

The HEAVENS project aimed at addressing software vulnerabilities, which could impact safety and security in vehicles. It developed a method for threat analysis and risk assessment, contained in the HEAVENS security model, which was updated in the HOLISEC project. The HEAVENS workflow consists of three main phases: (1) Threat analysis, (2) Risks assessment and (3) Security requirements.

The threat analysis requires as input the functional use case and identifies threats for each asset involved in the use case. Threats are also mapped to security attributes, e.g. which security attribute is endangered by a threat. For the threat identification, STRIDE and threat modelling are used. The approach is aimed at the concept phase where vulnerabilities are not yet known, e.g. threats are identified independent from vulnerabilities.

Risk assessment is done by ranking the impact (Impact Level, IL) on an asset and the potential of a threat (Threat Level, TL), and defining the risk (Security Level, SL) based on IL and TL. Threat levels are based on Common Criteria, impact is similar to EVITA but extended with impact on legal and regulatory assets.

3.3 Digital Twins for Security Life-Cycle Data Management

As a virtual model of real-world factory settings, the Digital Twin enables various simulation and testing of system's performances throughout its various lifecycle phases, from system and product design and integration, manufacturing, operation, maintenance, to its end-of-life services. Digital Twin models can be system-, product- or service-oriented, representing the essential components that enable the system's real-world behavior in various scenarios. These models can be simulated, analyzed and updated in order to predict performances of the system, and support a range of stakeholders in planning and designing, modifying, optimizing, and verifying industrial factory settings and processes.

Twins. The concept of twins was firstly used in NASA's Apollo program for building two identical space vehicles: the one to be sent in space, and the other one to mirror the conditions and performances of the vehicle in space, during the flight mission [BORO16]. The twin concept has been used in aircraft industries as a core for the optimization and validation technology of aircraft systems based on the integration of sensor data, historical maintenance data and all available historical/ fleet data [SHAF10][SHAF12].

Digital Twins. The further evolution of ''microchip, sensor and IT technologies'' [ABGD16] opened the way for the creation of smart products that can track product models along their lifecycle phases, merge and analyze the acquired sensor data and communicate their production and operating conditions [SAMW17]. Such technology evolution shifted the concept of twins from the aerospace industry into Smart Manufacturing [RHOM15], ensuring information exchange throughout the entire manufacturing lifecycle [ABGD16] [ROWL15], virtualization of manufacturing systems [SCRO16], decision support and system behavior-based predictions [KRAF16], as some of the major features of the Digital Twin.

The term Digital Twin was coined by M. Grieves in 2002 and evolved over time from "conceptual ideal for PLM (Product Lifecycle Management)", "the mirrored space model", "the information mirroring model", to today's notion of the Digital Twin. The term Digital Twin has been in wide use from 2011 and is defined as "a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level'' [GRIE14] [GRVI17]. The full overview of the Digital Twin definitions that appeared in literature is given in [NEFM17], mainly defining the Digital Twin as a "product digital counterpart of a physical product" [RHOM15] that is used for its simulation in a virtual world to predict future states of the system" [GABK16]. The authors in [HAAN18] define the Digital Twin as a comprehensive digital representation of an individual product, its properties, condition and behavior. Its core functionality is to support design tasks and/or to validate system properties through the multi-domain and multi-level simulations along all lifecycle phases, including operation support [BORO16]. According to the literature overview in [NEFM17] that explores the availability of simulations and simulation tools for the Digital Twin in Smart Manufacturing, existing simulations are focused on complex behavior of production or data exchange simulation, while simulations tools in manufacturing are not available.

In Smart Automotive industry, Digital Twins are defined as a lifecycle management and certification paradigm that incorporates models and simulations consisting of as-built vehicle states, loads and environments, and other vehicle-specific history [HOLN17]. The authors in [REMM13] look at the

Digital Twin as a simulation integrating an on-board health management system, maintenance history, historical vehicle and fleet data. Here, Digital Twin can mirror the entire lifecycle of a specific physical product, enabling significant gains in safety and reliability.

At present, the following types of the Digital Twin can be found in the literature [GRVI17]:

- Digital Twin Prototype (DTP): it includes information related to requirements of a physical object, its 3D model, Bill of Materials (BoM) and material specification, Bill of Processes (BoP), Bill of Services (BoS) and Bill of Disposal (BoD);
- Digital Twin Instance (DTI): it includes information such as 3D model with General Dimensioning & Tolerances (GD&T) describing geometry of the physical object and its components, a BoM that lists the object's components, a BoP that lists operations performed on the object and related measurements, operational states captured from sensor data;
- Digital Twin Environment (DTE): it enables operations on the Digital Twin to support either prediction of the future behaviour and performances, or interrogation for the histories and data correlation.

3.3.1 Open Source and Commercial Digital Twin Architectures and Tools

Some examples of existing commercial software tools that implement industrial Digital Twin technology are:

- GE has developed the Digital Twin of jet engine that enables a configuration of individual wind turbines, prior to procurement and construction. Each virtual turbine is fed data from its physical equivalent. The Digital Twin optimizes turbine-specific parameters, such as torque of the generator and speed of the blades. GE Digital Twin is based on Predix platform (www.predix.com) that delivers capabilities such as asset connectivity, edge technologies, analytics and Machine Learning (ML), Big Data Processing, Asset Performance Management (APM), and asset-centric Digital Twins [PRED18].
- PTC has developed smart Product Lifecycle Management (PLM) software called Windchill that supports processes such as a failure reporting, analysis and corrective action system (https://www.ptc.com/en/products/plm/plm-products/windchill).
- Dassault Systèmes (DS) has built an aerospace- and defence- manufacturing operations management software called Build to Operate, which enables monitoring, controlling, and validating of all aspects of manufacturing operations [DCX16].
- DXC Technology has developed the Digital Twin for predicting the performance of hybrid cars before committing the changes in the car manufacturing process [DXC17].
- Siemens has built Simcenter 3D for the Digital Twin [SIM17].

Open source implementations of Digital Twins are even more in their early stages. The currently available implementations are restricted to the following prototypes:

- Eclipse Ditto is an open source software solution that implements Digital Twins as IoT development patterns. Here, the Digital Twin mirrors physical assets/devices, provides services and context related to the product's environment, keeps real and digital worlds synchronized. (Eclipse Ditto project webpage: https://www.eclipse.org/ditto/index.html; GitHub page: https://github.com/eclipse/ditto)
- CPS Twinning is a framework for generating and executing Digital Twins that mirror CPSs. CPS Twinning generates Digital Twins from an AutomationML artifact and currently, it requires major manual adjustments. (The GitHub page of the project is available from: https://github.com/sbaresearch/cps-twinning).

3.3.2 Technologies Enabling Smart CPSs and Digital Twins

Each physical component of the CPS has its virtual representation that is called the Digital Twin. The core benefits of implementing the Digital Twin can be summarized as follows [ORACLE17]: it allows visibility in the manufacturing operations; it can be used to predict the future state of the machines; it can be used to simulate various conditions that would be impractical to create in real life; it can be used to connect with the backend business applications to support supply chain, financial decisions, etc. Technologically speaking, it combines AI, and real-time predictive analytics and algorithms performing on top of Big Data derived from IoT sensors and historical data. *The ultimate objective of the Digital Twin is to improve the design and execution in digital manufacturing through simulation, prediction of future states and intelligent decision-making related to various lifecycle phases.*

The design of the Digital Twin suggests three major components to be considered [ORACLE17]: (i) Asset modelling, (ii) Predictive analytics and decision making, and (iii) Lifecycle knowledge base, including real-time sensor data and historical data.

Asset Modelling

Asset modelling is about architecting of the Digital Twin: designing the structure of its assets (physical things) and components, measurable physical parameters and other digital manufacturing information about the assets (e.g. manufacturing date, maintenance history). Asset modelling adds value to connected sensor data and contributes to a range of new insights, e.g. obtaining an information on health of sensors, which can be performed through inferring, correlation and transformation of measured sensor values and asset states, conditions and maintenance records [KUAB17]. It may also include a different presentation (visualization) forms for different user groups, e.g. one group of users may require the insight in only operational data, while the others could be more focused on individual devices. Adding information such as metadata, nearby environmental conditions, maintenance data, service history, configuration and production data, external data, enterprise web services etc., contributes to a rich representation of the physical things (device/system) and further augments the Digital Twin.

Predictive Analytics and Decision-Making

Analytics for Digital Twins includes predictive and descriptive analysis of the behaviour of various assets. Predictive analytics is composed of a training phase (learning a model from training data) and a predicting phase (using the model for predicting future outcomes). The most used predictive models in ML belongs to the category of Supervised Learning, encompassing (i) classification

models for the evaluation of a discrete value (e.g. Logistic regression, Neural networks, Support Vector Machine (SVN)) and (ii) regression models for the evaluation of a numeric value (e.g. Linear regression model, Regression with regularization, Bayesian network and Naïve Bayes, K-Nearest Neighbour (KNN)) [BOBP15]. The more types of data the ML model can analyse and thus, learn the states that matter along the manufacturing path, the better the model will be. For example, availability of historical data is useful for ML models to learn the maintenance states of assets for predictive maintenance. However, continuous learning of the ML models requires a flow of real-time data. In addition to ML-based algorithms, there are models, such as [BOBP15]: Online Analytical Processing (OLAP), which is a part of the broader category of Business Intelligence; Graph Analysis, a method for the analysis and representation of complex networks; Text Analytics, a method for converting unstructured data into meaningful data for analysis to provide search features, sentiment analysis and fact based decision making; Time Series Processing, a method for analysing time series data in order to extract meaningful statistics and other data characteristics; Monte Carlo simulation method that uses repeated random sampling to generate simulated data for solving any problem with a probabilistic interpretation.

Lifecycle Knowledge Base

The Digital Twin knowledge base collects asset lifecycle data (e.g. time-series sensor data), data derived from analytics and decision-making algorithms, and historical data. The functionality of a Digital Twin improves over time as more data is accumulated and processed by algorithms. The prerequisite to the knowledge base creation is to have a proper foundation and integration platform in place, enabling the integration of multiple distinct data streams through standards and frameworks, their utilization and management [ORACLE17]. The Digital Twin can also be augmented by adding data from a variety of data sources, e.g. asset maintenance history from an Enterprise Resource Planning (ERP) system, account data from a Customer Relationship Management (CRM) system, environmental data, etc.

According to the size of a knowledge base, the authors in [BOBP15] differentiate among:

- partial Digital Twin, with a small number of data sources that can be combined to infer data (derivative data),
- clone Digital Twin, with a larger amount of meaningful and measurable data sources and
- augmented Digital Twin that enhances connected asset data with derivative data and correlated data obtained from analytics tools.

A partial Digital Twin is built on top of simplistic device models that could be implemented as JSON documents with a set of observed and reported attributes (e.g. speed of a machine) and a set of desired values (e.g. an application is setting the speed of a machine), which can be effectively correlated to detect operational abnormalities and instantly generate alerts. A clone Digital Twin is what is typically needed in industry: it is built on top of the product design and manufacturing information and reflects its physical properties and uses real-time data.

4. Secure and Safe IoT for Industrial Applications

4.1 Secure and safe IoT for Automated Driving

4.1.1 Market Aspects

The market introduction of ADAS/AD/HAD has shown that the primary challenges potentially impeding a faster market penetration are pricing, consumer understanding and safety/security issues. Technological challenges are not insignificant and will drive the delay between conditionally automated driving and fully automated capabilities. Figure 5 [McKin16] shows the disruption scenarios in the automated driving context.

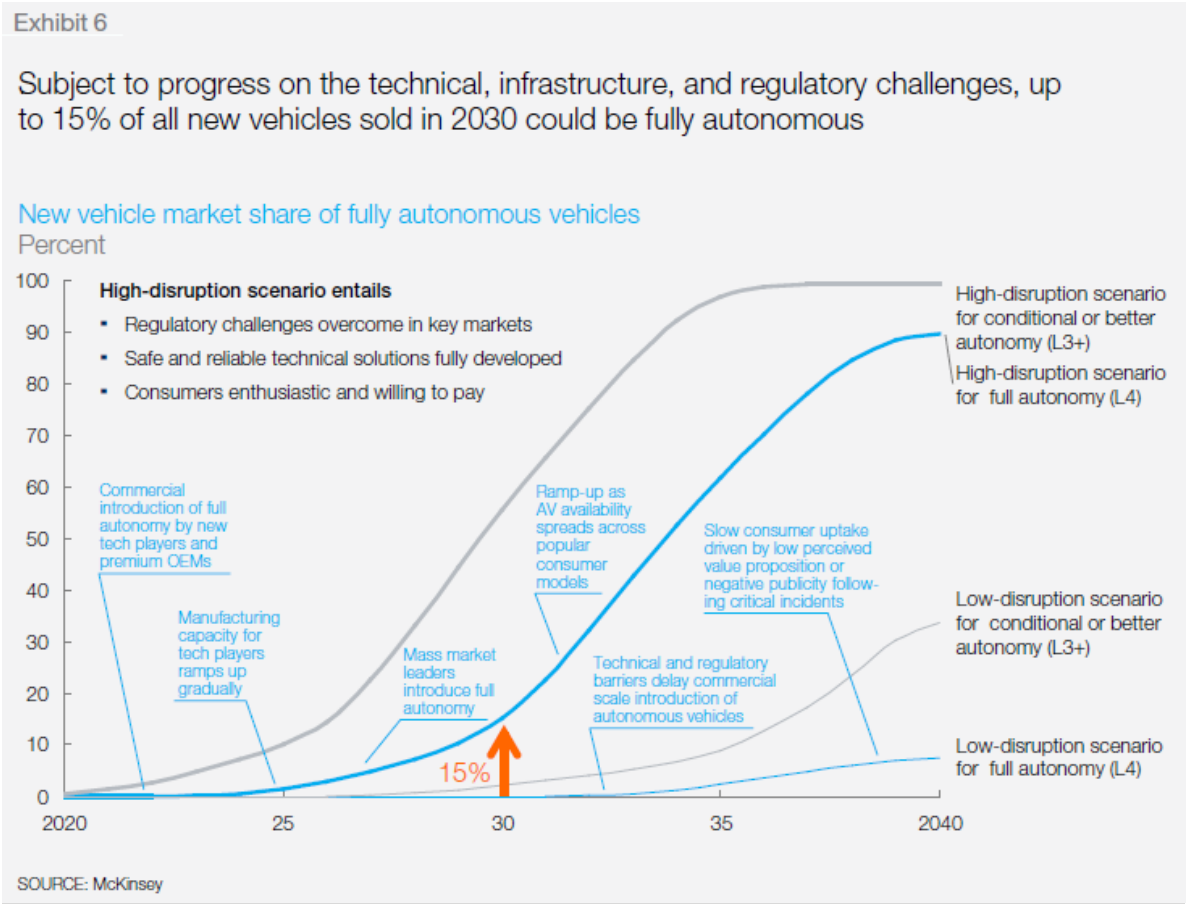

Figure 5: AD Disruption Scenarios.

From Figure 5, we can summarize the following scenarios:

- Best scenario:
  - Conditional autonomy (L3) will have a penetration rate of 10% by 2025
  - Highly automated driving (L4) will have a penetration rate of 2,5% by 2025

- Worst scenario:
  - Conditional autonomy (L3) will have a penetration rate of 2,5% by 2030
  - Highly automated driving (L4) will have a penetration rate of 2,5% by 2035

According to IHS Markit [IHS18], the number of vehicles sold with automated driving capabilities by 2040 will surpass the 33 million annually (Figure 6).

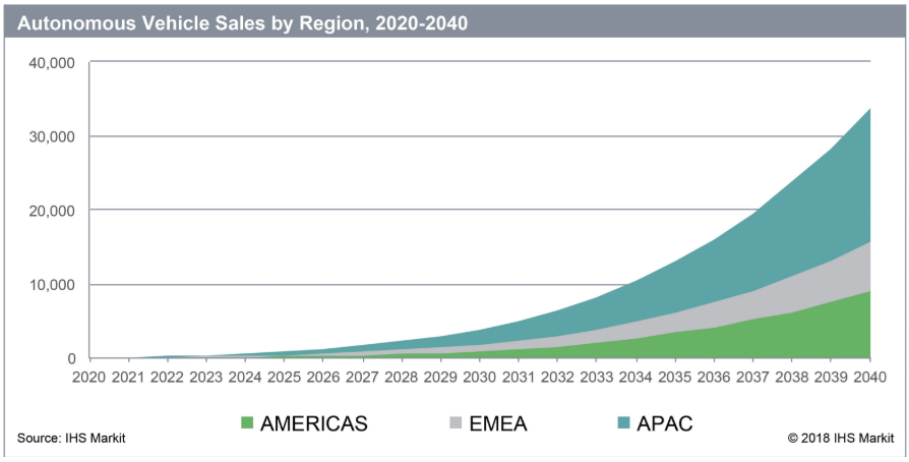

Figure 6: Autonomous Vehicles Sales by Region.

As it can be seen in the above charts, there is an uncertainty with regards on the market penetration rates of automated driving. However, all of them agree on a growth forecast in the future.

4.1.2 Classification of Automated Driving

The SAE defines 6 levels for AD systems. While the lowest level L0 denotes manual operation by the driver, the highest level L5 indicates that the entire door-to-door driving experience is handled by the system. The intermediate levels indicate which tasks are handled by the vehicle and the driver, respectively (see Table 3).

	L0: no automation	L1: driver assistance	L2: partial automation	L3: conditional automation	L4: high automation	L5: full automation
Control	Driver	Driver	Vehicle	Vehicle	Vehicle	Vehicle
Monitoring	Driver	Driver	Driver	Vehicle	Vehicle	Vehicle
Fallback	Driver	Driver	Driver	Driver (limited take-over time)	Vehicle (for defined use case)	Vehicle

Table 3: Autonomy levels according to SAE.

The tasks can be divided roughly into the following:

Control: The actuators (powertrain, brakes, steering) are controlled either by the vehicle or the driver. L1 systems only *support* the driver by making manoeuvres easier to execute, while L2+ systems can *control* longitudinal and/or lateral movement.

- Monitoring: While the vehicle can assume *control* for L2 systems, the driver still needs to permanently supervise the correct functioning of the system, i.e., they may take their hands off the steering wheel, but not their eyes off the road. For L3+ systems, the vehicle can also assume *responsibility* and the driver may take their eyes off the road. The vehicle thus needs to detect faults by itself.

- Fallback: For L1-L2 systems, the driver needs to be able to take over without warning. For L3 systems, the vehicle needs to continue operating at least with degraded functionality for a limited take-over time. For L4 systems, the vehicle needs to continue operating within the defined use case (e.g., as long as it is on the highway), while for L5 systems, the vehicle needs to be able to finish the mission (i.e., until the final destination is reached).

4.1.3 Current generation AD systems

For a long time, each feature (Parking Assistant, Adaptive Cruise Control, etc.) was hosted on a separate ECU with a dedicated sensor set (see Figure 7). As the number of features has increased considerably in the last years, this approach is no longer feasible due to high complexity and cost.

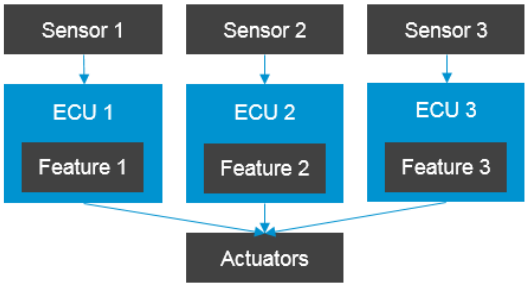

Figure 7: System architecture using dedicated ECUs for each feature.

Current system architectures consolidate most processing and features in a single, centralized ECU, allowing for more complex features at reduced cost. This ECU generally consists of several hosts (microcontrollers or SoCs) with different performance and safety characteristics (see Figure 8). Since all sensor data passes through the ECU, sensor fusion can be used for obtaining a more accurate model of the vehicle's environment. The central ECU can handle all processing, perception, prediction, and planning tasks.

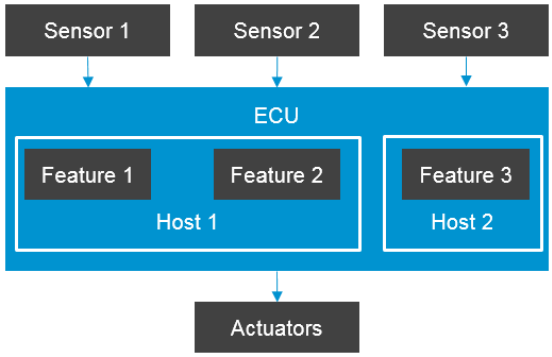

Figure 8: System architecture with a central ECU consolidating all features.

Most currently offered features are L1 or L2. Only a few L3 features are currently undergoing certification, all of them restricted to defined environments and simple use cases. This allows reducing the required ASIL for some components of the system, thus lowering development efforts. The following upcoming L3 features can serve as examples:

- Parking Pilot: The vehicle parks itself in a parking spot or garage without a driver inside.

- Traffic Jam Pilot: The vehicle drives autonomously on a highway without changing lanes and at less than 60 km/h.
- Highway Pilot: The vehicle drives autonomously on a highway changing lanes at a maximum speed of 130 km/h.

4.1.4 Safety monitoring using runtime verification methods

In the last decade assertion-based hardware monitoring has received an increasing attention. Generating hardware monitors from Property Specification Language (PSL) has been proposed by several research groups and it has been implemented first in the tools FoCs [DGG+05] developed by IBM and MBAC [BZ05, BZ06, BZ08] developed by Zilic and Boulé. On the same line of research is the work of Borrione et. al. in [BLMA+05] and Backasch et. al. in [BHW+13].

In order to allow rich specifications, we support past and future, timed and untimed temporal logic. In contrast to our work, all of them are focused on untimed digital specifications. FoCs generates monitors for SystemC simulations. MBAC adopts an automata-oriented approach which conceptually differs from our transducer-based compositional construction. Synthesizing hardware from formal specifications was successfully applied to obtain an arbiter for ARM's Advanced Microcontroller Bus Architecture (AMBA) Advanced High-performance Bus (AHB) bus for specifications given in PSL [BGJ+07]. The formal semantics of PSL does not include an explicit notion of time. On the other hand, formalisms such as STL and TRE allow precise definition of desired time intervals for the requirements.

Finkbeiner et al. in [FK09] present a technique to synthesize monitor circuits from LTL formulas with bounded and unbounded future operators. They allow only past-time specifications and evaluate their approach only with formulas with the lower time bound equal to zero. Claessen et al. [CES13] propose some efficient techniques to synthesize a LTL safety and liveness property checkers as circuits with sequential elements. The authors focus on model checking of hardware system design.

Another very popular extension of correctness monitors is the system health monitoring. In this context, a monitor is not only reporting violations, but rather recognizing trends in a behaviour of a system and estimating the system health in every time step. Such monitors, which rely on Bayesian network to estimate system health, are implemented in [MRS17].

Reinbacher et. al. in [RFB14, RRS14] propose hardware monitors from different fragments of Metric Temporal Logic. In [RFB14], the authors tackle only the past fragment of MTL using a transducer-based approach. The authors use an approach in which absolute time stamps are memorized. Hence, the resources needed for implementing their monitors depend on the duration of the emulation runtime. The authors develop a sophisticated architecture that targets reconfigurability of monitors. In [RRS14] the authors address the future fragment of MTL. They adopt a three-valued interpretation of the logic and produce a "maybe" output delaying a definite verdict until the formula can be really evaluated. This approach is suitable for estimating system health using a Bayesian network on top of the observers. Similar to their previous work, the authors evaluate their framework only on recorded data.

UPPAAL [BLL+95, LPY97] is a well-established tool for the verification of real-time systems which can be modelled with timed automata. This tool provides a description language for modelling, a simulator, and a model checker. In contrast, our goal is to create a standalone monitor in order to verify a discrete time system during runtime. Orthogonal to monitoring, an SMT-based approach to design and analysis of CPS was described in [CSRB13]. In that work, the authors show how to reduce several important verification and synthesis problems of CPS to exists-forall quantified propositional combinations of constraints which is then handled by a solver.

Recent breakthrough in state-of-the-art of runtime monitoring includes a scalable algorithm for implementing real-time safety monitors in hardware, from specifications expressed in Signal Temporal Logic [JBG+15]. This approach has the advantage that it scales well with the length of the trace, allows monitoring fast devices and relies on formal specifications with unambiguous semantics. Such correctness monitors rely on deterministic data and provide non-probabilistic, definitive verdicts. The monitors adopt a black-box approach: they do not require insights into the internal structure of the monitored system.

4.2 Secure IoT for Industry 4.0

4.2.1 Security Challenges in the IoT

With the constantly increasing number of IoT-connected devices also the related (security) challenges are growing further. The following Table 4 shows some of the current challenges in the IoT and their impact on security issues.

CHALLENGE	CHARACTERISTICS	SECURITY IMPACT/ISSUES
Large number of IoT devices of the same kind & accessible from one network access point	Possibility for millions of instances of each device & often one access to the internet is sufficient to reach any other device on the network.	Breaking one instance allows all similar devices to be broken down as well & potential creation of a huge attack surface through a single access point.
Many types of IoT devices for many use cases	Ever expanding types of IoT devices due to the constant appearance of new use cases.	Common IoT standard and interoperable security framework is not yet available.
Unmanaged lifetimes of IoT devices	As it's not centrally managed, lifetime of an IoT device will span an undefined number of years.	Assurance of updated security throughout the entire (undefined) lifetime of an IoT device currently a challenge.
Limited resources of IoT devices	Many IoT devices are limited in e.g. processing power, storage capability etc.	Implementation of standard security techniques to distribute the security burden among various IoT devices to achieve overall system security to be fully deployed.
Mixture of IoT devices for critical and non-critical applications in one network	Inter-connectedness of e.g. critical energy distribution facilities and non-critical sport	Easier access to critical applications via low security devices in the same system.

	monitoring wearables because of sharing the same network.	
Generation of huge amounts of personal data by IoT devices	IoT devices collect detailed data to send information to service providers, wherefore many of them can be associated with single individuals.	Collected data could potentially be used to violate the privacy of a single individual.

Table 4: Security Challenges in the IoT (NXP)

4.2.2 Security in Industry 4.0

One manifestation of the IoT is demonstrated by Industry 4.0. Hereby the fusion of production-technologies with ICT is addressed. It virtually can be described as the manifestation of the IoT in the industrial production environment and incorporates, for example, not only sensor systems and CPS but also business models and processes. In this context, amongst others, the following aspects are vital [VI40O18]:

- Data reliability & consistency
- Production automation
- Supply Chain Management & Integration
- System interoperability
- Machine (construction) optimization
- Quality optimization (maintenance)
- Interface optimization

Considering the above aspects, it becomes apparent that security is a critical factor also in Industry 4.0, and that the statements made in 1.2 are to a high degree also applicable for Industry 4.0. With regards to secure traceability solutions based on RFID technologies, on top of the above statements, two more aspects are relevant in terms of state-of-the-art:

- Coexistence and interoperability: the increasing use of wireless technologies and solutions at the same time raises challenges in terms of potential radio interference and the related dependability of wireless systems. Step-ups in both, methodological as well as measurement capabilities are required to master the increasing complexity and to ensure dependability by design. Secondly, advanced system architectures and interface concepts will be needed to ensure the optimum combination and interaction between different wireless technologies in a complex I4.0 environment. The improvements in terms of power consumption and sensitivity of components will significantly help to solve this challenge.
- Adequate security despite constrained resources: state-of-the-art security solutions are available for many high-end security requirements. However, IoT applications require step-ups, at least regarding two aspects. Primarily, integral end-to-end security concepts still need

to be developed, covering both, life-time and stakeholder considerations, as well as the diversity of devices from a security level perspective (heterogenous populations). Secondly, mapping and implementation of required security levels on resource constrained devices (like passive RFID-devices) require innovations in security algorithms as well as advances in low power mixed signal design and production of such devices.

In combination with sensor functionality, additional topics like energy harvesting come into play.

4.2.3 IIoT – Industrial Internet of Things

As Reiner Anderl in [Anderl2014] summarizes, Industry 4.0 technology aims at enabling communicating, intelligent and self-controlled systems. One of the key aspects of Industry 4.0 is the integration of cyber technologies into the production system in order to provide innovative services, such as Internet-based diagnostics, remote maintenance or pay-per-use services, in an efficient way. A state-of-the art production system is organized according to the automation pyramid shown in Figure 9 [Ruprechter2017]. Each level includes Industrial Control Systems (ICS) with an increasing number of components and subsystems from top to the control level. Figure 9 points out the current situation in security implementation, where a big need and a gap is identified at field level, control level, and supervision level. Furthermore, these are also the levels where the security relevance is very high, leading to a significant mismatch, which is intended to be solved by the project. Figure 10 [Ruprechter2017] illustrates different kinds of ICS in a factory with two production lines, down to robotic arms on each production line. In the past, the whole automation pyramid, meaning all the components, was located inside one production site and an external connection was neither needed nor desired. Recently the demand for external connections to a plant, a production line, or even to specific components has risen rapidly and the different applications enabled by this connection promise to lead to various new business models. Just to note, this affects both new production lines as well as the existing ones.

On the field level, Programmable Logic Control (PLCs) is used for automated cycles of actuators and sensors. PLCs produce output results from input conditions in real-time and usually have built-in communications ports enabling peer-to-peer communication for information exchange with other processors. This allows separate parts, e.g., production lines, of a complex process to have individual control while allowing the subsystems to coordinate over the communication link. These communication links are also often used for Human Machine Interface (HMI) devices such as PC-type workstations or keypads.

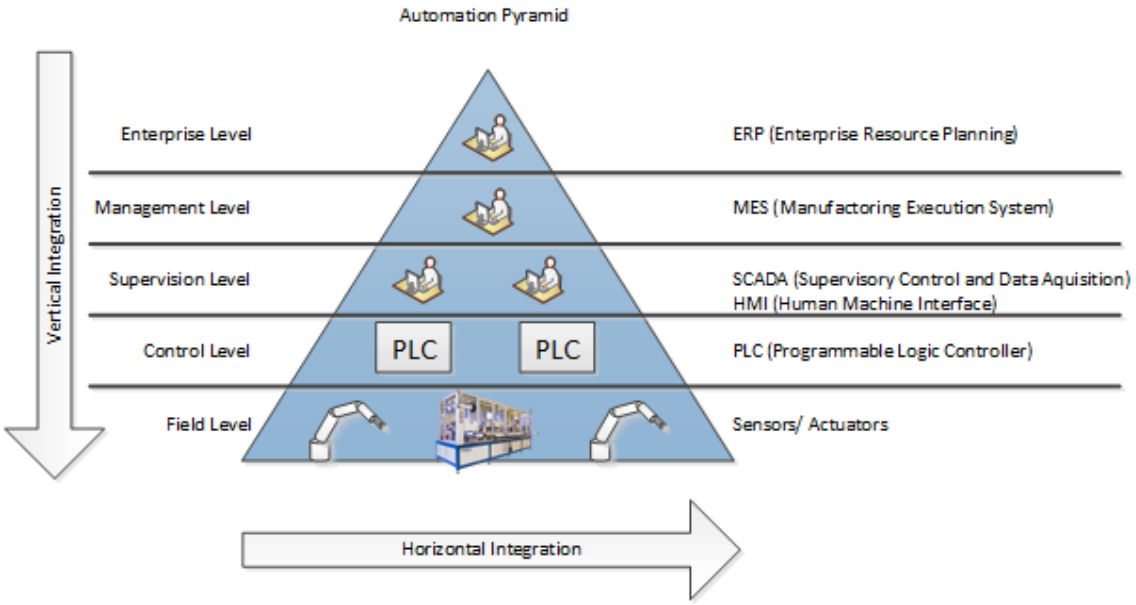

Figure 9: The automation pyramid from enterprise down to control and field level.

Over recent years, an increasing number of ICS have been equipped with Internet connectivity in order to benefit from the advantages of remote accessibility such as e.g., installation of software updates, or remote maintenance. This Internet connectivity is simply achieved by establishing a remote desktop connection via a Virtual Private Network (VPN) [Cruz2015] to an industrial PC connected to the production line. From a security perspective, an attacker that gains access to this remote PC by compromising the VPN credentials or via malware injected to the PC can thus get full control on the production line. Furthermore, a remote desktop connection is not helpful for the acquisition of real-time data from a production line as data needs to be manually transferred from the industrial production line PC via the remote desktop link.

Data acquisition within companies can currently be best described as “fog computing” [Stojmenovic2014]. Production-process related data is mostly used in an isolated network within the company, but not transferred to external servers or consumers. The term “fog computing” addresses the local storage on edge nodes, as opposed to the “cloud computing” paradigm where data is accessed on-demand via the Internet.

The main questions to be answered are

- How to protect production lines from undesired or deliberate interruption?
- How to protect production lines from undesired or deliberate changes?
- How to prevent safety mechanisms from being turned off or overridden?
- How can connections, readouts and changes be tracked?
- Which mechanisms can be introduced to prevent manipulation by users that have only reading access rights to machines and controllers?
- Which authorization levels should be introduced?

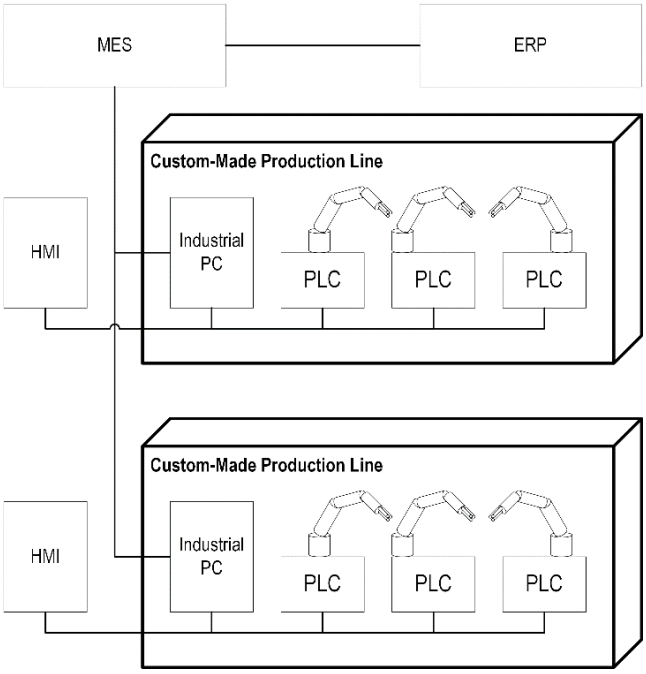

Figure 10: A factory side's industrial control systems with multiple production lines.

5. Conclusion

This deliverable provides a review of different technology roadmaps, surveys or recommendations to consolidate the technology approach of the IoT4CPS project. It presents an overview of the current state of the art in the context of Trustworthy IoT for CPS, both from the academic and from the industrial perspective. From the academic perspective, for example, the field of design and testing for reconfigurable networked embedded systems has been covered in relation to recovery and mitigation. STRIDE methodology was described as a state-of-the-art threat modelling methodology. In addition to that, several frameworks for security analysis such as ETSI Threat Vulnerability and Risk Analysis (eTVRA), EVITA, and HEAVENS were investigated. While analysing reference architecture models such as RAMI 4.0 and IIRA, we were able to align the research in IoT4CPS according to the perspectives and components of these models. In essence, this deliverable serves as a guidance for aligning various research topics in the project.

6. References

[ABGD16] [ABGD16] Abramovici, M., Göbel, J.C., Dang, H.B. Semantic Data Management for the Development and Continuous Reconfiguration of Smart Products and Systems. CIRP Ann 2016;65(1):185–8. 2016.

[AIM10] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A survey. Computer Networks, 54(15):2787 – 2805, 2010.

[ALRL04] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Trans. on Dependable and Secure Computing, 1:11–33, 2004.

[Anderl2014] Anderl, R.; "Industrie 4.0 - Advanced Engineering of Smart Products and Smart Production" in 19th International Seminar on High Technology, 2014.

[BF18] Ezio Bartocci and Yliès Falcone, editors. Lectures on Runtime Verification - Introductory and Advanced Topics, volume 10457 of Lecture Notes in Computer Science. Springer, 2018.

[BFMN18] Ezio Bartocci, Thomas Ferrère, Niveditha Manjunath, and Dejan Nickovic. Localizing faults in simulink/stateflow models with STL. In Proc. of HSCC 2018: the 21st International Conference on Hybrid Systems: Computation and Control, pages 197–206, 2018.

[BGJ+07] Roderick Bloem, Stefan Galler, Barbara Jobstmann, Nir Piterman, Amir Pnueli, and Martin Weiglhofer. Specify, compile, run: Hardware from psl. Electronic Notes in Theoretical Computer Science, 190(4):3 – 16, 2007. Proceedings of the Workshop on Compiler Optimization meets Compiler Verification (COCV 2007).

[BHW+13] R. Backasch, C. Hochberger, A. Weiss, M. Leucker, and R. Lasslop. Runtime verification for multicore soc with high-quality trace data. ACM Transactions on Design Automation of Electronic Systems, 18(2), 2013.

[BLL+95] Johan Bengtsson, Kim Guldstrand Larsen, Fredrik Larsson, Paul Pettersson, and Wang Yi. UPPAAL - a Tool Suite for Automatic Verification of Real-Time Systems. In Hybrid Systems III: Verification and Control, Proceedings of the DIMACS/SYCON Workshop, Ruttgers University, NJ, USA, pages 232–243, 1995.

[BLMA+05] D. Borrione, Miao Liu, K. Morin-Allory, P. Ostier, and L. Fesquet. On-line assertion-based verification with proven correct monitors. In Proc. of ITI 2005: the 3rd International Conference on Information and Communications Technology, pages 125–143, 2005.

[BMS14] Ismail Butun, Salvatore D Morgera, and Ravi Sankar. A survey of intrusion detection systems in wireless sensor networks. IEEE communications surveys & tutorials, 16(1):266–282, 2014.

[BNS13] Robin Bloomfield, Kateryna Netkachova, and Robert Stroud. Security-informed safety: If it's not secure, it's not safe. In Anatoliy Gorbenko, Alexander Romanovsky, and Vyacheslav Kharchenko, editors, Software Engineering for Resilient Systems, pages 17–32, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[BORO16] Boschert S, Rosen R (2016) Digital Twin – The Simulation Aspect. in Hehen- berger P, Bradley D, (Eds.) Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers, Springer International Publishing 2016, pp. 59–74.

[BRPA03] P. J. Brooke and R. F. Paige, "Fault Trees for Security System Design and Analysis," Computers & Security, pp. 256–264, 2003.

[BZ05] M. Boulé and Z. Zilic. Incorporating efficient assertion checkers into hardware emulation. In Proc. of ICCD, pages 221–228. IEEE Computer Society Press, 2005.

[BZ06] M. Boulé and Z. Zilic. Efficient automata-based assertion-checker synthesis of PSL properties. In Proc. of HLDVT, pages 69–76. IEEE, 2006.

[BZ08] M. Boulé and Z. Zilic. Automata-based assertion-checker synthesis of PSL properties. ACM Transactions on Design Automation of Electronic Systems, 13(1), 2008.

[C+09] B. H. C. Cheng et al. Software Engineering for Self-Adaptive Systems: A Research Roadmap. In Software Engineering for Self-Adaptive Systems, pages 1–26. Springer Verlag, Berlin, Heidelberg, 2009.

[CBF+ 16] Andrea Ceccarelli, Andrea Bondavalli, Bernhard Froemel, Oliver Hoeftberger, and Hermann Kopetz. Basic Concepts on Systems of Systems, pages 1–39. Springer International Publishing, Cham, 2016.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A Survey. ACM Comput. Surv., 41(3):15:1–15:58, July 2009.

[CES13] K. Claessen, N. Een, and B. Sterin. A circuit approach to ltl model checking. In Formal Methods in Computer-Aided Design (FMCAD), 2013, pages 53–60, Oct 2013.

[Cruz2015] Cruz, T., Barrigas, J., Proença, J., Graziano, A., Panzieri, S., Lev, L., & Simões, P. (2015, May). Improving network security monitoring for industrial control systems. In Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on (pp. 878-881). IEEE.

[CSRB13] Chih-Hong Cheng, Natarajan Shankar, Harald Ruess, and Saddek Bensalem. EFSMT: A logical framework for cyber-physical systems. CoRR, abs/1306.3456, 2013.

[DEPB12] Davis, J., Edgar, T., Porter, J., Bernaden, J., Sarli, M. Smart Manufacturing, Manufacturing Intelligence and Demand-Dynamic Performance, Comput. Chem. Eng. 47 (2012) 145–156.

[DGG+05] A. Dahan, D. Geist, L. Gluhovsky, D. Pidan, G. Shapir, Y. Wolfsthal, L. Benalycherif, R. Kamidem, and Y. Lahbib. Combining system level modeling with assertion-based verification. In Proc. of ISQED 2005: Sixth International Symposium on Quality of Electronic Design, pages 310–315. IEEE, 2005.

[FK09] B. Finkbeiner and L. Kuhtz. Monitor circuits for ltl with bounded and unbounded future. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5779 LNCS:60–75, 2009.

[FK15] Daniel J. Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. Transportation Research Part A: Policy and Practice, 77:167–181, 2015.

[FOMA09] I. N. Fovino, M. Masera, and A. D. Cian, “Integrating Cyber Attacks within Fault Trees,” Rel. Eng. & Sys. Safety, pp. 1394–1402, 2009.

[GABK16] Gabor, T., Belzner, L., Kiermeier, M. A Simulation-Based Architecture for Smart Cyber-Physical Systems. IEEE International Conference on Autonomic Computing (ICAC) 2016:374–379. (2016)

[GRIE14] M. Grieves. Digital Twin: Manufacturing Excellence through Virtual Factory Replication, 2014.

[GRVI17] Grieves, M., Vickers, J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. in Kahlen F-J. Flumerfelt, S., Alves, A., (Eds.) Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches, Springer International Publishing, pp. 85–113. (2017)

[GSRRU07] Debanjan Ghosh, Raj Sharman, H. Raghav Rao, and Shambhu Upadhyaya. Self-healing Systems - Survey and Synthesis. Decis. Support Syst., 42(4):2164–2185, January 2007.

[HAAN18] Haag, S. and Anderl, R. Digital Twin - Proof of Concept. Manufacturing Letters, (2018) Online: https://doi.org/10.1016/j.mfglet.2018.02.00

[Hel18] Edward Helmore. Uber shuts down self-driving operation in Arizona after fatal crash. The Guardian, May 23, 2018.

[Hol18] Peter Holley. Chrysler Fiat announces recall of nearly 5 million U.S. cars. The Guardian, May 25, 2018.

[HOLN17] Hochhalter, J., Leser, W.P., Newman, J.A. Coupling Damage-Sensing Particles to the Digital Twin Concept. NASA Center for AeroSpace Information. (2014)

[IHS18] IHS Markit, Autonomous Vehicle Sales to Surpass 33 Million Annually in 2040, Enabling New Autonomous Mobility in More Than 26 Percent of New Car Sales, January 2, 2018

[Ise06] Rolf Isermann. Fault-diagnosis systems: an introduction from fault detection to fault tolerance. Springer Science & Business Media, 2006.

[JAZD14] Jazdi, N. Cyber Physical Systems in the Context of Industry 4.0, In: Autom. Qual. Testing, Robot. 2014 IEEE Int. Conf., 2014: pp. 1–4.

[JBG+15] Stefan Jakšić, Ezio Bartocci, Radu Grosu, Reinhard Kloibhofer, Thang Nguyen, and Dejan Ničković. From signal temporal logic to FPGA monitors. In 13. ACM/IEEE International Conference on Formal Methods and Models for Codesign MEMOCODE 2015, Austin, TX, USA, September 21-23, 2015, pages 218–227, 2015.

[KAWJ13] Kagermann, H., Wahlster, W., Johannes, H. Recommendations for Implementing the Strategic [KHJM99] Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., Van Brussel, H. Reconfigurable Manufacturing Systems, Annals of the CIRP, 48 (2), 527-540, (1999).

[KLB+17] Samuel Kounev, Peter Lewis, Kirstie L. Bellman, Nelly Bencomo, Javier Camara, Ada Diaconescu, Lukas Esterle, Kurt Geihs, Holger Giese, Sebastian Götz, Paola Inverardi, Jeffrey O. Kephart, and Andrea Zisman. The Notion of Self-aware Computing, pages 3–16. Springer International Publishing, Cham, 2017.

[KLCK16] Kang, H.S., Lee, J.Y., Choi, S., Kim, H., Park, J.H., Son, J.Y., Kim, B.H., Noh, S.D. Smart Manufacturing: Past Research, Present Findings, and Future Directions. International Journal of Precision Engineering and Manufacturing-Green Technology, 3, pp. 111–128 (2016)

[Kop11] Hermann Kopetz. Real-Time Systems: Design Principles for Distributed Embedded Applications. Springer, New York, 2nd edition, 2011.

[KOSH10] Koren, Y., Shpitalni, M. Design of Reconfigurable Manufacturing Systems. Journal of Manufacturing Systems, 29(4), 130-141. (2010)

[KRAF16] Kraft, E.M. The Air Force Digital Thread/Digital Twin - Life Cycle Integration and Use of Computational and Experimental Knowledge, in: 54th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, 2016.

[L+13] R. de Lemos et al. Software Engineering for Self-Adaptive Systems: A Second Research Roadmap, pages 1–32. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Lap08] Jean-Claude Laprie. From Dependability to Resilience. In Dependable Systems and Networks (DSN 2008), 38th Annual IEEE/IFIP International Conference, 2008.

[LEBK15] Lee, J., Bagheri, B., Kao, H. A Cyber-Physical Systems Architecture for Industry 4.0-based Manufacturing Systems, Manuf. Lett. 3, 18–23. (2015)

[LEE08] Lee, E.A. Cyber Physical Systems: Design Challenges. In 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, 363–369. (2008)

[LEKY14] Lee, J., Kao, H., Yang, S. Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment. In: Procedia CIRP, Elsevier B.V., 2014: pp. 3–8.

[LFKF14] Lasi, H., Fettke, P., Kemper, H. G., Feld, T., Hoffmann, M. Industry 4.0. Business & Information Systems Engineering, 6(4), 239. (2014)

[LPY97] Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. UPPAAL in a nutshell. STTT, 1(1-2):134–152, 1997.

[LS09] Martin Leucker and Christian Schallhart. A brief account of runtime verification. The Journal of Logic and Algebraic Programming, 78(5):293 – 303, 2009.

[LS10] Edward A. Lee and Sanjit A. Seshia. An introductory textbook on cyber-physical systems. In Proceedings of the 2010 Workshop on Embedded Systems Education, WESE '10, pages 1:1–1:6, New York, NY, USA, 2010. ACM

[LSC+ 12] Insup Lee, Oleg Sokolsky, Sanjian Chen, John Hatcliff, Eunkyoung Jee, BaekGyu Kim, Andrew King, Margaret Mullen-Fortino, Soojin Park, Alexander Roederer, and Krishna K. Venkatasubra- manian. Challenges and research directions in medical cyber- physical systems. Proceedings of the IEEE, 100(1):75–90, 2012.

[MC14] Robert Mitchell and Ing-Ray Chen. A survey of intrusion detection techniques for cyber-physical systems. ACM Computing Surveys (CSUR), 46(4):55, 2014.

[McKin16] Paul Gao, Hans-Werner Kaas, Detlev Mohr und Dominik Wee, Automotive revolution – perspective towards 2030, McKinsey & Company Report, https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/disruptive-trends-that-will-transform-the-auto-industry/de-de, 2016

[MOEL16] Möller, D.P. Digital Manufacturing/Industry 4.0. In: Guide to Computing Fundamentals in Cyber-Physical Systems, 307-375. Springer International Publishing. (2016).

[MONO14] Monostori, L. Cyber-Physical Production Systems: Roots, Expectations and R&D Challenges. In Variety Management in Manufacturing Proceedings of the 47th CIRP Conference on Manufacturing Systems 17, 9–13. (2014)

[Mou15] Jad Mouawad. F.A.A. Orders Fix for Possible Power Loss in Boeing 787. New York Times, May 1, 2015.

[MRS17] Patrick Moosbrugger, Kristin Y. Rozier, and Johann Schumann. R2U2: monitoring and diagnosis of security threats for unmanned aerial systems. Formal Methods in System Design, 51(1):31–61, 2017.

[NEFM17] Negri, E., Fumagalli, L., Macchi, M. A Review of the Roles of Digital Twin in CPS-based Production Systems. In Proceedings of the 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy. Procedia Manuf 2017; 11:939–48. 2017.

[NXP] NXP IoT Whitepaper: From the INTERNET of THINGS to the INTERNET of TRUST

[PD11] Harald Psaier and Schahram Dustdar. A survey on self-healing systems: approaches and systems. Computing, 91(1):43–73, Jan 2011.

[PE16] Zoltan Papp and George Exarchakos, editors. Runtime Reconfiguration in Networked Embedded Systems - Design and Testing Practices. Internet of Things - Technology, Communications and Computing. Springer Science+Business Media Singapore, 2016.

[Raj12] Ragunathan Rajkumar. A cyber-physical future. Proceedings of the IEEE, 100(Special Centennial Issue):1309–1312, 2012.

[REMM13] Reifsnider, K., Majumdar, P., Multiphysics Stimulated Simulation Digital Twin Methods for Fleet Management. 54th AIAA/ASME/ ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 1578. (2013)

[RHOM15] Rios, J., Hernandez,, J.C., Oliva,, M., and Masb, F. Product Avatar as Digital Counterpart of a Physical Individual Product: Literature Review and Implications in an Aircraft System. In: ISPE CE: pp. 657–666. (2015).

[RKG+18] Denise Ratasich, Faiq Khalid, Florian Geissler, Radu Grosu, Muhammad Shafique, Ezio Bartocci. A Roadmap Towards Resilient Internet of Things for Cyber-Physical Systems. arXiv preprint arXiv:1810.06870, 2018.

[RLSS10] Ragunathan (Raj) Rajkumar, Insup Lee, Lui Sha, and John Stankovic. Cyber-physical systems: The next computing revolution. In Proc. of DAC '10: the 47th Design Automation Conference, pages 731–736, New York, NY, USA, 2010. ACM.

[RMJP16] Razzaque, M.A., Milojevic-Jevric, M., Palade, A. and Clarke, S. Middleware for Internet of Things: A Survey. In IEEE Internet of Things Journal, Vol. 3, No. 1 (2016)

[ROWL15] Rosen, R., von Wichert, G., Lo, G., Bettenhausen, K.D. About the Importance of Autonomy and Digital Twins for the Future of Manufacturing. IFAC- PapersOnLine 2015;48(3):567–72. 2015.

[RRS14] Thomas Reinbacher, Kristin Y. Rozier, and Johann Schumann. Temporal-logic based runtime observer pairs for system health management of real-time systems. In Proc. of TACAS 2014, volume 8413 of LNCS, pages 357–372. Springer-Verlag, 2014.

[Ruprechter2017] Ruprechter, Th. "Industrie 4.0" University of Applied Science Burgenland Lecture Master Program "Cloud Computing", 2017.

[SABA00] Sarma, S., Brock, D.L., Ashton, K. The Networked Physical World, 2000.

[SAMW17] Schleich, B., Anwer, N., Mathieu, L., Wartzack, S. Shaping the Digital Twin for Design and Production Engineering. In CIRP Annals - Manufacturing Technology. Pp. 141-144. (2017)

[SB10] S. Siva Sathya and K. Syam Babu. Survey of fault tolerant techniques for grid. Computer Science Review, 4(2):101 – 120, 2010.

[SCRO16] Schluse, M. , Rossmann, J. From Simulation to Experimentable Digital Twins - Simulation-based Development and Operation of Complex Technical Systems. In: Second IEEE International Symposium on Systems Engineering (ISSE 2016), October 3–5, Edinburgh, Scotland, pp. 273–278, IEEE, 2016.

[SHAF10] Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J., Wang, L. NASA Technology Roadmap: DRAFT Modeling, Simulation, Information Technology & Processing Roadmap Technology Area. (2010)

[SHAF12] Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J., Wang, L. NASA Technology Roadmap: Modeling, Simulation, Information Technology & Processing Roadmap Technology Area. (2012)

[STLI13] M. Steiner and P. Liggesmeyer, "Combination of Safety and Security Analysis - Finding Security Problems That Threaten The Safety of a System," in SAFECOMP, 2013.

[Stojmenovic2014] Stojmenovic, I., & Wen, S. (2014, September). The Fog computing paradigm: Scenarios and security issues. In Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on (pp. 1-8). IEEE.

[TAHO18] Tavčar, J. and Horváth, I. A Review of the Principles of Designing Smart Cyber-Physical Systems for Run-Time Adaptation: Learned Lessons and Open Issues. In IEEE Transactions on Systems, Man, and Cybernetics: Systems. (2018) Online available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8329014&isnumber=6376248

[VFG+11] Ovidiu Vermesan, Peter Friess, Patrick Guillemin, Sergio Gusmeroli, Harald Sundmaeker, Alessandro Bassi, Ignacio Soler Jubert, Margaretha Mazura, Mark Harrison, Markus Eisenhauer, et al. Internet of things strategic research roadmap. Internet of Things-Global Technological and Societal Trends, 1(2011):9–52, 2011.

[VI40O18] Verein Industrie 4.0 Österreich: Ergebnispapier „Forschung, Entwicklung und Innovation in der Industrie 4.0. Prioritäre Forschungsfelder & Maßnahmen zur Förderung der österreichischen Forschungslandschaft, Stand Juli 2018.

[WEGE17] Wegener, D. “Industrie 4.0” bedeutet die Verschmelzung von Office Floor mit Shop Floor. Zvei Die Elektoindustrie. (2017)

[Wey17] Danny Weyns. Software Engineering of Self-Adaptive Systems: An Organised Tour and Future Challenges. Springer, 2017.

[WGL+16] W. Eric Wong, Ruizhi Gao, Yihao Li, Rui Abreu, and Franz Wotawa. A survey on software fault localization. IEEE Trans. Software Eng., 42(8):707–740, 2016.

[WWLZ16] Wang, S., Wan, J., Li, D., Zhang, C. Implementing Smart Factory of Industrie 4.0: An Outlook. International Journal of Distributed Sensor Networks, 7. (2016)

[YOSS12] Yoon, J. S., Shin, S. J., & Suh, S. H. A Conceptual Framework for the Ubiquitous Factory. International Journal of Production Research, 50(8), 2174-2189. (2012)

[ZUEH10] Zuehlke, D. SmartFactory - Towards a Factory-of-Things. In Annual Review in Control. (2010). Online: http://foresight.ifmo.ru/ict/shared/files/201311/1_124.pdf