STEFAN SCHULTE

ADAPTIVE RESOURCE AND TASK SCHEDULING FOR THE INDUSTRIAL INTERNET

TECHNISCHE
UNIVERSITÄT
WIEN

Habilitationsschrift

ADAPTIVE RESOURCE AND TASK SCHEDULING FOR THE INDUSTRIAL INTERNET

Submitted to TU Wien
Faculty of Informatics

on behalf of receiving the *venia docendi* ("Lehrbefugnis")
in the area of "Informatik"

by

STEFAN SCHULTE
Gellertplatz 4/32, 1100 Vienna, Austria
mail@stefanschulte.com

Vienna, December 6, 2017

To my girls: Luise, Marla and Nora.

ABSTRACT

The proliferation of Information and Communications Technology and the Internet of Things provides the manufacturing industry with the means to realize inter-organizational business processes which are highly flexible and dynamic. With the corresponding advent of the *Industrial Internet*, the industry is undergoing substantial transformations, leading to new requirements regarding hardware and software support. Especially, industrial system landscapes have become volatile distributed systems which need to quickly adapt to changes in the processes themselves and in their environments. Therefore, a major research challenge is to devise methods and technologies for the efficient utilization of (cloud-based) distributed computational resources in the Industrial Internet.

In the constituent publications of this habilitation thesis, we present novel concepts, methods, and technologies to efficiently exploit cloud-based computational resources for utilization in the Industrial Internet and other smart systems. The main focus is on *elastic computing*, namely the realization, configuration, and optimization of elastic processes and elastic stream processing using cloud-based resources. The developed optimization approaches aim at cost efficiency, taking Quality of Service constraints into account.

As a secondary part of this thesis, research results from the field of *Service-oriented Computing* are presented. In mobile scenarios, it is necessary to consider the needs of potentially resource-limited client devices and adapt (cloud-based and Web) services for usage on such devices. For this, solutions for the efficient usage of distributed computational resources on mobile devices with limited capabilities are realized. To achieve this, methods to select and adapt Web service invocations aiming at reduced network usage are introduced. As another research contribution in Service-oriented Computing, approaches for the cost-aware composition of complex service-based processes are discussed.

ZUSAMMENFASSUNG

Die Verbreitung von Informations- und Kommunikationstechnik und Technologien des Internet of Things ermöglichen der verarbeitenden Industrie, interorganisationale Geschäftsprozesse zu realisieren, welche hochgradig flexibel und dynamisch sind. Durch die gleichzeitige Adaption von *Industrie-4.0*-Prinzipien und -Technologien ist die Industrie maßgeblichen Transformationen ausgesetzt, welche neue Anforderungen an den Hardware- und Softwaresupport von Produktionsprozessen bedeuten. Insbesondere stellen industrielle Systemlandschaften heute volatile verteilte Systeme dar, welche sich schnell an Änderungen in den Prozessen und den Produktionsumgebungen anpassen müssen. Daher stellt die Entwicklung entsprechender Methoden und Technologien für die effiziente Nutzung (Cloud-basierter) verteilter Rechenkapazitäten eine maßgebliche Forschungsaufgabe im Bereich Industrie 4.0 dar.

In den konstituierenden Publikationen dieser Habilitationsschrift werden daher neuartige Konzepte, Methoden und Technologien zur effizienten Nutzung Cloud-basierter Rechenkapazitäten vorgestellt. Die entsprechenden Forschungsergebnisse zielen insbesondere auf Industrie-4.0-Szenarien und vergleichbare „smarte“ Umgebungen ab. Der Hauptfokus liegt auf Verfahren und Technologien des *Elastic Computing*, namentlich die Realisierung, Konfiguration und Optimierung von elastischen Prozessen und elastischer Datenstromverarbeitung mithilfe von Cloud-Ressourcen. Dabei zielen die erarbeiteten Optimierungsansätze auf Kosteneffizienz und die Einhaltung von Dienstgütevereinbarungen ab.

Im zweiten Schwerpunkt dieser Habilitationsschrift werden Forschungsergebnisse aus dem Bereich *Service-oriented Computing* präsentiert. In mobilen Szenarien ist es notwendig, die Eigenschaften von Endgeräten zu berücksichtigen, welche potentiell Ressourcenlimitierungen aufweisen. Daher müssen (Cloud-basierte und Web) Services für die Verwendung auf solchen Geräten adaptiert werden. Daher werden Konzepte für die effiziente Nutzung von Services und verteilten Rechenkapazitäten auf Geräten mit limitiertem Leistungsvermögen entwickelt. Als Lösungsansatz werden dabei Methoden zur Auswahl und Adaption von Serviceinvokationen verwendet, welche auf die Reduktion der Netzwerklast abzielen. Als weiterer Forschungsbeitrag im Bereich Service-oriented Computing werden Ansätze zur kosteneffizienten Komposition von komplexen, Service-basierten Prozessen vorgestellt.

ACKNOWLEDGMENTS

This habilitation thesis would not have been possible without my roots as a researcher at TU Darmstadt. Especially, I'd like to thank my former colleagues Rainer Berbner, Julian Eckert, Kalman Graffi, Sebastian Kaune, Ulrich Lampe, André Miede (who is – amongst other things – also responsible for the very nice thesis template I am using), Apostolos Papageorgiou, Nicolas Repp, Andreas Reinhardt, Dieter Schuller, and Ralf Steinmetz, for their role in shaping me as a researcher.

Especially at the beginning of my research career, I have benefited significantly from the expertise of Korbinian von Blanckenburg and Mathias Uslar. I'd like to thank Ingo Weber for the very fruitful collaboration we had in the last 5 years and I hope that we will be able to continue this. Further thanks go to Srikumar Venugopal for our collaboration on elastic processes.

Since joining the Distributed Systems Group at TU Wien in 2011, I was able to collaborate with a number of excellent researchers, who helped me to keep track and to improve my research output. Especially, I'd like to thank Michael Borkowski, Elena Georgiana Copil, Christoph Hochreiner, Philipp Hoenisch, Waldemar Hummer, Christian Inzinger, Lukasz Juszczyk, Roman Khazankin, Philipp Leitner, Fei Li, Vitaliy Liptchinsky, Christoph Mayr-Dorn, Daniel Moldovan, Thomas Rausch, Johannes Schleicher, Ognjen Scekic, Olena Skarlat, Svetoslav Videnov, Michael Vögler, Philipp Waibel, and Rostyslav Zabolotnyi for our manifold collaborations in research, teaching, and beyond. Many thanks go also to our secretaries Christine Kamper, Margret Steinbuch, and Renate Weiss, for their support through the years. I'd also like to thank the Bachelor and Master students I have worked with.

Outside of the Distributed Systems Group, I have collaborated with a number of inspiring researchers at TU Wien, including Ivona Brandic, Detlef Gerhard, Christian Huemer, Gerti Kappel, Wolfgang Kastner, and Manuel Wimmer, to name just a few.

Also, I'd like to thank everyone who has worked with me on and in the EU FP7 and H2020 projects ADVENTURE, SIMPLI-CITY, and CREMA. These projects gave me the opportunity to work in exciting environments and with interesting people. Especially, many thanks to Sven Abels and Stuart Campbell for getting me in touch with "EU project business".

Last but not least, I want to thank my academic advisor Schahram Dustdar for his support and for giving me the freedom to establish my own research group at TU Wien.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ADVENTURE	ADaptive Virtual ENterprise ManufacTURing Environment
ANN	Artificial Neural Networks
AWS	Amazon Web Services
BPM	Business Process Management
BPMS	Business Process Management System
BRP	Block Rate Pricing

BSU	Billing Storage Unit
BTU	Billing Time Unit
CI	Continuous Integration
CPS	Cyber-Physical System
CPPS	Cyber-Physical Production System
CREMA	Cloud-based Rapid Elastic MAnufacturing
CRUD	Create, Read, Update, Delete
DAG	Directed Acyclic Graph
DBMS	Database Management System
ERP	Enterprise Resource Planning
FSPP	Fog Service Placement Problem
GPRS	General Packet Radio Service
ICT	Information and Communications Technology
ILP	Integer Linear Programming
IoS	Internet of Services
IoT	Internet of Things
KPI	Key Performance Indicator
LTE	Long Term Evolution
MES	Manufacturing Execution System
MILP	Mixed Integer Linear Programming
MML	Mobility Mediation Layer
OLS	Ordinary Least Squares
P2P	Peer-to-Peer
PESP	Platform for Elastic Stream Processing
QoE	Quality of Experience
QoS	Quality of Service
RC1	Research Challenge 1
RC2	Research Challenge 2
RC3	Research Challenge 3

RC4	Research Challenge 4
RC5	Research Challenge 5
RC6	Research Challenge 6
REST	Representational State Transfer
RIA	Research and Innovation Action
SaaS	Software-as-a-Service
SEME	Single-Entry-Multiple-Exit
SIPP	Service Instance Placement Problem
SLA	Service Level Agreement
SLO	Service Level Objective
SOAP	Simple Object Access Protocol *(Deprecated)*
SOC	Service-oriented Computing
SPE	Stream Processing Engine
SWF	Scientific Workflow
UMTS	Universal Mobile Telecommunications System
UNSW	University of New South Wales
ViePEP	Vienna Platform for Elastic Processes
VISP	Vienna Platform for Elastic Stream Processing
VM	Virtual Machine
WSDL	Web Service Description Language
WSN	Wireless Sensor Network
XaaS	Everything-as-a-Service

Part I

INTRODUCTION

1 INTRODUCTION

1.1 OVERVIEW

The proliferation of Information and Communications Technology (ICT) and the Internet of Things (IoT) leads to still ongoing disruptive changes in many industries, including but not limited to the manufacturing, logistics, mobility, healthcare, and energy domains [aca15; Deg16]. This *digital transformation* (or *digitalization*) has already started in the 1990s in areas like the retail and airline industries [ACY03], and is today influencing large parts of the manufacturing industry by adding new, ICT-driven services to existing products [VR88], changing existing business processes into "agile" processes [YSG99], or even replacing former business models [CLR10]. In the manufacturing industry, the transformations based on technical innovations have been assembled under the terms *Industrial Internet (of Things)* [Bru13; KDB16] and *Industry 4.0* [aca13]. The Industrial Internet is primarily rendered possible through the advent of Cyber-Physical Systems (CPSs), or more precisely Cyber-Physical Production Systems (CPPSs), which are "smart", interconnected manufacturing assets featuring embedded sensors and actuators, and are able to collect data and influence business processes based on these data [aca15; Mon14; Raj+10]. The technical progress in the manufacturing industry leads to new possibilities to monitor the supply chain, to make appropriate changes and customizations to business process models and instances if necessary, but also allows for new business models, e.g., based on mass customization [aca13; AIM10; Ng+15; Sch+14b].

Industrial Internet

In general, the advent of the IoT leads to the pervasion of business and private spaces with ubiquitous computing devices, which are connected to both the Internet and other devices, are available in many forms, and are able to act autonomously [AIM10]. IoT devices do not simply act as sensors or actuators, but feature computing, communication, and storage capacities [Bon+14; Mio+12]. In the Industrial Internet, the integration of ICT and IoT technologies has transformed the underlying IT systems: Today, industrial IT system landscapes have become distributed *smart systems*, which may comprise ever-increasing numbers of networked (IoT) objects [BXW14; Mio+12; Vög+17].

Internet of Things

One particular immanent property of these smart systems is their volatility, i.e., they are ever-changing in different dimensions in space and time, including the number of devices connected, the data to be processed, the number of tasks and processes executed, and there-

Volatile Systems

fore the need for computing capacities [Hoc+16a; Vög+17]. During runtime, transient and perpetual system changes occur, e.g., entities entering or leaving and therefore shifting the system boundaries, data sources issuing varying amounts of data, and computational resources becoming available or unavailable. This makes it necessary for the underlying distributed systems to quickly take into account these system changes, including the need to adapt computational resources based on the current and future demands of the system landscape [Lai+12; Sch+14b]. In particular, it is necessary to establish autonomic means to support self-adaptation and self-optimization with regard to the computational resources necessary to support a system [MBS13]. Therefore, a major research challenge is to devise methodologies and techniques for the efficient utilization of computational resources in the Industrial Internet and other smart systems.

Cloud Computing

With the advent of *cloud technologies* and *virtualization*, it is today possible to scale the computational resources necessary to support a smart system in an on-demand, utility-like fashion [Arm+10; Bot+16; Buy+09]. In contrast, permanently providing computational resources that are able to handle peak loads is not the best solution for smart systems, as this *overprovisioning* will lead to underutilization during non-peak times, leading to unnecessary cost [Arm+10]. Also, such an approach still leads to the risk that the amount of fixed computational resources is underestimated, e.g., if a smart system grows very fast or the number of users increase exponentially. This would lead to *underprovisioning* of computational resources [Arm+10].

Scalability for single applications and tasks has been a very lively field of research in recent years [SC16; Zha+15]. However, less attention has been paid to scalable smart systems. Mostly, the focus of research was on cost-efficient scalability of computational resources. Despite this, scalability of resources is only one aspect to be taken into account when adapting the amount of computational resources for the execution of a smart system [Dus+11]. Instead, we apply the

Elasticity

notion of *elasticity* in the constituent publications of this thesis. Herbst et al. differentiate between scalability and elasticity by stating that the latter is done in a timely and prompt manner, i.e., that elasticity is achieved in an autonomic manner at all times [HKR13]. We follow this approach by introducing autonomous elastic systems which are able to self-adapt and self-optimize.

Elasticity Dimensions

While many different elasticity dimensions have been proposed [LEB15], we adopt the model presented by Copil et al. [Cop+13]: In many approaches, elasticity is only regarded from the perspective of *resource elasticity* [Cop+13], which describes the ability to add or remove resources from a system, if necessary. While resource elasticity is a common way to describe the scalability of single applications as well as compositions of multiple applications, resources are not the only dimension that should be taken into account in the context of

smart systems. Notably, Quality of Service (QoS) criteria like runtime duration do not necessarily reflect resource elasticity in a (linear) way, so there might be no proportional relationship between involved resources and QoS [SDQ10]. Also, QoS may differ from user-perceived, subjective Quality of Experience (QoE) [CS13; Into8], which should also be taken into consideration. As a result, it is necessary to include *quality elasticity*, which describes the responsiveness of quality regarding changes in provided resources [Dus+11]. As a third elasticity dimension, many cloud providers make use of dynamic pricing models, which should also be taken into account if cloud resources are used in order to realize scalable processes. These dynamic pricing models are reflected in *cost elasticity*, i.e., the sensitivity of the cost with regard to the change of other parameters (here: quality and resources) [Dus+11]. Naturally, there is a trade-off between different elasticity dimensions, e.g., additional resources lead to higher cost and should increase the QoS. This trade-off may change over time [Mol+13].

Since 2010, we have made several contributions to the field of *elastic systems* by providing solutions for autonomous (cloud-based) resource allocation and task scheduling. Therefore, in the constituent publications of this habilitation thesis, we present novel concepts, methods, and technologies to efficiently exploit computational resources for utilization in the Industrial Internet and comparable smart systems.

In the following subsections, we will provide a brief overview of the underlying research challenges. For this, we exemplify research needs using the simplified overview of an elastic smart system in Figure 1[1]. Typical entities in such an elastic smart system include:

DATA SOURCES: Despite the fact that IoT devices offer computational capabilities, they are very often primarily seen as data sources, which may emit different amounts of data over time. In general, it is assumed that smart systems generate very large amounts of data [CML14], and despite the existence of some standard technologies on both hardware and software level [AlF+15], IoT-based data sources may nevertheless possess a very high level of technological heterogeneity [CML14]. IoT-based data sources in smart systems include, but are not limited to, Wireless Sensor Networks (WSNs), sensor nodes in general, but also smartphones [Chr+09].

STORAGE CAPABILITIES: The data emitted by the aforementioned data sources may directly be processed. This is usually requested in big data scenarios [CML14]. For this, e.g., stream processing principles can be applied [Bab+02]. Alternatively, data may be stored for later processing as structured, semi-structured, or un-

1 Applying an extended version of the Fundamental Modeling Concepts [KGT06].

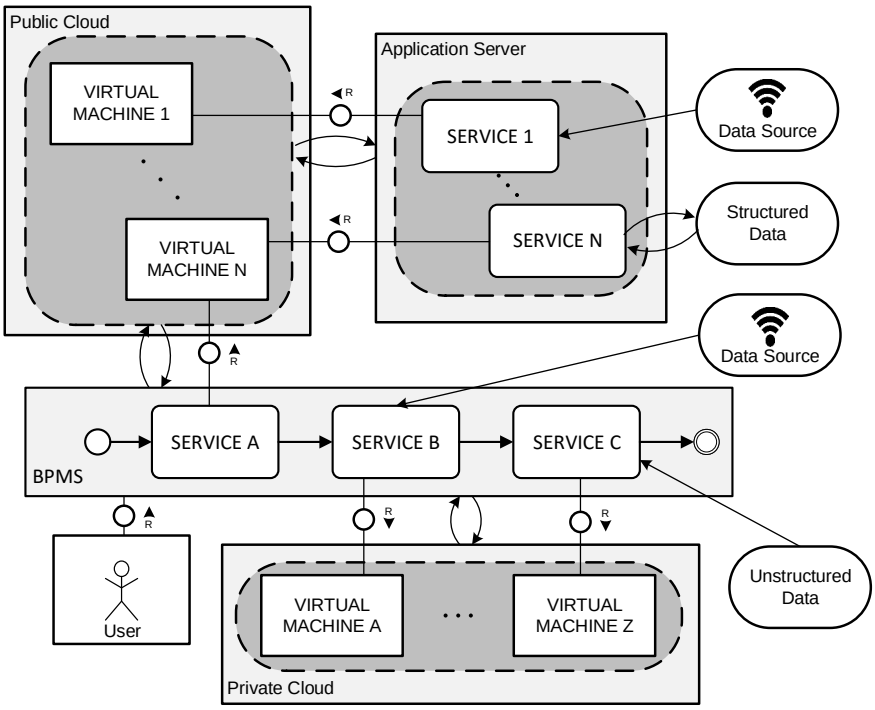

Figure 1: Example Elastic Smart System

structured data [CML14; Gub+13]. Since the number of data changes and data requests may be too large for handling in traditional Database Management Systems (DBMSs), solutions like NoSQL databases are very often applied for this [Cat11; CML14].

PROCESSING ENTITIES: Data processing as well as business logic and business intelligence based on data processing can be executed in different ways. As has already been mentioned before, data may be processed in an ad hoc manner, or later on, e.g., in a batch style. Also, data processing and business logic in smart systems might be provided by standalone entities [Hoc+15], e.g., single software services, or by a composition of such services, e.g., in a business process [MRM13]. In Figure 1, software services are used to depict such processing entities.

COMPUTATIONAL RESOURCES: Processing entities (in terms of services) have to be deployed on computational resources. With regard to the IoT, the cloud is very often named as *the* provider of computational resources [Bot+16]. Indeed, the constituent papers presented in this habilitation thesis mostly apply cloud-based computational resources for the deployment and execution of smart systems and the inherent entities. For this, a "white-box approach" is applied, i.e., the deployment of services on computational resources can be controlled by the service user or a service broker. As an alternative, it is also possible to ap-

ply a "blackbox approach". Using this approach, a usually externally hosted (Web) service with guaranteed QoS is invoked. This way of service provisioning is the "classical" approach in Service-oriented Computing (SOC) [Pap+07; PG03].

More recently, the usage of already existing computational resources in the IoT to deploy services has gained much attention by the research community and the industry. This approach has been labeled *fog computing* [Bon+14; Das+16]. While not depicted in Figure 1, this topic will be discussed in more detail in Section 3.3.2.

INVOKING ENTITY: Last but not least, the end user needs to interact with the processing entities, e.g., via a laptop or a mobile device. The user can directly invoke a service, or indirectly, e.g., via a Business Process Management System (BPMS) as depicted in the figure. It is also possible that the user directly interacts with an IoT entity, if the entity offers a suitable interface. This is not depicted in Figure 1, but again would be the usual approach in fog computing (see Section 3.3.2). In addition to human users, machines may also be service consumers.

1.2 RESEARCH CHALLENGES OF ADAPTIVE RESOURCE AND TASK SCHEDULING FOR THE INDUSTRIAL INTERNET

Although significant contributions have been made in the field of elastic computing (including, but not limited to the application of elastic computing in industrial smart systems), still, major research challenges exist. In particular, it is necessary *(i)* to find mechanisms to provide real-time information about (manufacturing) processes and *(ii)* to facilitate the provisioning of resources necessary to support smart system landscapes. This has to be done taking into account the volatility of smart systems and the IoT in general, and that the systems are potentially of very large scale, generating and processing big amounts of data.

In the following subsections, six research challenges, which form the foundation for the research work presented in the constituent papers of this thesis, are discussed. Naturally, these research challenges are not completely complementary, since various relationships between the single research challenges exist. Figure 2 provides an overview on how these research challenges interact with each other, which will be further discussed below.

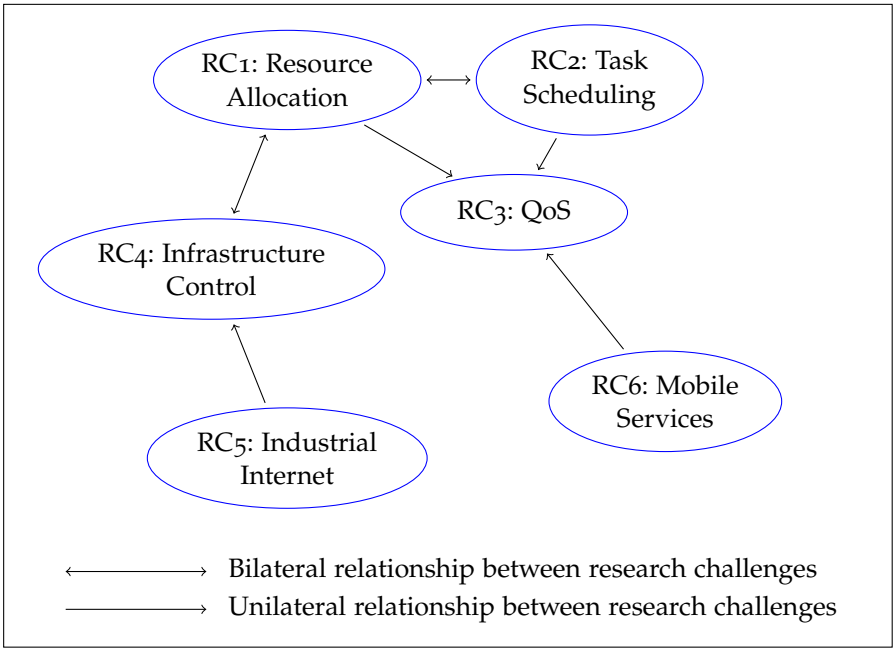

Figure 2: Research Challenges Overview

1.2.1 *Research Challenge 1 (RC1): How to efficiently allocate resources for elastic smart systems in volatile scenarios?*

In order to provide and invoke the manifold software services which are at the basis of any smart system, it is necessary to allocate the computational resources needed to execute them. With regard to the Industrial Internet and the IoT, these resources are usually estimated to be taken from the cloud [Bot+16; Gub+13], i.e., software services are hosted within Virtual Machines (VMs) or software containers. In the state of the art, the focus is mostly on the optimal allocation of resources for single services, e.g., [SC16; Zha+15], while composed services have gained less attention [Sch+15]. To allocate resources for elastic smart systems in volatile systems in an efficient way, e.g., cost-efficiently, it is necessary to predict *how many* resources are actually needed, *when* these resources are needed, and to take into account further constraints, e.g., non-functional properties like deadlines or privacy aspects (see RC3).

So far, the state of the art is missing optimization approaches which *(i)* are not limited to single services, but take into account composed services, which *(ii)* are part of extensive system landscapes, and can use computational resources concurrently, *(iii)* work in volatile system landscapes where user requests or data volume may change at any time, and *(iv)* take into account different types of resources, e.g., private cloud vs. public cloud resources.

Obviously, this research challenge is directly connected to QoS-aware task scheduling, which will be discussed in the next subsec-

tions (RC2 and RC3). Also, RC1 is related to the question how to actually control the computational resources needed in volatile systems, which will be discussed in Section 1.2.4 (RC4).

1.2.2 *Research Challenge 2 (RC2): How to schedule tasks on these resources?*

Once the amount of resources has been determined, it is necessary to schedule tasks on these resources. This research challenge is closely related to RC1, since in the case of elastic computational resources, the task scheduling relies on the resource allocation and there might be some control flow from the task scheduling back to the resource allocation, e.g., since some resources might not be necessary any longer as tasks have been preponed. In fact, many approaches which do not allow concurrent usage of computational resources by different services (see RC1) are aiming solely at task scheduling, since computational resources like VMs are leased for a particular task and then released again or used for another task, e.g., [Pan+10].

While there are a number of scheduling approaches for single applications, e.g., [Lei+12], solutions for composed services are again sparse. Existing approaches like, e.g., [BWE04; CP06], are providing task schedules for a fixed amount of resources, which is not sufficient in smart systems, as discussed above. It needs to be noted that resource allocation and task scheduling for composed services are a NP-hard problem [Str10] and there is no known way to decide such problems in polynomial time [GJ79]. In the case of volatile smart systems, resource allocations and task schedules often need to be found in short time. This makes it necessary to come up with heuristic or approximative solutions for task scheduling. Also, task scheduling makes it necessary to take into account QoS aspects, which are separately discussed in Section 1.2.3.

To summarize, it is necessary to find approaches which *(i)* find a task scheduling taking into account QoS constraints, *(ii)* feed back decision information into resource allocation, and *(iii)* find task schedules in polynomial time.

1.2.3 *Research Challenge 3 (RC3): How to take into account QoS properties?*

Without the consideration of QoS and other non-functional properties (e.g., privacy aspects), resource allocation (RC1) and task scheduling (RC2) could find solutions to the underlying decision problem which do not necessarily meet the demands of the user. For instance, a cost-optimal solution could be found which leads to a task schedule with very long service execution time. In order to avoid this, it is necessary to model the QoS demands of the users and take them

into account during resource allocation and task scheduling. For this, service providers and service consumers need to negotiate a Service Level Agreement (SLA) which includes a number of Service Level Objectives (SLOs) [KL03; PRS09]. Example SLO definitions include the already mentioned deadlines or privacy aspects (i.e., a particular data item or service is not allowed to be hosted in the public cloud), but also availability, downtime, or reliability [Car+04; KL03]. While the current state of the art mostly focuses on the QoS constraints for single applications [Lei+12; WGB11], there are also approaches for composed services. However, most of them focus on single QoS constraints *or* cost, e.g., [Juh+11], Pareto efficiency regarding time and cost, e.g., [Bes+13], or only take into account rather simple composition structures, e.g., [VVB13].

To summarize, there is a need *(i)* to integrate user-defined QoS constraints into resource allocation and task scheduling models, and *(ii)* to regard these constraints during optimization.

In addition to the already mentioned RC1 and RC2, QoS aspects also play an important role in the invocation of services on mobile devices. This will be discussed in more detail in Section 1.2.6.

1.2.4 *Research Challenge 4 (RC4): How to control the IT infrastructure?*

If solutions for QoS-aware resource allocation and task scheduling are in place (i.e., RC1-3), it is necessary to enact the required decisions, i.e., when to deploy which task on which resource, and which resources to lease. For this, a framework is necessary which is able to control both the execution environment in order to lease or release computational resources whenever necessary, *and* the actual application environment, e.g., through a BPMS [Wes12] or a Stream Processing Engine (SPE) [SÇZ05]. Basically, a number of cloud management systems have been proposed which take into account service requests and deploy required services on VMs, e.g., [Buy+09], often explicitly taking into account hybrid clouds or interclouds, e.g., [BRC10; Rod+10], or aiming at specific aspects like energy efficiency [BAB12; Mas+14]. Also, approaches for smart systems, e.g., smart cities [FPV14; KCB15; Li+13], have been proposed, however not taking into account the nature of composed services to be deployed and not focusing on optimization of resource allocation and task scheduling. For the Industrial Internet (see Section 1.2.5), conceptual frameworks have been described, e.g., [Ren+15; Xu12]. Other approaches for the Industrial Internet leave out the actual resource allocation and only focus on service selection, e.g., [Tao+13]. Furthermore, first approaches exist to deploy smart system applications on IoT-inherent computational resources [Vög+17]. Despite this, there is a lack of frameworks explicitly aiming at elastic processes and elastic stream processing.

To summarize, there is a need *(i)* to realize holistic frameworks not only focusing on the optimization of computational resources, but also *(ii)* taking into account the deployment needs of smart systems, and *(iii)* application areas like business processes or stream processing.

1.2.5 *Research Challenge 5 (RC5): How to support Industrial Internet processes conceptually and resource-wise?*

As stated above, due to the proliferation of ICT and IoT technologies, the manufacturing industry is currently undergoing substantial transformations, not only in terms of hardware, but also in terms of CPPS and the software and services used within production environments [aca13]. Simply putting in place IoT devices, i.e., "sensorizing" production environments, and pushing data to the cloud does not help to improve manufacturing processes or to enable new business models. To facilitate mass customization, to react to changing order situations in an ad hoc manner, and to realize short time-to-market [TST12], technological means need to be provided and exploited in order to enable flexibility and scalability of manufacturing processes. One particular approach to enable manufacturing processes which can be changed in a "plug-and-play manner" [Sch+14b] is *cloud manufacturing* [Xu12; Wu+13]. As the name implies, cloud manufacturing is based on principles originally formulated in the field of cloud computing: Leasing and releasing manufacturing assets in an on-demand, utility-like fashion, rapid elasticity of manufacturing process outputs through scaling leased assets up and down, and pay-per-use through metered service. By applying cloud manufacturing principles, companies are able to transform production-oriented manufacturing processes into service-oriented manufacturing networks.

While the basic principles of cloud manufacturing are easy to understand and have led to a number of conceptual solutions, there is a lack of concrete solutions in terms of IT frameworks which support the constitution of cloud manufacturing business networks [Wu+13; Xu12]. Such frameworks need *(i)* to incorporate the means to integrate and process data from arbitrary IoT-based data sources, *(ii)* to virtualize real-world manufacturing assets (analogous to how physical machines are abstracted into VMs), and *(iii)* to provide service and (cloud) resource management [HX15]. For the latter, the aspects discussed in RC4 need to be taken into account.

1.2.6 *Research Challenge 6 (RC6): How to adapt (Web) services for usage on mobile devices?*

Apart from (semi-)automating decision-making processes, one ultimate goal of Industrial Internet research efforts is to provide users like managers or shop floor workers with information, e.g., by providing Key Performance Indicators (KPIs) [Dav+12]. Also, data needs to be transferred to machines, e.g., on the shop floor.

As depicted in Figure 1, user interaction may be done via client software running on a laptop or on a smartphone. Despite the fact that today, mobile user clients like smartphones are very powerful computational devices, the actual connection bandwidth and throughput may still be an issue, especially in harsh (manufacturing) environments [GH09; GLH10]. In addition, not all mobile devices used in the Industrial Internet might be as powerful as smartphones. Even for smartphones, energy and bandwidth constraints persist [CCL09]. Therefore, it is a desirable goal to minimize the amount of data that needs to be transferred to mobile clients and to minimize the communication overhead in general.

In the case of Web services, a large number of approaches to optimize data transfer in order to ensure QoE or QoS have been proposed, e.g., by making sure that the best connection is selected in every situation [GJ03; KKP08], or the according content (i.e., service output) is adapted [LL02; Zha07].

Since the selection of the best adaptation mechanism is context-dependent, there is *(i)* a need for a mechanism which is able to choose the best adaptation mechanism based on the user or system context. In addition, *(ii)* caching and prefetching mechanisms might be useful in order to decrease the data transfer or make sure that data is available at all.

1.3 THESIS STRUCTURE

Based on the discussed research challenges, the main focus of the solutions presented in this thesis is on cloud-based computational resources for the processing of IoT data and the enactment of business processes.

Elastic Computing

For this, we present novel concepts, methods, and technologies to efficiently exploit and invoke elastic (cloud-based) computational resources (RC1, RC4) for utilization in the Industrial Internet and other smart systems (RC5), and for scheduling tasks accordingly on these resources (RC2). The main focus will be on elastic computing, namely the realization, configuration, and optimization of elastic processes and elastic stream processing using cloud-based resources. Optimization is aiming at cost efficiency, taking QoS constraints into account (RC3).

Mobile Services

Moreover, it is necessary to consider the needs of potentially resource-limited client devices and adapt (cloud-based) services for usage on such devices (RC6). Therefore, as a secondary part of this work, solutions for efficient usage of distributed computational resources on (mobile) devices with limited capabilities are discussed. To achieve this, methods to select and adapt Web service invocations aiming at reduced network usage are introduced.

The remainder of this thesis is organized as follows. Chapter 2 presents research challenges and contributions in resource allocation and task scheduling for elastic processes. Chapter 3 describes research questions and contributions regarding process support for the Industrial Internet. Chapter 4 focuses on QoS-aware adaptation and optimization in SOC. A brief summary and outlook on future research topics is given in Chapter 5.

The constitutional papers of this habilitation thesis can be found in Part II, which follows the structure of Chapters 2–4: Chapter 6 presents contributions on resource allocation and task scheduling for elastic processes, Chapter 7 focuses on solutions for stream processing and cloud manufacturing for the Industrial Internet, and Chapter 8 includes papers on QoS-aware service-oriented computing.

While the research presented in this thesis draws major motivation from the Industrial Internet, it needs to be noted that the research results are not limited to this particular application domain. Apart from [Sch+14b], the representative published scientific work presented in Part II is not directly aiming at the Industrial Internet, but has a more generic scope towards smart systems and generic process landscapes.

2

RESOURCE ALLOCATION AND TASK SCHEDULING FOR ELASTIC PROCESSES

2.1 OVERVIEW

Business Processes

The realization of scalable process landscapes is an important prerequisite to provide real-time information about real-world business processes, e.g., in the Industrial Internet, but also in other application domains. Such business processes are composed from single activities which can be provided by humans or by information systems [STD08; Wes12]. If an activity is supported by an information system, it can be an originary software service, e.g., providing data analytics based on IoT data [LL15], or a (value-added) software wrapper for a real-world entity, e.g., an industrial machine, which is accessed and controlled through this service [Gil+07; Xu12].

Computational Resources

The needs of the software parts of a process[1] in terms of computational resources depend on inherent characteristics of the services as well as their contexts. For the work presented in this thesis, the most important inherent characteristic is the actual complexity of the tasks carried out, while the most important service and process context aspects include the data input, user-defined constraints defined in an SLA, and the dynamic behavior of the resources. The characteristics of the computational resources needed for the execution of a single process depend on the complexity of the process model, which may be a combination of simple and complex patterns [Aal+03]. Also, with regard to computational resources, time constraints are of interest [EPR99], since these constraints need to be taken into account when deciding at which point in time which amount and kind of resources are needed. Time constraints can also be beneficial inputs during resource allocation and task scheduling, since they allow to derive the timing of *future* process activities in terms of the point in time by when an activity needs to be finished. Therefore, timing information allows to calculate how many computational resources will be needed at a future point in time. Indeed, the fact that tasks in a business process are part of a sequence is at the core of the resource allocation and task scheduling solutions presented in this work (see Section 2.3).

Process Landscapes

Business processes are not standalone entities, but part of potentially very large *business process landscapes*, i.e., it is not sufficient to take into account the demands of a single process instance, but it

1 Also known as *workflow* [Bec+99; LR99]. However, we will use the term process in this thesis.

is rather necessary to take care of the demands of the complete, often volatile ecosystem, i.e., the business process landscape. While this increases the complexity of the problem to be taken into account, it also allows, e.g., to share computational resources [Sch+15]. This may lead to better cost efficiency of approaches explicitly aiming at elastic processes compared to the ad hoc allocation and scheduling of single applications or single process instances.

As has already been discussed in Section 1, today's means to support scalable process landscapes rely on cloud-based computational resources [Sch+15]. This kind of resources is also at the core of our approaches to realize *elastic processes*. The conceptual and implemented solutions in this area can be partitioned into two fields: First, it is necessary to establish the basic means to realize *elastic BPMS*, i.e., methodologies and tools to control both the cloud and the actual business process landscapes (see RC4), and important helper functionalities, e.g., to predict the actual resource demands. These topics will be discussed in more detail in Section 2.2. Based on these basic results, it is possible to establish means to optimize resource allocation and task scheduling under QoS constraints (see RC1-3), which will be discussed in more detail in Section 2.3.

2.2 ELASTIC BPMS

Research in the field of elastic BPMS is based on the assumption that the concurrent control of a business process landscape in terms of process enactment and of the cloud infrastructure is beneficial both with regard to the cost-efficient assignment of computational resources as well as the non-functional requirements of the process landscape. The latter includes scalability and adherence to non-functional requirements (e.g., deadlines) of the overall process landscape based on the current and future resource demands and constraints of running and requested process instances. The goal is to realize the means for *autonomic* process landscapes, which self-configure and self-optimize themselves, following the vision of autonomic computing by Kephart and Chess [KC03]. Figure 3 shows the typical steps to be carried out in an autonomic system by a so-called *autonomic manager*. As it can be seen, an autonomic system needs the means to *monitor* the state of a system (in terms of resource utilization and non-functional behavior of services), *analyze* this state (e.g., with regard to pre-defined KPIs), *plan* according actions, and finally *execute* this plan before the loop starts allover again. This is done based on *knowledge* about the system which is stored in a database. It should be noted that in smart systems, different parts of the loop may be carried out at the same time, e.g., monitoring is done at all times and therefore also during analysis, planning, and execution.

Autonomic Process Landscapes

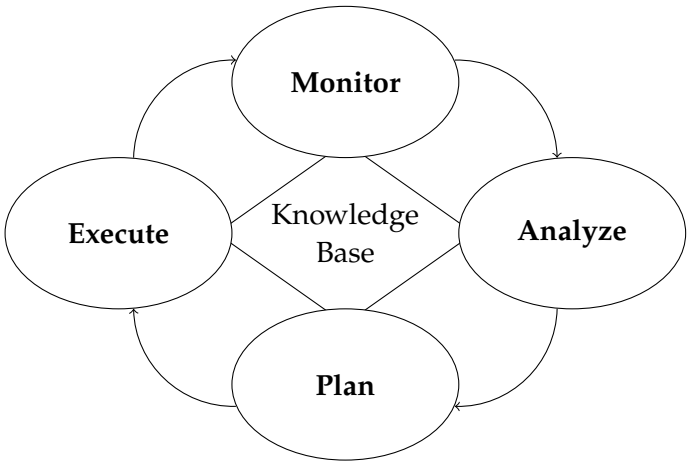

Figure 3: MAPE-K Loop (Adapted from [KC03])

In order to support autonomous elastic processes, it is necessary to build an elastic BPMS which covers all steps of the loop by becoming an autonomic manager. For this, the elastic BPMS needs to provide the means for *(i)* monitoring the process landscape *and* the computational resources in terms of CPU and RAM utilization, and the non-functional behavior of services in terms of execution time and availability, as well as to *(ii)* analyze the state of the system (including process requests and running processes) based on the monitored data. The goal of this analysis is to find out if there is currently any under- or overprovisioning regarding the computational resources (e.g., VMs) and to detect SLA violations in order to carry out according countermeasures if necessary, e.g., to provide further resources, redo task scheduling, or reinvoke a service. The next functionality to be supported is *(iii)* planning, which goes beyond the analysis by not only computing the current status of the system, but also taking into account knowledge about future process activities to be carried out in order to plan the resource allocation and task scheduling (see Section 2.3). Accordingly, functionalities are needed to *(iv)* execute the resource allocation and tasks scheduling.

To achieve this, we provide the following research contributions [BSH16; Hoe+13; Hoe+15a; Hoe+16; HSD13; Sch+13b; Sch+13c; Sch+13d; Sch+15; Wai+17; WHS16]:

Research Contributions

- We conceptualize and implement the Vienna Platform for Elastic Processes (ViePEP), which is a full-fledged research elastic BPMS covering the above-mentioned MAPE-K cycle for elastic processes (see Section 2.2.1).
- We develop mechanisms for the prediction of resource utilization of software services enacted in the cloud, which may be used for standalone services and in the context of elastic processes (see Section 2.2.2).

- We provide solutions for the redundant storage of data (see Section 2.2.3), which may be used in processes or standalone applications.

For the planning part, the BPMS depends on algorithms for resource allocation and task scheduling, which will be separately discussed in Section 2.3. An in-depth discussion of the state of the art in elastic Business Process Management (BPM) can be found in [Sch+15], which is one of the constituent papers of this habilitation thesis.

2.2.1 *The Vienna Platform for Elastic Processes*

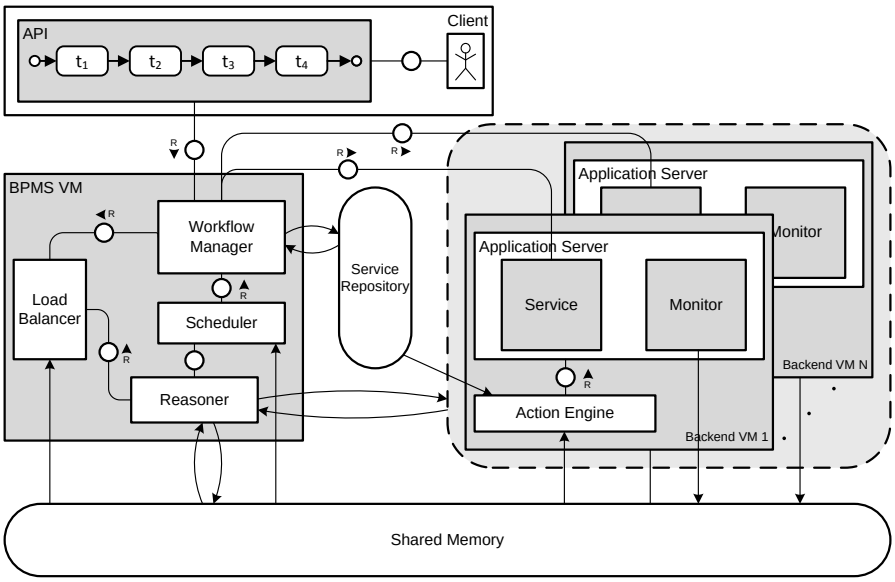

Figure 4: Vienna Platform for Elastic Processes [Sch+15]

To establish an elastic BPMS, we develop the functionalities to support process execution in the cloud in an elastic way. While different elasticity dimensions can be taken into account, the focus is on resource and quality elasticity. The resulting Vienna Platform for Elastic Processes (ViePEP) (see Figure 4) offers the following major functionalities.

- Clients are able to request the instantiation of arbitrary process models. Clients are able to define an according SLA for the single process instances and/or single process activities, i.e., services. The *BPMS VM* (see below) is able to serve different clients at the same time.
- The process landscape and the cloud-based computational resources are controlled by means of load balancing, scheduling, and reasoning, which wrap the algorithms for resource allocation and task scheduling (see Section 2.3). These functionalities are part of the BPMS VM.

- A service repository is used to store software artifacts necessary to execute a software-based elastic process.
- Data is shared between the elastic BPMS and the application servers via a distributed shared storage [Sch+13b; Sch+13c]. This storage serves as the knowledge base, and includes information about which service instances are running on which VMs, how many VMs are currently part of the system, which types of VMs are running, and information about requested and running processes and services.
- Finally, the elastic BPMS needs to provide the means to control application servers. The application servers are deployed on VMs (called *Backend VMs*) which host the software services to be executed as part of a process instance. Through the application servers, the Backend VMs provide the means to monitor service executions and resource utilization on the application server.

BPMS VM

As it can be seen, the BPMS VM is at the core of ViePEP. Within the BPMS VM, we provide functionalities for the deployment of processes (and their underlying software services), and to dynamically arrange incoming process requests on cloud-based computational resources: A Backend VM can be controlled by a BPMS VM, e.g., in order to instantiate additional service instances if necessary. Through the so-called *Action Engine* (located at the Backend VM), it is possible to start a new VM, terminate a VM if it is not needed any longer, duplicate an existing VM if another instance of the service hosted on the VM is needed, exchange the hosted service by another service, and move a running service to another Backend VM, e.g., with more or less computational resources. Thus, scaling in and out is supported.

Both BPMS VMs and Backend VMs could be hosted in a private cloud or a public cloud environment or a mixture thereof. All ViePEP components are loosely coupled, which allows to replace single functionalities with little effort. In fact, this permits to develop and integrate different optimization approaches for resource allocation and task scheduling, which will be discussed in Section 2.3. While ViePEP has been tested with OpenStack and Amazon Web Services (AWS), the interfaces are open and further cloud providers could be integrated with little effort.

To the best of our knowledge, ViePEP is the first elastic BPMS presented, thus contributing substantially to the establishment of a research community in the field of elastic processes. Also, in Schulte et al. [Sch+13b], a first systematic description of research needs in the field of elastic BPMS is provided. In the course of this analysis, the basic considerations for later optimization approaches are identified, e.g., exact vs. heuristic reasoning, global vs. local approaches, and continuous vs. interval reasoning. These basic considerations were

later on taken into account during the work on resource allocation and task scheduling algorithms.

2.2.2 *Resource Prediction*

[BSH16]: "Predicting Cloud Resource Utilization"

In order to derive how many computational resources are needed for carrying out single software services and therefore accumulations of these services, e.g., business processes, process landscapes, or stream processing topologies, it is necessary to predict resource utilization for the single services. For this, we conceptualize, implement and evaluate a predictor for cloud resource utilization. Our contributions [BSH16] are as follows:

Research Contributions

- We introduce and implement a machine learning-based approach for resource utilization prediction on a per-task and per-resource level.
- The presented solution applies Artificial Neural Networks (ANN) to compute predictions based on available historic data, i.e., past task executions.
- We evaluate the prediction approach using a real-world dataset.

The prediction is the basis for a *predictive* resource provisioning and task scheduling strategy [GB12], as will be further outlined in Section 2.3. This may lead to improved resource efficiency and overall response time for the system in question, compared to an ad hoc, reactive approach [Isl+12].

However, simple (linear) regression models are not sufficient for the prediction of cloud resource utilization, since cloud-based computational resources do not necessarily scale linearly with the cardinality of their inputs and may also exhibit some degree of uncertainty with regard to the computational power offered [Bor+14; Len+11]. Hence, after discussing different prediction algorithms, we select ANN models for our prediction approach. ANNs are inspired by biological neurons [Bis06]. We create an according system model which describes the type of task to be provisioned (here: service to be executed) and a vector of input data for this task. The system model also contains a list of potential resources for which utilization needs to be predicted. In the case of cloud-based computational resources, VMs are regarded as resource types. For each different type of resource, a prediction of utilization is returned by our prediction model.

During the evaluation of the resource utilization prediction model, we compare the results of our ANN-based approach with a linear regression model, which represents the state of the art. As test data set, compilation times from a publicly available Continuous Integration (CI) repository, namely *Travis CI*[2] is applied. We show that with

2 http://travis-ci.org

our model, we are able to reduce prediction errors (compared to linear regression) by 20% for the median case. For 72% of all tested cases, our model outperforms the baseline.

Future Work

It should be noted that the developed prediction mechanism may be applied in any setting where cloud-based computational resources may be applied. So far, the solution has been applied to single services (more precisely: compilation times in CI) [BSH16] while work in the field of elastic data stream processing is currently under review. First experiments have shown that there is a need for a prediction toolset applicable and customizable for different types of applications, e.g., elastic stream processing or elastic processes.

2.2.3 *Data Redundancy*

The usage of different service providers in order to avoid vendor and data lock-in effects as well as to decrease cost is a well-known basic principle for risk-aware IT operators who want to use cloud-based computational resources [Arm+10; Kur+11; TCB14]. This basic principle has also been applied to the field of cloud storages, e.g., [Ber+11]. If data is stored in a redundant way, it is possible to allow self-healing of an application or process which depends on this data, even if one particular cloud storage provider is not available anymore. For this, the data is recovered from still available storages and can then be used again.

While the basic means for data redundancy in the cloud, in terms of frameworks and software solutions, are already existing, there is a lack of cost-efficient approaches which optimize the storage usage with regard to redundancy. Therefore, we make the following contributions in this field [Wai+17; WHS16]:

Research Contributions

- We introduce local and global optimization models for cost-efficient data redundancy in the cloud. The Mixed Integer Linear Programming (MILP)-based models take into account QoS needs (e.g., availability) as defined by the data owners. In addition, a heuristic for the global optimization model is presented.
- We apply erasure coding [WK02] in order to achieve data redundancy.
- We implement a cloud storage middleware named *CORA* which can be applied by users in order to enact our optimization models.

[WHS16]: "Cost-Efficient Data Redundancy in the Cloud"

The goal of the formulated and implemented optimization models is to optimize the placement of arbitrary data objects on cloud storages with regard to the occurring cost. For this, the models consider different cost categories including data transfer and migration cost, the actual storage cost, and cost for Create, Read, Update, Delete (CRUD) activities (most importantly read and write requests). In contrast to the state of the art, our approach takes into account a sophisticated pricing model for public cloud storage providers, allowing to include Block Rate Pricing (BRP), Billing Storage Units (BSUs), and Billing Time Units (BTUs). This allows us to consider long-term storages like, e.g., Amazon S3 Standard – Infrequent Access[3], which are not covered in the state of the art.

The optimization models recognize historical data access patterns, i.e., information about past data accesses is applied to predict future data access. Here, the basic assumption is that data access patterns stay the same over a period of time [MEWoo]. In addition, the models ensure user-defined QoS requirements, namely availability, durability, and the vendor lock-in factor. The latter defines how many different storage providers should be used in a particular setting and therefore sets the parameters for erasure coding.

[Wai+17]: "Cost-Optimized Redundant Data Storage in the Cloud"

While a basic optimization model is implemented in [WHS16], a more sophisticated version is presented in [Wai+17]. The former applies the above-mentioned considerations, but conducts the optimization for single data objects, i.e., applies a localized optimization approach. Therefore, the major drawbacks of the basic model are the natural limitations of the localized optimization approach, which only finds a local optimum. Hence, we have added a global optimization model, which finds a global optimum for all data objects. Since the underlying decision problem is NP-hard [PBA12], it is not known if it is possible to find a solution to the global optimization model in polynomial time. Therefore, a heuristic optimization approach is presented for the global optimization problem. The heuristic is based on the idea that data chunks can be categorized into classes, based on the size and the outgoing traffic of a data chunk. A fitting storage for a class of data chunks is then computed for one representative data object, applying the localized optimization approach.

The optimization models and the heuristic are enacted from within *CORA*, which is a cloud-based storage middleware able to interact with arbitrary public and private cloud storages and to enact erasure coding to place data chunks on different cloud storages. SLOs are defined per file. CORA continuously monitors stored data objects and initiates a new optimization cycle if necessary. Optimization is done in case a CRUD operation needs to take place, the number of available cloud storages change, and additionally in regular, predefined intervals.

3 https://aws.amazon.com/s3/storage-classes/

The evaluations are based on an extensive real-world cloud storage access trace [Gra+15]. In the evaluation setup, the parameters for long-term and standard storage services from Amazon S3, Google Cloud Storage, as well as a private Swift-based cloud storage system are applied. The evaluation duration is set to 720 hours. As a baseline, the cost are calculated for a scenario where no cost-optimization is done. For the localized optimization model, we show that we are able to decrease the cost by 11% compared to the baseline. For the global optimization, cost savings are 31.36%, while the heuristic leads to a cost reduction of 24.61%.

Because the global optimization does not compute for large sets of data chunks, a second evaluation was conducted which only compares the heuristic approach to the baseline, however, with a larger number of data chunks. In this scenario, cost savings between 30.9% and 32.16% were achieved (depending on the parameter setting). In addition, the heuristic and the global optimization approach were compared with regard to their runtime performance. Here, it was shown that the heuristic is able so solve the problem >50 times faster than the global optimization approach.

Future Work

As mentioned above, the presented optimization models and the heuristic are generally applicable for cloud-based applications and also for elastic processes. Regarding the latter, it might be a promising approach to place data in the vicinity of software services used in processes. By taking into account the needs of process and data owners, data and processes could be co-located in order to decrease cost and latencies.

2.3 OPTIMIZATION OF ELASTIC PROCESSES

With the basic framework, technologies, and methodologies in place (see Section 2.2), it is possible to realize mechanisms for resource scheduling and task allocation to execute business processes on cloud-based computational resources. For this, we have iteratively conceptualized a number of solutions and optimization models for resource allocation and task scheduling for elastic processes [Hoe+13; Hoe+15a; Hoe+16; HSD13; Sch+13d]. The goal of these solutions is that the best-fitting resources (here: cloud-based computational resources) perform an activity (here: a process task) at the right time [KAV02] and under given user-defined constraints (here: process deadlines). Our contributions in this area are as follows:

Research Contributions

- We conceptualize and implement a number of solutions for (cost-efficient) resource allocation and task scheduling for elastic processes.
- We present both (MILP-based) optimal solutions and heuristics.

- We take into account complex process patterns, penalty cost, BTUs, and different types of VMs.
- We allow the optimization of elastic processes in public and private clouds, and mixtures thereof, i.e., hybrid clouds.

Notably, all approaches presented in this section presume that the services which represent single process tasks are hosted on different VMs and can be invoked concurrently from within different process instances. Also, all approaches aim at the optimization of a complete process landscape, not at resource allocation and task scheduling of single process instances.

[Hoe+13]: "Self-Adaptive Resource Allocation for Elastic Process Execution"

In an early proof-of-concept approach presented in [Hoe+13], the demand of an elastic process landscape regarding computational resources is calculated based on currently running process instances and incoming and queued process requests. For each process request, a deadline can be defined by the process owner.

In this approach, only sequential process models and one VM type are regarded, which simplifies the calculation of needed resources. For this calculation, the CPU utilization of a service invocation is of primary interest and therefore calculated using Ordinary Least Squares (OLS) linear regression. Based on the results of these calculations and the knowledge about running and upcoming service requests, this early solution derives the needed amount of computational resources to enact the requested process instances. ViePEP uses this information to lease an according amount of VMs for a particular timespan. BTUs are not taken into account. During runtime, the utilization of leased VMs in terms of CPU and RAM utilization is monitored and also taken into account for the scheduling of tasks.

This early solution is evaluated with regard to the cost for leasing VMs. For this, one particular process model is used and executed 20,000 times. The model includes five different tasks with varying resource demands. Three different process request arrival patterns are simulated to show the applicability of the solution in scenarios with constantly and rather arbitrarily incoming process requests. The results are compared to a baseline without optimization, i.e., a scenario where tasks are scheduled based on their deadlines and new VMs are leased/released if a particular upper/lower threshold is exceeded. By applying the basic resource allocation and task scheduling solution, cost savings can be achieved for all three arrival patterns. The savings are between 1.55% and 13.3%, depending on the pattern used in the evaluation runs. It should be noted that the total duration until all process models are enacted is higher if applying our solution in the case of arbitrary access patterns. In the case of constant process request arrivals, the usage of the baseline approach leads to a longer overall duration.

While the first solution presented in [Hoe+13] shows some promising evaluation results, it nevertheless does not provide an optimization of resource allocation and task scheduling. Instead, this solution simply makes sure that all processes are scheduled and that deadlines are met. Hence, in [HSD13], we extended our approach by mechanisms to ensure that leased resources are utilized as much as possible. Again, we apply OLS linear regression for the calculation of resource demands. In addition, OLS is also used for predicting the duration of service invocations.

[HSD13]: "Workflow Scheduling and Resource Allocation for Cloud-based Execution of Elastic Processes"

Based on this information, expected deadlines and necessary starting times for all service invocations can be calculated, i.e., ViePEP gets knowledge about which process steps need to be carried out at what point in time. Again, this allows to compute the amount of resources necessary at a particular point in time.

In general, tasks are scheduled so that they meet their individual deadlines. However, if spare resources are available in a particular time period, the task scheduler moves service invocations to an earlier timeslot, if this does not lead to a conflict with the intended control flow of a process model. By moving service invocations to earlier timeslots, it is possible to achieve a higher saturation of the leased computational resources. Scheduling and resource allocation are redone once the system landscape changes, e.g., if new process requests arrive or the resource utilization of the leased VMs is not as predicted. In addition, scheduling and allocation are done in regular intervals.

In the evaluation, we apply the same process model and amount of process instances as in [Hoe+13]. Once more, we use different process request arrival patterns and apply total execution duration and cost as evaluation metrics. The baseline is again following a threshold-based approach, however, this time taking into account the deadlines of the single tasks, while in [Hoe+13], this information is not regarded. Hence, the baseline already applies basic scheduling functionalities. Despite this, we are able to show that depending on the arrival pattern observed, a cost decrease of 1.4% to 8.3% can be achieved, compared to the baseline. Regarding the time savings, improvements between 4.3% and 6.8% were achieved.

While the work presented in [HSD13] provides a first systematic approach to resource allocation and task scheduling, there is no application of a formal optimization model. Therefore, in [Sch+13d], which is an invited extended version of [HSD13], we introduce a formal model for the cost-driven optimization of resource allocation of cloud-based computational resources for elastic processes. For this, we develop a cost model, provide the mechanisms to predict the cost, and to perform a cost/performance analysis. The objective function applies Integer Linear Programming (ILP).

[Sch+13d]: "Cost-Driven Optimization of Cloud Resource Allocation for Elastic Processes"

As outlined in Section 1.2.2, the decision problem for resource allocation and task scheduling is NP-hard, which renders the applicabil-

ity of the optimization model difficult for large-scale scenarios. Hence, a heuristic is also developed for the formal model.

While in the former papers, only sequential process models were allowed, the formal model now also integrates the AND, XOR, and loop patterns [Aal+03]. However, it is still assumed that the next process task to be executed is known. In addition, we foresee now different types of VMs which provide different computational resources in terms of number of CPUs, RAM, and bandwidth. Following well-known pricing schemes, e.g., from Amazon EC2, the cost of these VMs are proportional, i.e., if the amount of cores doubles, the leasing price doubles as well.

In the evaluation of the heuristic, we again apply the process model that has already been used in [Hoe+13; HSD13]. Also, the same threshold-based baseline as in [HSD13] is applied for comparison purposes. The baseline is not able to choose the best-fitting VM type. Two request arrival patterns are used in the implementation – a constant one and a linearly rising one.

With regard to cost, we are able to show a substantial decrease in cost if comparing the heuristic and the baseline, ranging between 16.5% and 22.6%, depending on the applied arrival pattern. The heuristic is also able to decrease the duration of all process executions substantially, by 22.3% to 25%.

While [Sch+13d] already provides a formal model for resource scheduling, the model is not implemented. A major limitation of the formal model is the fact that the next process step must be known in advance, i.e., complex process patterns are not supported during optimization of a resource allocation and task scheduling plan. Furthermore, the model is missing some aspects which are of interest in real-world settings: First, penalty cost, which the provider of ViePEP would have to pay if an SLA violation takes place, are not regarded. Second, BTUs have also not been regarded so far. Third, service deployment times and VM startup times have not been explicitly taken into account.

[Hoe+16]: "Optimization of Complex Elastic Processes"

Hence, a new optimization model is presented in [Hoe+16], which provides a worst-case analysis for complex process patterns (XOR-blocks, AND-blocks, repeat loops). This includes the support of interlaced, recursive patterns. The model also takes into account penalty cost and BTUs. The resulting model is named Service Instance Placement Problem (SIPP) and applies MILP. Its general goal is to minimize the cost of process enactments, taking into account leasing cost for computational resources *and* arising penalty cost. Optimization is done with regard to multiple time periods, i.e., multiple optimization steps are carried out. Penalty cost are calculated based on a linear penalty function. Notably, paying penalty cost essentially means that the deadline of a process instance is postponed, i.e., if penalties are paid, the time allowed to finish a process instance is extended.

For the evaluation, ten different process models from the SAP reference model [CK97; KT98] have been chosen. These process models feature different degrees of complexity in terms of number of process steps and process patterns. To carry out the single process steps, ten different software services are conceptualized and simulated. These services feature different requirements with regard to expected CPU load and service makespan. Two types of SLAs are applied, namely a lenient and a strict SLA. As SLO, the process deadline is taken into account. In addition, we apply different process request arrival patterns, a constant one and a rather arbitrary one. The evaluation results are compared to a threshold-based baseline.

The evaluation is carried out with regard to (i) cost, (ii) SLA adherence, and (iii) total makespan. We show that depending on the applied SLA type and arrival pattern, the SIPP reduces SLA violations up to 32.67%, decreases the cost by 35.65% to 47.66%, while the total makespan is increased by 2.92% to 22.08%.

[Hoe+15a]: "Cost-Efficient Scheduling of Elastic Processes in Hybrid Clouds"

In [Hoe+15a], the SIPP optimization model is extended by the means to take into account data transfer cost and times, and the possibility to instantiate services both in the public cloud and the private cloud. In contrast, our former work primarily aimed at the usage of computational resources from private clouds. By the integration of capabilities for hybrid clouds, it is possible to avoid vendor lock-ins, achieve low latency (due to geographical distribution), and wider resource availability. In the extended SIPP optimization model, the usage of private cloud resources is preferred, since these resources cause less cost than the leasing of public cloud resources.

For the evaluation, the same set of process models from the SAP reference model as in [Hoe+16] has been selected, while the ten different software services used for the process instances have been adapted (in terms of individual CPU load and makespans). The baseline is also extended, as the baseline used in [Hoe+16] is not able to differentiate between private and public cloud resources. Hence, the baseline is now able to fill the private cloud up to a particular threshold. Afterwards, resources from the public cloud are leased and used for process enactment. We apply the strict and lenient SLAs from [Hoe+16] and slightly adapt the two process request arrival patterns from our former work.

Again, the evaluation is carried out with regard to (i) cost, (ii) SLA adherence, and (iii) total makespan. We show that depending on the applied SLA type and arrival pattern, the extended SIPP reduces the total cost between 35.97% and 61.38%. We explicitly discuss the data transfer cost, and are able to show that these cost are decreased by 85.59% to 100% by the data transfer-aware SIPP, which also explains a major part of the savings with regard to total cost. The results with regard to SLA adherence are more mixed, since this value is actually *decreased* compared to the baseline (between minus 10.45 percentage

points and plus 5.24 percentage points). It should be noted that this can be traced back to the consideration of penalty cost in the SIPP, i.e., that from the perspective of cost efficiency, it is in many cases more meaningful to accept SLA violations but therefore have lower cost. Finally, as has already been observed in [Hoe+16], the total makespan is increased by the usage of the extended SIPP (by 1.02% to 11.27%).

Table 1: Overview of Contributions in Optimization of Elastic Processes

	Supported Process Patterns	VM Types	BTUs	Penalty Cost	Hybrid Clouds	Approach
[Hoe+13]	Sequences	☐	☐	☐	☐	Heuristic without rescheduling
[HSD13]	Sequences	☐	☐	☐	☐	Heuristic with rescheduling
[Sch+13d]	Sequences, XOR, AND, loops[1]	☑	☐	☐	☐	ILP & heuristic
[Hoe+15a]	Sequences, XOR, AND, loops	☑	☑	☑	☑	MILP
[Hoe+16]	Sequences, XOR, AND, loops	☑	☑	☑	☐	MILP

[1] Next step needs to be known

Overview

Table 1 provides an overview of the approaches discussed in Section 2.3, and exemplifies the iterative research approach used with regard to the optimization of elastic processes. As it can be seen, the different publications add particular functionalities to the optimization of elastic processes. Notably, [Hoe+16] content-wise is a predecessor of [Hoe+15a], but has nevertheless been published later (but accepted for publication earlier).

Future Work

In all of the approaches presented above, VMs are used as the unit of cloud-based computational resources. While this leads to very good results, it should be noted that VMs are rather coarse-grained and also need quite some startup time, which reduces the flexibility of an elastic process landscape. Therefore, the usage of containers [Hoe+15b; Pah+18] instead of VMs might be a promising solution.

Also, it should be noted that the (extended) SIPP as presented in [Hoe+15a; Hoe+16] can only be applied to rather small process land-

scapes, since the underlying optimization problem is NP-hard. Hence, it is necessary to develop heuristics for the SIPP, which allow to provide cost-efficient resource allocation and task scheduling for large-scale elastic process landscapes.

Also, there is a number of further process patterns, like OR-blocks or unstructured process components, which could be regarded within our optimization models (see Section 4.3). At the moment, unstructured process components cannot be taken into account for elastic process optimization.

Recently, the impact of cognitive computing on BPM has been discussed [HN16]. One particular topic in this area is the dynamic resource allocation based on cognitive capabilities [Rög+17]. This might also be a promising starting point for novel approaches in the field of elastic process enactment.

Table 2: Contributions of Publications to Research Challenges in Elastic Processes (☑: Primary Concern, ⊠: Secondary Concern, □: No Concern)

	RC1	RC2	RC3	RC4	RC5	RC6
Constituent Publications						
[Hoe+13]	☑	⊠	☑	☑	⊠	□
[Hoe+15a]	☑	☑	☑	☑	⊠	□
[Sch+15]	☑	☑	☑	☑	⊠	□
[Hoe+16]	☑	☑	☑	☑	⊠	□
Further Publications						
[HSD13]	☑	☑	☑	☑	⊠	□
[Sch+13b]	⊠	⊠	⊠	☑	⊠	□
[Sch+13c]	⊠	⊠	⊠	☑	⊠	□
[Sch+13d]	☑	☑	☑	☑	⊠	□
[BSH16]	⊠	⊠	⊠	⊠	□	□
[WHS16]	⊠	□	☑	☑	⊠	□
[Wai+17]	⊠	□	☑	☑	⊠	□

Research Challenges

Table 2 shows how the single publications contribute to the research challenges discussed in Section 1.2. Not surprisingly, the research outcomes in the field "Resource Allocation and Task Scheduling for Elastic Processes" primarily aim at RC1-RC4, while providing important foundations to answer RC5.

2.4 PUBLICATIONS

1. Philipp Hoenisch, Stefan Schulte, and Schahram Dustdar. "Workflow Scheduling and Resource Allocation for Cloud-based Execution of Elastic Processes." In: *6th IEEE International Con-*

ference on Service Oriented Computing and Applications (SOCA 2013). IEEE, 2013, pp. 1–8. URL: http://www.dx.doi.org/10.1109/SOCA.2013.44.

2. Philipp Hoenisch, Stefan Schulte, Schahram Dustdar, and Srikumar Venugopal. "Self-Adaptive Resource Allocation for Elastic Process Execution." In: *IEEE 6th International Conference on Cloud Computing (CLOUD 2013)*. IEEE, 2013, pp. 220–227. URL: http://dx.doi.org/10.1109/CLOUD.2013.126.

3. Stefan Schulte, Philipp Hoenisch, Srikumar Venugopal, and Schahram Dustdar. "Introducing the Vienna Platform for Elastic Processes." In: *Performance Assessment and Auditing in Service Computing Workshop (PAASC 2012) at 10th International Conference on Service Oriented Computing (ICSOC 2012)*. Vol. 7759. Lecture Notes in Computer Science. Springer, 2013, pp. 179–190. URL: http://dx.doi.org/10.1007/978-3-642-37804-1_19.

4. Stefan Schulte, Philipp Hoenisch, Srikumar Venugopal, and Schahram Dustdar. "Realizing Elastic Processes with ViePEP." In: *10th International Conference on Service Oriented Computing (ICSOC 2012) – Demos*. Vol. 7759. Lecture Notes in Computer Science. Springer, 2013, pp. 439–443. URL: https://doi.org/10.1007/978-3-642-37804-1_48.

5. Stefan Schulte, Dieter Schuller, Philipp Hoenisch, Ulrich Lampe, Schahram Dustdar, and Ralf Steinmetz. "Cost-Driven Optimization of Cloud Resource Allocation for Elastic Processes." In: *International Journal of Cloud Computing* 1.2 (2013), pp. 1–14. URL: http://hipore.com/ijcc/2013/IJCC-Vol1-No2-2013-pp1-14-Schulte.pdf.

6. Philipp Hoenisch, Christoph Hochreiner, Dieter Schuller, Stefan Schulte, Jan Mendling, and Schahram Dustdar. "Cost-Efficient Scheduling of Elastic Processes in Hybrid Clouds." In: *IEEE 8th International Conference on Cloud Computing (CLOUD 2015)*. IEEE, 2015, pp. 17–24. URL: http://dx.doi.org/10.1109/CLOUD.2015.13.

7. Stefan Schulte, Christian Janiesch, Srikumar Venugopal, Ingo Weber, and Philipp Hoenisch. "Elastic Business Process Management: State of the Art and Open Challenges for BPM in the Cloud." In: *Future Generation Computer Systems* 46 (2015), pp. 36–50. URL: http://dx.doi.org/10.1016/j.future.2014.09.005.

8. Michael Borkowski, Stefan Schulte, and Christoph Hochreiner. "Predicting Cloud Resource Utilization." In: *9th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2016)*. ACM, 2016, pp. 37–42. URL: http://dx.doi.org/10.1145/2996890.2996907.

9. Philipp Hoenisch, Dieter Schuller, Stefan Schulte, Christoph Hochreiner, and Schahram Dustdar. "Optimization of Complex Elastic Processes." In: *IEEE Transactions on Services Computing* 9.5 (2016), pp. 700–713. URL: http://dx.doi.org/10.1109/TSC.2015.2428246.

10. Philipp Waibel, Christoph Hochreiner, and Stefan Schulte. "Cost-Efficient Data Redundancy in the Cloud." In: *9th IEEE International Conference on Service Oriented Computing and Applications (SOCA 2016)*. IEEE, 2016, pp. 1–9. URL: http://dx.doi.org/10.1109/SOCA.2016.12.

11. Philipp Waibel, Johannes Matt, Christoph Hochreiner, Olena Skarlat, Ronny Hans, and Stefan Schulte. "Cost-Optimized Redundant Data Storage in the Cloud." In: *Service Oriented Computing and Applications* 11.4 (2017), pp. 411–426. URL: https://dx.doi.org/10.1007/s11761-017-0218-9.

2.4.1 *Constituent Publications*

Paper [2] was a joint work resulting from the Master thesis of Philipp Hoenisch at TU Wien, which has been written partially during an internship of Mr. Hoenisch at University of New South Wales (UNSW), where he was supervised by Srikumar Venugopal. Philipp Hoenisch later on became a PhD student funded by the EU project SIMPLICITY: The Road User Information System of the Future. The paper describes a first basic approach to minimize cost during enactment of business processes in the cloud. I was responsible for setting up the collaboration with UNSW and defining the topic of the Master thesis. I co-supervised the implementation work done during the Master thesis and wrote the complete paper. In addition, I was responsible for defining the evaluation setup.

Paper [6] was a joint work in the context of the Master thesis of Christoph Hochreiner at Wirtschaftsuniversität Wien, who later on became a PhD student funded by the EU project Cloud-based Rapid Elastic MAnufacturing (CREMA). The paper extends the original SIPP optimization model by the means to integrate hybrid clouds and data transfer. I was responsible for the original idea to extend the existing SIPP optimization problem while defining the topic of the Master thesis and for setting up the collaboration with Wirtschaftsuniversität Wien. I supervised and revised the complete paper written originally by Christoph Hochreiner and Philip Hoenisch and wrote the section on related work.

Paper [7] was a collaboration with Christian Janiesch, Srikumar Venugopal, Ingo Weber, and Philipp Hoenisch. The goal of this paper is to provide a comprehensive overview of the state of the art in elastic BPM. I acted as coordinating author and contributed to the background section on cloud computing and elasticity, described an

example scenario, and wrote the sections on scheduling and resource allocation as well as the corresponding section on future research directions. In addition, I contributed to the identification of challenges for elastic BPM and to the description of ViePEP and revised the complete paper.

For paper [9] , I was responsible for setting up the collaboration with TU Darmstadt and the basic idea of optimization of complex elastic processes. The paper provides the SIPP (see Section 2.3). To the actual paper, I contributed several sections and revised the sections written by the main author Philipp Hoenisch. In addition, I contributed to the development of the optimization model and set up the evaluation scenario.

2.4.2 *Further Publications*

Paper [1] contributes a first systematic approach to resource allocation and task scheduling for elastic processes (see Section 2.3). I co-supervised the implementation work done for the paper and wrote several sections of the paper. In addition, I revised the sections written by Philipp Hoenisch.

In papers [3; 4] , the original version of ViePEP (see Section 2.2.1) was presented. ViePEP is the main result of the Master thesis of Philipp Hoenisch (see above), for which I defined the topic and co-supervised the implementation work. I was the main author of both papers.

Paper [5] provides a first ILP-based optimization approach for elastic processes and a heuristic. I was responsible for setting up the collaboration with TU Darmstadt which led to this paper. I wrote most of the paper, revised the input by the co-authors, and contributed to setting up the optimization model.

The work presented in paper [8] has been implemented by my PhD student Michael Borkowski, who was also responsible for writing most of the paper. I contributed by defining the original idea of resource utilization prediction, which is at the core of this paper. In addition, I revised the sections written by Mr. Borkowski.

Papers [10; 11] present optimization approaches for the redundant storage of data in the cloud. The work presented has been implemented by my PhD student Philipp Waibel, while the original idea of optimized, redundant data storage in the cloud was defined by me during the work on the CREMA project proposal. I guided the implementation and writing work done by Mr. Waibel for both papers and contributed to the evaluation scenarios. In addition, I revised both papers.

3

PROCESS SUPPORT FOR THE INDUSTRIAL INTERNET

3.1 OVERVIEW

In Chapter 2, we have presented the basic means to realize elastic process landscapes. Such process landscapes are one important building block for the distributed smart systems which are at the core of the Industrial Internet. However, while providing an approach on how to *technically* realize the software interactions necessary in today's high-paced manufacturing processes, elastic processes only explicitly give insights on how these processes should actually be modeled and executed in the *real world*. For this, it is necessary to provide methods capable of handling the highly flexible and dynamic manufacturing processes of the future, in order to satisfy end user demands in a customer-centric way [Wu+13].

Cloud Manufacturing

One conceptual approach on how to flexibly react to customer demands and to be able to offer production capacities in a rapid way is to port successful concepts from the field of Everything-as-a-Service (XaaS) and cloud computing to manufacturing in order to mirror agile collaboration through flexible and scalable manufacturing processes. This concept is also known as *cloud manufacturing* [Xu12; Wu+13] (see Section 1.2.5).

Cloud manufacturing should not be mixed up with the pure application of cloud technologies in the manufacturing domain, e.g., by deploying and using Manufacturing Execution System (MES) or Enterprise Resource Planning (ERP) software in a Software-as-a-Service (SaaS) fashion. In brief, cloud manufacturing allows to lease and release manufacturing assets in an on-demand, utility-like fashion, enables rapid elasticity of manufacturing processes through scaling leased assets up and down if necessary, and by pay-per-use of the leased assets through metered service [Xu12]. Therefore, cloud manufacturing mirrors some essential characteristics of cloud computing as defined in "The NIST Definition of Cloud Computing" [MG11].

By modeling all process steps and manufacturing assets as cloud services, it is possible to realize cross-organizational manufacturing orchestrations and integrate distributed manufacturing resources as if they were all located on the same shop floor.

While there is a number of conceptual approaches to cloud manufacturing, there is a lack of concrete software systems, let alone BPMS, supporting this manufacturing paradigm in a holistic way [Xu12]. For this, it is necessary to allow the modeling and integration of arbitrary

manufacturing assets (e.g., CPPS) into processes, and the subsequent enactment of these business processes (see RC5). We will discuss our contributions to the field of cloud manufacturing in Section 3.2.

IoT Data Processing

One particular functionality in order to realize cloud manufacturing is the handling of potentially very large amounts of data which are generated online on the shop floor or as part of the supply chain. In fact, the handling of data streams in volatile smart systems is not a singularity in the manufacturing domain, but rather a prerequisite in smart systems in general, e.g., smart cities [Kol+14; Pui+16], smart grids [Sim+11], or smart healthcare [Cor+15], where a very large number of IoT devices might be deployed to sense and generate data. With the 50+ billion Internet-connected devices expected by CISCO [Eva11] and also with the 20.4 billion connected 'things' expected by Gartner [Mid+16] in 2020, an extremely large volume of data will be generated in a distributed fashion in many different application areas. Stream processing is one specific kind of distributed processing, which has been named one of the most common tasks in the IoT [Per+14b].

Analogous to the usage of principles from the field of elastic computing for the establishment of elastic process landscapes (as presented in Chapter 2), elastic computing is also a promising approach to support stream processing topologies (see RC1-4), i.e., choreographies of stream processing operators [Ged+08]. Our research contributions in the field of IoT data processing will be presented in Section 3.3.

3.2 CLOUD MANUFACTURING

As outlined above, the basic ideas of cloud manufacturing are easy to understand: Cloud manufacturing ports well-known principles from the field of cloud computing to the manufacturing domain in order to offer manufacturing services analogous to how cloud services are offered on the Internet [Xu12; Wu+13]. In order to realize cloud manufacturing, it is necessary to deliver the means to support elastic process landscapes (as discussed in Section 2), provide the means for data integration (as discussed in Section 3.3), enable virtualization (or encapsulation) of manufacturing assets into services (out of scope of the research presented in this thesis), and compose services into processes (as discussed in Section 4.3) [Xu12; Wu+13]. Of course, there are further cross-cutting concerns, e.g., regarding security and privacy, which are however out of the scope of this thesis.

CREMA: Cloud-based Rapid Elastic MAnufacturing

Current approaches to cloud manufacturing are applying a rather conceptual approach or only cover some of the topics mentioned above. Therefore, the EU H2020 Research and Innovation Action (RIA) Cloud-based Rapid Elastic MAnufacturing (CREMA)[1] aims at provid-

1 http://www.crema-project.eu

ing a holistic solution stack to cloud manufacturing, covering manufacturing virtualization and interoperability, a cloud manufacturing process and optimization framework, and the means for collaboration and stakeholder interaction. I was the overall proposal coordinator of CREMA and act as Scientific Leader during its runtime (2015–2017). The research presented in the following is either a result of CREMA or its unofficial predecessor ADaptive Virtual ENterprise ManufacTURing Environment (ADVENTURE)[2], for which I also was the proposal coordinator. For an overview of CREMA, we refer to [Sch+14b; Sch+16].

Our contributions in the field of cloud manufacturing are as follows [Sch+12b; Sch+14b]:

Research Contributions

- We provide basic considerations for the usage of service-oriented concepts in the manufacturing domain and therefore lay the foundations for cloud manufacturing.
- We propose a practical cloud manufacturing framework based on ViePEP.

[Sch+12b]: "Plug-and-Play Virtual Factories"

An early approach to define the basic concepts of cloud manufacturing (however, under the notion of "service-oriented virtual factories"), is presented in [Sch+12b]. In this invited paper, the concept of service-based, cross-organizational manufacturing processes is presented. Similar to cloud manufacturing, these processes have a modular structure and the single process steps are represented by services. Process models might be instantiated in both a semi-automatic or manual manner, depending on the project model's description of functional and non-functional requirements for the single services and the complete process. Furthermore, single process instances are part of a potentially extensive process landscape, where different process instances need to be executed concurrently.

While former approaches to define virtual factories have focused on partner-finding and factory-building, the proposed concept covers the complete process lifecycle. Also, first thoughts to integrate the means to monitor real-time data into process models is discussed. Furthermore, the paper defines a number of research challenges, namely semantic description formats for virtual factories, solutions for process execution, forecasting, adaptation and simulation, as well as the integration of process status data from IoT data sources.

[Sch+14b]: "Towards Process Support for Cloud Manufacturing"

While [Sch+12b] discusses service-oriented manufacturing processes on a rather high level, [Sch+14b] goes one step further by adopting the notion of cloud manufacturing and showing how cloud manufacturing can be integrated into a BPMS. Notably, within this paper, the aspect of elastic process landscapes is raised, and the usage of elastic computational resources for the enactment of manufacturing

2 http://www.fp7-adventure.eu

processes is discussed. In contrast, the work presented in [Sch+12b] relies on the "classic" approach to service-oriented processes, where the underlying resources are not taken into account (see Section 4.3).

To realize cloud manufacturing processes, the paper reviews the requirements towards a software framework for cloud manufacturing and explains how concepts from the field of elastic processes can be applied for this. In addition, the requirement to manage real-world manufacturing assets is discussed. Also, open research questions with regard to process modeling and service descriptions, extended service marketplaces, process monitoring, and trust and data security are elaborated.

Future Work

While the work presented in this section is based on the usage of cloud resources, not all potential users are interested in sending data to the cloud for processing. Also, from the perspective of network overhead, sending vast amounts of data to the cloud for processing, then sending results back to the user (which might be located close to the data sources), is not necessarily the best option. Therefore, as an alternative approach, the usage of IoT-inherent computational resources at the edge of the network is considered to be a promising approach [Geo+16], i.e., the application of fog computing principles as outlined in Section 3.3.2.

3.3 IOT DATA PROCESSING

As discussed in Section 3.2, the integration and exploitation of IoT devices and related technologies is a prerequisite for the realization of cloud manufacturing and for the Industrial Internet in general. Such systems are assumed to be ever-changing and volatile, i.e., generate different amounts of data at different points of time. In addition, change and volatility are also given with respect to the system boundaries and the entities within these system boundaries, i.e., the number and locations of entities in terms of data sinks and data sources. This is especially challenging if data needs to be processed in a continuous manner, since this means that the necessary computational resources need to be adapted during runtime.

Elastic Stream Processing

Hence, we propose the adoption of principles from elastic computing in order to realize *elastic stream processing* topologies. The usage of cloud-based computational resources as presented for elastic process landscapes in Section 2.3 is again a natural choice for the realization of elastic stream processing topologies. Our according research contributions are presented in Section 3.3.1.

With the advent of the IoT, private and business spaces become infused with omnipresent Internet-enabled devices, which provide data about their environments [Gub+13]. These devices do often not only offer sensing capabilities, but also provide computational, networking, and storage resources [AIM10]. As discussed above, the preva-

lent vision of how to process this data is to send it to cloud systems for processing, and then return it to the data sinks, which very often are located close to the data sources [Bot+16]. Naturally, this leads to communication overhead and corresponding latencies, since the architecture of the cloud with very large, centralize data centers, does not match the decentralized and distributed nature of the IoT. In addition, not all users (both private persons and businesses) may feel comfortable about sending data to a (public) cloud. In application areas like smart healthcare, such data transfer may even not be permitted by law [Pea13].

Edge Computing / Fog Computing

Instead of relying on the cloud-based processing of IoT data, the computing capabilities inherent to many IoT devices could be exploited as an alternative. As an example, computational and storage capacities provided by IoT devices in local proximity to one another can be shared and used synergistically [Das+16]. This approach is also known as *edge computing* [Shi+16]. The combination of (possibly virtualized) IoT services at the edge of the network with cloud-based data processing has gained much momentum in the research community and in industry recently under the term *fog computing* [Bon+12]. In short, fog computing provides a conceptual approach for virtualizing and orchestrating computational, networking, and storage capabilities in the IoT and in the cloud. It does so by providing IoT-based computational resources in a similar vein as physical resources are offered as VMs in the cloud [Das+16]. Fog computing is seen as a basic building block in future smart systems where data (pre-)processing or data filtering can be done "on-site", while big data tasks (or tasks which are in general too demanding to be executed on IoT devices) are offloaded to the cloud [Bon+14; SW14].

While the basic conceptual approach of fog computing may appear intuitive and a number of high-level conceptual approaches to realize this novel computing paradigm have already been proposed, there is a lack of research with regard to resource allocation in the fog. Our according research contributions are presented in Section 3.3.2.

To sum it up, in order to achieve data processing for and in the IoT as a prerequisite for the Industrial Internet, we provide the following research contributions:

Research Contributions

- We develop a platform to support elastic stream processing in a distributed, scalable manner. In addition, we propose optimization models for the deployment of elastic stream operators in hybrid clouds (see Section 3.3.1).
- We conceptualize a framework for fog computing and implement optimal and heuristic approaches for resource provisioning for IoT services in the fog (see Section 3.3.2).

3.3.1 *Elastic Stream Processing*

To cope with the volatility of smart systems and therefore the amount of streaming data to be processed, it is necessary to achieve scalability by distributing stream processing nodes [Che+03]. While solutions to *scalable* stream processing focus on the level of computational resources, *elastic* stream processing systems also need to take into account the trade-off between the amount of resources and the offered QoS levels [Hoc+15]. To build concrete solutions for elastic stream processing, it is first necessary to provide a framework able to orchestrate a number of self-contained processing nodes. Second, mechanisms for the cost-efficient provisioning of computational resources under QoS constraints are needed.

While there is related work in the area of elastic stream processing, there is a lack of holistic solutions taking into account the deployment of stream processing operators on geographically dispersed computational resources, necessary reconfigurations at runtime, cost-efficient resource allocation for stream processing nodes, and the usage of fine-grained computational resources. Therefore, our contributions in the field of elastic stream processing are as follows [Hoc+16a; Hoc+17]:

Research Contributions

- We implement an elastic stream processing platform named Platform for Elastic Stream Processing (PESP) which allows to deploy processing nodes in hybrid clouds.
- We introduce an optimization model for cost-efficient real-time stream processing.

[Hoc+16a]: "Elastic Stream Processing for the Internet of Things"

The main contribution in terms of the provided PESP platform is the possibility to add and remove cloud-based computational resources during system runtime [Hoc+16a]. This is not supported by state of the art solutions which rely on fixed amounts of computational resources, e.g., Apache Storm[3] or Apache Spark[4]. To achieve runtime adaptations, the platform provides the following major functionalities [Hoc+16a]:

- PESP enables the distributed deployment of stream processing nodes in a hybrid cloud. This reduces transfer times and distances between data sources and stream processing operators.
- The platform facilitates self-configuration of stream processing operators during runtime, e.g., in order to react to volatile data rates or failures in the stream processing topology.
- PESP allows the integration of arbitrary algorithms for the allocation of computational resources.

3 http://storm.apache.org
4 http://spark.apache.org

To achieve this, PESP follows the requirements for real-time stream processing engines defined by Stonebraker, Çetintemel, and Zdonik [SÇZ05].

The actual platform is composed of an arbitrary number of *Operator Nodes*. Each of these nodes represents one particular stream processing operator within a stream processing topology. Each Operator Node maintains a number of *Processing Nodes*, which host the actual stream processing logic. The Operator Nodes buffer incoming data, thus allowing to cope with changing data rates, and distribute the load to the adjunct Processing Nodes. A *Reasoner* is used to optimize the throughput of data items while controlling the amount of Processing Nodes and therefore the decision if Processing Nodes should be spawned or deleted. For this, the Reasoner gets information about the incoming data objects and the system loads of the Processing Nodes.

While arbitrary optimization models could be implemented in the Reasoner, we introduce one particular optimization model [Hoc+16a]. This optimization model aims at cost-efficient resource allocation for elastic stream processing based on monitoring information about the system status and the incoming data volume. The goal is to minimize cost for VMs while processing data in real-time. For this, an optimization problem is defined and an according model is implemented. The optimization model takes into account BTUs, and allows to deploy Processing Nodes in hybrid clouds.

The platform and the optimization model are evaluated using an example scenario from the transportation domain. Hybrid cloud resources are used to enact this scenario, with Operator Nodes and Processing Nodes being bundled within separate VMs. For privacy reasons, not all stream processing operators are allowed to run in the public cloud. As test data set, the T-drive trajectory data sample [Yua+10] is applied. From this data set, 75 rides are selected. The performance of the proposed optimization model is compared against two baselines, which both rely on a fixed amount of computational resources. One baseline underprovides resources, while the second baseline overprovides resources. The performance of the different approaches is compared with regard to cost, total makespan, average duration, and SLA adherence.

Overall, we are able to show that our model, if compared to the overprovisioning baseline, allows to reduce cost by 18.9% while however decreasing the SLA adherence by 72% and increasing the average duration by 120%. Compared to the underprovisioning baseline, the model allows an improvement of the average duration of 72% and improves SLA adherence by 28%, while leading to a cost increase of 16.41%. Importantly, this shows that our model is able to support the trade-off between cost and QoS, while approaches with fixed resources have to go for one particular goal.

[Hoc+17]: "Cost-Efficient Enactment of Stream Processing Topologies"

The results presented in [Hoc+16a] are extended in [Hoc+17] by the means to deploy stream processing operators not only using VMs, but also utilizing container technologies like Docker. Thus, resources can be leased and allocated in a more fine-grained manner. In addition, runtime adaptations can be performed faster, since containers introduce a smaller startup overhead compared to VMs.

To achieve its goals, [Hoc+17] introduces an additional resource abstraction layer on top of the VMs. For this, a platform capable to control containers is implemented, and an optimization problem is formulated. The goal of the designed optimization problem is cost efficiency while maximizing resource utilization in the public cloud and minimizing the number of necessary reconfigurations of the enacted stream processing topology. Also, QoS constraints defined in SLAs are regarded by taking into account the maximum duration for a particular data processing task. The resulting model takes into account BTUs and allows penalty cost if SLAs are violated.

The optimization problem is reduced to an unbounded knapsack problem, which is known to be NP-hard [APR00]. Hence, a heuristic is developed to find solutions to the optimization problem. The heuristic takes into account historical data about prior deployments of stream processing operators in order to find a cost-efficient resource allocation.

For the evaluation, the heuristic is implemented and integrated into the Vienna Platform for Elastic Stream Processing (VISP) [Hoc+16b], which is a significant extension of the already discussed PESP. For the evaluation scenario, three different sensor types (generating different types and amounts of data) and nine different operator types (each with different resource metrics) are deployed. The heuristic is evaluated using three different configurations in terms of BTU durations. Four different load scenarios with varying arrival patterns are applied during the evaluation runs. As evaluation metrics, cost, three different levels of SLA adherence, the mean time to adapt until an operator type is back to real-time processing, and the number of scaling and migration operations are regarded. When assessing the evaluation results, it should be taken into account that the approach presented in [Hoc+16a], which is used as a baseline for [Hoc+17], already leads to excellent results compared to an over- or underprovisioning scenario.

The evaluation results show that depending on the arrival pattern and the configuration, a cost reduction of up to 36% can be achieved, if compared to the baseline. However, in some cases, the baseline leads to lower cost. Regarding the SLA adherence, the different configurations of the new approach perform in general better than the baseline, with an improvement of up to 25%.

Future Work

The presented results [Hoc+16a; Hoc+17] relies on cloud-based computational resources. Instead, some nodes of the stream processing topology could be hosted on IoT-inherent resources, following the principles of edge and fog computing. In addition, particular tasks, e.g., for prefiltering of data before the actual stream processing takes place, could be done at the edge of the network. Also, the proposed solution could be extended in order to be capable of placing the stream processing nodes in a hybrid cloud.

3.3.2 *Fog Computing*

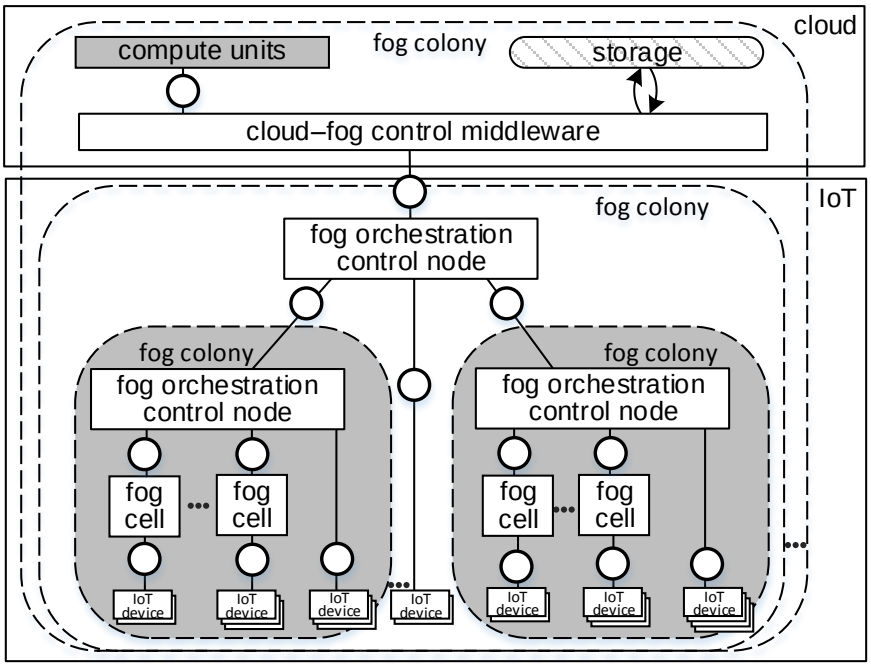

Figure 5: Fog Computing Framework Overview [Ska+16]

As discussed above, fog computing is a very recent research topic. Hence, it is not surprising that the number of resource allocation approaches specially dedicated to fog computing are still quite manageable. Especially, there is a lack of formal models to describe fog resources and, thereupon, resource allocation solutions able to take into account a formal system model.

Therefore, our contributions in the field of fog computing are as follows [Ska+16; Ska+17a; Ska+17b]:

Research Contributions

- We conceptualize a framework for resource allocation in the fog, covering the levels of IoT devices, computational resources at the edge of the network, and the cloud.
- We introduce different approaches to the QoS-aware and resource-efficient usage of fog-based computational resources. Both heuristics and optimal solutions are presented.

[Ska+16]: "Resource Provisioning for IoT Services in the Fog"

In [Ska+16], we propose a first resource allocation heuristic for fog computing, which aims at a high utilization of IoT-inherent computational resources at the edge of the network. Offloading to the cloud is only performed if there is a lack of IoT resources. For this, we first define a conceptual framework to fog computing, which covers three distinct layers, namely *(i)* IoT devices which do not offer computational resources, but, e.g., may act as data sources, *(ii) fog cells*, which are virtualized IoT devices capable of hosting arbitrary software services and of controlling IoT devices, and *(iii)* the cloud, which may also be used to run software services if the fog cells do not provide enough computational resources to deploy all necessary services. This concept is formally defined as a system model.

As it can be seen in Figure 5, fog cells can be orchestrated into *fog colonies*, which are micro data centers made up from an arbitrary number of fog cells. A cloud-based *cloud-fog control middleware* is introduced which acts as an intermediary between fog and cloud and controls the fog cells. If the middleware is not available and for the control of the fog cells within one fog colony, *fog orchestration control nodes* are used. As it can be seen, fog colonies may span the IoT *and* the cloud, i.e., might be deployed on edge devices, cloud-based computational resources, or a mixture thereof. Importantly, both the cloud-fog control middleware and the fog orchestration control nodes may issue and enact resource provisioning plans.

The goal of the resource allocation heuristic presented in [Ska+16] is to distribute services among the different fog cells while utilizing the cells as much as possible and minimizing the latency caused by data transfers between the fog cells and the cloud. The heuristic does not foresee regular optimization intervals nor is it applied because particular events occurred, e.g., a number of additional service requests were issued. As a foundation, a first fit approach is applied [Bre89].

The optimization model is analyzed using the simulation environment *CloudSim* [Cal+11], which is extended to handle the proposed system model. Four standard provisioning policies already provided by CloudSim are applied as a baseline *without* the first fit heuristic, and the results are compared to the application of the standard provisioning policies *with* the first fit heuristic. These standard provisioning policies apply time-sharing and space-sharing for the fog cells and for service requests, respectively. In addition, the fog-based approaches are compared to a pure cloud-based approach. The evaluation scenario is made up from a fog colony with 100 fog cells (each running on a separate IoT device), which have to serve 1,000 service requests. If the IoT-based computational resources are not sufficient, cloud resources are used to fulfill the service requests. As evaluation metrics, average round-trip time per service request, total delays for the execution of all requests, and the makespan are applied. The

cloud-based approach is also analyzed with regard to cost, while for the fog-based scenarios, no cost occur, since we assume ownership of the according IoT devices.

If comparing the baseline against the first fit heuristic, the average round-trip times stay the same for all four provisioning approaches, while the total delays can be decreased by ~39% for the time-sharing of fog cells, while for space-sharing, there are no significant differences between baseline and heuristic. If compared to the cloud-based approach, the total delays can be reduced by 92.39% to 94.99%. Regarding the total makespan, the baseline provisioning approaches and the heuristic do not show any significant differences, while the cloud-based approach performs ~39% faster, due to the more powerful resources available. However, this also leads to cost for the leasing of cloud-based computational resources.

[Ska+17b]: "Towards QoS-aware Fog Service Placement"

The work presented in [Ska+16] provides a first system model and a rather simple heuristic for resource allocation, which does not explicitly take into account QoS concerns. Therefore, we extend this work in [Ska+17b] by augmenting the optimization model to take into account dedicated QoS constraints in terms of deadlines for the fulfillment of single service requests. For this, the Fog Service Placement Problem (FSPP) optimization model is formulated, which is implemented as an ILP problem. Like the first fit heuristic presented in [Ska+16], the FSPP again aims at maximizing the utilization of IoT-based computational resources. Optimizations are done at periodic intervals. The FSPP is a decentralized optimization problem, i.e., optimization is done for single fog colonies. Requests may be forwarded to other fog colonies or the cloud if additional computational resources are needed.

As a prerequisite for the FSPP, a dedicated application model is formulated, which includes a particular, user-defined deadline until when the application's services need to be finished. The response time of an application takes into account deployment times, communication delays in the fog, and actual runtimes of the underlying services. By taking into account the estimated response times for application requests, it is possible to prioritize requests with a closer deadline. The according numbers have to be derived from historical data. In the FSPP, a worst-case estimation of response times is performed.

The FSPP is evaluated using the *iFogSim* simulator [Gup+17], which is an extension of CloudSim used in [Ska+16]. As a baseline, the first fit heuristic presented in [Ska+16] is applied. In addition, a pure cloud-based approach is also used for comparisons. Four different sense-process-actuate application requests are regarded within the evaluation. Not all services are allowed to run on all available fog resources. As metrics, the response time and the cost are regarded.

If compared to the baseline, the FSPP leads to a reduction in response times of 0% to 51.98%, depending on the analyzed applica-

tion. If compared to the cloud-based approach, there is an increase of response times between 3.06% and 118.32%. Since we assume ownership of fog resources, no cost accrue for the baseline. If comparing the FSPP with the cloud-based approach, the cost for computational resource are decreased by 35.51%. Last but not least, the baseline leads to deadline violations in two out of four cases, while both the FSPP and the cloud approach do not lead to any deadline violations.

[Ska+17a]: "Optimized IoT Service Placement in the Fog"

While the exact solution provided by the FSPP leads to optimal results, the underlying decision problem is NP-complete. Hence, we present a heuristic to solve the FSPP in [Ska+17a], which is an invited, peer-reviewed extension of [Ska+16]. The main contribution of this paper is a genetic algorithm used as heuristic approach to allocate resources for the deployment of services and to schedule service requests in the fog. In general, the genetic algorithm is designed analogous to the FSPP regarding its optimization goal. The work follows the application model presented in [Ska+17b].

The evaluation is done analogous to the one presented in [Ska+17b]. The first fit heuristic, the FSPP, and the genetic algorithm are compared with regard to response times (and deadline violations, respectively) and cost. Again, a pure cloud-based approach is also implemented for comparison reasons. In total, five applications are used in the experimental setup.

It can be shown that the genetic algorithm leads on average to slightly higher response times than the optimization approach. However, if taking into account the standard deviation, there is no significant difference between the FSPP and the genetic algorithm. However, with regard to cost, the FSPP leads to 59.09% less cost compared to the genetic algorithm. It should be noted that the cost accruing for the cloud-based approach are >20 times higher than for the genetic algorithm.

Future Work

All approaches discussed so far [Ska+16; Ska+17a; Ska+17b] are based on a rather centralized fog system model, since the control functionalities are implemented in dedicated fog cells, which could become bottlenecks. Instead, it is also possible to apply a distributed, Peer-to-Peer (P2P)-based approach to implement the fog orchestration functionalities. This would lead to higher fault tolerance, but also involves higher coordination efforts between the involved fog cells.

Fog computing and IoT-based computational resources could also be used in order to enact (complex) service compositions, e.g., Scientific Workflows (SWFs) or elastic processes as discussed in Section 2.3 [Nar+17]. The decomposition of applications to run them at the edge of the network and in the cloud is also known as osmotic computing [Vil+16], and offers a number of interesting research questions regarding resource allocation and task scheduling, service configuration, or service orchestrations.

Last but not least, one interesting research direction could be the development of fog-native applications, i.e., applications which are build from services which may roam different types of computational resources in the fog.

Table 3: Contributions of Publications to Research Challenges in the Industrial Internet (☑: Primary Concern, ⊠: Secondary Concern, □: No Concern)

	RC1	RC2	RC3	RC4	RC5	RC6
Constituent Publications						
[Sch+14b]	⊠	⊠	□	⊠	☑	□
[Hoc+16a]	☑	□	☑	☑	⊠	□
Further Publications						
[Sch+12b]	□	□	□	□	☑	□
[Ska+16]	☑	□	⊠	☑	⊠	⊠
[Ska+17a]	☑	☑	☑	☑	⊠	⊠
[Ska+17b]	☑	☑	☑	☑	⊠	⊠
[Hoc+17]	☑	⊠	☑	☑	⊠	□

Research Challenges

Table 3 shows how the single publications contribute to the research challenges discussed in Section 1.2. As it can be seen, the focus of the papers related to IoT data processing (as presented in Section 3.3) is on RC1, and partially on RC2-RC3, while the focus of the papers related to cloud manufacturing (as presented in Section 3.2) is primarily on RC5.

3.4 PUBLICATIONS

1. Stefan Schulte, Dieter Schuller, Ralf Steinmetz, and Sven Abels. "Plug-and-Play Virtual Factories." In: *IEEE Internet Computing* 16 (5 2012). Invited Paper, pp. 78–82. URL: http://dx.doi.org/10.1109/MIC.2012.114.
2. Stefan Schulte, Philipp Hoenisch, Christoph Hochreiner, Schahram Dustdar, Matthias Klusch, and Dieter Schuller. "Towards Process Support for Cloud Manufacturing." In: *18th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2014)*. IEEE, 2014, pp. 142–149. URL: http://dx.doi.org/10.1109/EDOC.2014.28.
3. Christoph Hochreiner, Michael Vögler, Stefan Schulte, and Schahram Dustdar. "Elastic Stream Processing for the Internet of Things." In: *IEEE 9th International Conference on Cloud Computing (CLOUD 2016)*. IEEE, 2016, pp. 100–107. URL: http://dx.doi.org/10.1109/CLOUD.2016.21.

4. Olena Skarlat, Stefan Schulte, Michael Borkowski, and Philipp Leitner. "Resource Provisioning for IoT Services in the Fog." In: *9th IEEE International Conference on Service Oriented Computing and Applications (SOCA 2016)*. IEEE, 2016, pp. 32–39. URL: http://dx.doi.org/10.1109/SOCA.2016.10.

5. Christoph Hochreiner, Michael Vögler, Stefan Schulte, and Schahram Dustdar. "Cost-Efficient Enactment of Stream Processing Topologies." In: *PeerJ Computer Science* NN (NN 2017), NN–NN. URL: https://doi.org/10.7717/peerj-cs.141.

6. Olena Skarlat, Matteo Nardelli, Stefan Schulte, Michael Borkowski, and Philipp Leitner. "Optimized IoT Service Placement in the Fog." In: *Service Oriented Computing and Applications* 11.4 (2017), pp. 427–443. URL: https://dx.doi.org/10.1007/s11761-017-0219-8.

7. Olena Skarlat, Matteo Nardelli, Stefan Schulte, and Schahram Dustdar. "Towards QoS-aware Fog Service Placement." In: *IEEE/ACM 1st International Conference on Fog and Edge Computing (ICFEC 2017)*. IEEE, 2017, pp. 89–96. URL: http://dx.doi.org/10.1109/ICFEC.2017.12.

3.4.1 *Constituent Publications*

I was the main author of [2]. As such, I conceptualized the research idea of "practical cloud manufacturing", which was in fact an idea defined by me for the proposal of the CREMA project. Furthermore, I wrote all sections of the paper completely by myself, apart from the section on related work and on ViePEP, where some further input has been contributed by the co-authors. In addition, I revised the input provided by the co-authors.

The work presented in paper [3], has been implemented by Christoph Hochreiner. I contributed by defining the original idea of elastic stream processing, which is at the core of this paper. In addition, I supervised the implementation work and revised the sections written by Mr. Hochreiner. This paper presents a platform and optimization approach for the deployment of a streaming topology in the cloud.

3.4.2 *Further Publications*

I was the main author of [1], which presents an early, service-based approach to cloud manufacturing. The paper is based on the basic ideas of the EU FP7 project ADVENTURE, for which I wrote the proposal. For the paper, I wrote all sections and incorporated the feedback received by the co-authors.

In paper [4] , an early approach to resource provisioning in the fog is presented. I conceptualized the original research idea that there is

a need for resource allocation in the fog in order to enact services in the Industrial Internet "on-site" instead of the cloud. I supervised the implementation work presented in this paper and revised the paper, which was mostly written by my PhD student Olena Skarlat. Paper [6] is an invited, peer-reviewed extension of [4], where I again supervised the implementation and revised the paper.

Paper [5] provides a solution to cost-efficient enactment of stream processing topologies in the cloud. As outlined above, the basic idea of elastic stream processing has been conceptualized by me. I revised this paper, which was mostly written by Christoph Hochreiner.

Paper [7] is a follow up to [4]. Again, the basic research idea of resource optimization has been defined by me, but the concrete implementation work and writing for this paper have been done by Olena Skarlat. I supervised the implementation work and revised the paper.

4

QOS-AWARE ADAPTATION AND OPTIMIZATION IN SERVICE-ORIENTED COMPUTING

4.1 OVERVIEW

SOC serves as the conceptual and technical foundation for cloud computing [DWC10], with especially the SaaS service model being a technology-agnostic variant of the original vision of Web service-based computing [LZV08]. This original vision is based on a well-defined stack of protocols and standards, which is known as the *WS-* stack* [Wee+05]. The goal of this rich set of standards is to support interoperability and platform-independence through self-description. In contrast, cloud service models as well as today's RESTful approach to Web service delivery are not based on particular protocols and standards, but make use of basic Representational State Transfer (REST) principles [PZL08; zNS05].

Mobile Services

The communication overhead caused by the application of Web service standards like SOAP or the Web Service Description Language (WSDL) is especially an issue in mobile scenarios, e.g., if Web services are invoked from a mobile device like a smartphone. While mobile devices become more and more powerful in terms of the data volume that can be processed, the mobile bandwidth is growing to the same degree [CCL09]. Furthermore, in the IoT, devices which are *not* as powerful as a modern smartphone might invoke services. These devices possess only limited bandwidth, CPU power, memory, or energy resources [Per+14a]. Because of the resulting gap between the computational power of mobile devices and the data to be processed, it is reasonable to decrease the communication overhead and to identify ways to reduce the computational efforts of the mobile device. In the case of the WS-* stack, this means that approaches to adapt the used communication protocols and description standards need to be found. Our according contributions will be presented in Section 4.2.1.

In addition, e.g., in manufacturing scenarios, where harsh environments persist [GH09; GLH10], but also in in-vehicle scenarios [SK13], wireless network connections might not be stable. Therefore, the reduction of transferred data as well as the prefetching of data while the network quality is high are promising approaches. Our according contributions to data caching and data prefetching for Web and cloud services will be discussed in Section 4.2.2.

Service Composition

One particular research topic in QoS-aware SOC is the composition of services in order to execute business processes under QoS constraints (usually defined in an SLA) and in a cost-efficient way. This can be seen as a predecessor to the elastic processes presented in Section 2, however with the difference that methods and solutions for elastic processes manage the actual resource allocation and task scheduling on cloud-based computational resources, while service composition is based on the assumption that arbitrary software services are available on the Web and can be invoked with given QoS guarantees [DS05; Str10]. While a large number of papers on service composition has been published before, e.g., [LHD13; TKM04], important aspects like complex process patterns, probabilistic QoS values, or probabilistic process execution paths have not been covered yet. Therefore, we propose novel optimization algorithms for the QoS-aware cost optimization of service compositions. Our according contributions will be discussed in Section 4.3.

Regarding the research challenges discussed in Section 1.2, the research presented in the field of mobile services primarily aims at RC6, while the work in the field of service composition aims at resource allocation (in terms of services), i.e., RC1, and QoS-awareness, i.e., RC3.

4.2 MOBILE SERVICES

As outlined above, service invocations in mobile settings may be subject to issues arising from harsh environments and unstable network conditions. Therefore, solutions are needed which take into account the particularities of mobile service invocations and act as intermediaries between the actual services and the mobile environment [AB06]. These intermediaries may offer different functionalities in order to, e.g., decrease the communication overhead [Pap+10], lessen the computational efforts needed at the mobile client by offloading particular tasks to the cloud [FLR13], prefetch data before it is actually needed [Hig+12], or offer other functionalities to make service invocations reliable in mobile settings. Within this section, two approaches are presented to improve the user experience during mobile service invocation, namely Web service adaptation (in terms of the used communication protocols), and data caching and prefetching.

Web Service Adaptation

First, adaptation mechanisms for Web service invocation will be discussed: A number of different adaptation mechanisms have been proposed for mobile service invocation, aiming at selecting the best connection in every situation [GJ03; KKP08], at adapting the actual content to be transferred [LL02; Zha07], or at choosing a particular communication protocol [Pap+10]. For the latter, a number of solutions have been proposed for the mobile invocation of (Web) services on mobile devices, e.g., Wireless SOAP [ADJ05], SOAP-over-UDP [LPT05], or proprietary solutions [AKM07].

Still, the question remains which adaptation mechanism is the best choice in a particular context. Different factors play a role here, including, but not limited to the capabilities of the mobile device, the used standards and protocols, and the type and quality of the network. Hence, no single adaptation mechanism is the best-performing in all possible system contexts [Pap+10]. Since the number of potential service users are huge, the number of possible invocation settings are also very large, which further complicates the selection of a well-fitting adaptation mechanism.

Caching

Second, caching mechanisms might be helpful to decrease the amount of transmitted data in mobile service invocations. Client-side caching prevents the repeated transfer of data, especially if there has been no change in the data. One aspect to be taken into account during caching is data freshness, i.e., if data is still up-to-date when it is actually consumed. While 100% freshness can be achieved if the Web service provider supports client-side caching, e.g., [Li+08], this does not enable caching for arbitrary services. For this, the usage of a proxy-based approach is promising and will be further observed.

Prefetching

While caching aims at the avoidance of unnecessary data transfer, prefetching mechanisms aim at making sure that data is available at all even if the network quality is weak or if there is no network connection. For this, different aspects like the type of mobile application, the user context (especially the current and future location of the user), and the current and future network quality need to be taken into account. While existing approaches to data prefetching provide solutions for ad hoc data prefetching, there is a lack of solutions which schedule data prefetching based on the context.

To sum it up, we provide the following research contributions aiming at the reliable usage of Web service technologies in mobile settings [Pap+11a; Pap+11b; Pap+12; Hum+14; Pap+14; BSH16]:

Research Contributions

- We conceptualize and implement a middleware for mobile service invocations (see Section 4.2.1).
- We use the middleware in order to apply our developed mechanisms for Web service adaptation in mobile settings. These mechanisms provide decision support with regard to selectable adaptation mechanisms aiming at minimization of communication overhead (see Section 4.2.1).
- In addition, we provide the means to decrease wireless communication overhead by providing proxy-based, client-side caching for service requests, while guaranteeing 100% freshness of the service responses (see Section 4.2.2).
- We provide solutions for data prefetching in mobility settings. These solutions are able to compute which data are needed at a particular point in time, based on the user context. Hence, data

can be requested before it is actually needed, e.g., if the network connection is unstable (see Section 4.2.2).

4.2.1 *Web Service Adaptation*

In order to determine which Web service adaptation mechanism is preferable in a particular mobile context, we have come up with an according decision support solution, i.e., a toolset helping users like operators or developers to (semi-)automatically select the best adaptation mechanism in a particular situation.

[Pap+11a]: "Always Best Served: On the Behaviour of QoS- and QoE-based Algorithms for Web Service Adaptation"

In [Pap+11a], we present the Mobility Mediation Layer (MML), which is a middleware acting as intermediary in settings with mobile service invocations. For this, the MML offers an interface for mobile devices, a connection to externally hosted services, a context adapter which helps to collect context data for single service invocations, and several further components. While the MML can support arbitrary adaptation mechanisms, we focus on adaptations in terms of reduction of communication overheads.

These arbitrary adaptation mechanisms can be deployed as proxies within the MML. Proxies have been selected as technology for adaptation mechanisms, since they allow to intercept a service invocation request without modifications of the actual service code or service interface [AB06]. In brief, the proxies are used in order to replace heavyweight service calls from a mobile device to a Web service with proxied service calls implementing a particular service adaptation mechanism. Multiple proxies, i.e., adaptation mechanisms, may exist for the same service.

The MML allows to propose the best alternative on how to invoke a service in terms of which proxy to generate for a particular service request. This proposal can be accepted or rejected by the user or could also be used to automatically generate a new proxy or select an existing one. To compute scorings for the different possible service adaptations, the system context is taken into account. The context is composed from connection (bandwidth, latency, packet loss, stability), device (CPU power), application (service call frequency, service call dependence), and service (message size, overhead ratio, processing time) attributes. For each service call, all attributes are stored in a database of historical service requests.

This knowledge base can then be used by QoS-based and QoE-based scoring algorithms to propose the best proxy for an incoming service request. For this, the QoS-based approach provides a utility function that determines the vector distance between the "ideal" setting for the selection of a particular proxy, and the actual setting within a particular service request. The QoE-based approach is based on implicit user feedback, which is given by actual proxy selections from the past. A Bayesian network is used to learn the best proxy

from the past user behavior, i.e., to identify scores for the different possible proxies.

Both approaches are evaluated with regard to their reliance on complete information. For this, four different adaptation mechanisms and 12 services with different attributes have been chosen. Two scenarios with historical data about 1,000 and 5,000 service requests were evaluated, respectively. To show how the approaches perform given incomplete data, 25% of the connection-related attributes, 25% of the device- and application-related attributes, and 25% of the user feedback were set to be "unknown" in separate evaluation runs. As baseline, a scenario with complete information is used.

Overall, scoring outputs (if compared to the baseline) may deviate to about one third because of the missing information. Within the evaluation, we show that for the QoS-based scoring algorithm, the connection-related attributes are of primary interest, i.e., if part of this information is missing, the scoring performs worse. For the QoE-based scoring algorithm, the missing device- and application-related attributes are leading to higher deviations than the communication-related attributes. However, naturally, the QoE-based approach is most affected if information about the user feedback is missing. We also show that the QoE-based scoring algorithm depends to quite some degree on the size of the historical data set, while the QoS-related approach is less affected by this.

While the work presented in [Pap+11a] gives a first overview of the QoS- and QoE-based scoring algorithms, it only provides a short description of the approaches. Also, while the behavior with regard to incomplete data has been evaluated, no mechanism to handle missing data has been proposed. As a result, it is necessary to provide complete service request logs to achieve good scoring results. However, due to transmission errors, reluctance, or unavailability of particular context data sources, this information might not be available.

[Pap+14]: "Decision Support for Web Service Adaptation"

Therefore, within [Pap+14], which is an invited, peer-reviewed extension of [Pap+11a], we present full information (including the complete mathematical models) about the scoring algorithms. In addition, we add different imputation algorithms for the handling of incomplete data, namely case deletion, modus/mean imputation, random hot deck, distance function matching, and combinations thereof.

Within the evaluation, we show how the integrated imputation mechanisms significantly decrease the error imposed by the missing data. For this, imputation is applied during executions of the QoS-based scoring algorithm. Different test scenarios are applied, which are based on varying methods to generate incomplete data: This can be done either randomly or following the patterns of unreliable data sources or unreliable data collection. For all scenarios, the resulting missingness is ~25%. As performance metric, we apply the normalized error imposed by the missing values on the result of the QoS-

based scoring algorithm. As historical knowledge base, 10,000 service requests are taken into account. Like in [Pap+11a], we apply six different services with different capabilities during the evaluation.

We show that the normalized error if no imputation is used is between ~9.5% and ~15%, depending on the particular error. This error can be decreased to less than 1% if multiple imputation is applied, however at the cost of a rather high runtime of the imputation algorithm. Both case deletion and random hot deck lead to mean normalized errors of less than 2% and at the same time offer less runtime overhead.

Future Work

The solutions presented in [Pap+11a; Pap+14] are mainly aiming at mobile devices like smartphones. However, they could be applied for arbitrary resource-limited wireless devices like sensor nodes in the IoT. Therefore, it is worth investigating how the presented adaptation mechanisms (respectively the decision support for selecting the best-fitting mechanism) could help to solve problems in IoT scenarios where devices may be limited by similar constraints but possess even smaller computational power. This could also be an interesting research question for mobile (or wireless) fog devices with little computational power.

4.2.2 *Caching and Prefetching*

Caching

While the solutions presented in Section 4.2.1 can be applied in order to decrease the data transferred for particular service invocations, the actual data transmissions take place in any case. Another approach to decrease the amount of data to be transferred is to actually omit particular data transmissions by identifying if the transmission has to take place again if there has already been the same service request in the past. When interacting with static Web content like Websites, client-side caching is used to achieve this [PB03; Wan99]. While client-side caching could be used with standard Web service technologies like SOAP and WSDL, there is a risk of receiving data which is not fresh any longer, since Web services make use of complex message patterns [TR03]. In fact, with standard Web service technologies from the WS-* stack, it is necessary to send a new service request in order to make sure that the service response provides up-to-date data. While it is not possible to circumvent this new service request, it is possible to decrease the additional data transfer between the mobile client and the server-based service by quite some extent.

[Pap+11b]: "Enhancing the Caching of Web Service Responses on Wireless Clients"

In [Pap+11b], we present a novel approach which verifies the freshness of a caching result during runtime, i.e., directly before the cached data is intended to be processed by a mobile client. For this, the MML presented in Section 4.2.1 is applied in order to enable proxied mobile service invocations. However, instead of enriching service invocations with decision support for adaptations (as presented in

[Pap+11a; Pap+14]), the proxies are used to verify the freshness of cached service responses. The usage of proxies allows to apply the caching mechanism for arbitrary services without modifications of the server-based Web services themselves. Therefore, the proposed client-side caching can also be applied for externally hosted services.

In the approach presented, it is the goal to decrease wireless data transfers by conducting these additional service requests only between a server-based proxy (located in the MML) and the service provider. This means that a service consumer (here: a mobile device) sends a service request not to the Web service, but to the proxy. The proxy forwards the request to the actual service and analyzes if the content of the response is identical to an already cached response. Only if a change has happened, the full service response is forwarded to the mobile device which originally issued the service request. Else, a status code is provided to the client, which decreases the amount of data to be transferred by some degree, since data from earlier service invocations is cached at the client and can be reused.

During the proxy generation, SOAP request and response wrappers are generated, which are enriched with information about service modifications, a status code, and an identification tag. This information serves as foundation to achieve cacheability for arbitrary services. Together, these modifications allow to automatically check the freshness during service invocations. If a cached service response is not fresh any longer, the service response is forwarded by the proxy to the mobile device.

In order to evaluate this approach, the solution is compared to two scenarios with direct service requests and no caching and with caching, respectively. As evaluation metrics, the saved bandwidth and the user-perceived latency are compared for all three approaches. During the evaluation runs, different types of network connections, i.e., General Packet Radio Service (GPRS), Universal Mobile Telecommunications System (UMTS), Long Term Evolution (LTE), Web services (in terms of response sizes), and workloads (in terms of number of requests) are tested. As evaluation parameters, the cache hit ratio and the response sizes are used. The goal of the evaluation is to demonstrate how the saved bandwidth changes with different cache hit ratios and response sizes.

The evaluation shows that 25% of the bandwidth can be saved even with small cache hit ratios and response sizes. With regard to the user-perceived latency, it is demonstrated that the reduction primarily depends on the applied types of network connections. Significant latency reductions can be achieved despite the usage of a proxy and the fact that two service request/response interactions take place for each original interaction – one from the mobile device to the proxy and another one from the proxy to the server-based Web service.

[Pap+12]: "Lightweight Wireless Web Service Communication Through Enhanced Caching Mechanisms"

[Pap+12] is an invited extension of [Pap+11b], which primarily provides an enhanced evaluation of the proposed caching solution. In addition, the description of the automatic proxy generation is significantly augmented by additional information. However, the basic contribution in terms of the client-side caching solution stays the same. While [Pap+11b] focuses on selected interactions with two real-world Web services, the extended evaluation presented in [Pap+12] makes use of artificial yet realistic Web service call traces, which provide different levels of dynamicity and response size. The outcomes confirm the evaluation results presented in [Pap+11b].

Prefetching

The work discussed so far in this section has focused on the decrease of data transmission sizes, which might become handy, e.g., in order to save computational resources (for parsing excessive responses) at low-powered IoT devices, and to decrease the general data transfer, which is especially helpful in scenarios with a lot of mobile devices. In scenarios where network connections are not reliable, it might however be a better idea to provoke additional data transfers while the connection is still reliable. This is also known as data prefetching. In brief, data prefetching describes the querying and gathering of data *before* it is actually needed. For this, it is necessary to predict which data will be requested by applications in the future [Hig+12]. In the case of mobile devices, data prefetching is not only done to decrease user-perceived latency [Sch+11a], but also to make applications work even if the network connection is weak or fails.

[Hum+14]: "Context-Aware Data Prefetching in Mobile Service Environments"

To achieve an appropriate level of QoE in mobility scenarios, we present an approach to context-aware data prefetching in mobile service environments [Hum+14]. This work is a result of the EU FP7 project SIMPLI-CITY and focuses on in-vehicle usage of mobile devices and the prefetching of data from cloud services for a moving user.

We assume that the route of the mobile device is known, e.g., from a GPS navigation device. By combining future locations of the user and knowledge about network quality at different locations, it is possible to derive which data should be prefetched. The prefetching is application-aware, i.e., takes into account typical access patterns of mobile applications. For this, different categories of applications are identified. These categories feature different levels of importance, time criticality, and access patterns. Based on this information and the user context data, a particular (pre-)fetching strategy is automatically selected. For this, a formal model for the prefetching strategy is developed.

For the evaluation, real-world vehicle traces and network coverage maps are used. The evaluation is conducted taking into account six services which represent different categories of mobile applications. In the evaluation, we show how often prefetching is utilized in order

to increase the user experience, and how often the presented solution does not prefetch data even though this would have been necessary. Also, the data freshness is analyzed.

[Bor+16]: "Prediction-Based Prefetch Scheduling in Mobile Service Applications"

The approach presented in [Hum+14] performs prefetching in an ad hoc manner, i.e., there is no scheduling of prefetching for different applications running on a mobile device. Therefore, in [Bor+16], a second approach to data prefetching in mobility scenarios is presented, which uses calculus to schedule prefetching. The goal of the scheduling approach is to make sure that data is prefetched not too late, but as late as possible (to maximize the data freshness), and to avoid unnecessary data transfers caused by prefetching data continuously. The approach is again based on knowledge about mobility patterns of the mobile device, the resulting expected network quality, and assumptions on how often particular applications request data from an online service.

The evaluation is conducted with regard to user-perceived response times. In the according simulation, real-world vehicle traces and measurements of network quality are applied. During the simulation, one particular service is used, while service request intervals are varied. As a baseline, an approach to data buffering which does not take into account the perceived location of the user is applied. We are able to show how the prefetching solution leads to avoidance of buffer underruns, while the baseline cannot prevent such underruns, i.e., results in lack of data at particular points of time. Second, it is analyzed how inaccurate network quality predictions influence the prefetching results. We show for different scenarios how the response times and data freshness are influenced by too pessimistic and too optimistic predictions. If comparing our location-based prefetching approach with two baselines which apply no prefetching and location-independent prefetching, respectively, we can show that for many scenarios, our approach leads to better response times and data freshness even if there are some prediction inaccuracies.

Future Work

As has already been discussed in Section 4.2.1, the solutions presented in this section primarily aim at caching and data prefetching at powerful mobile devices like smartphones. However, in the IoT, devices with less resources become ubiquitous. Therefore, it is worth to investigate if the mechanisms presented here for caching and prefetching might also be applied for IoT devices in general or in the fog in order to decrease the amount and size of data transmissions in the IoT.

While energy efficiency has not been regarded explicitly in the work presented in this section, it should be taken into account with regard to fog and IoT devices with limited energy storages, since the reduction of used bandwidth is not necessarily the most important factor regarding resource consumption [BBV09]. In addition, the approaches to caching and prefetching could also be combined in order

to ensure an even better user experience. At the moment, the presented approaches are decoupled. In the future, a smart mechanism could actually decide if in a particular context it is a better idea to cache or prefetch data.

4.3 QOS-AWARE SERVICE COMPOSITION

(QoS-aware) service composition has been one of the major research fields in SOC for years and has been widely recognized in the literature [DS05; Gar+16; LDB16; She+14; Str10]. The basic idea of service composition is that in an upcoming Internet of Services (IoS), several (Web) services offer the same functionality, however, with different non-functional capabilities (e.g., throughput, response time, availability) and cost. Hence, the goal of QoS-aware service composition is to find an optimal set of services to be composed under user-defined QoS constraints and (in most cases) at minimal cost. This optimal set of services can then be used to execute a business process which is modeled as a service composition.

Apart from conceptual groundwork, especially optimization approaches have been in the focus of the research on service composition. Both optimal and heuristic approaches have been proposed [Str10], and different variants of QoS parameters to be taken into account, process patterns, and optimization approaches have been investigated. A particular focus was also on the area of semantic Web service composition, where semantic information from the service descriptions were exploited to compose the services, e.g., [CKK15; ELM16].

Despite this, a number of open research questions remain. First, while a number of approaches support complex, structured process patterns (e.g., XOR-blocks, AND-blocks, and repeat loops), there is a lack of solutions for OR-blocks and unstructured process patterns like Directed Acyclic Graphs (DAGs) and Single-Entry-Multiple-Exit (SEME) patterns. Second, probabilities for different paths in a service composition have received little attention. Also, most approaches assume deterministic QoS behavior during service composition, despite the fact that real-world services do often show a stochastic QoS behavior [Ros+09].

In brief, we provide the following research contributions in the field of QoS-aware service composition:

Research Contributions

- We propose optimization approaches which take into account OR-blocks and unstructured process patterns in service compositions.
- We allow the consideration of process path probabilities, thus avoiding that a worst-case analysis needs to be performed dur-

ing optimization. Instead, we provide the possibility to perform an average-case analysis.

- We define a solution to take into account probabilistic QoS values during service composition.
- We present both (ILP- and MILP-based) optimal and heuristic solutions to the service composition problem.

[Sch+11b]: "Optimization of Complex QoS-aware Service Compositions"

In [Sch+11b], we introduce an optimization approach for complex QoS-aware service compositions which explicitly takes into account OR-blocks and unstructured process patterns in terms of DAGs. Therefore, the presented optimization approach extends the related work, which is limited to structured business processes made up from XOR-blocks, AND-blocks, and repeat loops. In contrast to other approaches, which perform a worst-case analysis during service composition, we allow for path probabilities, i.e., the likelihood that a particular path through a process model is chosen during process execution. While other optimal solutions compute every possible execution path, this is not necessary with the approach at hand. Thus, the optimization problem is reduced without losing the capability to identify the optimal service composition under the given QoS constraints.

To achieve this, process models are formally defined and rules for well-formed models are set up. To be able to handle unstructured components during composition, process models are parsed into a hierarchy of process patterns, which have a unique entry node and a unique exit node. For this, the refined process structure tree method is applied [VVK09]. Based on the process models, a system model which provides the foundation for optimization is compiled. As QoS parameters and constraints, the system model considers service execution time, reliability, and throughput of services. In addition, cost are regarded. This makes it necessary to define QoS aggregation functions for summation, multiplication, and min/max operators. For OR- and XOR-blocks, an average-case analysis is performed based on the path probabilities. The possible runs of a DAG are rewritten into an XOR-block, thus allowing to apply the corresponding aggregation functions. If the process steps within a split and join are not arranged sequentially, a recursive pattern interlacing technique is applied to resolve this.

Like in the optimization models presented for elastic processes (see Section 2.3), the goal of the objective function is cost efficiency. The basic non-linear optimization problem is linearized by replacing non-linear aggregations. Thus, the optimization problem can be solved by using ILP, leading to an optimal service execution plan. To increase the scalability of the proposed optimization model, we also apply MILP, which may however lead to invalid solutions. Therefore, a heuristic selection strategy is also applied, which identifies only valid solutions based on the MILP approach.

In the evaluation, the ILP, MILP, and heuristic approach are compared with regard to computation time and solution quality (in terms of cost) for one particular process model. As a baseline, a brute-force algorithm is applied, which finds the optimal solution. To show the performance with regard to scalability, the number of candidate services per task as well as the number of tasks are varied during the evaluation. We show that the runtime performance of the ILP approach increases with the number of tasks and candidate services, while the increases for the MILP and heuristic approaches are significantly smaller. Overall, our three approaches lead to better computation times than the brute-force approach. With regard to cost, the ILP and brute-force approaches lead to the best solution quality, which is not surprising, since both approaches provide an optimal solution. The heuristic solution leads to significantly higher cost, while the MILP-based approach does not always lead to a valid solution and therefore cannot be compared to the other approaches.

While [Sch+11b] integrates probabilities with regard to the paths a particular process instance could follow, probabilistic QoS values are not regarded. Instead, the hard, deterministic QoS bounds defined by service providers are the foundation for the optimization. However, in reality, hard bounds do not realistically mirror the non-functional behavior of Web services [Mie+10; Ros+09] and distributed computational resources in general [Oli+05]. Hence, an optimal service composition execution plan, which has been computed based on deterministic QoS values, may need to be replanned. This may result in significant computational overhead during process runtime.

[Sch+12a]: "Cost-driven Optimization of Complex Service-based Workflows for Stochastic QoS Parameters"

Therefore, in [Sch+12a], we extend our former work presented in [Sch+11b] by integrating the means to take into account probabilistic QoS values during optimization. Also, penalty cost are regarded, i.e., a service provider or service broker has to pay a fee, if the constraints agreed upon in an SLA are not met. We apply the position of a service broker within this paper. The service broker receives service requests from consumers and selects based on the user-defined constraints which services to invoke from different service providers. The goal of the service broker is to meet the constraints of all service consumers at minimal cost or to pay a penalty in order to maximize profits.

To take into account probabilistic QoS values, our former approach, i.e., the optimization step as presented in [Sch+11b], is extended by a simulation step and an adaptation step. For the QoS aggregation, a worst-case analysis is performed. It should be noted that the work presented in [Sch+12a] does not take into account OR-blocks, repeat loops, and DAGs, but is limited to sequences, XOR-, and AND-blocks. The resulting objective function is again linearized and can therefore be solved by the application of ILP techniques.

During the simulation step, the calculated 'optimal' (under the assumption of deterministic QoS values) execution plan is used to com-

pute the expected runtime behavior of the service composition. To simulate this, the simulation relies on the availability of probabilistic QoS specifications for the offered services. The simulation step is carried out a predefined number of times, thus allowing to draw specific values for all QoS parameters from the probability distribution. Also, the followed paths within a process model (in case of XOR-blocks) are drawn based on their path probabilities.

In the adaptation step, a greedy heuristic is applied in order to mitigate the risk of potentially occurring constraint violations. The goal of the heuristic is to reduce the weighted empirical standard deviation of a QoS value in order to minimize uncertainty and risk during service invocations. For this, the results from the simulation are taken into account. The goal of the heuristic is to exclude these services from the execution path which could lead to high penalty cost due to unexpected runtime behavior. For the accruing penalty cost, we assume linear penalty fees. After the possible adaptation has been computed, a comparison with the original execution plan is performed and the simulation and adaptation steps might be repeated.

In the evaluation, we assess the effect of different parameterizations of the adaptation step in order to show how this influences the overall cost for the service broker. For this, one process model is applied. The different parameter settings lead to a cost reduction of 6%-8.5%. However, the application of the simulation and adaptation steps also leads to additional computation time (up to 10 times higher, depending on the settings).

[Sch+13a]: "Optimizing Complex Service-based Workflows for Stochastic QoS Parameters"

[Sch+13a] is an invited extension of [Sch+12a]. Several enhancements have been made. Most notably, additional complex process patterns are regarded, i.e., OR-blocks and repeat loops. Also, we introduce SEME loops [Dum+10] as a novel, unstructured pattern not regarded within the related work before. In order to take into account these patterns, additional aggregation mechanisms for both average-case and worst-case analysis are provided.

Compared to [Sch+12a], two additional adaptation heuristics are developed. The first one adapts to specific aggregations of mean values and standard deviations while the second one provides an adaptation based on unweighted differences in standard deviations; the heuristic from [Sch+12a] is based on weighted differences in standard deviations.

Within the evaluation, we reuse the process model from [Sch+12a] and utilize two additional process models. Again, the goal of the evaluation is to assess the performance of the proposed optimization approach with regard to computation time and cost. In addition to the independent variables used in [Sch+12a] (i.e., greed, annealing, penalty fee, process model), also the degree of conservativeness is taken into account. This variable indicates which quantile of the distribution function for each QoS parameter is assumed in the deter-

ministic optimization and therefore describes how conservative the service broker is with regard to risks. For the independent variables that have already been applied in [Sch+12a], additional settings are evaluated. Also, different distributions such as the triangle distribution have been used accounting for stochastic QoS behavior.

We are able to show that depending on the process model, the settings of the independent variables, and the applied adaptation heuristic, cost reductions of up to 18% can be achieved, with the newly introduced adaptation heuristics performing better than the adaptation heuristic from [Sch+12a]. This is especially the case for more complex process models. On the downside, this increase in cost efficiency comes at the price of computational overhead.

[Sch+14a]: "Towards Heuristic Optimization of Complex Service-based Workflows for Stochastic QoS Attributes"

While [Sch+12a; Sch+13a] take into account SEME loops, DAGs as presented in [Sch+11b] are not regarded. We extend our optimization approach in [Sch+14a] to address DAGs. Again, a service broker scenario is applied and we make use of the three steps optimization, simulation, and adaptation. For the adaptation step, we apply a novel genetic algorithm instead of the previously introduced heuristics from [Sch+12a; Sch+13a].

Within the evaluation, we evaluate the genetic algorithm with regard to cost and computation time. A comparison with an ILP-based solution and the adaptation heuristic from [Sch+12a] is performed. We use one complex process model for the evaluation. As independent variables, greed, annealing, and penalty fees are applied. We are able to show that by applying the genetic algorithm, the cost can be reduced by up to 28.4% (compared to the ILP-based baseline), depending on the setting of the independent variables. However, the adaptation heuristic even performs better, with a cost reduction of up to 30.7% (compared to the ILP-based baseline). Once again, this comes at the price of additional computation time: The evaluated adaptation heuristic increases the computation time by a factor of 22.2–55.1, while the genetic algorithm leads to an increase by a factor of 3.6–10.2.

Overview

Table 4 provides an overview of the approaches discussed in Section 4.3, and exemplifies the iterative research approach used with regard to the optimization of complex service compositions.

Future Work

While research on service compositions has led to a huge amount of publications, the technical focus has shifted to elastic processes. As discussed in Section 2.3, we have provided several approaches to this field in recent years. Nevertheless, there are certain features which have been covered in service composition, but not in the field of elastic processes yet, e.g., probabilistic process paths, probabilistic QoS values, average-case aggregation (instead of worst-case aggregation) of QoS values, or DAG and SEME process patterns.

In turn, recent discussions have brought service composition back to the attention of the research community [Bou+17], however, with a

Table 4: Overview of Contributions in Optimization of Service Compositions (☑: Covered, ⊠: Partially Covered, □: Not Covered)

	Sequence, XOR, AND	Loops	OR	DAGs	SEME	Probabilities	Penalty Cost	Approach
[Sch+11b]	☑	☑	☑	☑	□	⊠	□	ILP, MILP, MILP+heuristic
[Sch+12a]	☑	□	□	□	□	☑	☑	ILP, adaptation heuristic
[Sch+13a]	☑	☑	☑	□	☑	☑	☑	ILP, adaptation heuristics
[Sch+14a]	☑	☑	☑	☑	□	☑	☑	ILP, genetic algorithm

largely different focus. Instead of the emphasis on Web services and the WS-* stack, today, the focus is rather on the composition of large amounts of services in a system which does not necessarily represent a (business) process. Instead, e.g., a Smart City could be seen as a large-scale ecosystem composed from services [Xu+15]. Despite the conceptual differences between process-based service compositions and this new type of large-scale service compositions, there is a need to ensure QoS, leading to new research questions in the field.

Research Challenges

Table 5 shows how the single publications presented in Chapter 4 contribute to the research challenges discussed in Section 1.2. Naturally, the focus of the papers related to mobile services (see Section 4.2) is on RC6. For the publications explicitly aiming at mobile service invocations, i.e., [Pap+11a; Pap+11b; Pap+12; Pap+14], it should be noted that RC4 is only covered at the level of communication overhead and the applied proxy concept to decrease and avoid data transfers. These four publications also provide basic concepts on how to decrease data transfer in the IoT, thus partially contributing to RC5. [Bor+16; Hum+14] provide basic concepts which could be applied to the IoT, therefore implicitly contributing to RC5, but do not control the infrastructure, i.e., do not aim at RC4.

For the papers related to QoS-aware service composition (see Section 4.3), the focus is on RC1 and RC3, since resources (in terms of services) for the execution of processes are selected and a particular focus is on adherence to QoS constraints. RC5 is partially taken into account, since service compositions could be applied to enact industrial processes.

Table 5: Contributions of Publications to Research Challenges in Service-oriented Computing (☑: Primary Concern, ⊠: Secondary Concern, □: No Concern)

	RC1	RC2	RC3	RC4	RC5	RC6
Constituent Publications						
[Sch+12a]	☑	□	☑	□	⊠	□
[Pap+14]	□	□	□	⊠	⊠	☑
Further Publications						
[Pap+11a]	□	□	□	⊠	⊠	☑
[Pap+11b]	□	□	□	⊠	⊠	☑
[Sch+11b]	☑	□	☑	□	⊠	□
[Pap+12]	□	□	□	⊠	⊠	☑
[Sch+13a]	☑	□	☑	□	⊠	□
[Hum+14]	□	□	□	□	⊠	☑
[Sch+14a]	☑	□	☑	□	⊠	□
[Bor+16]	□	□	□	□	⊠	☑

4.4 PUBLICATIONS

1. Apostolos Papageorgiou, André Miede, Dieter Schuller, Stefan Schulte, and Ralf Steinmetz. "Always Best Served: On the Behaviour of QoS- and QoE-based Algorithms for Web Service Adaptation." In: *PERCOM Workshops – 8th International Workshop on Managing Ubiquitous Communications and Services (MUCS 2011)*. IEEE Computer Society, Washington, DC, USA, 2011, pp. 76–81. URL: http://dx.doi.org/10.1109/PERCOMW.2011.5766975.
2. Apostolos Papageorgiou, Marius Schatke, Stefan Schulte, and Ralf Steinmetz. "Enhancing the Caching of Web Service Responses on Wireless Clients." In: *9th IEEE International Conference on Web Services (ICWS 2011)*. IEEE, 2011, pp. 9–16. URL: http://dx.doi.org/10.1109/ICWS.2011.52.
3. Dieter Schuller, Artem Polyvyanyy, Luciano Garcıa-Bañuelos, and Stefan Schulte. "Optimization of Complex QoS-aware Service Compositions." In: *9th International Conference on Service Oriented Computing (ICSOC 2011)*. Vol. 7084. Lecture Notes in Computer Science. Springer, 2011, pp. 452–466. URL: http://dx.doi.org/10.1007/978-3-642-25535-9.
4. Apostolos Papageorgiou, Marius Schatke, Stefan Schulte, and Ralf Steinmetz. "Lightweight Wireless Web Service Communication Through Enhanced Caching Mechanisms." In: *Inter-*

national Journal of Web Services Research 9.2 (2012), pp. 42–68. URL: http://dx.doi.org/10.4018/jwsr.2012040103.

5. Dieter Schuller, Ulrich Lampe, Julian Eckert, Ralf Steinmetz, and Stefan Schulte. "Cost-driven Optimization of Complex Service-based Workflows for Stochastic QoS Parameters." In: *10th IEEE International Conference on Web Services (ICWS 2012)*. IEEE, 2012, pp. 66–73. URL: http://dx.doi.org/10.1109/ICWS.2012.50.

6. Dieter Schuller, Ulrich Lampe, Julian Eckert, Ralf Steinmetz, and Stefan Schulte. "Optimizing Complex Service-based Workflows for Stochastic QoS Parameters." In: *International Journal of Web Services Research* 10.4 (2013), pp. 1–38. URL: http://dx.doi.org/10.4018/ijwsr.2013100101.

7. Waldemar Hummer, Stefan Schulte, Philipp Hoenisch, and Schahram Dustdar. "Context-Aware Data Prefetching in Mobile Service Environments." In: *IEEE Fourth International Conference on Big Data and Cloud Computing (BDCloud 2014)*. IEEE, 2014, pp. 214–221. URL: http://dx.doi.org/10.1109/BDCloud.2014.104.

8. Apostolos Papageorgiou, André Miede, Stefan Schulte, Dieter Schuller, and Ralf Steinmetz. "Decision Support for Web Service Adaptation." In: *Pervasive and Mobile Computing* 12 (2014), pp. 197–213. URL: http://dx.doi.org/10.1016/j.pmcj.2013.10.004.

9. Dieter Schuller, Melanie Siebenhaar, Ronny Hans, Olga Wenge, Ralf Steinmetz, and Stefan Schulte. "Towards Heuristic Optimization of Complex Service-based Workflows for Stochastic QoS Attributes." In: *12th IEEE International Conference on Web Services (ICWS 2014)*. IEEE, 2014, pp. 361–368. URL: http://www.dx.doi.org/10.1109/ICWS.2014.59.

10. Michael Borkowski, Olena Skarlat, Stefan Schulte, and Schahram Dustdar. "Prediction-Based Prefetch Scheduling in Mobile Service Applications." In: *IEEE 5th International Conference on Mobile Services (MS 2016)*. IEEE, 2016, pp. 41–48. URL: http://dx.doi.org/10.1109/MS.2016.13.

4.4.1 *Constituent Publications*

Paper [5] is one of the major outcomes of the PhD studies of Dieter Schuller, who was a member of my research group at TU Darmstadt. The paper discusses the inclusion of probabilistic QoS values during service composition. I supervised the implementation work and revised the paper.

Paper [8] is a major outcome of the PhD studies of Apostolos Papageorgiou. I co-supervised Dr. Papageorgiou's PhD thesis. For the

paper, I acted along with André Miede and Apostolos Papageorgiou as the main author. For this, I wrote and revised major parts of the paper based on the input from Dr. Papageorgiou's PhD thesis. In addition, I supervised the implementation work done by Apostolos Papageorgiou.

4.4.2 *Further Publications*

Paper [1] presents a first version of our work on decision support for Web service adaptation. I supervised the implementation work and revised the paper. In addition, I contributed to the conceptualization of the MML, which was originally described in the proposal of the *Green Mobility* project, for which I was the main scientific author.

Papers [2; 4] present a client-side caching approach for Web services which guarantees freshness of the cached service responses. [4] is an invited extension of [2]. This work was part of the PhD studies of Apostolos Papageorgiou. I guided the implementation and writing work of Mr. Papageorgiou for these papers. Also, as described above, I contributed to the conceptualization of the MML which is again used as technical framework for mobile Web service invocations. In addition, I revised both papers.

Paper [3] was another major outcome of the PhD studies of Mr. Schuller. Within the paper, an approach to take into account complex process patterns like DAGs and OR-blocks during service composition optimization is discussed. The paper was written together with an international team of experts. I supervised the implementation work leading to this publication and revised the paper.

Paper [6] is an invited extension of [5]. Again, I supervised the implementation work and revised the paper.

Paper [7] is a result of the SIMPLI-CITY project. I described the idea of data prefetching in mobility scenarios first in the according project proposal. Later on, this idea was implemented and evaluated by Waldemar Hummer for this paper. I contributed the introduction, the mobility scenario, and the discussion of the related work to the paper. In addition, I made contributions to the system model and the conceptualization of the prefetching strategies. Also, I revised the remaining sections, which were written by Waldemar Hummer.

Paper [9] is an extension of [3; 5; 6]. Apart from the contributions to the former work, I revised the paper and supervised the implementation, which was conducted by Dieter Schuller.

Paper [10] is a result of the Master thesis of Michael Borkowski, which I supervised. Again inspired by the SIMPLI-CITY project, I defined the topic for this Master thesis. As the academic supervisor, I oversaw the implementation and evaluation work of Michael Borkowski. For the paper, I substantially revised the introduction and supervised the overall writing process.

5 SUMMARY AND OUTLOOK

5.1 SUMMARY

Within this thesis, different contributions to adaptive resource and task scheduling for the Industrial Internet have been presented. In the following paragraphs, a brief summary of the contributions of the presented publications will be given. For this, we follow the research challenges discussed in Section 1.2.

RC1: HOW TO EFFICIENTLY ALLOCATE RESOURCES FOR ELASTIC SMART SYSTEMS IN VOLATILE SCENARIOS? Resource allocation for smart systems has been regarded within the presented research in manifold ways, including the allocation of cloud-based computational resources for the enactment of elastic processes [Hoe+13; HSD13; Sch+13d; Hoe+15a; Sch+15; Hoe+16], elastic stream processing [Hoc+16a; Hoc+17], fog computing [Ska+16; Ska+17a; Ska+17b], cloud manufacturing [Sch+14b], and within service compositions [Sch+11b; Sch+12a; Sch+13a; Sch+14a].

RC2: HOW TO SCHEDULE TASK ON THESE RESOURCES? Resource allocation (i.e., RC1) and task scheduling are very often regarded together. This is also the case in many of the contributions presented in this thesis, i.e., with regard to elastic processes [Hoe+13; HSD13; Sch+13d; Hoe+15a; Sch+15; Hoe+16], fog computing [Ska+16; Ska+17a; Ska+17b], and elastic stream processing [Hoc+17].

RC3: HOW TO TAKE INTO ACCOUNT QOS PROPERTIES? QoS properties are used as constraints in many of the presented research contributions, including most of the ones already mentioned above for RC1-RC2 in this discussion. This includes our contributions to the fields of elastic processes [Hoe+13; HSD13; Sch+13d; Hoe+15a; Sch+15; Hoe+16], redundant data storages in the cloud [WHS16; Wai+17], elastic stream processing [Hoc+15; Hoc+16a; Hoc+17], service compositions [Sch+11b; Sch+12a; Sch+13a; Sch+14a], and fog computing [Ska+16; Ska+17a; Ska+17b].

RC4: HOW TO CONTROL THE IT INFRASTRUCTURE? In order to enact the methods and solutions provided with regard to RC1-RC3, it is necessary to control the according IT infrastructure. For this, conceptual and technical contributions have been presented for the fields of elastic processes [Hoe+13; HSD13; Sch+13b; Sch+13c; Sch+13d; Hoe+15a;

Sch+15; Hoe+16], redundant data storages in the cloud [WHS16; Wai+17], cloud manufacturing [Sch+14b], elastic stream processing [Hoc+16a; Hoc+17], fog computing [Ska+16; Ska+17a; Ska+17b], and mobile service invocations [Pap+11a; Pap+11b; Pap+12; Pap+14].

RC5: HOW TO SUPPORT INDUSTRIAL INTERNET PROCESSES CONCEPTUALLY AND RESOURCE-WISE? While almost all research contributions presented in this paper contribute to this research challenge indirectly, some publications explicitly focus on this field by introducing approaches to realize cloud manufacturing [Sch+12b; Sch+14b].

RC6: HOW TO ADAPT (WEB) SERVICES FOR USAGE ON MOBILE DEVICES? The adaptation of (Web) services for usage on mobile devices like smartphones has been discussed within this thesis with regard to the adaptation of communication protocols [Pap+11a; Pap+14], data prefetching [Hum+14; Bor+16], and caching [Pap+11b; Pap+12]. The goal of all of these research contributions is to decrease communication overhead and to provide data even in the case of low network quality or unavailable network access. In addition, this is a topic in fog computing, where IoT-based mobile devices are applied [Ska+16; Ska+17a; Ska+17b].

Different methodologies have been utilized to answer the discussed research challenges. However, a particular focus was on the usage of optimization techniques (ILP, MILP, and heuristics), especially for RC1-RC3. For all research challenges, functional prototypes have been implemented; for RC4-RC5, the focus of the contributions is on the required software frameworks. With regard to RC6, the primary methodological approach relies on the application of machine learning algorithms, however, optimization approaches are also regarded.

5.2 OUTLOOK

Possible future work directions have already been discussed in Chapters 2-4. In the following paragraphs, we will take these ideas into account and provide a bigger picture on possible future research directions in the Industrial Internet.

Containers

With regard to adaptive resource and task scheduling for the Industrial Internet, two particular major trends can currently be observed. First, instead of using the rather coarse-grained computational resources offered by VMs, there is a trend towards the usage of software container technologies like Docker. We have already provided some first solutions in this field for general container placement [Hoe+15b] and elastic stream processing [Hoc+17], but containers could also replace VMs for the enactment of elastic processes, allowing to control the computational resources on a finer-grained level. For this, it is

necessary to conceptualize and implement new optimization models. Also, it is necessary to develop software frameworks which are able to control and manage a container-based process landscape.

Cognitive Systems

Fog computing could also become an important foundation for the establishment of cognitive systems. Such systems provide cognitive capabilities through data processing and machine learning. Especially in industry-related scenarios, cognitive systems are estimated to provide a disruptive paradigm shift by automating tasks and by supporting users during decision-making.

The current state of the art in cognitive systems (e.g., the well-known IBM Watson) is based on well-defined system boundaries and centralized approaches. In contrast, IoT-based systems feature system-inherent volatility and uncertainty, since system boundaries are not fixed and may quickly change over time and space. For fog- and IoT-based cognitive systems, novel methodologies are needed which are able to make these systems possible even in the case of uncertain system boundaries and non-deterministic changes in the system and its context. Hence, research is needed with regard to modeling of fog-based cognitive systems, and the conceptualization and deployment of distributed, large-scale cognitive systems.

Blockchain

One particular topic in the Industrial Internet and in the IoT in general is the verification of actions, i.e., tracking and verification of actions performed by the involved parties. In the world of cryptocurrencies like Bitcoin, a similar problem arises, since it is necessary to track and verify monetary transactions in order to avoid that particular funds are spent more than once or by parties not owning these funds. For this, Blockchain technologies are applied. While Blockchain technologies are not appropriate in all possible settings [ET17], it is worth investigating how Blockchain technologies could be applied in the Industrial Internet for verification, tracking, and other purposes. While we have provided a first contribution to the specific field of business process verification [Pry+17] and have contributed to a community paper on BPM and the Blockchain [Men+17], research in this field is still at its beginning and manifold research possibilities exist, e.g., to replace publish-subscribe mechanisms by Blockchains in order to make sure that data is really provided to the subscribers, or to enable data provenance by the means of Blockchains. Last but not least, there is a need for Blockchain-native applications, i.e., software which inherently takes into account Blockchain capabilities, but also regards the uncertainty arising because of the usage of Blockchains.

BIBLIOGRAPHY

[Aal+03] Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Bartek Kiepuszewski, and Alistair P. Barros. "Workflow Patterns." In: *Distributed and Parallel Databases* 14.1 (2003), pp. 5–51. URL: https://doi.org/10.1023/A:1022883727209.

[AB06] Mustafa Adaçal and Ayse Basar Bener. "Mobile Web Services: A New Agent-Based Framework." In: *IEEE Internet Computing* 10.3 (2006), pp. 58–65. URL: https://doi.org/10.1109/MIC.2006.59.

[aca13] acatech. *Recommendations for implementing the strategic initiative INDUSTRIE 4.0*. Final report of the Industrie 4.0 Working Group. 2013.

[aca15] acatech. *Living in a networked world: Integrated research agenda Cyber-Physical Systems (agendaCPS)*. Ed. by Eva Geisberger and Manfred Broy. 2015.

[ACY03] Angela Andal-Ancion, Phillip A. Cartwright, and George S. Yip. "The Digital Transformation of Traditional Business." In: *MIT Sloan Management Review* 44.3 (2003).

[ADJ05] Naresh Apte, Keith Deutsch, and Ravi Jain. "Wireless SOAP: Optimizations for Mobile Wireless Web Services." In: *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web (WWW '05)*. ACM, 2005, pp. 1178–1179. URL: http://doi.acm.org/10.1145/1062745.1062927.

[AIM10] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The Internet of Things: A survey." In: *Computer Networks* 54 (15 2010), pp. 2787–2805. URL: https://doi.org/10.1016/j.comnet.2010.05.010.

[AKM07] Erwin Aitenbichler, Jussi Kangasharju, and Max Mühlhäuser. "MundoCore: A light-weight infrastructure for pervasive computing." In: *Pervasive and Mobile Computing* 3 (4 2007), pp. 332–361. URL: https://doi.org/10.1016/j.pmcj.2007.04.002.

[AlF+15] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications." In: *IEEE Communication Surveys & Tutorials* 17.4 (2015), pp. 2347–2376. URL: https://doi.org/10.1109/COMST.2015.2444095.

[APR00] Rumen Andonov, Vincent Poirriez, and Sanjay Rajopadhye. "Unbounded knapsack problem: Dynamic programming revisited." In: *European Journal of Operational Research* 123 (2 2000), pp. 394–407. URL: https://doi.org/10.1016/S0377-2217(99)00265-9.

[Arm+10] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. "A View of Cloud Computing." In: *Communications of the ACM* 53 (4 2010), pp. 50–58. URL: https://doi.org/10.1145/1721654.1721672.

[Bab+02] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. "Models and Issues in Data Stream Systems." In: *Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '02)*. ACM, 2002, pp. 1–16. URL: http://doi.acm.org/10.1145/543613.543615.

[BAB12] Anton Beloglazov, Jemal H. Abawajy, and Rajkumar Buyya. "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing." In: *Future Generation Computer Systems* 28.5 (2012), pp. 755–768. URL: http://dx.doi.org/10.1016/j.future.2011.04.017.

[BBV09] Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. "Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications." In: *9th ACM SIGCOMM Conference on Internet Measurement (IMC '09)*. ACM, 2009, pp. 280–293. URL: http://doi.acm.org/10.1145/1644893.1644927.

[Bec+99] Jörg Becker, Christoph von Uthmann, Michael zur Muehlen, and Michael Rosemann. "Identifying the Workflow Potential of Business Processes." In: *32nd Annual Hawaii International Conference on System Sciences (HICSS-32)*. IEEE, 1999. URL: https://doi.org/10.1109/HICSS.1999.772962.

[Ber+11] David Bermbach, Markus Klems, Stefan Tai, and Michael Menzel. "MetaStorage: A Federated Cloud Storage System to Manage Consistency-Latency Tradeoffs." In: *IEEE 4th International Conference on Cloud Computing (CLOUD 2011)*. IEEE, 2011, pp. 452–459. URL: https://doi.org/10.1109/CLOUD.2011.62.

[Bes+13] Kahina Bessai, Samir Youcef, Ammar Oulamara, Claude Godart, and Selmin Nurcan. "Scheduling Strategies for Business Process Applications in Cloud Environments." In: *International Journal of Grid and High Performance Computing* 5.4 (2013), pp. 65–78. URL: http://dx.doi.org/10.4018/ijghpc.2013100105.

[Biso6] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag New York, 2006.

[Bon+12] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. "Fog Computing and Its Role in the Internet of Things." In: *MCC Workshop on Mobile Cloud Computing*. ACM, 2012, pp. 13–16. URL: https://doi.org/10.1145/2342509.2342513.

[Bon+14] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. "Fog Computing: A Platform for Internet of Things and Analytics." In: *Big Data and Internet of Things: A Roadmap for Smart Environments*. Ed. by Nik Bessis and Ciprian Dobre. Vol. 546. Studies in Computational Intelligence. Cham, Switzerland: Springer, 2014, pp. 169–186. URL: https://doi.org/10.1007/978-3-319-05029-4_7.

[Bor+14] Amir Hossein Borhani, Philipp Leitner, Bu-Sung Lee, Xiaorong Li, and Terence Hung. "WPress: An Application-Driven Performance Benchmark for Cloud-Based Virtual Machines." In: *18th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2014)*. IEEE, 2014, pp. 101–109. URL: https://doi.org/10.1109/EDOC.2014.23.

[Bor+16] Michael Borkowski, Olena Skarlat, Stefan Schulte, and Schahram Dustdar. "Prediction-Based Prefetch Scheduling in Mobile Service Applications." In: *IEEE 5th International Conference on Mobile Services (MS 2016)*. IEEE, 2016, pp. 41–48. URL: http://dx.doi.org/10.1109/MS.2016.13.

[Bot+16] Alessio Botta, Walter de Donato, Valerio Persico, and Antonio Pescapé. "Integration of Cloud Computing and Internet of Things: A Survey." In: *Future Generation Computer Systems* 56 (2016), pp. 684–700. URL: https://doi.org/10.1016/j.future.2015.09.021.

[Bou+17] Athman Bouguettaya et al. "A Service Computing Manifesto: The Next 10 Years." In: *Communications of the ACM* 60 (4 2017), pp. 64–72. URL: http://doi.acm.org/10.1145/2983528.

[BRC10] Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N. Calheiros. "InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services." In: *10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2010), Part I*. Vol. 6081. Lecture Notes in Computer Science. Springer, 2010, pp. 13–31. URL: http://dx.doi.org/10.1007/978-3-642-13119-6_2.

[Bre89] Richard P. Brent. "Efficient implementation of the first-fit strategy for dynamic storage allocation." In: *ACM Transactions on Programming Languages and Systems* 11.3 (1989), pp. 388–403. URL: https://doi.org/10.1145/65979.65981.

[Bru13] Jon Bruner. *Industrial Internet*. Sebastopol, CA, USA: O'Reilly Media Inc., 2013.

[BSH16] Michael Borkowski, Stefan Schulte, and Christoph Hochreiner. "Predicting Cloud Resource Utilization." In: *9th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2016)*. ACM, 2016, pp. 37–42. URL: http://dx.doi.org/10.1145/2996890.2996907.

[Buy+09] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." In: *Future Generation Computer Systems* 25.6 (2009), pp. 599–616. URL: https://doi.org/10.1016/j.future.2008.12.001.

[BWE04] Gregório Baggio, Jacques Wainer, and Clarence A. Ellis. "Applying Scheduling Techniques to Minimize the Number of Late Jobs in Workflow Systems." In: *19th Symposium on Applied Computing (SAC 2004)*. ACM, 2004, pp. 1396–1403. URL: https://doi.org/10.1145/967900.968180.

[BXW14] Zhuming Bi, Li Da Xu, and Chengen Wang. "Internet of Things for Enterprise Systems of Modern Manufacturing." In: *IEEE Transactions on Industrial Informatics* 20 (2 2014), pp. 1537–1546. URL: https://doi.org/10.1109/TII.2014.2300338.

[Cal+11] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A.F. De Rose, and Raykumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." In: *Software – Practice and Experience* 41 (1 2011), pp. 23–50. URL: https://doi.org/10.1002/spe.995.

[Car+04] Jorge S. Cardoso, Amit P. Sheth, John A. Miller, Jonathan Arnold, and Krys Kochut. "Quality of service for workflows and web service processes." In: *Journal of Web Semantics* 1.3 (2004), pp. 281–308. URL: http://dx.doi.org/10.1016/j.websem.2004.03.001.

[Cat11] Rick Cattell. "Scalable SQL and NoSQL data stores." In: *ACM SIGMOD Record* 39.4 (2011), pp. 12–27. URL: https://doi.org/10.1145/1978915.1978919.

[CCL09] Claudia Canali, Michele Colajanni, and Riccardo Lancellotti. "Performance Evolution of Mobile Web-Based Services." In: *IEEE Internet Computing* 13 (2 2009), pp. 60–68. URL: https://doi.org/10.1109/MIC.2009.43.

[Che+03] Mitch Cherniack, Hari Balakrishnan, Magdalena Balazinska, Donald Carney, Ugur Cetintemel, Ying Xing, and Stan Zdonik. "Scalable Distributed Stream Processing." In: *First Biennial Conference on Innovative Data Systems Research (CIDR 2003)*. www.cidrdb.org, 2003. URL: http://www-db.cs.wisc.edu/cidr/cidr2003/program/p23.pdf.

[Chr+09] Delphine Christin, Andreas Reinhardt, Parag S. Mogre, and Ralf Steinmetz. "Wireless Sensor Networks and the Internet of Things: Selected Challenges." In: *8th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze*. 2009, pp. 31–34.

[CK97] Thomas Aidan Curran and Gerhard Keller. *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1997.

[CKK15] Xiaoqi Cao, Patrick Kapahnke, and Matthias Klusch. "SPSC: Efficient Composition of Semantic Services in Unstructured P2P Networks." In: *12th European Semantic Web Conference (ESWC 2015)*. Vol. 9088. Lecture Notes in Computer Science. Springer, 2015, pp. 455–470. URL: https://doi.org/10.1007/978-3-319-18818-8_28.

[CLR10] Michael Chui, Markus Löffler, and Roger Roberts. "The Internet of Things." In: *McKinsey Quarterly* 2 (2010), pp. 1–9.

[CML14] Min Chen, Shiwen Mao, and Yunhao Liu. "Big Data: A Survey." In: *Mobile Networks and Applications* 19.2 (2014), pp. 171–209. URL: http://dx.doi.org/10.1007/s11036-013-0489-0.

[Cop+13] Georgiana Copil, Daniel Moldovan, Hong-Linh Truong, and Schahram Dustdar. "SYBL: An Extensible Language for Controlling Elasticity in Cloud Cpplications." In: *13th IEEE/ACM International Symposium on Cluster, Cloud and*

Grid Computing (CCGrid 2013). IEEE, 2013, pp. 112–119. URL: https://doi.org/10.1109/CCGrid.2013.42.

[Cor+15] Rudyar Cortés, Xavier Bonnaire, Olivier Marin, and Pierre Sens. "Stream Processing of Healthcare Sensor Data: Studying User Traces to Identify Challenges from a Big Data Perspective." In: *4th International Workshop on Body Area Sensor Networks (BASNet-2015)*. Vol. 52. Procedia Computer Science. Elsevier, 2015, pp. 1004–1009. URL: https://doi.org/10.1016/j.procs.2015.05.093.

[CP06] Carlo Combi and Giuseppe Pozzi. "Task Scheduling for a Temporal Workflow Management System." In: *Thirteenth International Symposium on Temporal Representation and Reasoning (TIME 2006)*. IEEE, 2006, pp. 61–68. URL: https://doi.org/10.1109/TIME.2006.26.

[CS13] Pedro Casas and Raimund Schatz. "Quality of Experience in Cloud services: Survey and measurements." In: *Computer Networks* 68 (2013), pp. 149–165. URL: https://doi.org/10.1016/j.comnet.2014.01.008.

[Das+16] Amir Vahid Dastjerdi, Harshit Gupta, Rodrigo N. Calheiros, Soumya K. Ghosh, and Rajkumar Buyya. "Fog Computing: Principles, Architectures, and Applications." In: *Internet of Things: Principles and Paradigms*. Ed. by Rajkumar Buyya and Amir Vahid Dastjerdi. Cambridge, MA, USA: Morgan Kaufmann, 2016. Chap. 4, pp. 61–75.

[Dav+12] Jim Davis, Thomas Edgar, James Porter, John Bernaden, and Michael Sarli. "Smart manufacturing, manufacturing intelligence and demand-dynamic performance." In: *Computers & Chemical Engineering* 47 (2012), pp. 145–156. URL: https://doi.org/10.1016/j.compchemeng.2012.06.037.

[Deg16] Christophe Degryse. *Digitalisation of the economy and its impact on labour markets*. Working Paper 2016.02. Brussels, Belgium: european trade union institute, 2016.

[DS05] Schahram Dustdar and Wolfgang Schreiner. "A survey on web services composition." In: *International Journal of Web and Grid Services* 1.1 (2005), pp. 1–30. URL: https://doi.org/10.1504/IJWGS.2005.007545.

[Dum+10] Marlon Dumas, Luciano Garcıa-Bañuelos, Artem Polyvyanyy, Yong Yang, and Liang Zhang. "Aggregate Quality of Service Computation for Composite Services." In: *8th International Conference on Service-Oriented Computing (ICSOC 2010)*. Vol. 6470. Lecture Notes in Computer Science. Springer, 2010, pp. 213–227. URL: https://doi.org/10.1007/978-3-642-17358-5_15.

[Dus+11] Schahram Dustdar, Yike Guo, Benjamin Satzger, and Hong-Linh Truong. "Principles of Elastic Processes." In: *IEEE Internet Computing* 15 (5 2011), pp. 66–71. URL: https://doi.org/10.1109/MIC.2011.121.

[DWC10] Tharam Dillon, Chen Wu, and Elizabeth Chang. "Cloud Computing: Issues and Challenges." In: *24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010)*. IEEE, 2010, pp. 27–33. URL: http://dx.doi.org/10.1109/AINA.2010.187.

[ELM16] Rik Eshuis, Freddy Lécué, and Nikolay Mehandjiev. "Flexible Construction of Executable Service Compositions from Reusable Semantic Knowledge." In: *ACM Transactions on the Web* 10.1 (2016), 5:1–5:27. URL: http://doi.acm.org/10.1145/2842628.

[EPR99] Johann Eder, Euthimios Panagos, and Michael Rabinovich. "Time Constraints in Workflow Systems." In: *11th International Conference on Advanced Information Systems Engineering (CAiSE'99)*. Vol. 1626. Lecture Notes in Computer Science. Springer, 1999, pp. 286–300. URL: https://doi.org/10.1007/3-540-48738-7_22.

[ET17] Jacob Eberhardt and Stefan Tai. "On or Off the Blockchain? Insights on Off-Chaining Computation and Data." In: *6th European Conference on Service-Oriented and Cloud Computing (ESOCC 2017)*. Vol. 10465. Lecture Notes in Computer Science. Springer, 2017, pp. 3–15. URL: https://doi.org/10.1007/978-3-319-67262-5_1.

[Eva11] Dave Evans. *The Internet of Things – How the Next Evolution of the Internet Is Changing Everything*. CISCO White Paper. 2011. URL: http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.

[FLR13] Niroshinie Fernando, Seng Wai Loke, and J. Wenny Rahayu. "Mobile cloud computing: A survey." In: *Future Generation Computer Systems* 29 (1 2013), pp. 84–106. URL: https://doi.org/10.1016/j.future.2012.05.023.

[FPV14] Maria Fazio, Antonio Puliafito, and Massimo Villari. "IoT4S: a new architecture to exploit sensing capabilities in smart cities." In: *International Journal of Web and Grid Services* 10.2/3 (2014), pp. 114–138. URL: https://doi.org/10.1504/IJWGS.2014.060255.

[Gar+16] Martin Garriga, Cristian Mateos, Andres Flores, Alejandra Cechich, and Alejandro Zunino. "RESTful service composition at a glance: A survey." In: *Journal of Network and Computer Applications* 60 (2016), pp. 32–53. URL: https://doi.org/10.1016/j.jnca.2015.11.020.

[GB12] Guilherme Galante and Luis Carlos Erpen De Bona. "A Survey on Cloud Computing Elasticity." In: *IEEE Fifth International Conference on Utility and Cloud Computing (UCC 2012)*. IEEE, 2012, pp. 263–270. URL: https://doi.org/10.1109/UCC.2012.30.

[Ged+08] Buğgra Gedik, Henrique Andrade, Kun-Lung Wu, Philip S. Yu, and Myungcheol Doo. "SPADE: The System S Declarative Stream Processing Engine." In: *International Conference on Management of Data (SIGMOD)*. ACM, 2008, pp. 1123–1134. URL: https://doi.org/10.1145/1376616.1376729.

[Geo+16] Dimitrios Georgakopoulos, Prem Prakash Jayaraman, Maria Fazio, Massimo Villari, and Rajiv Ranjan. "Internet of Things and Edge Cloud Computing Roadmap for Manufacturing." In: *IEEE Cloud Computing* 3 (4 2016), pp. 66–73. URL: https://doi.org/10.1109/MCC.2016.91.

[GH09] Vehbi C. Gungor and Gerhard P. Hancke. "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches." In: *IEEE Transactions on Industrial Electronics* 56 (10 2009), pp. 4258–4265. URL: https://doi.org/10.1109/TIE.2009.2039455.

[Gil+07] Virgilio Gilart-Iglesias, Francisco Macia-Perez, Diego Marcos-Jorquera, and Francisco Jose Mora-Gimeno. "Industrial Machines as a Service: Modelling industrial machinery processes." In: *5th IEEE International Conference on Industrial Informatics (INDIN 2007)*. IEEE, 2007, pp. 737–742. URL: https://doi.org/10.1109/INDIN.2007.4384865.

[GJ03] Eva Gustafsson and Annika Jonsson. "Always best connected." In: *IEEE Wireless Communications* 10 (1 2003), pp. 49–55. URL: https://doi.org/10.1109/MWC.2003.1182111.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W.H. Freeman & Co., 1979.

[GLH10] Vehbi C. Gungor, Bin Lu, and Gerhard P. Hancke. "Opportunities and Challenges of Wireless Sensor Networks in Smart Grid." In: *IEEE Transactions on Industrial Electronics* 57 (10 2010), pp. 3557–3564. URL: https://doi.org/10.1109/TIE.2009.2039455.

[Gra+15] Raúl Gracia-Tinedo, Yongchao Tian, Josep Sampé, Hamza Harkous, John Lenton, Pedro García-López, Marc Sánchez-Artigas, and Marko Vukolic. "Dissecting UbuntuOne: Autopsy of a Global-scale Personal Cloud Back-end." In: *2015 Internet Measurement Conference (IMC '15)*. ACM, 2015,

pp. 155–168. URL: https://doi.org/10.1145/2815675.2815677.

[Gub+13] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. "Internet of Things (IoT): A vision, architectural elements, and future directions." In: *Future Generation Computer Systems* 29 (7 2013), pp. 1645–1660. URL: https://doi.org/10.1016/j.future.2013.01.010.

[Gup+17] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K. Ghosh, and Rajkumar Buyya. "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog Computing Environments." In: *Software: Practice and Experience* 47 (9 2017), pp. 1275–1296. URL: https://doi.org/10.1002/spe.2509.

[Hig+12] Brett D. Higgins, Jason Flinn, T. J. Giuli, Brian Noble, Christopher Peplin, and David Watson. "Informed Mobile Prefetching." In: *10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*. ACM, 2012, pp. 155–168. URL: http://doi.acm.org/10.1145/2307636.2307651.

[HKR13] Nikolas Roman Herbst, Samuel Kounev, and Ralf H. Reussner. "Elasticity in Cloud Computing: What It Is, and What It Is Not." In: *10th International Conference on Autonomic Computing (ICAC'13)*. USENIX Association, 2013, pp. 23–27. URL: https://www.usenix.org/conference/icac13/technical-sessions/presentation/herbst.

[HN16] Richard Hull and Hamid R. Motahari Nezhad. "Rethinking BPM in a Cognitive World: Transforming How We Learn and Perform Business Processes." In: *14th International Conference on Business Process Management (BPM 2016)*. Vol. 9850. Lecture Notes in Computer Science. Springer, 2016, pp. 3–19. URL: https://doi.org/10.1007/978-3-319-45348-4_1.

[Hoc+15] Christoph Hochreiner, Stefan Schulte, Schahram Dustdar, and Freddy Lécué. "Elastic Stream Processing for Distributed Environments." In: *IEEE Internet Computing* 19 (6 2015). Invited Paper, pp. 54–59. URL: http://dx.doi.org/10.1109/MIC.2015.118.

[Hoc+16a] Christoph Hochreiner, Michael Vögler, Stefan Schulte, and Schahram Dustdar. "Elastic Stream Processing for the Internet of Things." In: *IEEE 9th International Conference on Cloud Computing (CLOUD 2016)*. IEEE, 2016, pp. 100–107. URL: http://dx.doi.org/10.1109/CLOUD.2016.21.

[Hoc+16b] Christoph Hochreiner, Michael Vögler, Philipp Waibel, and Schahram Dustdar. "VISP: An Ecosystem for Elastic Data Stream Processing for the Internet of Things." In: *20th International Enterprise Distributed Object Computing Conference (EDOC 2016)*. IEEE, 2016, pp. 19–29. URL: https://doi.org/10.1109/EDOC.2016.7579390.

[Hoc+17] Christoph Hochreiner, Michael Vögler, Stefan Schulte, and Schahram Dustdar. "Cost-Efficient Enactment of Stream Processing Topologies." In: *PeerJ Computer Science* NN (NN 2017), NN–NN. URL: https://doi.org/10.7717/peerj-cs.141.

[Hoe+13] Philipp Hoenisch, Stefan Schulte, Schahram Dustdar, and Srikumar Venugopal. "Self-Adaptive Resource Allocation for Elastic Process Execution." In: *IEEE 6th International Conference on Cloud Computing (CLOUD 2013)*. IEEE, 2013, pp. 220–227. URL: http://dx.doi.org/10.1109/CLOUD.2013.126.

[Hoe+15a] Philipp Hoenisch, Christoph Hochreiner, Dieter Schuller, Stefan Schulte, Jan Mendling, and Schahram Dustdar. "Cost-Efficient Scheduling of Elastic Processes in Hybrid Clouds." In: *IEEE 8th International Conference on Cloud Computing (CLOUD 2015)*. IEEE, 2015, pp. 17–24. URL: http://dx.doi.org/10.1109/CLOUD.2015.13.

[Hoe+15b] Philipp Hoenisch, Ingo Weber, Stefan Schulte, Liming Zhu, and Alan Fekete. "Four-fold Auto-Scaling on a Contemporary Deployment Platform using Docker Containers." In: *13th International Conference on Service Oriented Computing (ICSOC 2015)*. Vol. 9435. Lecture Notes in Computer Science. Springer, 2015, pp. 316–323. URL: http://dx.doi.org/10.1007/978-3-662-48616-0_20.

[Hoe+16] Philipp Hoenisch, Dieter Schuller, Stefan Schulte, Christoph Hochreiner, and Schahram Dustdar. "Optimization of Complex Elastic Processes." In: *IEEE Transactions on Services Computing* 9.5 (2016), pp. 700–713. URL: http://dx.doi.org/10.1109/TSC.2015.2428246.

[HSD13] Philipp Hoenisch, Stefan Schulte, and Schahram Dustdar. "Workflow Scheduling and Resource Allocation for Cloud-based Execution of Elastic Processes." In: *6th IEEE International Conference on Service Oriented Computing and Applications (SOCA 2013)*. IEEE, 2013, pp. 1–8. URL: http://www.dx.doi.org/10.1109/SOCA.2013.44.

[Hum+14] Waldemar Hummer, Stefan Schulte, Philipp Hoenisch, and Schahram Dustdar. "Context-Aware Data Prefetching in Mobile Service Environments." In: *IEEE Fourth International Conference on Big Data and Cloud Computing*

(BDCloud 2014). IEEE, 2014, pp. 214–221. URL: http://dx.doi.org/10.1109/BDCloud.2014.104.

[HX15] Wu He and Lida Xu. "A state-of-the-art survey of cloud manufacturing." In: *International Journal of Computer Integrated Manufacturing* 28 (3 2015), pp. 239–250. URL: http://dx.doi.org/10.1080/0951192X.2013.874595.

[Into8] International Telecommunication Union. *Vocabulary for performance and quality of service. Amendment 2: New definitions for inclusion in Recommendation ITU-T P.10/G.100*. ITU-T: Telecommunication Standardization Sector of ITU, 2008.

[Isl+12] Sadeka Islam, Jacky Keung, Kevin Lee, and Anna Liu. "Empirical prediction models for adaptive resource provisioning in the cloud." In: *Future Generation Computer Systems* 28 (1 2012), pp. 155–162. URL: https://doi.org/10.1016/j.future.2011.05.027.

[Juh+11] Ernst Juhnke, Tim Dörnemann, David Bock, and Bernd Freisleben. "Multi-objective Scheduling of BPEL Workflows in Geographically Distributed Clouds." In: *IEEE 4th International Conference on Cloud Computing (CLOUD 2011)*. IEEE, 2011, pp. 412–419. URL: https://doi.org/10.1109/CLOUD.2011.24.

[KAV02] Akhil Kumar, Wil M. P. Van Der Aalst, and Erik M. W. Verbeek. "Dynamic Work Distribution in Workflow Management Systems: How to Balance Quality and Performance." In: *Journal of Management Information Systems* 18 (3 2002), pp. 157–193.

[KC03] Jeffrey O. Kephart and David M. Chess. "The Vision of Autonomic Computing." In: *IEEE Computer* 36 (1 2003), pp. 41–50. URL: https://doi.org/10.1109/MC.2003.1160055.

[KCB15] Farzad Khodadadi, Rodrigo N. Calheiros, and Rajkumar Buyya. "A Data-Centric Framework for Development and Deployment of Internet of Things Applications in Clouds." In: *Tenth IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2015)*. IEEE, 2015, pp. 1–6. URL: http://dx.doi.org/10.1109/ISSNIP.2015.7106952.

[KDB16] Farzad Khodadadi, Amir Vahid Dastjerdi, and Rajkumar Buyya. "Internet of Things: An Overview." In: *Internet of Things: Principles and Paradigms*. Ed. by Rajkumar Buyya and Amir Vahid Dastjerdi. Cambridge, MA, USA: Morgan Kaufmann, 2016. Chap. 1, pp. 3–27.

[KGT06] Andreas Knöpfel, Bernhard Gröne, and Peter Tabeling. *Fundamental Modeling Concepts: Effective Communication of IT Systems*. Hoboken, NJ, USA: Wiley, 2006.

[KKP08] Meriem Kassar, Brigitte Kervella, and Guy Pujolle. "An overview of vertical handover decision strategies in heterogeneous wireless networks." In: *Computer Communications* 31 (10 2008), pp. 2607–2620. URL: https://doi.org/10.1016/j.comcom.2008.01.044.

[KL03] Alexander Keller and Heiko Ludwig. "The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services." In: *Journal of Network and Systems Management* 11 (1 2003), pp. 57–81. URL: https://doi.org/10.1023/A:1022445108617.

[Kol+14] Sefki Kolozali, Maria Bermúdez-Edo, Daniel Puschmann, Frieder Ganz, and Payam M. Barnaghi. "A Knowledge-Based Approach for Real-Time IoT Data Stream Annotation and Processing." In: *2014 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, and IEEE Cyber, Physical and Social Computing (iThings/GreenCom/CPSCom 2014)*. IEEE, 2014, pp. 215–222. URL: https://doi.org/10.1109/iThings.2014.39.

[KT98] Gerhard Keller and Thomas Teufel. *SAP R/3 Process Oriented Implementation: Iterative Process Prototyping*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.

[Kur+11] Tobias Kurze, Markus Klems, David Bermbach, Alexander Lenk, Stefan Tai, and Marcel Kunze. "Cloud Federation." In: *The Second International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING 2011)*. IARIA, 2011, pp. 32–38.

[Lai+12] Yuanjun Laili, Fei Tao, Lin Zhang, and Bhaba R. Sarker. "A study of optimal allocation of computing resources in cloud manufacturing systems." In: *International Journal of Advanced Manufacturing Technology* 63.5–8 (2012), pp. 671–690. URL: https://doi.org/10.1007/s00170-012-3939-0.

[LDB16] Angel Lagares Lemos, Florian Daniel, and Boualem Benatallah. "Web Service Composition: A Survey of Techniques and Tools." In: *ACM Computing Surveys* 48 (3 2016). URL: https://doi.org/10.1145/2831270.

[LEB15] Sebastian Lehrig, Hendrik Eikerling, and Steffen Becker. "Scalability, Elasticity, and Efficiency in Cloud Computing: a Systematic Literature Review of Definitions and Metrics." In: *11th International ACM SIGSOFT Conference*

on Quality of Software Architectures (QoSA'15). ACM, 2015, pp. 83–92. URL: http://doi.acm.org/10.1145/2737182.2737185.

[Lei+12] Philipp Leitner, Waldemar Hummer, Benjamin Satzger, Christian Inzinger, and Schahram Dustdar. "Cost-Efficient and Application SLA-Aware Client Side Request Scheduling in an Infrastructure-as-a-Service Cloud." In: *IEEE 5th International Conference on Cloud Computing (CLOUD 2012)*. IEEE, 2012, pp. 213–220. URL: https://doi.org/10.1109/CLOUD.2012.21.

[Len+11] Alexander Lenk, Michael Menzel, Johannes Lipsky, Stefan Tai, and Philipp Offermann. "What Are You Paying For? Performance Benchmarking for Infrastructure-as-a-Service Offerings." In: *IEEE 4th IEEE International Conference on Cloud Computing (CLOUD 2011)*. IEEE, 2011, pp. 484–491. URL: https://doi.org/10.1109/CLOUD.2011.80.

[LHD13] Philipp Leitner, Waldemar Hummer, and Schahram Dustdar. "Cost-Based Optimization of Service Compositions." In: *IEEE Transactions on Services Computing* 6.2 (2013), pp. 239–251. URL: https://doi.org/10.1109/TSC.2011.53.

[Li+08] Wubin Li, Zhuofeng Zhao, Kaiyuan Qi, Jun Fang, and Weilong Ding. "A Consistency-Preserving Mechanism for Web Services Response Caching." In: *6th IEEE International Conference on Web Services (ICWS 2008)*. IEEE, 2008, pp. 683–690. URL: https://doi.org/10.1109/ICWS.2008.60.

[Li+13] Fei Li, Michael Vögler, Markus Claessens, and Schahram Dustdar. "Efficient and Scalable IoT Service Delivery on Cloud." In: *IEEE 6th International Conference on Cloud Computing (CLOUD 2013)*. IEEE, 2013, pp. 740–747. URL: http://dx.doi.org/10.1109/CLOUD.2013.64.

[LL02] Wai Yip Lum and Francis C.M. Lau. "A Context-Aware Decision Engine for Content Adaptation." In: *IEEE Pervasive Computing* 1 (3 2002), pp. 41–49. URL: https://doi.org/10.1109/MPRV.2002.1037721.

[LL15] In Lee and Kyoochun Lee. "The Internet of Things (IoT): Applications, investments, and challenges for enterprises." In: *Business Horizons* 58 (1 2015), pp. 431–440. URL: https://doi.org/10.1016/j.bushor.2015.03.008.

[LPT05] Kwong Yuen Lai, Thi Khoi Anh Phan, and Zahir Tari. "Efficient SOAP Binding for Mobile Web Services." In: *30th IEEE Conference on Local Computer Networks (LCN*

2005). IEEE, 2005. URL: https://doi.org/10.1109/LCN.2005.62.

[LR99] Frank Leymann and Dieter Roller. *Production Workflow: Concepts and Techniques*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.

[LZV08] Phillip A. Laplante, Jia Zhang, and Jeffrey Voas. "What's in a Name? Distinguishing between SaaS and SOA." In: *IT Professional* 10 (3 2008). URL: https://doi.org/10.1109/MITP.2008.60.

[Mas+14] Toni Mastelic, Ariel Oleksiak, Holger Claussen, Ivona Brandic, Jean-Marc Pierson, and Athanasios V. Vasilakos. "Cloud Computing: Survey on Energy Efficiency." In: *ACM Computing Surveys* 47.2 (2014), 33:1–33:36. URL: http://dx.doi.org/10.1145/2656204.

[MBS13] Michael Maurer, Ivona Brandic, and Rizos Sakellariou. "Adaptive resource configuration for Cloud infrastructure management." In: *Future Generation Computer Systems* 29.2 (2013), pp. 472–487. URL: http://dx.doi.org/10.1016/j.future.2012.07.004.

[Men+17] Jan Mendling et al. *Blockchains for Business Process Management – Challenges and Opportunities*. arXiv report 1704.03610. arXiv, 2017. URL: https://arxiv.org/abs/1704.03610.

[MEW00] Anirban Mahanti, Derek Eager, and Carey Williamson. "Temporal locality and its impact on Web proxy cache performance." In: *Performance Evaluation* 42.2 (2000), pp. 187–203. URL: https://doi.org/10.1016/S0166-5316(00)00032-8.

[MG11] Peter Mell and Timothy Grance. *The NIST Definition of Cloud Computing*. Special Publication 800-145. Recommendations of the National Institute of Standards and Technology, 2011. URL: http://dx.doi.org/10.6028/NIST.SP.800-145.

[Mid+16] Peter Middleton, James F. Hines, Bettina Tratz-Ryan, Eric Goodness, Dean Freeman, Masatsune Yamaji, Angela McIntyre, Anurag Gupta, Denise Rueb, and Tracy Tsai. *Forecast: Internet of Things – Endpoints and Associated Services, Worldwide, 2016*. Gartner, Inc., 2016. URL: https://www.gartner.com/doc/3558917/forecast-internet-things--endpoints.

[Mie+10] André Miede, Ulrich Lampe, Dieter Schuller, Julian Eckert, and Ralf Steinmetz. "Evaluating the QoS Impact of Web Service Anonymity." In: *IEEE 8th European Conference on Web Services (ECOWS'10)*. IEEE, 2010, pp. 75–82. URL: https://doi.org/10.1109/ECOWS.2010.8.

[Mio+12] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. "Internet of things: Vision, applications and research challenges." In: *Ad Hoc Networks* 10 (7 2012), pp. 1497–1516. URL: https://doi.org/10.1016/j.adhoc.2012.02.016.

[Mol+13] Daniel Moldovan, Georgiana Copil, Hong-Linh Truong, and Schahram Dustdar. "MELA: Monitoring and Analyzing Elasticity of Cloud Services." In: *IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom 2013)*. IEEE, 2013, pp. 80–87. URL: http://dx.doi.org/10.1109/CloudCom.2013.18.

[Mon14] László Monostori. "Cyber-physical Production Systems: Roots, Expectations and R&D Challenges." In: *Procedia CIRP* 17 (2014), pp. 9–13. URL: https://doi.org/10.1016/j.procir.2014.03.115.

[MRM13] Sonja Meyer, Andreas Ruppen, and Carsten Magerkurth. "Internet of Things-Aware Process Modeling: Integrating IoT Devices as Business Process Resources." In: *25th International Conference on Advanced Information Systems Engineering (CAiSE 2013)*. Vol. 7908. Lecture Notes in Computer Science. Springer, 2013, pp. 84–98. URL: https://dx.doi.org/10.1007/978-3-642-38709-8_6.

[Nar+17] Matteo Nardelli, Stefan Nastic, Schahram Dustdar, Massimo Villari, and Rajiv Ranjan. "Osmotic Flow: Osmotic Computing + IoT Workflow." In: *IEEE Cloud Computing* 4 (2 2017), pp. 68–75. URL: https://doi.org/10.1109/MCC.2017.22.

[Ng+15] Irene Ng, Kimberley Scharf, Ganna Pogrebna, and Roger Maull. "Contextual variety, Internet-of-Things and the choice of tailoring over platform: Mass customisation strategy in supply chain management." In: *International Journal of Production Economics* 159 (2015), pp. 76–87. URL: https://doi.org/10.1016/j.ijpe.2014.09.007.

[Oli+05] Adam J. Oliner, Larry Rudolph, Ramendra K. Sahoo, José E. Moreira, and Manish Gupta. "Probabilistic QoS Guarantees for Supercomputing Systems." In: *2005 International Conference on Dependable Systems and Networks (DSN 2005)*. IEEE, 2005, pp. 634–643. URL: https://doi.org/10.1109/DSN.2005.80.

[Pah+18] Claus Pahl, Antonio Brogi, Jacopo Soldani, and Pooyan Jamshidi. "Cloud Container Technologies: a State-of-the-Art Review." In: *IEEE Transactions on Cloud Computing* NN.NN (2018), NN. URL: https://doi.org/10.1109/TCC.2017.2702586.

[Pan+10] Suraj Pandey, Linlin Wu, Siddeswara Mayura Guru, and Rajkumar Buyya. "A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments." In: *24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010)*. IEEE, 2010, pp. 400–407. URL: https://doi.org/10.1109/AINA.2010.31.

[Pap+07] Mike P. Papazoglou, Paolo Traverso, Schahram Dustdar, and Frank Leymann. "Service-Oriented Computing: State of the Art and Research Challenges." In: *IEEE Computer* 40.11 (2007), pp. 38–45. URL: http://dx.doi.org/10.1109/MC.2007.400.

[Pap+10] Apostolos Papageorgiou, Jeremias Blendin, André Miede, Julian Eckert, and Ralf Steinmetz. "Study and Comparison of Adaptation Mechanisms for Performance Enhancements of Mobile Web Service Consumption." In: *6th World Congress on Services (SERVICES 2010)*. IEEE, 2010, pp. 667–670. URL: http://dx.doi.org/10.1109/SERVICES.2010.52.

[Pap+11a] Apostolos Papageorgiou, André Miede, Dieter Schuller, Stefan Schulte, and Ralf Steinmetz. "Always Best Served: On the Behaviour of QoS- and QoE-based Algorithms for Web Service Adaptation." In: *PERCOM Workshops – 8th International Workshop on Managing Ubiquitous Communications and Services (MUCS 2011)*. IEEE Computer Society, Washington, DC, USA, 2011, pp. 76–81. URL: http://dx.doi.org/10.1109/PERCOMW.2011.5766975.

[Pap+11b] Apostolos Papageorgiou, Marius Schatke, Stefan Schulte, and Ralf Steinmetz. "Enhancing the Caching of Web Service Responses on Wireless Clients." In: *9th IEEE International Conference on Web Services (ICWS 2011)*. IEEE, 2011, pp. 9–16. URL: http://dx.doi.org/10.1109/ICWS.2011.52.

[Pap+12] Apostolos Papageorgiou, Marius Schatke, Stefan Schulte, and Ralf Steinmetz. "Lightweight Wireless Web Service Communication Through Enhanced Caching Mechanisms." In: *International Journal of Web Services Research* 9.2 (2012), pp. 42–68. URL: http://dx.doi.org/10.4018/jwsr.2012040103.

[Pap+14] Apostolos Papageorgiou, André Miede, Stefan Schulte, Dieter Schuller, and Ralf Steinmetz. "Decision Support for Web Service Adaptation." In: *Pervasive and Mobile Computing* 12 (2014), pp. 197–213. URL: http://dx.doi.org/10.1016/j.pmcj.2013.10.004.

[PB03] Stefan Podlipnig and László Böszörményi. "A Survey of Web Cache Replacement Strategies." In: *ACM Computing Surveys* 35.4 (2003), pp. 374–398. URL: http://doi.acm.org/10.1145/954339.954341.

[PBA12] Thanasis G. Papaioannou, Nicolas Bonvin, and Karl Aberer. "Scalia: An Adaptive Scheme for Efficient Multi-cloud Storage." In: *International Conference on High Performance Computing, Networking, Storage and Analysis (SC '12)*. IEEE/ACM, 2012, 20:1–20:10. URL: https://doi.org/10.1109/SC.2012.101.

[Pea13] Siani Pearson. "Privacy, Security and Trust in Cloud Computing." In: *Privacy and Security for Cloud Computing*. Ed. by Siani Pearson and George Yee. Computer Communications and Networks. London, UK: Springer, 2013. Chap. 1, pp. 3–42. URL: https://doi.org/10.1007/978-1-4471-4189-1_1.

[Per+14a] Charith Perera, Prem Prakash Jayaraman, Arkady B. Zaslavsky, Dimitrios Georgakopoulos, and Peter Christen. "MOSDEN: An Internet of Things Middleware for Resource Constrained Mobile Devices." In: *47th Hawaii International Conference on System Science (HICSS'14)*. IEEE, 2014, pp. 1053–1062. URL: https://doi.org/10.1109/HICSS.2014.137.

[Per+14b] Charith Perera, Arkady B. Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. "Context Aware Computing for The Internet of Things: A Survey." In: *IEEE Communications Surveys and Tutorials* 16.1 (2014), pp. 414–454. URL: https://doi.org/10.1109/SURV.2013.042313.00197.

[PG03] Mike P. Papazoglou and Dimitrios Georgakopoulos. "Service-oriented Computing." In: *Communications of the ACM* 46 (10 2003), pp. 24–28. URL: http://doi.acm.org/10.1145/944217.944233.

[PRS09] Pankesh Patel, Ajith H. Ranabahu, and Amit P. Sheth. *Service Level Agreement in Cloud Computing*. Tech. rep. 78. Kno.e.sis Publications, 2009. URL: http://corescholar.libraries.wright.edu/knoesis/78.

[Pry+17] Christoph Prybila, Stefan Schulte, Christoph Hochreiner, and Ingo Weber. "Runtime Verification for Business Processes Utilizing the Bitcoin Blockchain." In: *Future Generation Computer Systems* NN.NN (2017), NN–NN. URL: https://doi.org/10.1016/j.future.2017.08.024.

[Pui+16] Dan Puiu et al. "CityPulse: Large Scale Data Analytics Framework for Smart Cities." In: *IEEE Access* 4 (2016), pp. 1086–1108. URL: https://doi.org/10.1109/ACCESS.2016.2541999.

[PZL08] Cesare Pautasso, Olaf Zimmermann, and Frank Leymann. "RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision." In: *17th International Conference on World Wide Web (WWW 2008)*. ACM, 2008, pp. 805–814. URL: http://doi.acm.org/10.1145/1367497.1367606.

[Raj+10] Ragunathan Rajkumar, Insup Lee, Lui Sha, and John Stankovic. "Cyber-physical Systems: The Next Computing Revolution." In: *47th Design Automation Conference (DAC '10)*. ACM, 2010, pp. 731–736. URL: http://doi.acm.org/10.1145/1837274.1837461.

[Ren+15] Lei Ren, Lin Zhang, Fei Tao, Chun Zhao, Xudong Chai, and Xinpei Zhao. "Cloud manufacturing: from concept to practice." In: *Enterprise IS* 9.2 (2015), pp. 186–209. URL: http://dx.doi.org/10.1080/17517575.2013.839055.

[Rod+10] Luis Rodero-Merino, Luis Miguel Vaquero Gonzalez, Victor Gil, Fermín Galán, Javier Fontán, Rubén S. Montero, and Ignacio Martín Llorente. "From infrastructure delivery to service management in clouds." In: *Future Generation Computer Systems* 26.8 (2010), pp. 1226–1240. URL: http://dx.doi.org/10.1016/j.future.2010.02.013.

[Rög+17] Maximilian Röglinger, Johannes Seyfried, Simon Stelzl, and Michael zur Muehlen. "Cognitive Computing: What's in for Business Process Management? An Exploration of Use Case Ideas." In: *1st Workshop on Cognitive Business Process Management (CBPM) at the 15th International Conference on Business Process Management (BPM 2017)*. 2017, NN–NN.

[Ros+09] Sidney Rosario, Albert Benveniste, Stefan Haar, and Claude Jard. "Probabilistic QoS and Soft Contracts for Transaction-Based Web Services Orchestrations." In: *IEEE Transactions on Services Computing* 1.4 (2009). URL: https://doi.org/10.1109/TSC.2008.17.

[SC16] Sukhpal Singh and Inderveer Chana. "QoS-Aware Autonomic Resource Management in Cloud Computing: A Systematic Review." In: *ACM Computing Surveys* 48.3 (2016), 42:1–42:46. URL: http://doi.acm.org/10.1145/2843889.

[Sch+11a] Daniel Schreiber, Andreas Göb, Erwin Aitenbichler, and Max Mühlhäuser. "Reducing User Perceived Latency with a Proactive Prefetching Middleware for Mobile SOA Ac-

cess." In: *International Journal of Web Services Research* 8.1 (2011), pp. 68–85. URL: https://doi.org/10.4018/jwsr.2011010104.

[Sch+11b] Dieter Schuller, Artem Polyvyanyy, Luciano García-Bañuelos, and Stefan Schulte. "Optimization of Complex QoS-aware Service Compositions." In: *9th International Conference on Service Oriented Computing (ICSOC 2011)*. Vol. 7084. Lecture Notes in Computer Science. Springer, 2011, pp. 452–466. URL: http://dx.doi.org/10.1007/978-3-642-25535-9.

[Sch+12a] Dieter Schuller, Ulrich Lampe, Julian Eckert, Ralf Steinmetz, and Stefan Schulte. "Cost-driven Optimization of Complex Service-based Workflows for Stochastic QoS Parameters." In: *10th IEEE International Conference on Web Services (ICWS 2012)*. IEEE, 2012, pp. 66–73. URL: http://dx.doi.org/10.1109/ICWS.2012.50.

[Sch+12b] Stefan Schulte, Dieter Schuller, Ralf Steinmetz, and Sven Abels. "Plug-and-Play Virtual Factories." In: *IEEE Internet Computing* 16 (5 2012). Invited Paper, pp. 78–82. URL: http://dx.doi.org/10.1109/MIC.2012.114.

[Sch+13a] Dieter Schuller, Ulrich Lampe, Julian Eckert, Ralf Steinmetz, and Stefan Schulte. "Optimizing Complex Service-based Workflows for Stochastic QoS Parameters." In: *International Journal of Web Services Research* 10.4 (2013), pp. 1–38. URL: http://dx.doi.org/10.4018/ijwsr.2013100101.

[Sch+13b] Stefan Schulte, Philipp Hoenisch, Srikumar Venugopal, and Schahram Dustdar. "Introducing the Vienna Platform for Elastic Processes." In: *Performance Assessment and Auditing in Service Computing Workshop (PAASC 2012) at 10th International Conference on Service Oriented Computing (ICSOC 2012)*. Vol. 7759. Lecture Notes in Computer Science. Springer, 2013, pp. 179–190. URL: http://dx.doi.org/10.1007/978-3-642-37804-1_19.

[Sch+13c] Stefan Schulte, Philipp Hoenisch, Srikumar Venugopal, and Schahram Dustdar. "Realizing Elastic Processes with ViePEP." In: *10th International Conference on Service Oriented Computing (ICSOC 2012) – Demos*. Vol. 7759. Lecture Notes in Computer Science. Springer, 2013, pp. 439–443. URL: https://doi.org/10.1007/978-3-642-37804-1_48.

[Sch+13d] Stefan Schulte, Dieter Schuller, Philipp Hoenisch, Ulrich Lampe, Schahram Dustdar, and Ralf Steinmetz. "Cost-Driven Optimization of Cloud Resource Allocation for Elastic Processes." In: *International Journal of Cloud Com-*

puting 1.2 (2013), pp. 1–14. URL: http://hipore.com/ijcc/2013/IJCC-Vol1-No2-2013-pp1-14-Schulte.pdf.

[Sch+14a] Dieter Schuller, Melanie Siebenhaar, Ronny Hans, Olga Wenge, Ralf Steinmetz, and Stefan Schulte. "Towards Heuristic Optimization of Complex Service-based Workflows for Stochastic QoS Attributes." In: *12th IEEE International Conference on Web Services (ICWS 2014)*. IEEE, 2014, pp. 361–368. URL: http://www.dx.doi.org/10.1109/ICWS.2014.59.

[Sch+14b] Stefan Schulte, Philipp Hoenisch, Christoph Hochreiner, Schahram Dustdar, Matthias Klusch, and Dieter Schuller. "Towards Process Support for Cloud Manufacturing." In: *18th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2014)*. IEEE, 2014, pp. 142–149. URL: http://dx.doi.org/10.1109/EDOC.2014.28.

[Sch+15] Stefan Schulte, Christian Janiesch, Srikumar Venugopal, Ingo Weber, and Philipp Hoenisch. "Elastic Business Process Management: State of the Art and Open Challenges for BPM in the Cloud." In: *Future Generation Computer Systems* 46 (2015), pp. 36–50. URL: http://dx.doi.org/10.1016/j.future.2014.09.005.

[Sch+16] Stefan Schulte, Michael Borkowski, Christoph Hochreiner, Matthias Klusch, Aitor Murguzur, Olena Skarlat, and Philipp Waibel. "Bringing Cloud-based Rapid Elastic Manufacturing to Reality with CREMA." In: *Workshop on Intelligent Systems Configuration Services for Flexible Dynamic Global Production Networks (FLEXINET) at the 8th International Conference on Interoperability for Enterprise Systems and Applications (I-ESA 2016)*. ISTE Ltd., 2016, pp. 407–413.

[SÇZ05] Michael Stonebraker, Ugur Çetintemel, and Stanley B. Zdonik. "The 8 Requirements of Real-Time Stream Processing." In: *SIGMOD Record* 34.4 (2005), pp. 42–47. URL: http://doi.acm.org/10.1145/1107499.1107504.

[SDQ10] Jörg Schad, Jens Dittrich, and Jorge-Arnulfo Quiané-Ruiz. "Runtime Measurements in the Cloud: Observing, Analyzing, and Reducing Variance." In: *Proceedings of the VLDB Endowment* 3.1 (2010), pp. 460–471. URL: https://doi.org/10.14778/1920841.1920902.

[She+14] Quan Z. Sheng, Xiaoqiang Qiao, Athanasios V. Vasilakos, Claudia Szabo, Scott Bourne, and Xiaofei Xu. "Web services composition: A decade's overview." In: *Information Sciences* 280 (2014), pp. 218–238. URL: https://doi.org/10.1016/j.ins.2014.04.054.

[Shi+16] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. "Edge Computing: Vision and Challenges." In: *IEEE Internet of Things Journal* 3 (5 2016), pp. 637–646. URL: https://doi.org/10.1109/JIOT.2016.2579198.

[Sim+11] Yogesh Simmhan, Baohua Cao, Michail Giakkoupis, and Viktor K. Prasanna. "Adaptive Rate Stream Processing for Smart Grid Applications on Clouds." In: *2nd International Workshop on Scientific Cloud Computing (ScienceCloud '11)*. ACM, 2011, pp. 33–38. URL: http://doi.acm.org/10.1145/1996109.1996116.

[SK13] Vasilios A. Siris and Dimitrios Kalyvas. "Enhancing mobile data offloading with mobility prediction and prefetching." In: *ACM SIGMOBILE Mobile Computing and Communications Review* 17 (1 2013), pp. 22–29. URL: https://doi.org/10.1145/2502935.2502940.

[Ska+16] Olena Skarlat, Stefan Schulte, Michael Borkowski, and Philipp Leitner. "Resource Provisioning for IoT Services in the Fog." In: *9th IEEE International Conference on Service Oriented Computing and Applications (SOCA 2016)*. IEEE, 2016, pp. 32–39. URL: http://dx.doi.org/10.1109/SOCA.2016.10.

[Ska+17a] Olena Skarlat, Matteo Nardelli, Stefan Schulte, Michael Borkowski, and Philipp Leitner. "Optimized IoT Service Placement in the Fog." In: *Service Oriented Computing and Applications* 11.4 (2017), pp. 427–443. URL: https://dx.doi.org/10.1007/s11761-017-0219-8.

[Ska+17b] Olena Skarlat, Matteo Nardelli, Stefan Schulte, and Schahram Dustdar. "Towards QoS-aware Fog Service Placement." In: *IEEE/ACM 1st International Conference on Fog and Edge Computing (ICFEC 2017)*. IEEE, 2017, pp. 89–96. URL: http://dx.doi.org/10.1109/ICFEC.2017.12.

[STD08] Daniel Schall, Hong-Linh Truong, and Schahram Dustdar. "Unifying Human and Software Services in Web-Scale Collaborations." In: *IEEE Internet Computing* 12 (3 2008), pp. 62–68. URL: https://doi.org/10.1109/MIC.2008.66.

[Str10] Anja Strunk. "QoS-Aware Service Composition: A Survey." In: *IEEE 8th European Conference on Web Services (ECOWS'10)*. IEEE, 2010, pp. 67–74. URL: https://doi.org/10.1109/ECOWS.2010.16.

[SW14] Ivan Stojmenovic and Sheng Wen. "The Fog Computing Paradigm: Scenarios and Security Issues." In: *2014 Federated Conference on Computer Science and Information Sys-*

tems (FedCSIS). IEEE, 2014, pp. 1–8. URL: https://doi.org/10.15439/2014F503.

[Tao+13] Fei Tao, Yuanjun Laili, Lida Xu, and Lin Zhang. "FC-PACO-RM: A Parallel Method for Service Composition Optimal-Selection in Cloud Manufacturing System." In: *IEEE Transactions on Industrial Informatics* 9.4 (2013), pp. 2023–2033. URL: http://dx.doi.org/10.1109/TII.2012.2232936.

[TCB14] Adel Nadjaran Toosi, Rodrigo N. Calheiros, and Rajkumar Buyya. "Interconnected Cloud Computing Environments: Challenges, Taxonomy, and Survey." In: *ACM Computing Surveys* 47 (1 2014), 7:1–7:47. URL: https://doi.org/10.1145/2593512.

[TKM04] Stefan Tai, Rania Khalaf, and Thomas A. Mikalsen. "Composition of Coordinated Web Services." In: *ACM/IFIP/USENIX International Middleware Conference (Middleware 2004)*. Vol. 3231. Lecture Notes in Computer Science. Springer, 2004, pp. 294–310. URL: https://doi.org/10.1007/978-3-540-30229-2_16.

[TR03] Douglas B. Terry and Venugopalan Ramasubramanian. "Caching XML Web Services for Mobility." In: *ACM Queue – Wireless* 1 (3 2003), pp. 70–78. URL: https://doi.org/10.1145/846057.864024.

[TST12] Marco Taisch, Bojan Stahl, and Giacomo Tavola. "ICT in Manufacturing: Trends and Challenges for 2020 – An European View." In: *10th IEEE International Conference on Industrial Informatics (INDIN 2012)*. IEEE, 2012, pp. 941–946. URL: https://doi.org/10.1109/INDIN.2012.6301312.

[Vil+16] Massimo Villari, Maria Fazio, Schahram Dustdar, Omer F. Rana, and Rajiv Ranjan. "Osmotic Computing: A New Paradigm for Edge/Cloud Integration." In: *IEEE Cloud Computing* 3 (6 2016), pp. 76–83. URL: https://doi.org/10.1109/MCC.2016.124.

[Vög+17] Michael Vögler, Johannes M. Schleicher, Christian Inzinger, and Schahram Dustdar. "Optimizing Elastic IoT Application Deployments." In: *IEEE Transactions on Services Computing* NN (2017), NN–NN. URL: https://doi.org/10.1109/TSC.2016.2617327.

[VR88] Sandra Vandermerwe and Juan Rada. "Servitization of business: Adding value by adding services." In: *European Management Journal* 6 (4 1988), pp. 314–324. URL: https://doi.org/10.1016/0263-2373(88)90033-3.

[VVB13] Ruben Van den Bossche, Kurt Vanmechelen, and Jan Broeckhove. "Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds." In: *Future Generation Computer Systems* 29.4 (2013), pp. 973–985. URL: https://doi.org/10.1016/j.future.2012.12.012.

[VVK09] Jussi Vanhatalo, Hagen Völzer, and Jana Koehler. "The refined process structure tree." In: *Data & Knowledge Engineering* 68 (9 2009), pp. 793–818. URL: https://doi.org/10.1016/j.datak.2009.02.015.

[Wai+17] Philipp Waibel, Johannes Matt, Christoph Hochreiner, Olena Skarlat, Ronny Hans, and Stefan Schulte. "Cost-Optimized Redundant Data Storage in the Cloud." In: *Service Oriented Computing and Applications* 11.4 (2017), pp. 411–426. URL: https://dx.doi.org/10.1007/s11761-017-0218-9.

[Wan99] Jia Wang. "A Survey of Web Caching Schemes for the Internet." In: *ACM SIGCOMM Computer Communication Review* 29 (5 1999), pp. 36–46. URL: https://doi.org/10.1145/505696.505701.

[Wee+05] Sanjiva Weerawarana, Francisco Curbera, Frank Leymann, Tony Storey, and Donald F. Ferguson. *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2005.

[Wes12] Mathias Weske. *Business Process Management: Concepts, Languages, Architectures*. 2nd. Berlin Heidelberg, Germany: Springer, 2012. URL: https://doi.org/10.1007/978-3-642-28616-2.

[WGB11] Linlin Wu, Saurabh Kumar Garg, and Rajkumar Buyya. "SLA-Based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments." In: *11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2011)*. IEEE, 2011, pp. 195–204. URL: https://doi.org/10.1109/CCGrid.2011.51.

[WHS16] Philipp Waibel, Christoph Hochreiner, and Stefan Schulte. "Cost-Efficient Data Redundancy in the Cloud." In: *9th IEEE International Conference on Service Oriented Computing and Applications (SOCA 2016)*. IEEE, 2016, pp. 1–9. URL: http://dx.doi.org/10.1109/SOCA.2016.12.

[WK02] Hakim Weatherspoon and John Kubiatowicz. "Erasure Coding Vs. Replication: A Quantitative Comparison." In: *Revised Papers from the First International Workshop on P2P Systems (IPTPS 2002)*. Vol. 2429. Lecture Notes in Com-

puter Science. Springer, 2002, pp. 328–338. URL: https://doi.org/10.1007/3-540-45748-8_31.

[Wu+13] Dazhong Wu, Matthew John Greer, David W. Rosen, and Dirk Schaefer. "Cloud manufacturing: Strategic vision and state-of-the-art." In: *Journal of Manufacturing Systems* 32.4 (2013), pp. 564–579. URL: https://doi.org/10.1016/j.jmsy.2013.04.008.

[Xu+15] Xiaofei Xu, Quan Z. Sheng, Liang-Jie Zhang, Yushun Fan, and Schahram Dustdar. "From Big Data to Big Service." In: *IEEE Computer* 48 (7 2015), pp. 80–83. URL: https://doi.org/10.1109/MC.2015.182.

[Xu12] Xun Xu. "From cloud computing to cloud manufacturing." In: *Robotics and Computer-Integrated Manufacturing* 28 (1 2012), pp. 75–86. URL: https://doi.org/10.1016/j.rcim.2011.07.002.

[YSG99] Yahaya Y Yusuf, M Sarhadi, and Angappa Gunasekaran. "Agile Manufacturing: The Drivers, Concepts and Attributes." In: *International Journal of Production Economics* 62 (1–2 1999), pp. 33–43. URL: https://doi.org/10.1016/S0925-5273(98)00219-9.

[Yua+10] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. "T-Drive: Driving Directions Based on Taxi Trajectories." In: *18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10)*. ACM, 2010, pp. 99–108. URL: https://doi.org/10.1145/1869790.1869807.

[Zha+15] Zhi-Hui Zhan, Xiao Fang Liu, Yue-Jiao Gong, Jun Zhang, Henry Shu-Hung Chung, and Yun Li. "Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches." In: *ACM Computing Surveys* 47.4 (2015), 63:1–63:33. URL: http://doi.acm.org/10.1145/2788397.

[Zha07] Dongsong Zhang. "Web Content Adaptation for Mobile Handheld Devices." In: *Communications of the ACM* 50.2 (2007), pp. 75–79. URL: https://doi.org/10.1145/1216016.1216024.

[zNS05] Michael zur Muehlen, Jeffrey V. Nickerson, and Keith D. Swenson. "Developing web services choreography standards—the case of REST vs. SOAP." In: *Decision Support Systems* 40 (1 2005), pp. 9–29. URL: https://doi.org/10.1016/j.dss.2004.04.008.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/