Towards Function and Data Shipping in Manufacturing Environments: How Cloud Technologies leverage the 4th Industrial Revolution

Michael Falkenthal[1], Uwe Breitenbücher[1], Maximilian Christ[2],
Christian Endres[1], Andreas W. Kempa-Liehr[2,3],
Frank Leymann[1], Michael Zimmermann[1]

[1]Institute of Architecture of Application Systems,
University of Stuttgart, Germany,
lastname@iaas.uni-stuttgart.de

[2]Blue Yonder GmbH,
Germany,
maximilian.christ@blue-yonder.com

[3]Freiburg Material Research Center,
University of Freiburg, Germany,
kempa-liehr@fmf.uni-freiburg.de

BibTeX

```
@inproceedings{Falkenthal2016,
    author    = {Falkenthal, Michael and Breitenb{\"{u}}cher, Uwe and Christ,
                 Maximilian and Endres, Christian and Kempa-Liehr, Andreas W. and
                 Leymann, Frank and Zimmermann, Michael},
    booktitle = {Proceedings of the 10\textsuperscript{th} Advanced Summer
                 School on Service Oriented Computing},
    pages     = {16--25},
    publisher = {IBM Research Division},
    title     = {{Towards Function and Data Shipping in Manufacturing
                 Environments : How Cloud Technologies leverage the 4th
                 Industrial Revolution}},
    year      = {2016}
}
```

Towards Function and Data Shipping in Manufacturing Environments: How Cloud Technologies leverage the 4$^{\text{th}}$ Industrial Revolution

Michael Falkenthal[1], Uwe Breitenbücher[1], Maximilian Christ[2],
Christian Endres[1], Andreas W. Kempa-Liehr[2,3],
Frank Leymann[1], and Michael Zimmermann[1]

[1] University of Stuttgart, Institute of Architecture of Application Systems
Universitätsstr. 38, 70569 Stuttgart, Germany
`[lastname]@iaas.uni-stuttgart.de`
[2] Blue Yonder GmbH
Ohiostr. 8, 76149 Karlsruhe, Germany
`maximilian.christ@blue-yonder.com`
[3] University of Freiburg, Freiburg Materials Research Center
Stefan-Meier-Str. 21, 79104 Freiburg, Germany
`kempa-liehr@fmf.uni-freiburg.de`

Abstract. Advances in the field of cloud computing and the Internet of Things are boosting the 4$^{\text{th}}$ industrial revolution. New research and developments foster the emergence of smart services, which augment conventional machinery to become smart cyber-physical systems. The resulting systems are characterized by providing preemptive functionality to automatically react on circumstances and changes in their physical environment. In this paper we sketch our vision of how to automatically provision smart services in manufacturing environments, whereby the paradigms of function and data shipping are specifically considered. To base this approach upon a clear understanding of influences, we point out key challenges in the context of smart services for Industry 4.0.

Keywords: data shipping, function shipping, fourth industrial revolution, cyber-physical systems, TOSCA

1 Introduction and Background

The availability of cheap sensors and the increasing connectivity between devices are the drivers behind the accelerating availability of data in industrial operations [2,20]. The evolving Internet of Things (IoT) is formed by *"embedded devices (Things) with Internet connectivity, allowing them to interact with each other, services, and people on a global scale"* [15] and is a central enabler of Industry 4.0, which heavily relies on predicting future device states by combining the knowledge of device attributes with historic and current sensor readings.

"Industry 4.0 is a collective term for technologies and concepts of value chain organization" [11] and is a synonym for the 4th industrial revolution.

An important application of Industry 4.0 is the anticipation of future device states in the context of predictive maintenance [14]. Its task is the discrimination of properly working machines from those, which are likely to evolve a specific risk of failure. Collected data might describe a fleet of machines, with each machine being characterized by certain time invariant data (e.g., geo-coordinates, date of putting into operation), time variant control parameters (e.g., the currently used tool head), collections of regularly updated sensor data (e.g., time series containing pressure and temperature measurements), and the results of some successive inspection reports, which might indicate specific technical flaws. On basis of the inspection reports, the machines can be divided into two groups: Machines for which a specific technical flaw has been observed and machines, for which this flaw has not been observed yet. The optimization task of predictive maintenance is to determine the relation of statistical health factors to operating costs and failure risks by means of machine learning algorithms [21].

This optimization task is provided by so-called *smart services*, which are not only *"reactive or even proactive"* but actually *"fundamentally preemptive"* and *"based upon hard field intelligence"* given by *"awareness and connectivity"* [1]. The vast amount of mounted sensors on machinery enables a smart service to perform detailed analyses of production steps [9] and to subsequently influence the production flow by adapting machine configurations and adjustments in an automated fashion. As a consequence, these kind of machines become smart cyber-physical systems. It is expected that this approach will contribute significantly to the expectations being associated with Industry 4.0.

In such scenarios, huge amounts of metering data are generated. In case of critical operations, this data has to be processed in parallel in order to react to the process under optimization in a timely manner [12]. Accordingly, in many applications it is insufficient or almost impossible to transfer the data to a central data store or a public cloud environment providing adequate processing power and storage for analyzing the data [10]. Further, latencies and limited network bandwidth make centralized processing unsuitable. Instead, analytics or data aggregation functionality has to be provisioned as close to the data sources as possible. Such a scenario, where functionality is shipped and provisioned close to the data sources, is called *function shipping*.

While some applications benefit from the function shipping approach, also other use cases exist that do not require strict reaction times for processing analysis. Instead, the sensor data from different sources and respective meta-information need to be merged, which raises the demand for self-documenting file formats [18]. In these scenarios, the data have to be transferred to a powerful central execution environment, either self-hosted or in a public cloud environment. This approach is called *data shipping*, because the data is transferred to the functionality that has to process the data. Summarizing, different provisioning strategies for providing and running analytics functionality seem to be feasible to cover the needs of smart service development and operation.

However, there is still a lack of proper technologies and tools in order to efficiently support the development and provisioning of smart services. Further, the concepts of function and data shipping have to be particularly applied to manufacturing environments considering the specific challenges such as handling of secret production data or time constraints for processing the data.

Thus, in this paper we introduce a standards-based vision of how to automatically provision smart services in manufacturing environments, whereby the paradigms of function and data shipping are specifically considered. To base this approach upon a clear understanding of related issues, we point out key challenges in the context of smart services for Industry 4.0. The envisioned approach is currently implemented in the course of the project SePiA.Pro[4] (*Service Platform for intelligently optimizing Applications in Production and Manufacturing*) but should be applicable to smart services in general [19].

The remainder of this paper is structured as following: We discuss the key challenges for provisioning smart services in Section 2. We sketch a self-contained and secure packaging format for smart services in Section 3 and discuss how it addresses the identified challenges and how it enables to establish function and data shipping in the context of smart services for Industry 4.0. In Section 4, we explain how to utilize the packaging format by a toolchain and conclude this paper in Section 5.

2 Key Challenges

There are several key challenges, which have to be addressed in order to develop and run smart services for, respectively, in manufacturing environments. We categorize the identified challenges into *organizational challenges* and *technical challenges*. Organizational challenges mainly arise from the connection of formerly disconnected sets of data and reflect issues concerning the collaboration of former organizationally distinct (legal) entities like different departments, organizational units, subsidiaries, or even different companies. The technical challenges, in contrast, address difficulties that are driven by different technologies, statistical aspects of the machine learning algorithms and programming principles. In SePiA.Pro, we will investigate and address both types of challenges.

2.1 Organizational Challenges

Restricted Access to Smart Services Since smart services contain analytics algorithms that process detailed metering data from production processes, critical intellectual property of a company in the form of information about actual processing steps and produced parts could be reengineered from the algorithm implementations. Thus, access to a smart service, specifically the implementations of the contained analytics algorithms has to be restricted to authorized personal only. Neither non-authorized personal shall be able to inspect nor to configure the algorithms.

---

[4] http://projekt-sepiapro.de

Data Security As mentioned in the former section, smart services process data that contain trade secrets and confidential information. Thus, specifically in the case of data shipping, where data are transferred to an analytics environment, it has to be assured that access to the data, the data sources, and the persistency layers is strictly controlled. This implies that shipped data have to be encrypted.

Data Ownership The discussed data security aspects are closely related to data ownership issues. While smart services can process data close to the machinery, there might also be situations where data has to be transferred to an external execution environment, such as a public cloud or the data center of the smart service developer. The latter case especially emerges if smart service developers establish new business models, whereby they also offer processing environments for the analysis of the metered data as a service, besides the mere development and integration of smart services. Thus, the data owners, which are typically the companies that operate the metered machinery, must not forfeit control on where their data is sent to.

Algorithms as Intellectual Property Even though a customer purchases and uses a smart service, this does not necessarily imply that they also obtain the rights to reuse, manipulate, or adapt an algorithm contained in a smart service. So, the intellectual property of the developer of a smart service has to be respected by restricting access to the smart service itself.

Smart Service Integrity In order to enable new business models for smart service developers, an app store like concept seems to be necessary. Developers need the possibility to offer their smart services via platforms such as public repositories or marketplaces, which allow customers to easily search, purchase, and utilize smart services. This will boost the acceptance and success of smart services. However, if smart services are provided that way, their integrity has to be assured. This means that a smart service developer needs a means to assure that algorithms and data contained in a smart service are not corrupted or manipulated from any third-party.

2.2 Technical Challenges

Use Case-specific Function and Data Shipping In many industrial applications of machine learning algorithms the volume of the generated data forbids their transport to centralized databases or computing centers [10]. Instead, techniques for an efficient reduction of the data volume and local data analysis close to the machinery are needed [5]. Thus, the ability to ship and execute functionality, either to analyze or to aggregate data as close to the production environment and machinery as possible, is inevitable.

Fast Adaption of Analytics Algorithms There are two reasons why the machine learning parts of smart services need to adapt quickly. Firstly, the algorithms are mostly tailored to specific analysis scenarios. Thus, they handle intrinsic characteristics of the data to process. If there are changes in the underlying dynamics and the characteristics of the metered data, the algorithms have to be retrained in order to deal with such concept drifts. For example, it could be that a new product is produced on the same machinery. Secondly, the machine learning algorithms have to be adapted and reconfigured if the production environment or the optimization objectives itself are changing. Here, one could think of a physical reordering of the existing machinery as an exemplary cause for the adaption of the machine learning model.

Stream and Batch Processing of Data Most of the metering data from machinery is not captured in a static but a streaming fashion resulting in more and more timely annotated data sources. To unleash the full potential of this data for use cases such as predictive maintenance it is necessary to process it on the fly, which typically results in stream processing approaches. Besides, there are also batch processing scenarios, which require to store raw data in order to be processed and analyzed later on. For example, the training procedures for most machine learning algorithms are working in a batch fashion.

Heterogeneous Technologies Smart services are complex compositions of different technologies in order to meter data from machinery and process them based on machine learning algorithms. Thus, it is up to the smart service developer which technologies and libraries to use for implementing the analytics algorithms. Further, the technical circumstances of the production environments and the machinery as well as the available IT infrastructures, platforms, middlewares, and applications on the customer side increase the number of heterogeneous technologies and software artifacts that are required in order to implement a smart service. Exemplarily, the component to fetch metering data from machinery might be designed to communicate via OPC-UA[5] with metered machinery in a specific production environment, while MQTT[6] has to be used in another one. To enable the efficient development of smart services, developers are forced to design the different components of a smart service as replaceable as possible.

Modularity In order to gain savings in terms of development time and, thus, time to market, components of a smart service have to be self-contained and interoperable with other components by stable interfaces. To raise great benefits regarding development costs for creating new smart services or adapting existing ones to new production environments, components of a smart service need to be packaged to be easily reusable.

---

[5] https://opcfoundation.org/about/opc-technologies/opc-ua/

[6] http://mqtt.org

Smart Service Provisioning As a result of the formerly described challenges, smart services have to be designed as analytics artifacts that can be dynamically changed. This can be based on changes of the analyzed data, which has direct impact on the used machine learning algorithms. Moreover, this can also be owing to changing technical circumstances at customer side or the emergence of new technologies. Thus, smart services should be fully automatically provisionable in order to efficiently deal with their dynamic character.

3 Self-Contained Packaging Format for Smart Services

In order to enable efficiently developing, shipping, and deploying smart services, a self-contained packaging format for smart services is inevitable. Thus, in SePiA.Pro, one of the main objectives is to develop a *Smart Service Archive (SMAR)* format that enables bundling all artifacts of a smart service. The conceptual structure of a SMAR is depicted in Figure 1. The Smart Service Archive format is based on TOSCA [17,16], an OASIS standard that enables describing applications to be provisioned in a portable manner. In particular, SMARs are based on the TOSCA *Cloud Service Archive (CSAR)*, which is a standardized archive format for packaging all required data to enable the automated provisioning of the respective application. TOSCA and CSARs are explained in the following in order to point out how the standardized format has to be enhanced and extended to support the packaging of smart services.

CSARs contain several data required to automatically provision an application. First, a CSAR contains a *topology model*, which is a directed graph describing the structure of the application to be provisioned. The topology model consists of nodes, which represent the components of the application such as Webservers and virtual machines, and edges between these nodes, which represent the dependencies between the components. Nodes are called *node templates*, the dependencies are called *relationship templates*. The TOSCA standard provides a means to specify types for node and relationship templates in the form of so called *node types* and *relationship types*, which allows to specify the semantics of the respective templates on a type level. Those types enable to populate a topology model by different manifestations and instances of a node or relationship, respectively. Therefore, they are reusable in arbitrary topology models and provide *reusable building blocks* for creating new applications. The actual artifacts that implement the components, such as Java classes or other binaries of an analytics algorithm, can be placed into the archive by means of so called *deployment artifacts*. These artifacts are associated with the corresponding node template for which they provide the implementation. Through this format, TOSCA tackles challenges regarding the reusability of components by means of node types. However, based on the presented key challenges in Section 2, it seems to be valuable to also treat algorithms and data along with policies as first level modeling concepts for smart services. Therefore, the SMAR format needs to extend the CSAR format by (i) *algorithms*, (ii) *data*, and (iii) *data policies* as conceptual first level modeling entities, as depicted in Figure 1. Although TOSCA already

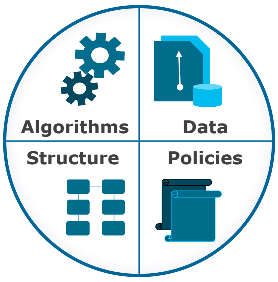

Fig. 1. Concept of the Smart Service Archive (SMAR)

supports policies for cloud applications in general [17,22] and description languages, such as USDL [3], can be combined with TOSCA to add non-functional service descriptions [8], a strict data policy approach for manufacturing data is missing. Treating algorithms and data as first level modeling concepts allows to enable function and data shipping approaches, either by clearly defining how functionality has to be shipped close to the data, or by enforcing the provisioning of smart services based on policies and further rules through an adequate toolchain. Therefore, SePiA.Pro will research on how already available TOSCA modeling concepts can be generally reused, which of the key challenges outlined in this paper can be translated into already existing modeling concepts of TOSCA, and where the TOSCA modeling approach eventually has to be extended by new concepts to support the requirements of smart services.

4 Function and Data Shipping for Manufacturing Environments

The capability to ship functionality as close to metered machinery, or to ship data to powerful analytics environments is vital to tailor smart services to different use cases. Besides the archive format for smart services, as introduced in Section 3, also a *Smart Service Ecosystem* to enable (i) modeling, (ii) shipping as well as (iii) provisioning and managing smart services is required and one of the main deliverables of the SePiA.Pro project.

To support these features, the planned toolchain will base on tools from the *OpenTOSCA Ecosystem*[7], which is an open source toolchain that supports modeling and provisioning of CSARs. The open-source TOSCA modeling tool *Winery*[8] [13], which supports modeling of topology models using the visual notation Vino4TOSCA [7], will be extended and enhanced to a *Smart Service Design Platform* in order to support modeling of SMARs. Further, a *Smart Service Repository and Self-Service Portal* will be developed based on OpenTOSCA's *Vinothek* [6] in order to efficiently ship smart services packaged in the SMAR

---

[7] https://www.iaas.uni-stuttgart.de/OpenTOSCA/

[8] https://projects.eclipse.org/projects/soa.winery

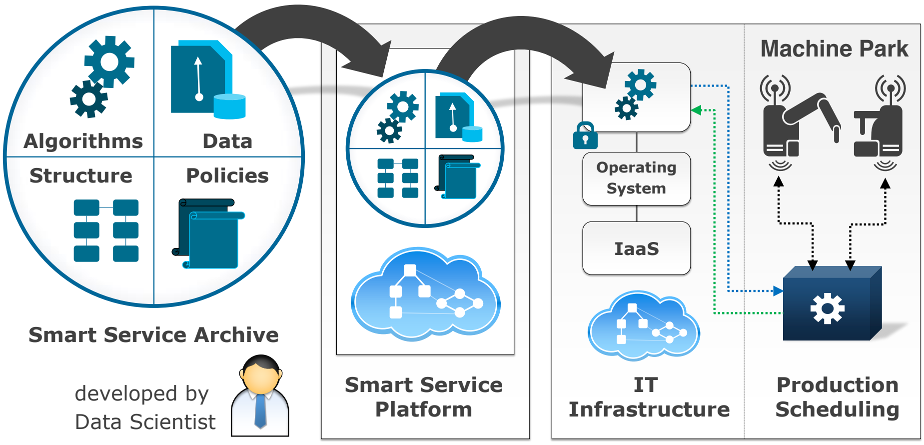

Fig. 2. Function and Data Shipping enabled by the Self-Contained Packaging Format for Smart Services

format on the one hand, and to trigger the provisioning of smart services by end users on the other hand. Finally, the *OpenTOSCA Container* [4], an open-source runtime environment for CSARs, will be extended and enhanced in order to be able to process the new SMAR format.

The Smart Service Ecosystem will support the development and application of smart services as shown in Figure 2. The SMAR format is the key enabler to automatically provision smart services. As shown on the left of the figure, a data scientist can develop an analytics algorithm and put it into a SMAR along with references to data that have to be processed. References to data can point to files also contained in the archive, e.g., if the algorithm requires static data that does not change over time. But they can also point to sensors from machinery that constantly deliver metering data. The data can be secured by adding specific data policies. These could declare, e.g., that the metering data must not leave the IT infrastructure of the customer that will run the smart service. All these steps in Figure 2 will be supported by the Smart Service Design Platform[9].

SMARs can then be deployed on a Smart Service Platform that includes the extended OpenTOSCA Container. The Smart Service Platform is capable of provisioning new instances of a smart service, which is bundled in a deployed SMAR. This is conceptually illustrated in Figure 2 by the IT infrastructure box, where an application stack is sketched that consists of an operating system, installed on a virtual machine in a private infrastructure as a service cloud. The analytics algorithms developed by the data scientist are provisioned on top of the operating system and wired to the sensors in the machine park via a production scheduling system. In this scenario, the production scheduling system wraps

---

[9] SMARs that are packaged in this manner can be uploaded to a Smart Service Repository and Self-Service Portal to ease the shipping to customers. For the sake of simplicity this step is omitted in Figure 2.

the access to metering data from machinery as well as to functionality that enables to adjust and to configure machinery programmatically. The connection is established bidirectionally to receive metering data from the machinery and to also send commands for adjustments of the machinery back to the production scheduling system. Thus, this provisioning flow indicates how machinery can be augmented by smart services and how functionality can be automatically shipped and provisioned closely to the sensors.

Other scenarios could cause to provision the analytics stack along with the algorithms in a public cloud environment, e.g., if the data to process is allowed to physically leave the IT infrastructure of the data owner. In this case, functionality is shipped automatically to the respective public cloud and, besides, also data is shipped to the analytics stack in the public cloud environment.

Finally, scenarios are possible, where the analytics stack is provisioned at a data center of the data scientist. This last scenario shows that also pure data shipping approaches are possible by the depicted toolchain in combination with the SMAR format since the extended format allows also packaging data only.

5 Conclusion

The presented key challenges are the objects of investigation for the project SePiA.Pro. A huge potential of smart services to generate business value in the field of Industry 4.0 on the one hand and the missing formats to provision smart services on the other hand are motivating this research project. SePiA.Pro will transfer the presented challenges into research questions in order to refine and enhance the TOSCA standard to the field of smart services in production environments. The resulting format will enable function and data shipping scenarios in the context of Industry 4.0. Although the project is still in its opening stages at the time of writing this paper, the presented vision of a new packaging format for smart services along with an OpenTOSCA-based toolchain show how cloud computing technologies may boost developments in the sector of manufacturing.

Acknowledgments. This work is partially funded by the project SePiA.Pro (01MD16013F) of the BMWi program Smart Service World.

References

1. Allmendinger, G., Lombreglia, R.: Four strategies for the age of smart services. Harvard Business Review 83(10), 131 (2005)
2. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer Networks 54(15), 2787–2805 (2010)
3. Barros, A., Oberle, D. (eds.): Handbook of Service Description: USDL and Its Methods. Springer (2012)
4. Binz, T., Breitenbücher, U., Haupt, F., Kopp, O., Leymann, F., Nowak, A., Wagner, S.: OpenTOSCA – A Runtime for TOSCA-based Cloud Applications. In: Proceedings of the 11th International Conference on Service-Oriented Computing (ICSOC 2013). pp. 692–695. Springer (2013)

5. Bolón-Canedo, V., Snchez-Maroo, N., Alonso-Betanzos, A.: Feature Selection for High-Dimensional Data. Artificial Intelligence: Foundations, Theory, and Algorithms, Springer (2015)
6. Breitenbücher, U., Binz, T., Kopp, O., Leymann, F.: Vinothek - A Self-Service Portal for TOSCA. In: Proceedings of the 6th Central-European Workshop on Services and their Composition (ZEUS 2014). pp. 69–72. CEUR-WS.org (2014)
7. Breitenbücher, U., Binz, T., Kopp, O., Leymann, F., Schumm, D.: Vino4TOSCA: A Visual Notation for Application Topologies based on TOSCA. In: On the Move to Meaningful Internet Systems: OTM 2012 (CoopIS 2012). pp. 416–424. Springer (2012)
8. Cardoso, J., Binz, T., Breitenbücher, U., Kopp, O., Leymann, F.: Cloud computing automation: Integrating usdl and tosca. In: Proceedings of the 25th International Conference on Advanced Information Systems Engineering (CAiSE 2013). Springer (2013)
9. Chaouchi, H.: The Internet of Things: Connecting Objects. ISTE, Wiley
10. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems 29(7), 1645–1660
11. Hermann, M., Pentek, T., Otto, B.: Design Principles for Industrie 4.0 Scenarios. In: Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS). pp. 3928–3937 (2016)
12. Kempa-Liehr, A.W.: Performance analysis of concurrent workflows. Journal of Big Data 2(10), 1–14 (2015)
13. Kopp, O., Binz, T., Breitenbücher, U., Leymann, F.: Winery – A Modeling Tool for TOSCA-based Cloud Applications. In: Proceedings of the 11th International Conference on Service-Oriented Computing (ICSOC 2013). pp. 700–704. Springer (2013)
14. Mobley, R.K.: An introduction to predictive maintenance. Elsevier Inc., 2 edn.
15. Mukhopadhyay, S.C. (ed.): Internet of Things, Smart Sensors, Measurement and Instrumentation, vol. 9. Springer International Publishing (2014)
16. OASIS: Topology and Orchestration Specification for Cloud Applications (TOSCA) Primer Version 1.0 (2013)
17. OASIS: Topology and Orchestration Specification for Cloud Applications (TOSCA) Version 1.0 (2013)
18. Riede, M., Schueppel, R., Sylvester-Hvid, K.O., Kühne, M., Röttger, M.C., Zimmermann, K., Liehr, A.W.: On the communication of scientific data: The full-metadata format. Computer Physics Communications 181(3), 651–662 (2010)
19. Smart Service Welt Working Group/acatech (Eds.): Smart Service Welt Recommendations for the Strategic Initiative Web-based Services for Businesses. Tech. rep., Berlin (2015)
20. Sundmaeker, H., Guillemin, P., Friess, P., Woelffle, S.: Vision and challenges for realising the internet of things. European Commission Information Society and Media (2010)
21. Susto, G.A., Schirru, A., Pampuri, S., McLoone, S., Beghi, A.: Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. Transactions on Industrial Informatics 11(3), 812–820 (2015-06)
22. Waizenegger, T., Wieland, M., Binz, T., Breitenbücher, U., Haupt, F., Kopp, O., Leymann, F., Mitschang, B., Nowak, A., Wagner, S.: Policy4TOSCA: A Policy-Aware Cloud Service Provisioning Approach to Enable Secure Cloud Computing. In: On the Move to Meaningful Internet Systems: OTM 2013 Conferences. pp. 360–376. Springer (2013)