

Master Thesis:

Design and Development of a Fog Service  
Orchestration Engine for Smart Factories

Master Thesis  
from

Markus Paeschke

Supervisor: Prof. Dr.-Ing. Thomas Magedanz  
Dr.-Ing. Alexander Willner  
Mathias Brito

# Acknowledgments

thank your supervisors

thank your colleagues

thank your family and friends

Berlin, February 20, 2017

Markus Paeschke  
Trachtenbrodtstr. 32  
10409 Berlin

I hereby declare that the following thesis “Design and Development of a Fog Service Orchestration Engine for Smart Factories” has been written only by the undersigned and without any assistance from third parties.

Furthermore, I confirm that no sources have been used in the preparation of this thesis other than those indicated in the thesis itself.

Berlin, February 20, 2017

---

Markus Paeschke



# Contents

<b>List of Figures</b>	<b>iii</b>
<b>List of Tables</b>	<b>iv</b>
<b>Quellcodeverzeichnis</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background and Motivation . . . . .	1
1.2 Problem Statement . . . . .	1
1.3 Assumptions and Scope . . . . .	1
1.4 Objectives and Contributions . . . . .	1
1.5 Methodology and Outline . . . . .	1
<b>2 State of the art</b>	<b>2</b>
2.1 Internet of Things . . . . .	2
2.1.1 Industry 4.0 and Smart Factories . . . . .	3
2.1.2 Cyber Physical Systems . . . . .	5
2.1.3 Fog Computing . . . . .	5
2.2 Virtualization . . . . .	5
2.2.1 Virtual Machines . . . . .	6
2.2.2 Container Virtualization . . . . .	6
2.2.3 Container Orchestration . . . . .	7
2.2.4 Network Function Virtualization . . . . .	7
2.3 Conclusion . . . . .	7
<b>3 Requirements Analysis</b>	<b>8</b>
3.1 Introduction . . . . .	8
3.2 System requirements . . . . .	8
3.3 Technologies . . . . .	8
3.4 Use-Case-Analysis . . . . .	8
3.5 Delineation from existing solutions . . . . .	8
3.6 Conclusion . . . . .	8
<b>4 Design</b>	<b>9</b>
4.1 Introduction . . . . .	9
4.2 Development environment . . . . .	9
4.3 Evaluation of existing frameworks . . . . .	9
4.3.1 Docker . . . . .	9

4.3.2	Docker Swarm . . . . .	9
4.3.3	Kubernetes . . . . .	9
4.3.4	Open Baton . . . . .	9
4.3.5	ETSI MANO . . . . .	9
4.3.6	TOSCA . . . . .	9
4.4	Architecture of the system . . . . .	9
4.4.1	Orchestration layer . . . . .	9
4.4.2	Constraint layer . . . . .	9
4.4.3	User interface . . . . .	9
4.5	Conclusion . . . . .	9
<b>5</b>	<b>Implementation</b>	<b>10</b>
5.1	Introduction . . . . .	10
5.2	Project structure . . . . .	10
5.3	Used external libraries . . . . .	10
5.4	Custom code . . . . .	10
5.5	Implementation of the orchestration layer . . . . .	10
5.6	Implementation of the constraint layer . . . . .	10
5.7	Implementation of the user interface . . . . .	10
5.8	Conclusion . . . . .	10
<b>6</b>	<b>Evaluation</b>	<b>11</b>
6.1	Introduction . . . . .	11
6.2	Experimental Validation . . . . .	11
6.3	Performance Evaluation . . . . .	11
6.4	Observational Validation . . . . .	11
6.5	Deployments . . . . .	11
6.6	Code Verification . . . . .	11
6.7	Comparative Analysis . . . . .	11
6.8	Conclusion . . . . .	11
<b>7</b>	<b>Summary and Further Work</b>	<b>12</b>
7.1	Overview . . . . .	12
7.2	Conclusion and Impact . . . . .	12
7.3	Outlook . . . . .	12
	<b>Acronyms</b>	<b>I</b>
	<b>Glossary</b>	<b>II</b>
	<b>Bibliography</b>	<b>III</b>

# List of Figures

1	Horizontal vs. Vertical Integration . . . . .	4
2	Structure traditional VMs vs. Docker . . . . .	7

# List of Tables

1	Design principles of each Industry 4.0 component . . . . .	3
---	--	---



# List of Listings

# Chapter 1

## Introduction

1.1 Background and Motivation

1.2 Problem Statement

1.3 Assumptions and Scope

1.4 Objectives and Contributions

1.5 Methodology and Outline

# Chapter 2

## State of the art

This chapter will give an overview into the background and concepts of this thesis. In the first section the Internet of Things and related subtopics like Smart Factories and Smart Cities are considered. Cyber Physical Systems, which are important for the development of Smart Factories are also covered in this section. Virtualization in general is the main topic of the second section. First we dive into the area of Virtual Machines, followed by Container Virtualization. Both are related to each other and sharing some basic ideas. Container Orchestration as an own subsection shows some possibilities of Container Virtualization. The last subsection Network Function Virtualization concludes with an introduction into the virtualization of network node functions to create communication services.

### 2.1 Internet of Things

The Internet of Things (IoT) has been a subject of great media- and economically growth in the recent years. In the year 2008 the number of devices which are connected to the Internet was higher than the human population.[Eva11, cf.] Cisco Internet Business Solutions Group predicted that the number will grow up to 50 billion in 2020, this equates to around 6 devices per person.[Eva11, cf.] Most of today's interactions are Human-to-Human (H2H) or Human-to-Machine (H2M) communication. The IoT on the other hand aims for the Machine-to-Machine (M2M) communication. This allows every physical device to be interconnected and to communicate with each other. These devices are also called "Smart Devices". Creating a network where all physical objects and people are connected via software is one primary goal of the IoT.[RD15, cf.][KKL13, cf.] When objects are able to capture and monitor their environment, a network can perceive external stimuli and respond to them.[cf. Itu05, p. 40] Therefore a new dimension of information and communication technology will be created, where users have access to everything at any time, everywhere. In addition to smart devices, subcategories are also emerging from the IoT which, in addition to the physical devices, also describe technologies such as protocols and infrastructures. The "Smart Home" has been a prominent topic in media and business for many years. Smart City or Industrie 4.0 are also becoming established and are increasingly popular. But the Internet started with the appearance of bar codes and Radio Frequency Identification (RFID) chips.[KKL13, cf.] The second step, which is more or less the current situation, sensors, physical devices, technical devices, data and software are connected to each other.[KKL13, cf.] This was achieved, in particular,

by cloud computing, which provides the highly efficient memory and computing power that is indispensable for such networks.[RD15, cf.] The next step could be a "Cognitive Internet of Things", which enables easier object and data reuse across application areas, for example through interoperable solutions, high-speed Internet connections and a semantic information distribution.[KKL13, cf.] Just as the omnipresent information processing in everyday life, also known as "Ubiquitous Computing", which was first mentioned in the "The Computer for the 21st Century"[Wei91, cf.] by Marks Weiser, it will take some time until it is ubiquitous.

### 2.1.1 Industry 4.0 and Smart Factories

The industry as an changing environment is currently in the state of the so called "fourth industrial revolution". The first industrial revolution was driven by steam powered machines. Mass production and division of labor was the primary improvement of the second industrial revolution, whereas the third revolution was characterized by using electronics and the integration of Information Technology (IT) into manufacturing processes.[cf. LPS16, p. 1] In the recent years the size, cost and power consumption of chipsets are reduced which made it possible to embed sensors into devices and machines much easier and cheaper.[cf. BHSW16, p. 1] The "Industrie 4.0" is the fourth step in this evolution and was first mentioned at the Hannover Fair in 2011.[cf. LPS16, p. 1] "Industrie 4.0 is a collective term for technologies and concepts of value chain organization."[cf. HPO15, p. 11]

Significantly higher productivity, efficiency, and self-managing production processes where everything from machines up to goods can communicate and cooperate with each other directly are the visions of the Industry 4.0.[Lyd16, cf.] It also aims for an intelligent connection between different companies and units. Autonomous production and logistics processes creating a real-time lean manufacturing ecosystem that is more efficient and flexible.[Lyd16, cf.] "This will facilitate smart value-creation chains that include all of the life-cycle phases of the product from the initial product idea, development, production, use, and maintenance to recycling."[Lyd16] At the end, the system can use customer wishes in every step in the process to be flexible and responsive.[Lyd16, cf.]

	Cyber-Physical Systems	Internet of Things	Internet of Services	Smart Factory
Interoperability	X	X	X	X
Virtualization	X	-	-	X
Decentralization	X	-	-	X
Real-Time Capability	-	-	-	X
Service Orientation	-	-	X	-
Modularity	-	-	X	-

Table 1: Design principles of each Industry 4.0 component.[cf. HPO15, p. 11]

Table 1 shows the six design principles which can be from the Industrie 4.0 components. They can help companies to identify and implement Industry 4.0 scenarios.[cf. HPO15, p. 11]

1. *Interoperability* Cyber-Physical System (CPS) of various manufacturers are connected with each other. Standards will be the key success factor in this area.[cf. HPO15, p. 11]
2. *Virtualization* CPS are able to monitor physical processes via sensors. The resulting

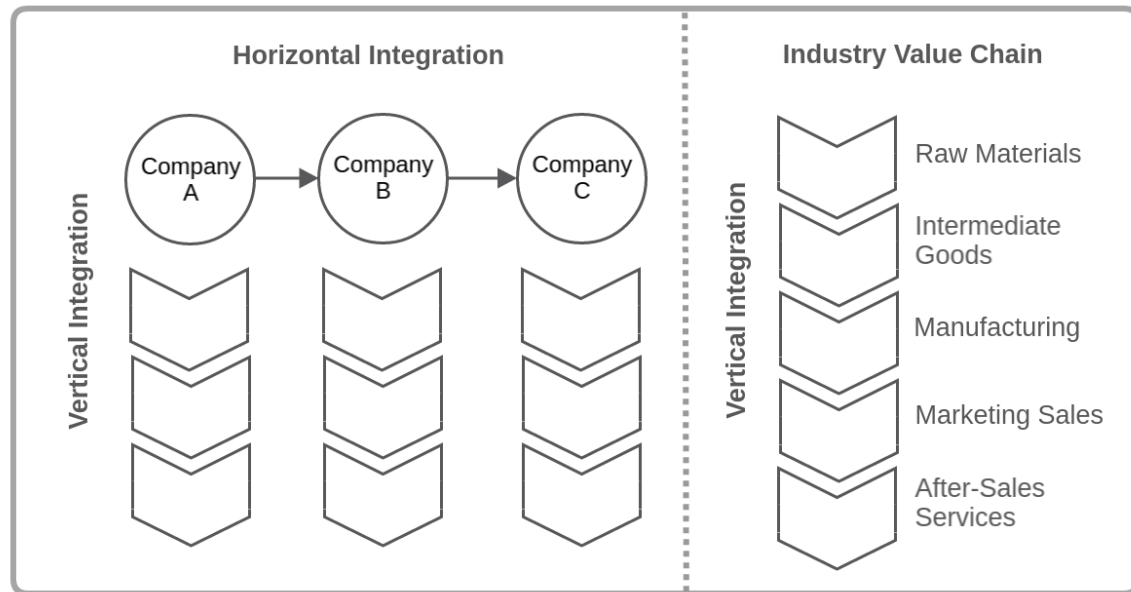


Figure 1: Horizontal vs. Vertical Integration. Adapted from: [Jur13]

data is linked to virtual plant and simulation models. These models are virtual copies of physical world entities.[cf. HPO15, p. 11]

3. *Decentralization* CPS are able to make decisions on their own, for example when RFID chips send the necessary working steps to the machine. Only in cases of failure the systems delegate task to a higher level.[cf. HPO15, p. 11]
4. *Real-Time Capability* Data has to be collected and analyzed in real time and the status of the plant is permanently tracked and analyzed. This enables the CPS to react to a failure of a machine and can reroute the products to another machine.[cf. HPO15, p. 11]
5. *Service Orientation* CPS are available over the Internet of Services (IoS) and can be offered both internally and across company borders to different participants. The manufacturing process can be composed based on specific customer requirements.[cf. HPO15, p. 11]
6. *Modularity* The system is able to be adjusted in case of seasonal fluctuations or changed product characteristics, by replacing or expanding individual modules.[cf. HPO15, p. 11]

Another important aspect of Industry 4.0 is the implementation of process automation with the focused on three distinct aspects. Starting with the vertical integration, which contains the connection and communication of subsystems within the factory enables flexible and adaptable manufacturing systems.[cf. Vbw14, p. 7 ff.] The horizontal integration, as the second aspect, enables technical processes to be integrated in cross-company business processes and to be synchronized in real time through multiple participants to optimize value chain outputs.[cf. Vbw14, p. 7 ff.] Finally end-to-end engineering, planning, and process control for each step in the production process.[Lyd16, cf.]

Figure 1 illustrates this concept. The left side shows the whole production process over company boundaries on the horizontal scale, as well as the industry value chain on the vertical scale which is specific for each company. On the right side there is an exemplary industry value chain which starts with the raw materials and ends with the sale of the product to illustrates an more specific example of the vertical integration.

guter abschluss -> moving production steps to each machine -> modular, without core system, locally in the modules -> communication path growth shorter -> self organisation -> decentralized

as well as highly flexible, individualized and resource friendly mass production

### 2.1.2 Cyber Physical Systems

As we already now, in smart factories every physical device is connected to each other. Everything can be captured and monitored in each step of a production process. With CPSs every physical entity has a digital representation in the virtual system.[cf. Poo10, p. 1363] Before a Cyber System (CS) was passive, which means it has no communication between the physical and the virtual world.[cf. Poo10, p. 1364] While new technologies in the physical world, like new materials, hardware and energy, are developed, the technologies in the virtual worlds are also being improved, for example through the use of new protocols, networking, storage and computing technologies.[cf. Poo10, p. 1364] This adds more intelligence in such systems, as well as a much more flexible and modular structure. A CPS can organize production automatically and autonomously, which eliminate the need of having a central process control.[? , cf.] Thereby the system can handle lossy signals and short range radio technologies, which are widely used in such a context.[YMSG<sup>+</sup>14, cf.] **write more**

### 2.1.3 Fog Computing

## 2.2 Virtualization

According to the National Institute of Standards and Technology (NIST) the definition of virtualization is: "Virtualization is the simulation of the software and/or hardware upon which other software runs. This simulated environment is called a virtual machine (VM)."[SSH11]. This means a Virtual Machine (VM), also referred as guest system, can be executed in a real system, which is referred as host system. Basically there are two types of virtualization: Process virtualization where the virtualizing software also known as Virtual Machine Monitor (VMM) is executed by the host Operating System (OS) and only an application will be executed inside the guest OS and on the other side there is the the system virtualization where the whole OS as well as the application are running inside the virtualizing software. Figure 1 illustrate both concepts. **right figure** Examples for process virtualization could be the Java Virtual Machine, the .Net framework or Docker, where VMWare, Oracle Virtual Box, XEN or Microsoft Hyper-V are only some examples for system virtualization. **footnotes or bibs?** The benefits of all virtualization techniques are the rapid provisioning of resources which could be Random Access Memory (RAM), disk storage, computation power or network bandwidth. Beside that, no human interaction is necessary during the provisioning process. Elasticity which scales a system in a cost-efficient manner in both directions, up and down.

Customer as well as the provider profit from such a system. Security based on the isolation of the VMs is another huge benefit. Different processes can not interfere with each other and the data of a single user can not be accessed by other users of the same hardware. A challenge despite all the mentioned benefits is the performance. Running VMs increases the overhead and reduces the overall performance of a system. Therefore the specific use case have to consider these behavior.

### 2.2.1 Virtual Machines

VMs are the core virtualization mechanism in cloud computing. There are also two different designs for hardware virtualization. The first and more popular type for cloud computing is the *bare-metal virtualization*. It needs only a basic OS to schedule VMs. The hypervisor runs directly on the hardware of the machine. This is more efficient, but requires special device drivers to be executed. The other type is the *hosted virtualization*. Unlike the first type the VMM run as a host OS process and the VMs as a process supported by the VMM. No special drivers are needed for these type of virtualization, but by comparison the overhead is much bigger. For both types, the performance limitation remains. Each VM need a full guest OS image in addition to binaries and libraries which are necessary for the application to be executed.[cf. PL15, p. 381] If only a single application should be executed which only need some binaries and libraries, these virtualization method is too bloated.

### 2.2.2 Container Virtualization

Container virtualization which is also known as System-level virtualization, is the second virtualization machanism. It based on fast and lightweight process virtualization to encapsulate an entire application with its dependencies into a ready-to-deploy virtual container.[cf. TRA15, p. 72]

a.[CMF<sup>+</sup>16] b.[AHA<sup>+</sup>16]

#### Linux Containers

a.[SPF<sup>+</sup>07]

#### Docker

"First, we must know what exactly Docker is and does. Docker is a container management system that helps easily manage Linux Container (LXC) in an easier and universal fashion. This lets you create images in virtual environment ons your laptop and run commands or operations against them. The actions you do to the containers that you run in these environments locally on your own machine will be the same commands or operations you run against them when they are running in your production environment. This helps in not having to do things differently when you go from a development environment like that on your local machine to a production environment like that on your local machine to a production environment on your server. Now, let's take a look at the differences between Docker containers and the typical virtual machine environments.

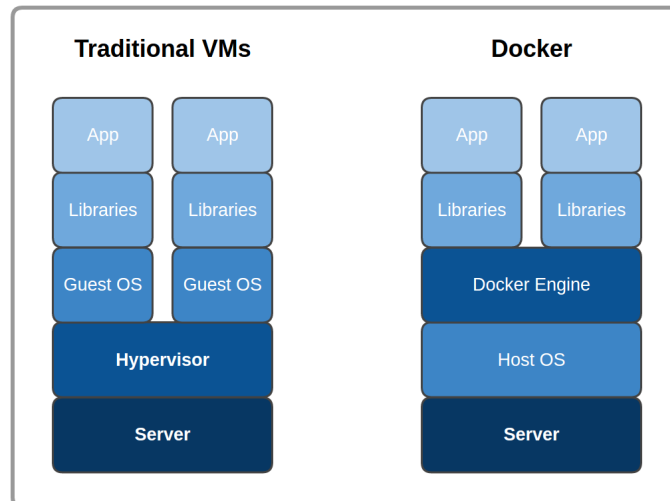


Figure 2: Structure traditional VMs vs. Docker. Adapted from: [Gal15, p. 2]

In the following illustration, we can see the typical Docker setup on the right-hand side versus the typical VM setup on the left-hand side:

"This illustration gives us a lot of insight into the biggest key benefit of Docker, that is, there is no need for a complete operating system every time we need to bring up a new container, which cuts down on the overall size of containers. Docker relies on using the host OS's Linux kernel (since almost all the versions of Linux use the standard kernel models) for the OS it was built upon, such as Red Hat, CentOS, Ubuntu, and so on. For this reason, you can have almost any Linux OS as your host operating system (Ubuntu in the previous illustration) and be able to layer other OSes on top of the host. For example, in the earlier illustration, we could have Red Hat running for one app (the one on the left) and Debian running for the other app (the one on the right), but there would never be need to actually install Red Hat or Debian on the host. Thus, another benefit of Docker is the size of an images when they are born. They are not built with the largest piece: the kernel or the operating system. This makes them incredibly small, compact, and easy to ship." [Gal15]

## 2.2.3 Container Orchestration

Kubernetes

Docker Swarm

## 2.2.4 Network Function Virtualization

## 2.3 Conclusion



## Chapter 3

# Requirements Analysis

3.1 Introduction

3.2 System requirements

3.3 Technologies

3.4 Use-Case-Analysis

3.5 Delineation from existing solutions

3.6 Conclusion

# Chapter 4

## Design

### 4.1 Introduction

### 4.2 Development environment

### 4.3 Evaluation of existing frameworks

#### 4.3.1 Docker

#### 4.3.2 Docker Swarm

#### 4.3.3 Kubernetes

#### 4.3.4 Open Baton

#### 4.3.5 ETSI MANO

#### 4.3.6 TOSCA

### 4.4 Architecture of the system

#### 4.4.1 Orchestration layer

#### 4.4.2 Constraint layer

#### 4.4.3 User interface

### 4.5 Conclusion

## Chapter 5

# Implementation

5.1 Introduction

5.2 Project structure

5.3 Used external libraries

5.4 Custom code

5.5 Implementation of the orchestration layer

5.6 Implementation of the constraint layer

5.7 Implementation of the user interface

5.8 Conclusion

## Chapter 6

# Evaluation

6.1 Introduction

6.2 Experimental Validation

6.3 Performance Evaluation

6.4 Observational Validation

6.5 Deployments

6.6 Code Verification

6.7 Comparative Analysis

6.8 Conclusion

## Chapter 7

# Summary and Further Work

### 7.1 Overview

### 7.2 Conclusion and Impact

### 7.3 Outlook

## Acronyms

<b>CPS</b>	Cyber-Physical System
<b>CS</b>	Cyber System
<b>H2H</b>	Human-to-Human
<b>H2M</b>	Human-to-Machine
<b>IoS</b>	Internet of Services
<b>IoT</b>	Internet of Things
<b>IT</b>	Information Technology
<b>M2M</b>	Machine-to-Machine
<b>NIST</b>	National Institute of Standards and Technology
<b>OS</b>	Operating System
<b>RAM</b>	Random Access Memory
<b>RFID</b>	Radio Frequency Identification
<b>VM</b>	Virtual Machine
<b>VMM</b>	Virtual Machine Monitor

## Glossary

Algorithmus a

Chiffrierung a

Dechiffrierung a

# Bibliography

- [AHA<sup>+</sup>16] ANDERSON, J. ; HU, H. ; AGARWAL, U. ; LOWERY, C. ; LI, H. ; APON, A.: Performance considerations of network functions virtualization using containers. In: *2016 International Conference on Computing, Networking and Communications (ICNC)*, 2016, S. 1–7
- [BHSW16] BRITO, M. S. D. ; HOQUE, S. ; STEINKE, R. ; WILLNER, A.: Towards Programmable Fog Nodes in Smart Factories. In: *2016 IEEE 1st International Workshops on Foundations and Applications of Self\* Systems (FAS\*W)*, 2016, S. 236–241
- [CMF<sup>+</sup>16] CELESTI, A. ; MULFARI, D. ; FAZIO, M. ; VILLARI, M. ; PULIAFITO, A.: Exploring Container Virtualization in IoT Clouds. In: *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2016, S. 1–6
- [Eva11] EVANS, D.: The Internet of Things: How the Next Evolution of the Internet Is Changing Everything / Cisco Internet Business Solutions Group (IBSG). Version: April 2011. [http://www.cisco.com/c/dam/en\\_us/about/ac79/docs/innov/IoT\\_IBSG\\_0411FINAL.pdf](http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf). 2011. – Forschungsbericht. – Accessed: 2017-02-12
- [Gal15] GALLAGHER, S.: *Mastering Docker*. Packt Publishing Ltd., 2015. – ISBN 978-1-78528-703-9
- [HPO15] HERMANN, M. ; PENTEK, T. ; OTTO, B.: Design Principles for Industrie 4.0 Scenarios: A Literature Review / Technische Universität Dortmund - Fakultät Maschinenbau. Version: Jan 2015. [http://www.thiagobranquinho.com/wp-content/uploads/2016/11/Design-Principles-for-Industrie-4\\_0-Scenarios.pdf](http://www.thiagobranquinho.com/wp-content/uploads/2016/11/Design-Principles-for-Industrie-4_0-Scenarios.pdf). 2015. – Forschungsbericht. – Accessed: 2017-02-12
- [Itu05] INTERNATIONAL TELECOMMUNICATION UNION: ITU Internet Reports: The Internet of Things. Version: November 2005. <https://www.itu.int/net/wsis/tunis/newsroom/stats/The-Internet-of-Things-2005.pdf>. 2005. – Forschungsbericht. – Accessed: 2017-02-12
- [Jur13] JUREVICIUS, O.: *Vertical Integration*. <https://www.strategicmanagementinsight.com/topics/vertical-integration.html>. Version: April 2013. – Accessed: 2017-02-13
- [KKL13] KRAMP, T. ; KRANENBURG, R. van ; LANGE, S.: Introduction to the Internet of



- Things. In: *Enabling Things to Talk* Bd. 1, Springer-Verlag Berlin Heidelberg, 2013. – ISBN 978–3–642–40402–3, S. 1–10
- [LPS16] LOM, M. ; PRIBYL, O. ; SVITEK, M.: Industry 4.0 as a part of smart cities. In: *2016 Smart Cities Symposium Prague (SCSP)*, 2016, S. 1–6
- [Lyd16] LYDON, B.: Industry 4.0: Intelligent and flexible production. In: *InTech Magazine* (2016), May-June. <https://www.isa.org/intech/20160601/>. – Accessed: 2017-02-13
- [PL15] PAHL, C. ; LEE, B.: Containers and Clusters for Edge Cloud Architectures – A Technology Review. In: *2015 3rd International Conference on Future Internet of Things and Cloud*, 2015, S. 379–386
- [Poo10] POOVENDRAN, R.: Cyber Physical Systems: Close Encounters Between Two Parallel Worlds. In: *Proceedings of the IEEE* 98 (2010), Aug, Nr. 8, S. 1363–1366. <http://dx.doi.org/10.1109/JPROC.2010.2050377>. – DOI 10.1109/JPROC.2010.2050377. – ISSN 0018–9219
- [RD15] RUI, J. ; DANPENG, S.: Architecture Design of the Internet of Things Based on Cloud Computing. In: *2015 Seventh International Conference on Measuring Technology and Mechatronics Automation*, 2015. – ISSN 2157–1473, S. 206–209
- [SPF<sup>+</sup>07] SOLTESZ, S. ; PÖTZL, H. ; FIUCZYNSKI, M. E. ; BAVIER, A. ; PETERSON, L.: Container-based Operating System Virtualization: A Scalable, High-performance Alternative to Hypervisors. In: *Proceedings of the 2Nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*. New York, NY, USA : ACM, 2007 (EuroSys '07). – ISBN 978–1–59593–636–3, 275–287
- [SSH11] SCARFONE, K. ; SOUPPAYA, M. ; HOFFMAN, P.: Guide to Security for Full Virtualization Technologies / National Institute of Standards and Technology. Version: Jan 2011. <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-125.pdf>. 2011. – Forschungsbericht. – Accessed: 2017-02-19
- [TRA15] TOSATTO, A. ; RUIU, P. ; ATTANASIO, A.: Container-Based Orchestration in Cloud: State of the Art and Challenges. In: *2015 Ninth International Conference on Complex, Intelligent, and Software Intensive Systems*, 2015, S. 70–75
- [Vbw14] VBW VEREINIGUNG DER BAYERISCHEN WIRTSCHAFT E. V.: Dienstleistungspotenziale im Rahmen von Industrie 4.0. Version: Mar 2014. <http://www.forschungsnetzwerk.at/downloadpub/dienstleistungspotenziale-industrie-4.0-mar-2014.pdf>. 2014. – Forschungsbericht. – Accessed: 2017-02-12
- [Wei91] WEISER, M.: *The Computer for the 21st Century*. <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>. Version: September 1991. – Accessed: 2017-02-12
- [YMSG<sup>+</sup>14] YANNUZZI, M. ; MILITO, R. ; SERRAL-GRACIÀ, R. ; MONTERO, D. ; NEMIROVSKY, M.: Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing. In: *2014 IEEE 19th International Workshop*

*on Computer Aided Modeling and Design of Communication Links and Networks  
(CAMAD)*, 2014. – ISSN 2378–4865, S. 325–329

