Cloud topology and orchestration using TOSCA: A systematic literature review

Published in the Proceedings of the 7th European Conference on Service-Oriented and Cloud Computing (ESOCC), pp. 207-215, 2018

Julian Bellendorf and Zoltán Ádám Mann

paluno – The Ruhr Institute for Software Technology
University of Duisburg-Essen, Essen, Germany
{firstname.lastname}@paluno.uni-due.de

Abstract. Topology and Orchestration Specification for Cloud Applications (TOSCA) is a standard for specifying the topology of cloud applications, their deployment on cloud resources, and their orchestration. In recent years, the cloud research community has shown increasing interest in TOSCA, leading to an increasing number of publications. These publications address different topics around TOSCA, e.g., devise cloud orchestration methods using TOSCA, extend the language of TOSCA, or present tools for manipulating TOSCA models. To help researchers and practitioners overview this multifaceted area of research, this paper presents the results of a systematic survey of the relevant literature.

Keywords: TOSCA · cloud · topology · orchestration.

1 Introduction

Cloud applications may comprise many components with different technical dependencies and constraints, making their deployment and ongoing management complicated and error-prone [27]. Also, the interoperability between management tools has become challenging [4]. Thus, the need arose to describe cloud applications and related management tasks on a higher level of abstraction, in a standardized format. To address these issues, the Organization for the Advancement of Structured Information Standards (OASIS) published the Topology and Orchestration Specification for Cloud Applications (TOSCA) standard in 2013 [30]. TOSCA is a modeling language addressing the deployment and portability of applications, and the reusability of application components [2, 13]. The original TOSCA specification was based on XML; the simplified TOSCA profile, released in 2016, used YAML [32].

TOSCA describes (i) the structure of composite cloud applications as *topology graphs* and (ii) *management plans* for deploying and maintaining cloud applications. In topology graphs, nodes represent components, which include management operations e.g. for creating, configuring or starting the component [19]. The topology graph also contains relationships between components; e.g., a "hosted-on" relation indicates the allocation of virtual to physical components.

In *imperative processing*, a Management Plan defines the management operations and their execution order, using a workflow language such as Business Process Model and Notation [31] or Business Process Execution Language [29]. In *declarative processing*, no Management Plans are defined; instead, a runtime system infers the necessary steps for typical operations (e.g., deployment) from the application topology based on some conventions [5].

TOSCA has played various roles in different research approaches: some used TOSCA as part of a more general methodology, others extended the modeling capabilities of TOSCA or designed tools to manipulate TOSCA models. The multifaceted use of TOSCA and the growing number of relevant papers make it hard to track all related research. The aim of this paper is to give an overview of the use of TOSCA in the research community. We performed a systematic literature survey to devise a taxonomy of the main research topics that have been addressed in connection with TOSCA.

To identify relevant papers, we first used Scopus[1] with the search string `"Topology and Orchestration Specification for Cloud Applications" AND (ABS(tosca) OR TITLE (tosca))`. Here, `ABS(tosca) OR TITLE (tosca)` means that the word TOSCA must be contained in the abstract or title, ensuring that TOSCA is a main aspect of the paper. In addition, we are looking for the full term (Topology and Orchestration Specification for Cloud Applications) to exclude papers that use the word TOSCA in another meaning. We focused on the period 2012–2017, since the first TOSCA papers were published in 2012, and 2017 was the last full year until the time of writing. We found 89 papers. We excluded 6 very short papers (less than 4 pages). Using Google Scholar[2], we found 8 more papers that were not in Scopus but fit our search string. This led to a total of 91 papers. Afterwards, we read each paper and categorized it using open coding. By continuously refining our coding scheme, we built up a taxonomy in a bottom-up fashion. Finally, we analyzed the results to identify focal points of existing research and directions for further research.

2 Survey results

Fig. 1 presents the taxonomy that we developed based on the analyzed papers. On the highest level, we categorized the papers based on their *main contribution regarding TOSCA*. We identified the following categories:

1. **Tools**: papers describing a tool for TOSCA. Further categorization is possible based on the type of tool; in particular, this includes modeling tools, tools for deployment automation, and run-time environments.
2. **Extension of language**: approaches that extend the TOSCA language. The extension may relate to topologies, Management Plans, or both. Furthermore, extensions aiming at a visual notation also belong to this category.

[1] https://www.scopus.com
[2] https://scholar.google.com

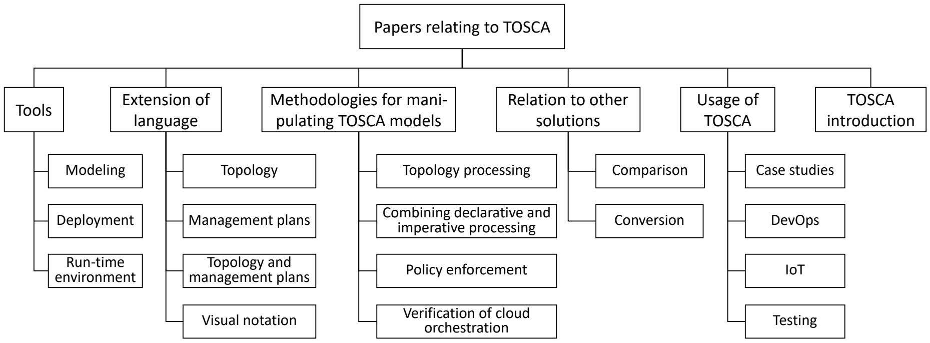

Fig. 1. Taxonomy for categorizing the processed papers

3. **Methodologies for manipulating TOSCA models**: papers describing a methodology for processing TOSCA models. The identified sub-categories are: papers about processing TOSCA topology models, approaches that combine declarative and imperative processing, solutions to enforce some given policies, and verification of cloud orchestration.
4. **Relation to other solutions**: papers about a relation – comparison or conversion – between TOSCA and some other technique.
5. **Usage of TOSCA**: papers demonstrating the use of TOSCA. This includes case studies that showcase the practical applicability of TOSCA, papers about the contribution of TOSCA to DevOps, papers about the use of TOSCA in an IoT (Internet of Things) setting, as well as papers about the use of TOSCA models for testing purposes.
6. **TOSCA introduction**: papers that introduce TOSCA or some of the concepts within TOSCA.

The following paragraphs describe some representative papers in each category of the taxonomy of Fig. 1, except for the category "TOSCA introduction." (Due to space limitation we cannot describe all found papers in detail.)

1) Tools The Tools category consists of papers mainly presenting tools for describing, deploying, and instantiating cloud applications using TOSCA. Prototypes that only serve to evaluate an approach are not assigned to this category.

Kopp et al. present the web-based modeling tool Winery [23]. First, Winery includes the Topology Modeler, with which components can be combined to form an application topology. Second, Winery contains the Element Manager, which can be used to create, modify, and delete components. Kopp et al. propose an extension to Winery that can be used to model Management Plans [24].

Binz et al. describe OpenTOSCA, a runtime for imperative processing of TOSCA applications [1]. This tool executes the defined Management Plans respectively the operations described within the nodes. Wettinger et al. present an extension to OpenTOSCA in the form of a unified invocation interface [40].

Breitenbücher et al. present Vinothek, which offers the user an interface for providing an instance of an application [7]. For this purpose, the user is offered the set of applications without having to deal with the technical details.

Katsaros et al. also provide a tool to deploy and manage software components [21]. The execution environment TOSCA2Chef parses TOSCA documents and deploys the components described in OpenStack Clouds using the Opscode Chef configuration management software and BPEL processes.

2) Extension of language Brogi et al. extend TOSCA by means for specifying the behavior of the application components when executing the management operations defined in the nodes [9]. Considering the effects of the operations to be performed and the states that the components assume after execution, makes the validation of Management Plans possible. Breitenbücher et al. propose a visual notation to unify the presentation of nodes within the topology [8]. Kopp et al. extend the BPMN to provide direct access to the topology elements [22]. The extension, called BPMN4TOSCA, can be transformed into standards-compliant BPMN.

3) Methodologies for manipulating TOSCA models Various approaches have been proposed to work with TOSCA models. Some focus on the processing of topologies, whereas others also take Management Plans into account to enforce policies, to verify the orchestration of the components, or to combine declarative and imperative processing.

Processing of Topologies Brogi and Soldani describe an approach that involves matching between individual Node Types and Service Templates [11, 12]. This matching allows sets of Node Types to be grouped together in a topology to reduce its complexity. In addition, proven combinations of Node Types can be reused in new application topologies. Service Templates which do not fit exactly can be adapted to create a template that matches exactly to a given node type.

Binz et al. observed that container components (e.g. virtual machines) are often underutilized by a single application component, so that additional components can be hosted by these containers [3]. For this purpose, the topologies from two applications that use the same container components are merged into one topology in which both applications retain their respective functionality. Saatkamp et al. present an approach for adapting the application topology when a provider specifies a new offer and certain components of the application need to be migrated to new container components [33].

Multiple approaches for topology completion were proposed, with the aim that an application developer only has to model the business-relevant components and the underlying infrastructure is automatically added. The approach of Hirmer et al. is based on a repository of nodes and relationships to fill in incomplete topologies [20]. Brogi et al. present an approach to collect information about suitable cloud offerings by crawling the network and storing their TOSCA representation in a repository [10]. This representation can be used by application developers to complete the topology. Soldani et al. propose TOSCAMart, an approach to reuse proven topologies in new environments [35]. The developer of a composite application defines a node in the topology that describes the requirements for the fragment being inserted. TOSCAMart then selects a solution suitable for these requirements from a repository of various existing topologies.

Combining declarative and imperative processing Breitenbücher et al. propose an approach that combines declarative and imperative processing to achieve hybrid processing [5, 6]. The defined application topologies are interpreted and finally, the associated Management Plans are generated. Calcaterra et al. present a similar approach, also based on interpreting a topology and providing the appropriate Management Plan [14].

Policy Enforcement Waizenegger et al. present a TOSCA runtime extension to enforce policies describing non-functional requirements, specifically security properties, such as the encryption of a database or the geographic positioning of privacy-related data [37, 38]. Policies can be defined using both single-node management operations and Management Plans.

Verification of Cloud Orchestration Yoshida et al. describe an approach to the formal verification of TOSCA topologies that can be used to test the achievement of a target state in the declarative processing of the TOSCA model [42]. The execution of the management operation is described by a state transition system in which a state with a certain property is to be reached. Chareonsuk and Vatanawood use Model Checking to verify security properties for imperative processing [15]. The approach of Tsigkanos and Kehrer is about defining patterns and anti-patterns and finally checking their presence or absence in the topology of a service template so that quality aspects can be proven [36].

4) Relation to other solutions A comparison between TOSCA and the Heat Orchestration Template (HOT) is provided by Di Martino et al. [17]. HOT is the template format used to define the structure of an application for declarative processing by the OpenStack orchestrator Heat. The main difference between the two approaches is that HOT is declarative while TOSCA supports both declarative and imperative processing. Also, similarities are shown, for example, both provide a catalog of nodes and resources that can be composed to applications.

Yongsiriwit et al. address the interoperability of standards for describing cloud resources: TOSCA, Open Cloud Computing Interface (OCCI) and Cloud Infrastructure Management Interface (CIMI) [41]. For interoperability, ontologies are defined that describe the resources noted in each standard. In addition, an upper-level ontology is presented to describe cloud resources regardless of the used standards. Using inference rules, the special descriptions can be translated into this higher-level format and vice versa, which also allows the translation from one standard to another. Using the upper ontology, a knowledge base could be created, providing insights into relationships and possible inconsistencies.

5) Usage of TOSCA This category consists of approaches that use the existing TOSCA notation. Kostoska et al. present a case study of the use of TOSCA for specifying the University Management System iKnow [25]. This system offers professors and students a platform to exchange electronic information and provide electronic services. Besides a detailed description of node and relationship types, this paper also mentions the challenges of using TOSCA for the specification of this application.

A different domain for using TOSCA is the specification of Internet of Things (IoT) applications. Li et al. show how TOSCA can be used for an IoT application: an Air Handling Unit (AHU) that controls air circulation in modern buildings [26]. Da Silva et al. demonstrate the feasibility of defining IoT applications using TOSCA, in the context of different technologies [16]. In another paper, Da Silva et al. address the multitude of sensor data produced in IoT [34]. The authors describe how Complex Event Processing Systems can be deployed using TOSCA to process the incoming data and efficiently use network resources.

3 Discussion

Our survey shows the *versatility* of TOSCA: its use in different domains (also beyond cloud computing), for different purposes, in different phases of the service lifecycle, by different groups of users. This versatility is mainly due to (i) the possibility to define custom types for nodes, relationships, and capabilities and (ii) the possibility to define and manipulate partial topologies. However, this versatility also poses the risk of the proliferation of incompatible TOSCA dialects. Hence we expect that *interoperability* will play an increasingly important role.

Some further topics received limited attention so far and represent important targets for future research. First, given the enormous importance of *security* in cloud computing, it is striking that very few papers address it so far (although several authors mentioned it as future work [41, 33]). Also, TOSCA support for other related topics like *data protection* needs to be investigated [28]. Second, the topic of *verification and validation* (V&V) is also addressed by few papers. Given the importance of V&V, we expect to see more work on how TOSCA can be used to improve V&V. Third, partial topologies open many possibilities for *optimization*, from which only a little has been investigated, mainly in connection with cost minimization. Many other aspects of optimization, e.g., related to performance and reliability, are yet to be explored. Finally, TOSCA has been shown to be useful in areas such as IoT and DevOps [39]. We expect to see TOSCA being applied to new domains like *network function virtualization* [18] or *fog and edge computing*.

Acknowledgment. This work received funding from the European Union's Horizon 2020 research and innovation programme (grant agreement 731678 (RestAssured)).

References

1. Binz, T., Breitenbücher, U., Haupt, F., Kopp, O., Leymann, F., Nowak, A., Wagner, S.: OpenTOSCA – A runtime for TOSCA-based cloud applications. In: Proc. ICSOC 2013. pp. 692–695 (2013)
2. Binz, T., Breitenbücher, U., Kopp, O., Leymann, F.: TOSCA: Portable Automated Deployment and Management of Cloud Applications. In: Advanced Web Services. pp. 527–549 (2014)

3. Binz, T., Breitenbücher, U., Kopp, O., Leymann, F., Weiß, A.: Improve resource-sharing through functionality-preserving merge of cloud application topologies. In: Proc. CLOSER 2013. pp. 96–103 (2013)
4. Binz, T., Breiter, G., Leyman, F., Spatzier, T.: Portable Cloud Services Using TOSCA. IEEE Internet Computing **16**(3), 80–85 (2012)
5. Breitenbücher, U., Binz, T., Képes, K., Kopp, O., Leymann, F., Wettinger, J.: Combining declarative and imperative cloud application provisioning based on TOSCA. In: Proc. IC2E 2014. pp. 87–96 (2014)
6. Breitenbücher, U., Binz, T., Kopp, O., Képes, K., Leymann, F., Wettinger, J.: Hybrid TOSCA provisioning plans: Integrating declarative and imperative cloud application provisioning technologies. In: Proc. CLOSER 2015. pp. 239–262 (2016)
7. Breitenbücher, U., Binz, T., Kopp, O., Leymann, F.: Vinothek – a self-service portal for TOSCA. In: Proc. ZEUS 2014. pp. 72–75 (2014)
8. Breitenbücher, U., Binz, T., Kopp, O., Leymann, F., Schumm, D.: Vino4TOSCA: A visual notation for application topologies based on TOSCA. In: Proc. OTM 2012. pp. 416–424 (2012)
9. Brogi, A., Canciani, A., Soldani, J.: Modelling and analysing cloud application management. In: Proc. ESOCC 2015. pp. 19–33 (2015)
10. Brogi, A., Cifariello, P., Soldani, J.: DrACO: Discovering available cloud offerings. Computer Science - Research and Development **32**(3-4), 269–279 (2017)
11. Brogi, A., Soldani, J.: Matching cloud services with TOSCA. In: Proc. ESOCC 2013. pp. 218–232 (2013)
12. Brogi, A., Soldani, J.: Finding available services in TOSCA-compliant clouds. Science of Computer Programming **115-116**, 177–198 (2016)
13. Brogi, A., Soldani, J., Wang, P.: TOSCA in a nutshell: Promises and perspectives. In: Proc. ESOCC 2014. pp. 171–186 (2014)
14. Calcaterra, D., Cartelli, V., Di Modica, G., Tomarchio, O.: Combining TOSCA and BPMN to enable automated cloud service provisioning. In: Proc. CLOSER 2017. pp. 159–168 (2017)
15. Chareonsuk, W., Vatanawood, W.: Formal verification of cloud orchestration design with TOSCA and BPEL. In: Proc. ECTI-CON 2016. pp. 1–5 (2016)
16. Da Silva, A., Breitenbücher, U., Hirmer, P., Képes, K., Kopp, O., Leymann, F., Mitschang, B., Steinke, R.: Internet of things out of the box: Using TOSCA for automating the deployment of IoT environments. In: Proc. CLOSER 2017. pp. 330–339 (2017)
17. Di Martino, B., Cretella, G., Esposito, A.: Defining cloud services workflow: a comparison between TOSCA and OpenStack Hot. In: Proc. CISIS 2015. pp. 541–546 (2015)
18. Dräxler, S., Karl, H., Mann, Z.A.: Joint optimization of scaling and placement of virtual network services. In: Proc. CCGrid 2017. pp. 365–370 (2017)
19. Haupt, F., Leymann, F., Nowak, A., Wagner, S.: Lego4TOSCA: Composable building blocks for cloud applications. In: Proc. CLOUD 2014. pp. 160–167 (2014)
20. Hirmer, P., Breitenbücher, U., Binz, T., Leymann, F.: Automatic topology completion of TOSCA-based cloud applications. In: Proc. Informatik 2014. pp. 247–258 (2014)
21. Katsaros, G., Menzel, M., Lenk, A., Rake-Revelant, J., Skipp, R., Eberhardt, J.: Cloud application portability with TOSCA, Chef and Openstack: Experiences from a proof-of-concept implementation. In: Proc. IC2E 2014. pp. 295–302 (2014)
22. Kopp, O., Binz, T., Breitenbücher, U., Leymann, F.: BPMN4TOSCA: A domain-specific language to model management plans for composite applications. In: Proc. BPMN 2012. pp. 38–52 (2012)

23. Kopp, O., Binz, T., Breitenbücher, U., Leymann, F.: Winery – a modeling tool for TOSCA-based cloud applications. In: Proc. ICSOC 2013. pp. 700–704 (2013)
24. Kopp, O., Binz, T., Breitenbücher, U., Leymann, F., Michelbach, T.: A domain-specific modeling tool to model management plans for composite applications. In: Proc. ZEUS 2015. pp. 51–54 (2015)
25. Kostoska, M., Chorbev, I., Gusev, M.: Creating portable TOSCA archive for iKnow university management system. In: Proc. FedCSIS 2014. pp. 761–768 (2014)
26. Li, F., Vögler, M., Claeßens, M., Dustdar, S.: Towards automated IoT application deployment by a cloud-based approach. In: Proc. SOCA 2013. pp. 61–68 (2013)
27. Mann, Z.Á.: Resource optimization across the cloud stack. IEEE Transactions on Parallel and Distributed Systems **29**(1), 169–182 (2018)
28. Mann, Z.Á., Metzger, A.: Optimized cloud deployment of multi-tenant software considering data protection concerns. In: Proc. CCGrid 2017. pp. 609–618 (2017)
29. OASIS: Web Services Business Process Execution Language Version 2.0 (Apr 2007), OASIS Standard
30. OASIS: Topology and Orchestration Specification for Cloud Applications Version 1.0. (Nov 2013), http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html., OASIS Standard
31. OMG: Business Process Model and Notation (BPMN) Version 2.0. (Jan 2011), OMG Document Number: formal/2011-01-03
32. Palma, D., Rutkowski, M., Spatzier, T.: TOSCA Simple Profile in YAML Version 1.0 (Dec 2016), http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/TOSCA-Simple-Profile-YAML-v1.0.html., OASIS Standard
33. Saatkamp, K., Breitenbücher, U., Kopp, O., Leymann, F.: Topology splitting and matching for multi-cloud deployments. In: Proc. CLOSER 2017. pp. 247–258 (2017)
34. Franco da Silva, A., Hirmer, P., Breitenbücher, U., Kopp, O., Mitschang, B.: Customization and provisioning of complex event processing using TOSCA. Computer Science – Research and Development pp. 1–11 (2017)
35. Soldani, J., Binz, T., Breitenbücher, U., Leymann, F., Brogi, A.: ToscaMart: A method for adapting and reusing cloud applications. Journal of Systems and Software **113**, 395–406 (2016)
36. Tsigkanos, C., Kehrer, T.: On formalizing and identifying patterns in cloud workload specifications. In: Proc. WICSA 2016. pp. 262–267 (2016)
37. 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: Proc. OTM 2013. pp. 360–376 (2013)
38. Waizenegger, T., Wieland, M., Binz, T., Breitenbücher, U., Leymann, F.: Towards a policy-framework for the deployment and management of cloud services. In: Proc. SECURWARE 2013. pp. 14–18 (2013)
39. Wettinger, J., Behrendt, M., Binz, T., Breitenbücher, U., Breiter, G., Leymann, F., Moser, S., Schwertle, I., Spatzier, T.: Integrating configuration management with model-driven cloud management based on TOSCA. In: Proc. CLOSER 2013. pp. 437–446 (2013)
40. Wettinger, J., Binz, T., Breitenbücher, U., Kopp, O., Leymann, F., Zimmermann, M.: Unified invocation of scripts and services for provisioning, deployment and management of cloud applications based on TOSCA. In: Proc. CLOSER 2014. pp. 559–568 (2014)
41. Yongsiriwit, K., Sellami, M., Gaaloul, W.: A semantic framework supporting cloud resource descriptions interoperability. In: Proc. CLOUD 2016. pp. 585–592 (2017)

42. Yoshida, H., Ogata, K., Futatsugi, K.: Formalization and verification of declarative cloud orchestration. In: Proc. ICFEM 2015. pp. 33–49 (2015)