Computing manuscript No.
(will be inserted by the editor)

Specification of cloud topologies and orchestration using TOSCA: A survey

Julian Bellendorf · Zoltán Ádám Mann

Received: date / Accepted: date

Abstract Topology and Orchestration Specification for Cloud Applications (TOSCA) is an OASIS standard for specifying the topology of cloud applications, their deployment on physical or virtual cloud resources, and their orchestration. In recent years, the cloud research community has shown significant interest in TOSCA, leading to an increasing number of related publications. Such publications address a wide-ranging set of topics around TOSCA, e.g., devise sophisticated cloud orchestration methodologies 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. We have processed over 120 papers and categorized them, leading to a taxonomy with 6 categories and 19 subcategories. The analysis of the results reveals several notable tendencies, as well as areas requiring future research.

Keywords TOSCA · cloud computing · cloud topology · cloud orchestration.

1 Introduction

With the widespread adoption of cloud computing, more and more applications are deployed in a cloud setting. For complex applications, comprising many components with different technical dependencies and constraints, also

Accepted for publication in *Computing*, 2019

J. Bellendorf
paluno – The Ruhr Institute for Software Technology
University of Duisburg-Essen
Essen, Germany

Z. Á. Mann
paluno – The Ruhr Institute for Software Technology
University of Duisburg-Essen
Essen, Germany

making use of different platform components, the deployment and ongoing management tasks have become dauntingly complicated and error-prone using traditional tools [84]. Also, the lack of interoperability between different tools became a severe limitation [14]. Thus, the need arose to describe cloud applications and related management tasks on a higher level of abstraction, in a standardized format.

To address the above-mentioned issues, the Organization for the Advancement of Structured Information Standards (OASIS[1]) published the TOSCA standard. TOSCA is a modeling language addressing the deployment and portability of cloud applications, as well as the interoperability and reusability of individual components of these applications [12].

Besides its industrial adoption[2], TOSCA also sparked significant interest in the research community. It 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 support the manipulation of TOSCA models. This multifaceted use of TOSCA and the growing number of relevant papers – which have been published in a wide variety of conferences and journals – make it hard to track all related research. Therefore, the aim of this paper is to give an overview of the use of TOSCA in the research community. We performed a systematic literature review to devise a taxonomy of the main research topics that have been addressed in connection with TOSCA in recent years. The analysis of the results reveals some trends and also gives hints about promising directions for future research.

This work is a significantly extended version of our previous brief TOSCA survey [7], extending both the breadth and depth of its coverage and also further developing the taxonomy and the analysis of the results.

The rest of the paper is organized as follows. Section 2 provides a brief introduction to TOSCA. Section 3 describes the used survey methodology. Section 4 gives an overview of the results, followed by the description of the details in Section 5. The findings are discussed in Section 6, while Section 7 concludes the paper.

2 A short introduction to TOSCA

In 2013, the OASIS published the Topology and Orchestration Specification for Cloud Applications (TOSCA) standard [91]. The concept of TOSCA includes two aspects: (i) the structural description of a composite cloud application and its deployment as a topology graph and (ii) the so-called management plans, which specify how the management operations of the components are used for setting up, managing, and orchestrating a composite cloud application.

The description of the *topology* of an application essentially consists of two types of elements, i.e., nodes and relationships. *Nodes* represent individual components and also define management operations for those components,

[1] https://www.oasis-open.org/
[2] See, e.g., https://wiki.oasis-open.org/tosca/TOSCA-implementations

such as for creating, configuring, or starting a component [63]. *Relationships* describe the relationships of the individual components to each other. For example, a “hosted-on” relationship can be used to allocate virtual components to the respective physical resources. Nodes and relationships are represented as node templates and relationship templates in a topology template. The term template means that the respective element may be instantiated several times. The type of a template defines its semantics [12].

TOSCA foresees two types of processing for the deployment and orchestration of composite applications [17]. In *imperative processing*, an explicit management plan defines the exact management operations and their execution order. TOSCA uses existing workflow languages such as BPMN (Business Process Model and Notation [92]) and BPEL (Business Process Execution Language [90]) for the definition of management plans. For *declarative processing*, no management plans are defined. Instead, a runtime system interprets the application topology and infers the necessary steps for typical operations (e.g., deployment) based on appropriate conventions.

The original TOSCA specification was based on XML. The simplified TOSCA profile, released in 2016, used YAML, “to simplify the authoring of TOSCA service templates” [93].

For a more in-depth introduction to TOSCA, we refer to existing introductory papers. In particular, Binz et al. [12] give an overview of TOSCA and name the challenges that TOSCA addresses, while Brogi et al. [34] present a compact overview of TOSCA and the underlying goals of the specification. Besides, Binz et al. [14] show how to use existing workflow technologies in the context of TOSCA to ensure the portability of management plans. Haupt et al. [63] focus on making the most of the capabilities provided by TOSCA for the design of node types.

3 Survey methodology

For our literature survey, we used a systematic methodology [24]. Fig. 1 gives an overview of the process, structured into three phases: planning, conducting, and documenting the survey. In the following, each step is explained.

Based on the high-level objectives described in Section 1, we identified the following research questions for our survey:

- RQ1: What are the main contributions of existing research with respect to TOSCA?
- RQ2: Are there noticeable trends in the topics of TOSCA research over the years?
- RQ3: Which topics need further research?

We used a combination of several methods to identify the set of relevant publications. First, we used keyword search to identify an initial set of possibly

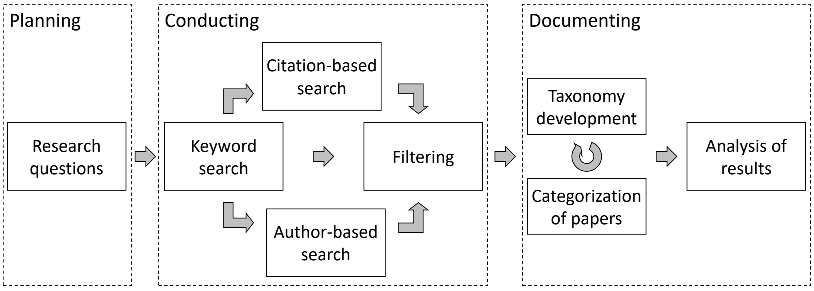

Fig. 1 Overview of survey methodology

relevant papers. For this purpose, we used the Scopus database[3] and applied the following search string:

> "Topology and Orchestration Specification for Cloud Applications" AND (TITLE-ABS-KEY(TOSCA))

Here, TITLE-ABS-KEY means that the given expression (TOSCA) must be contained in the title, abstract, or keywords of the paper. Searching for the word TOSCA in the title, abstract and keywords ensures that TOSCA is a main aspect of the respective paper. In addition, we are looking for the full version of the term (Topology and Orchestration Specification for Cloud Applications) in the text to exclude papers that use the word TOSCA in another meaning. We focused on the period from 2012 to 2018 (also including the years 2012 and 2018). 2012 was the year in which the first TOSCA papers were published. We considered all papers that were published before 1st January 2019.

In addition to Scopus, we used Google Scholar[4] to find papers that could not be found in the Scopus database, but fit the described search string. Moreover, to ensure that we do not miss relevant papers that compare TOSCA and other modeling languages but do not mention the keyword TOSCA so prominently, we also repeated the search with a slightly changed search string, in which we replaced "TOSCA" with "comparison" and "compare". In an additional step, we looked at the papers that are referenced in the papers found so far. Furthermore, we looked at the work of the members of the Technical Committee listed in the membership roster of the OASIS[5].

To filter the relevant papers, we used the following criteria:

- Inclusion criteria:
  IN1: Scientific papers published in international peer-reviewed outlets.
  IN2: Studies in which TOSCA plays a central role.
- Exclusion criteria:
  EX1: Studies in which TOSCA is of only marginal relevance.
  EX2: Short papers (less than 4 pages in double-column format).

[3] https://www.scopus.com
[4] https://scholar.google.com
[5] https://www.oasis-open.org/committees/tosca/

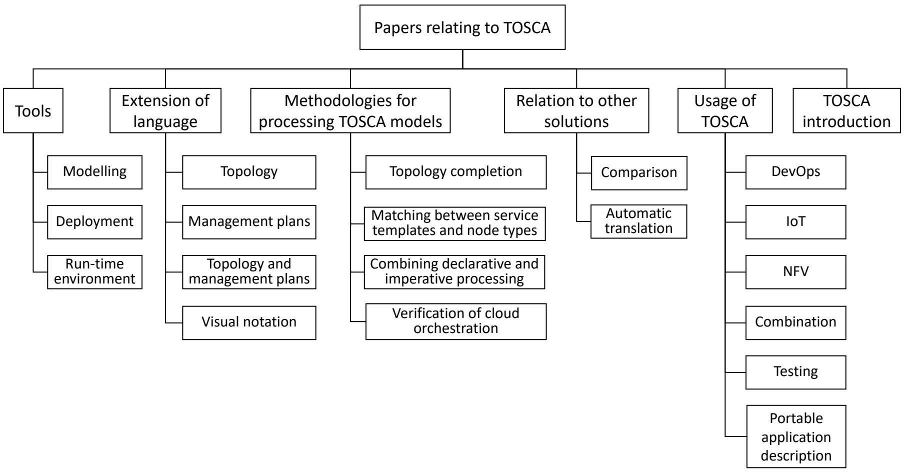

Fig. 2 Taxonomy for categorizing the processed papers

EX3: Non-English papers.

To assess the relevance of a publication regarding TOSCA (criteria IN2 and EX1), we relied on information given by the authors of that publication. Accordingly, the key indicator was the mention of the term TOSCA in the title, in the abstract or in the keywords. Exceptions are the papers that compare TOSCA and other modeling languages, as explained above.

The searching and filtering process resulted in a total of 124 papers. Afterwards, we read each paper and categorized them using open coding. Parallel to the processing of the papers, we continuously refined our coding scheme, building up a taxonomy in a bottom-up fashion, and also refining the categorization of already processed papers. Both the taxonomy and the categorization of the papers were regularly discussed between the authors to ensure consistency. In addition, an earlier version of the taxonomy was presented at a conference [7]; the feedback from the conference reviewers and from discussions at the conference has also been embodied in the final version of the taxonomy.

When the taxonomy was finished and each paper categorized according to this taxonomy, RQ1 could be answered. In the last step, we analyzed the results to identify focal points of existing research, trends, as well as possible directions for further research. The results of this analysis delivered answers to RQ2 and RQ3.

4 Survey results

Fig. 2 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:

The **Tools** category consists of papers that describe a tool for TOSCA. For this purpose only mature tools are considered. Prototypes that are only used to validate an approach can be found in the other categories. Further categorization is possible based on the type of tool regarding its use in the lifetime of a cloud application. The application designer uses **modeling** tools to describe the components of an application and the relationships between them. **Deployment** tools are used to install and launch an application. Finally, the **run-time environment** is responsible for the management operations such as for creating, starting, or configuring components.

The category **Extension of language** consists of extensions to the standard defined by OASIS, for example to define and enforce policies. This extension does not refer to the definition of new node or relationship types, but to the way in which a node type is defined. The further categorization is based on the aspect of TOSCA, to which the respective approach refers. That may be the **topology** or the **management plans**. It is also possible that one approach applies to both aspects. The **visual notation** sub-category is about the graphical representation of cloud topologies. Such a notation is not provided for in the TOSCA standard, so that it represents a separate category.

The category **Methodologies for processing TOSCA models** contains methods for processing TOSCA without describing an extension of the language. Four subcategories have been defined for this category. There may be several reasons for the need to **complete a topology**. For example, a designer may focus on defining the logical components of an application, and the assignment to physical components could eventually be automated, thus completing the topology. The **Matching between service templates and node types** sub-category refers to grouping multiple nodes into a single one. This one node represents a service template consisting of several nodes. Topologies of cloud applications can quickly become confusing, so it makes sense to group individual nodes together. In addition, a node can be defined as a placeholder, which can later be replaced by a proven configuration of nodes. Another problem with topologies that include a variety of components is the complexity of the associated management plans. The definition of management plans is therefore prone to error in applications with a variety of components. **Combining declarative and imperative processing** is a possible way to cope with this complexity. Approaches in the category **Verification of cloud orchestration** ensure the correct deployment of the cloud application. This includes for example the execution of management operations in the correct order.

The category **Relation to other solutions** contains two types of relationships between TOSCA and other modeling languages. One refers to the **comparison**, the other to an **automatic translation** between TOSCA and another modeling language.

The **Usage of TOSCA** category describes approaches that use TOSCA but do not extend the language defined by the standard or the way TOSCA models are processed. The sub-categories refer to the domain in which TOSCA is used (papers about the contribution of TOSCA to **DevOps**, papers about the use of TOSCA in an **IoT (Internet of Things)** or **NFV (Network Function Virtualization)** setting) or the purpose for which TOSCA is used (papers about a **combination** with other tools, papers about the use of

Category	Sub-category	Paper
Tools	Modelling	[75] [76]
	Deployment	[19]
	Run-time environment	[11] [69] [124] [125]
Extension of language	Topology	[4] [15] [16] [25] [26] [27] [101] [116] [132] [134]
	Management plans	[74] [107] [117]
	Topology and management plans	[22] [60] [61] [68] [78] [119] [120]
	Visual notation	[20]
Methodologies for processing TOSCA models	Topology completion	[1] [13] [29] [35] [36] [47] [55] [64] [65] [94] [100] [102] [104] [110]
	Matching between service templates and node types	[31] [32] [33]
	Combining declarative and imperative processing	[17] [18] [21] [42] [43] [44] [87]
	Verification of cloud orchestration	[30] [39] [49] [115] [131]
Relation to other solutions	Comparison	[8] [48] [52] [53] [54] [57] [73] [96] [109] [121]
	Automatic translation	[9] [10] [40] [130]
Usage of TOSCA	Portable application description	[5] [28] [37] [38] [41] [46] [56] [58] [59] [62] [66] [70] [72] [77] [79] [88] [89] [95] [97] [98] [99] [111] [112] [113] [114] [118] [128] [133]
	DevOps	[71] [123] [126] [127]
	IoT	[50] [81] [103] [108]
	Testing	[106] [129]
	Combination	[45] [51]
	Network Function Virtualization	[2] [3] [67] [105] [122]
TOSCA introduction		[12] [14] [23] [34] [63] [80] [82] [83]

Table 1 Overview of the found papers, assigned to the categories of the taxonomy

TOSCA models for **testing** purposes). Several papers in this category use TOSCA generally for the **portable description of applications** and cannot be assigned to a specific sub-category.

Finally, the category **TOSCA introduction** includes papers that introduce TOSCA or some of the concepts within TOSCA.

Table 1 gives an overview of the papers of each category.

5 Detailed results

The following subsections describe selected papers in each category of the taxonomy of Fig. 2, except for the category "TOSCA introduction." Due to space limitations, we cannot describe all papers in detail. One criterion for selecting papers to describe was to prefer papers with a high number of citations, either overall (e.g., [11,17]) or within a category (e.g., [65]). There were also some sub-categories with few papers, so that all those papers could be described (e.g., sub-categories “Modelling” and “Deployment” in the “Tools” category).

5.1 Tools

The Tools category consists of papers whose main contribution is a tool for describing, deploying, or instantiating cloud applications using TOSCA. Prototypes that only serve to illustrate or evaluate an approach are not assigned to this category.

Kopp et al. [75] present the web-based modeling tool Winery. Winery includes the Topology Modeler, with which components can be combined to form an application topology, and the Element Manager, which can be used to create, modify, and delete the components contained in the topologies. In addition to Topology Modeler and Element Manager, Winery also includes a repository to store the data for the other two components. Kopp et al. [76] also propose an extension to Winery that can be used to model management plans. However, this approach was only implemented as a prototype.

Binz et al. [11] describe OpenTOSCA, a runtime for imperative processing of TOSCA applications . This runtime serves to execute the defined management plans respectively the operations described within the nodes. Wettinger et al. [124,125] present an extension to OpenTOSCA in the form of a unified invocation interface. This interface provides the logic to perform management operations and hides the technical details of the underlying technologies. The interface is called from the management plans to provision an application.

Breitenbücher et al. [19] add Vinothek to the above two tools. Vinothek offers the user an interface for providing an instance of an application. For this purpose, the user is offered the set of applications without having to deal with technical details. The selected application is started by a connected runtime.

Katsaros et al. [69] also provide a tool to deploy and manage software components. 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.

5.2 Extension of language

This subsection deals with papers that extend TOSCA as a modeling language. The extension may affect the topology elements defined in the standard and/or the languages used by TOSCA to describe management plans.

Brogi et al. [25–27] extend TOSCA with means for specifying the behavior of the application components when executing the management operations defined for the nodes. Considering the effects of the operations to be performed and the states that the components assume after execution, the validation of management plans is possible. In a valid management plan, all actions are performed in the correct order. In an invalid plan, for example, the start process could be defined to run before installing a component.

Zimmermann et al. [132] define an extension to the topology modeling in TOSCA, regarding the business operations of the individual components, which should be displayed in a technology-agnostic manner. The application deployment described in this approach includes a service bus for communication between the individual components, respectively for calling these business operations using identifiers and interface descriptions.

Other papers focus on expanding the language of the management plans. Kopp et al. [74] extend the Business Process Model and Notation (BPMN) to provide direct access to the topology elements. The extended version of BPMN, called BPMN4TOSCA, can be transformed into standards-compliant BPMN. Vetter [117] shows an approach that describes the management plans through XML nets. This description allows the search for errors in the defined plans by finding anti-patterns. An anti-pattern could be for instance a missing or unneeded action.

Waizenegger et al. [119, 120] present a prototypical implementation of 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. Policies can be defined using both single-node management operations and management plans. Developers can create different instances of a policy so that the user can choose a specific version without having to adjust the entire template.

Breitenbücher et al. [20] propose a visual notation to unify the presentation of nodes within the topology, to avoid misunderstandings in the interpretation of the topologies defined in XML . This involves the presentation of nodes and relationship templates, as well as groups for combining multiple elements into a single element to reduce the complexity of large representations.

5.3 Methodologies for processing TOSCA models

Various approaches have been proposed to work with TOSCA models. Some focus on the processing of topologies, whereas others define a matching between service templates and node types, combine declarative and imperative processing, or verify the orchestration of the components.

5.3.1 Topology completion

Binz et al. [13] propose sharing resources to save costs. The rationale is that container components (for example, virtual machines) are often underutilized by a single application component so that additional components can be hosted by these containers. For this purpose, the topologies from two applications that use the same container components are merged into a topology after a feasibility check has been performed. This process results in a topology in which both applications retain their respective functionality. Saatkamp et al. [102] present another approach to save costs when running cloud applications, by 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. The approach checks whether the capabilities offered match the requirements of the given application components, and the topology is split based on the providers' offerings.

Automatic completion of TOSCA topology models has also been considered. The aim is that an application developer only has to model the business relevant components and the underlying infrastructure is automatically added. This leads to a reduction of the effort because the developer does not have to know the technical details of the infrastructure and platform components to deploy an application. Multiple approaches have been proposed to achieve this aim. The approach of Hirmer et al. [65] is based on a repository of nodes and relationships. This repository is used by a prototypical extension of Winery to complete incomplete topologies. The developer can follow the completion of the topology step by step and make adjustments if necessary. The approach of Panarello et al. [94] also tracks the completion of application topologies. This involves distributing the components of an application through the offerings of the providers of a federation. A Decision-Making component selects the appropriate offerings and a Topology Completion Manager completes the incomplete topology accordingly. Also Brogi et al. [29] present an approach for a related purpose. It uses the prototype DrACO, which receives information about suitable cloud offerings through crawling the network and storing their TOSCA representation in a repository. This representation can be used by the application developer to complete the topology. Similarly, the OpenTOSCA Injector presented by Saatkamp et al. [104] takes incomplete application topologies and chooses suitable node templates which are stored in the providers Repository to complete the topology. This approach is based on the requirements that are defined for a component and the capabilities offered by the components in the repository to meet them.

Soldani et al. [110] present TOSCAMart, an approach to reuse proven topologies in new environments. TOSCAMart is based on a repository of existing topologies provided to an application developer for the development of a new composite application. The developer defines a node in the new topology that describes the requirements for the fragment being inserted. TOSCAMart then selects a suitable solution for these requirements from the repository. Matching between a given node and the fragments from the repository is based on the definitions for *exact* and *plug-in* matching mentioned in 5.3.2.

TOSCA Node Types point to artifacts implementing installation and deployment operations. Selecting an artifact that meets the requirements of the user requires technology-specific expertise. Combining artifacts to deploy complex applications can be difficult and error-prone. To solve this problem, Endres et al. [55] present a method that crawls open-source repositories and transforms found artifacts into technology-agnostic topology models. These topology models facilitate the understanding of the functionality of the artifacts and can be combined to model the deployment of complex applications.

5.3.2 Matching between service templates and node types

Brogi and Soldani [31–33] describe an approach that involves matching between individual Node Types and Service Templates. 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. For this purpose, the authors define *(i) exact matching* between Node Type and Service Template and propose further definitions for *(ii) plug-in*, *(iii) flexible* and *(iv) white-box matching*. *Exact matching* between Service Template and Node Type exists if the requirements and offered services are perfectly compatible with each other. In the case of *plug-in matching*, the requirements of the Service Template are weaker and the capabilities offered higher than those of the Node Type. The other two types, *flexible* and *white-box matching*, involve bridging non-essential, syntactic differences between feature names and missing Service Template requirements or capabilities. Service Templates with the other three properties can be adapted to create a new template that fits exactly a given node type.

5.3.3 Combining declarative and imperative processing

Breitenbücher et al. [17,18] propose an approach that combines declarative and imperative to hybrid processing. Declarative processing may not be appropriate for larger applications because only general components and predefined operations known by the runtime can be executed. The disadvantage of imperative processing is that creating management plans is time-consuming, expensive, and error-prone. To overcome their respective drawbacks, the two kinds of processing are combined to a hybrid approach. The defined application topologies are interpreted and finally the associated management plans are generated. These can be finally adjusted by the developer.

Calcaterra et al. [42] present a similar approach, also based on interpreting a topology and providing the appropriate management plan. To do this, a converter translates the YAML model into a BPMN model, which is used to deploy the application. In addition, the approach still provides the ability for cloud providers to define and deliver their offerings so that users can choose the offering that best suits their needs. However, the presented prototype of this application only translates some of these functions.

5.3.4 Verification of Cloud Orchestration

Yoshida et al. [131] describe an approach to the formal verification of TOSCA topologies that can be used to test the achievement of a particular target state in declarative processing of the TOSCA model. 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.

Also Chareonsuk and Vatanawood [49] deal with the formal verification of cloud orchestration design. Their approach, unlike that of Yoshida et al., considers imperative processing. For this purpose, a CSAR file containing the TOSCA topology and associated BPEL processes is converted into a formal model, which is then processed by a model checker. The WSDL description of the used web services is used to take into account the signatures of the individual method invocations for verification. The authors use the description of safety properties as linear temporal logic formulae for model checking.

The approach of Tsigkanos and Kehrer [115] is about defining patterns and anti-patterns and finally checking their presence or absence in the topology of a TOSCA service template, so that quality aspects can be proven.

Brogi et al. [30,39] present the latest approach to validate a TOSCA application topology. This approach describes the topology of an application as a directed graph, where the nodes represent the components and the edges represent the relationships between them. In order to validate the topology, it is checked whether the requirements of a source node are met by a target node or its capabilities. For this check, the requirements, relationships, and capabilities are translated individually into formal conditions and then combined to define the notion of validity. Finally, Sommelier, an open-source prototype, is presented for the validation of application topologies.

5.4 Relation to other solutions

This category includes papers discussing (i) the comparison between TOSCA and other methods for describing cloud systems, and (ii) the automatic translation between TOSCA and another modeling language.

5.4.1 Comparison between TOSCA and other methods for describing cloud systems

A comparison between TOSCA and the Heat Orchestration Template (HOT) is provided by Di Martino et al. [52]. 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 does not support workflow definition, which is possible with TOSCA: HOT is purely declarative, whereas TOSCA also supports imperative processing using management plans. Also, similarities are shown, for example, both provide a catalog of nodes and resources that can be composed to applications.

For the translation of TOSCA into the HOT format, OpenStack has developed the project HEAT-Translator. This takes an input in a format not compatible with HEAT and translates it into HOT format. Among others, Tricomi et al. [114] present a deployment approach using the HEAT-Translator.

5.4.2 Automatic translation between TOSCA and another modeling language

Yongsiriwit et al. [130] address the interoperability between different standards for describing cloud resources: TOSCA, Open Cloud Computing Interface (OCCI) and Cloud Infrastructure Management Interface (CIMI). 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.

5.5 Usage of TOSCA

This category consists of approaches that use the existing TOSCA notation. Some papers present a case study of the use of TOSCA. In others, TOSCA is not the main focus, but only a means to achieve some goal. Some further papers discuss integrating TOSCA with other concepts to combine their benefits.

Kostoska et al. [77] present a case study of the use of TOSCA for specifying the University Management System iKnow. This system offers professors and students a platform to exchange electronic information and provide electronic services. The paper describes how the components of this application are defined using TOSCA. Examples show excerpts from the XML specification of node and relationship types, as well as artifacts that can be used for deployment and installation. Also, the challenges of using TOSCA for the specification of this application are mentioned.

A different domain for using TOSCA is the specification of Internet of Things (IoT) applications. This area is addressed by various authors. Li et al. [81] show how TOSCA can be used in the context of an IoT application, namely an Air Handling Unit (AHU) that controls the condition and circulation of air in modern buildings. The authors give examples of how to define the components of the application and describe their experience using TOSCA for this use case. Also Da Silva et al. [50] demonstrate the feasibility of defining IoT applications using TOSCA, in the context of different technologies. In another paper, Da Silva et al. [108] address the multitude of sensor data produced in IoT scenarios. The authors describe how Complex Event Processing Systems can be deployed using TOSCA to process the incoming data and efficiently use network resources.

Wettinger et al. [126, 127] show the use of TOSCA in the context of DevOps. A core principle of DevOps is to automate the deployment process to enable continuous software delivery. The authors address the challenge of combining different DevOps artifacts. For this purpose, a framework is presented to search public repositories for such artifacts, and to convert them to TOSCA format to ensure a uniform representation and to enable their combination. Specifically, the transformation of Chef cookbooks and Juju charms is described. In another paper, Wettinger et al. [123] integrate Configuration Management with Model-Driven Cloud Management in the context of DevOps. Model-Driven Cloud Management provides an overview of the structure of a complex application and supports developers in handling necessary infrastructure changes,

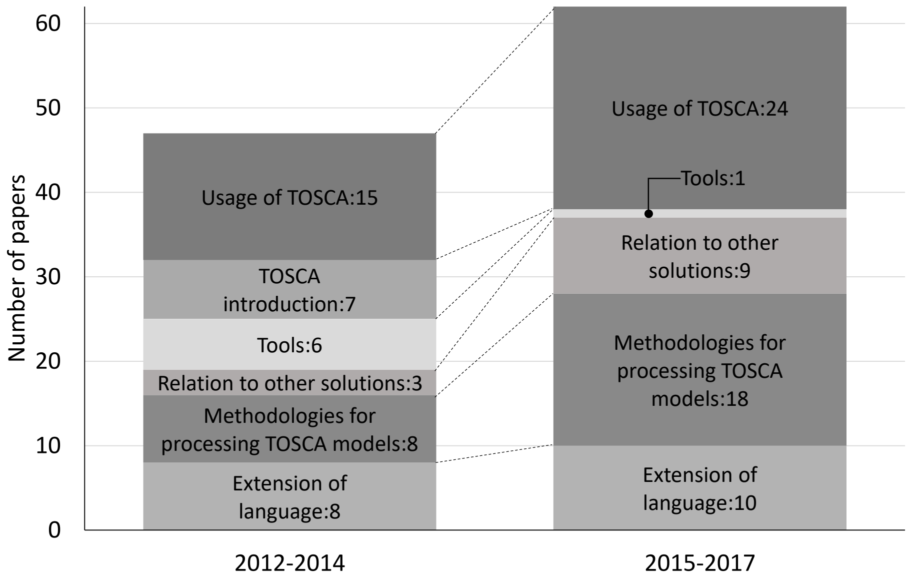

Fig. 3 Evolution of the number of papers belonging to the categories of the taxonomy

abstracting from lower-level actions (like installation or configuration of components), which are provided by Configuration Management. The paper deals with the integration of TOSCA as a vehicle for Model-Driven Cloud Management and Chef as a Configuration Management tool. Integration aims at defining a holistic service model and ensuring its portability.

Cardoso et al. [45] integrate TOSCA with the service description language USDL. USDL is used to describe service offerings of an application, so that service discovery and selection can be simplified. TOSCA is used to deploy and manage the components of an application. The paper describes how the two languages can be combined so that parts of the lifecycle of an application can be automated.

Sampaio et al. [106] define an approach to performing performance tests for cloud application topologies. For the tests, different variants of a topology are specified with the help of TOSCA. After being deployed, these variants are evaluated on the basis of performance parameters chosen by the user.

6 Discussion

After the detailed results of the literature review (RQ1), we now examine the lessons learned and present some further insight.

6.1 Temporal evolution of the addressed topics

To answer RQ2, we analyzed the distribution of papers among the categories of the taxonomy, also taking into account the evolution along the time axis. Fig. 3 shows the distribution of papers among the categories for the two halves of the investigated time period (first respectively second column of the diagram). We considered the years 2012 to 2017 for this figure, and divided them into two periods of equal length. It can be observed that the two biggest categories are "Usage of TOSCA" and "Methodologies for processing TOSCA models." This holds for both halves of the period. Also, both categories grew significantly over time. The biggest relative growth, however, was produced by the "Relation to other solutions" category, which grew from 3 to 9. Also the "Extension of language" category grew considerably. A possible explanation is that such papers address rather specific and advanced topics that gain more relevance when the given field of study matures.

There were also two categories that diminished. "TOSCA introduction" papers were only published in the first half of the period when TOSCA was new. More interestingly, also the number of "Tools" papers decreased considerably. A possible explanation for this phenomenon could be that the basic tools are already available to handle TOSCA models, which made it more interesting for the research community to focus on methodology issues. Tools are still produced, but rather to illustrate an approach than as core contributions.

6.2 Types of processing

In Section 2 we described the two types of processing for the deployment of composite applications: declarative and imperative processing. Section 5.3.3 showed that several papers considered the combination of declarative and imperative to hybrid processing. To further elaborate on RQ2, Fig. 4 shows the evolution of the number of approaches belonging to the different types of processing. Beyond imperative, declarative, and hybrid processing, there are two further categories called "both" respectively "undefined". "Both" refers to approaches that use both imperative and declarative processing, i.e., both variants of the processing are described in a paper. "Undefined" includes papers that do not describe exactly what type of processing is used in an approach. The figure shows the percentage of the processing types for each year.

Looking at the development, it is clear that the proportion of imperative processing has decreased significantly over the years. At the beginning of the analyzed period, this was 83%, in 2015 only 64% and in 2018 finally only 13%. In contrast, the proportion of declarative processing has increased significantly. In 2012, no approach was described in this category, while in 2015 the share increased to 14%. In 2018, this category contains 60% of the approaches. The remaining three categories account for only a small fraction of the total number of papers. The increase in declarative processing can be explained, on the one hand, by the error-proneness encountered in connection with the creation

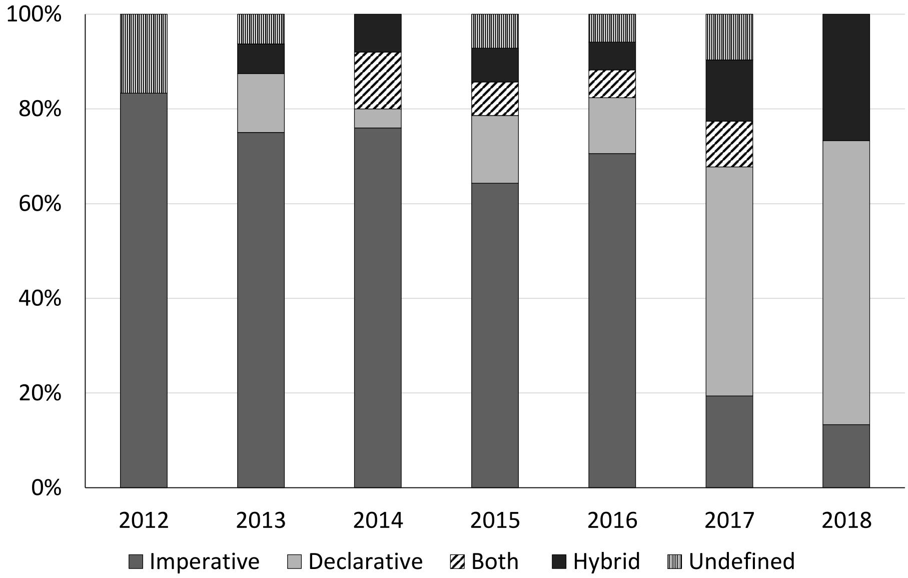

Fig. 4 Evolution of the number of approaches using different types of processing

of management plans [17] and, on the other hand, by the experience that applications can apparently be provided just by using the topology [30,36].

6.3 Topics for future research

In this sub-section, we address RQ3. Qualitatively, the findings of our survey show the *versatility* of TOSCA: its use in many 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. Indeed, most surveyed papers use some additional types (and this is also planned for future research [88,50]). Partial topologies make it possible to define an application without the technical details of the physical components and to optimize the deployment (which is also planned to be used in the future by several authors [94,29,108]). However, this versatility also poses the risk of the proliferation of different and incompatible TOSCA dialects. Hence we expect that *interoperability* will play an increasingly important role.

Furthermore, the survey results show that the following topics received limited attention so far and represent important topics for future research:

- Given the huge 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 [106,130,102]). Also, TOSCA support for other related topics like *privacy* and *data protection* needs to be investigated [85].

- The topic of *verification and validation* (V&V) is also addressed by a few papers. Since V&V activities make up a significant part of the IT budget of an organization, we expect to see more work on how TOSCA can be used to make V&V more efficient and effective.
- 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 [6,86].
- TOSCA has been shown to be useful in areas such as IoT and DevOps. We expect to see TOSCA being applied to new domains like *fog and edge computing* or *mobile computing*. TOSCA could be used as a framework for integrating different technologies, e.g. to support mobile cloud computing.
- Support for *hierarchical* topologies as well as hierarchical orchestration approaches seems to be a promising way to handle very large cloud systems and also to reuse artifacts between models of different systems. Hence we expect more research on supporting hierarchization of TOSCA models.

6.4 Threats to validity

We now discuss threats to the validity of our literature survey and the drawn conclusions. The *internal validity* of our study is influenced by the classification of the papers in the taxonomy; a bias in the classification is thus a threat to internal validity. To avoid subjective bias, we discussed the classification of the papers between us, and also adjusted the taxonomy several times so as to find the best fit. The iterative development of the taxonomy helped to ensure that it is suitable for an analysis of the current state of research.

A threat to the *external validity* of our study is that some important references may not have been found. To minimize this risk, we combined various search methods (keyword search with multiple keywords in multiple databases, search along references, search along key authors) to find all important TOSCA-relevant papers.

The *conclusion validity* of our discussion concerning future research topics is influenced by our subjective view on the TOSCA-related literature. To minimize this subjective influence, we focus in Section 6.3 on the topics that are named as future research topics in the literature.

A threat to the *construct validity* in our analysis of temporal changes lies in the comparison of different periods. To mitigate the threat of comparing periods of different length, we do not consider the year 2018 in our consideration of the temporal evolution of the addressed topics in Section 6.1. Similarly, in Section 6.2, only entire years are compared.

The above threats were further mitigated by presenting a preliminary version of the study in a relevant conference, and incorporating the feedback from the reviewers and discussions at the conference in this final version.

7 Conclusions

This paper presented a systematic literature review on TOSCA. The identified papers addressed a variety of topics, which we grouped into six categories (some further subdivided into several sub-categories). The categories with the highest number of papers were: (1) papers reporting on the use of TOSCA, (2) methodologies for processing TOSCA models, and (3) extensions of TOSCA.

Our survey showed the versatility of TOSCA, but also discovered areas that are hardly explored so far and may represent targets for future research. Examples include security and privacy aspects, as well as verification and validation in connection with TOSCA models. Hence we expect TOSCA to remain a relevant topic for future research in cloud computing and beyond.

Acknowledgements This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant no. 731678 (RestAssured).

References

1. Alonso, J., Orue-Echevarria, L., Escalante, M.: Cloud compliant applications: A reference framework to assess the maturity of software applications with respect to cloud. In: IEEE 9th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Environments. pp. 41–45 (2015)
2. Antonenko, V., Smeliansky, R., Ermilov, A., Plakunov, A., Pinaeva, N., Romanov, A.: C2: general purpose cloud platform with NFV life-cycle management. In: 9th IEEE International Conference on Cloud Computing Technology and Science. pp. 353–356 (2017)
3. Antonenko, V., Smeliansky, R., Ermilov, A., Romanov, A., Pinaeva, N., Plakunov, A.: Cloud infrastructure for researchers basing on NFV management and orchestration. In: XXVI International Symposium on Nuclear Electronics & Computing. pp. 75–81 (2017)
4. Antoniades, D., Loulloudes, N., Foudoulis, A., Sophokleous, C., Trihinas, D., Pallis, G., Dikaiakos, M., Kornmayer, H.: Enabling cloud application portability. In: 8th IEEE/ACM International Conference on Utility and Cloud Computing. pp. 354–360 (2015)
5. Artac, M., Borovsak, T., Nitto, E.D., Guerriero, M., Perez-Palacin, D., Tamburri, D.A.: Infrastructure-as-code for data-intensive architectures: A model-driven development approach. In: IEEE 15th International Conference on Software Architecture. pp. 156–165 (2018)
6. Bartók, D., Mann, Z.Á.: A branch-and-bound approach to virtual machine placement. In: Proceedings of the 3rd HPI Cloud Symposium "Operating the Cloud". pp. 49–63 (2015)
7. Bellendorf, J., Mann, Z.Á.: Cloud topology and orchestration using TOSCA: A systematic literature review. In: 7th European Conference on Service-Oriented and Cloud Computing. pp. 207–215 (2018)
8. Bergmayr, A., Breitenbücher, U., Ferry, N., Rossini, A., Solberg, A., Wimmer, M., Kappel, G., Leymann, F.: A systematic review of cloud modeling languages. ACM Comput. Surv. **51**(1), 22:1–22:38 (2018)
9. Bergmayr, A., Breitenbücher, U., Kopp, O., Wimmer, M., Kappel, G., Leymann, F.: From architecture modeling to application provisioning for the cloud by combining UML and TOSCA. In: 6th International Conference on Cloud Computing and Services Science. pp. 97–108 (2016)

10. Bergmayr, A., Troya, J., Neubauer, P., Wimmer, M., Kappel, G.: UML-based cloud application modeling with libraries, profiles, and templates. In: 2nd International Workshop on Model-Driven Engineering on and for the Cloud. pp. 56–65 (2014)
11. Binz, T., Breitenbücher, U., Haupt, F., Kopp, O., Leymann, F., Nowak, A., Wagner, S.: OpenTOSCA – a runtime for TOSCA-based cloud applications. In: 11th International Conference on Service-Oriented Computing. pp. 692–695 (2013)
12. Binz, T., Breitenbücher, U., Kopp, O., Leymann, F.: TOSCA: portable automated deployment and management of cloud applications. In: Advanced Web Services, pp. 527–549. Springer (2014)
13. Binz, T., Breitenbücher, U., Kopp, O., Leymann, F., Weiß, A.: Improve resource-sharing through functionality-preserving merge of cloud application topologies. In: 3rd International Conference on Cloud Computing and Services Science. pp. 96–103 (2013)
14. Binz, T., Breiter, G., Leyman, F., Spatzier, T.: Portable cloud services using TOSCA. IEEE Internet Computing **16**(3), 80–85 (2012)
15. Bonchi, F., Brogi, A., Canciani, A., Soldani, J.: Behaviour-aware matching of cloud applications. In: 10th International Symposium on Theoretical Aspects of Software Engineering. pp. 117–124 (2016)
16. Bonchi, F., Brogi, A., Canciani, A., Soldani, J.: Simulation-based matching of cloud applications. Science of Computer Programming **162**, 110–131 (2018)
17. Breitenbücher, U., Binz, T., Képes, K., Kopp, O., Leymann, F., Wettinger, J.: Combining declarative and imperative cloud application provisioning based on TOSCA. In: 2014 IEEE International Conference on Cloud Engineering. pp. 87–96 (2014)
18. 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: 5th International Conference on Cloud Computing and Services Science. pp. 239–262 (2015)
19. Breitenbücher, U., Binz, T., Kopp, O., Leymann, F.: Vinothek – A self-service portal for TOSCA. In: 6th Central-European Workshop on Services and their Composition. pp. 69–72 (2014)
20. 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. pp. 416–424 (2012)
21. Breitenbücher, U., Binz, T., Kopp, O., Leymann, F.: Pattern-based runtime management of composite cloud applications. In: 3rd International Conference on Cloud Computing and Services Science. pp. 475–482 (2013)
22. Breitenbücher, U., Binz, T., Kopp, O., Leymann, F., Wieland, M.: Policy-aware provisioning of cloud applications. In: Proceedings of the Seventh International Conference on Emerging Security Information, Systems and Technologies. pp. 86–95 (2013)
23. Breiter, G., Behrendt, M., Gupta, M., Moser, S., Schulze, R., Sippli, I., Spatzier, T.: Software defined environments based on TOSCA in IBM cloud implementations. IBM Journal of Research and Development **58**(2/3) (2014)
24. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software **80**(4), 571–583 (2007)
25. Brogi, A., Canciani, A., Soldani, J.: Modelling and analysing cloud application management. In: 4th European Conference on Service Oriented and Cloud Computing. pp. 19–33 (2015)
26. Brogi, A., Canciani, A., Soldani, J., Wang, P.: Modelling the behaviour of management operations in cloud-based applications. In: International Workshop on Petri Nets and Software Engineering. pp. 191–205 (2015)
27. Brogi, A., Canciani, A., Soldani, J., Wang, P.: A Petri net-based approach to model and analyze the management of cloud applications. Transactions on Petri Nets and Other Models of Concurrency **11**, 28–48 (2016)
28. Brogi, A., Carrasco, J., Cubo, J., Di Nitto, E., Durán, F., Fazzolari, M., Ibrahim, A., Pimentel, E., Soldani, J., Wang, P., D'Andria, F.: Adaptive management of applications across multiple clouds: The SeaClouds approach. CLEI Electronic Journal **18**(1) (2015)
29. Brogi, A., Cifariello, P., Soldani, J.: DrACO: Discovering available cloud offerings. Computer Science – Research and Development **32**(3-4), 269–279 (2017)

30. Brogi, A., Di Tommaso, A., Soldani, J.: Validating TOSCA application topologies. In: 5th International Conference on Model-Driven Engineering and Software Development. pp. 667–678 (2017)
31. Brogi, A., Soldani, J.: Matching cloud services with TOSCA. In: Advances in Service-Oriented and Cloud Computing – Workshops of ESOCC 2013. pp. 218–232 (2013)
32. Brogi, A., Soldani, J.: Reusing cloud-based services with TOSCA. In: Informatik 2014. pp. 235–246 (2014)
33. Brogi, A., Soldani, J.: Finding available services in TOSCA-compliant clouds. Science of Computer Programming **115-116**, 177–198 (2016)
34. Brogi, A., Soldani, J., Wang, P.: TOSCA in a nutshell: Promises and perspectives. In: 3rd European Conference on Service-Oriented and Cloud Computing. pp. 171–186 (2014)
35. Brogi, A., Neri, D., Rinaldi, L., Soldani, J.: From (incomplete) TOSCA specifications to running applications, with Docker. In: Software Engineering and Formal Methods – SEFM 2017 Collocated Workshops. pp. 491–506 (2017)
36. Brogi, A., Neri, D., Rinaldi, L., Soldani, J.: Orchestrating incomplete TOSCA applications with Docker. Science of Computer Programming **166**, 194–213 (2018)
37. Brogi, A., Rinaldi, L., Soldani, J.: TosKer: Orchestrating applications with TOSCA and Docker. In: Advances in Service-Oriented and Cloud Computing – Workshops of ESOCC 2017. pp. 130–144 (2017)
38. Brogi, A., Rinaldi, L., Soldani, J.: Tosker: A synergy between TOSCA and docker for orchestrating multicomponent applications. Softw., Pract. Exper. **48**(11), 2061–2079 (2018)
39. Brogi, A., Tommaso, A.D., Soldani, J.: Sommelier: A tool for validating TOSCA application topologies. In: 5th International Conference on Model-Driven Engineering and Software Development. pp. 1–22 (2017)
40. Brunelière, H., Alshara, Z., Alvares, F., Lejeune, J., Ledoux, T.: A model-based architecture for autonomic and heterogeneous cloud systems. In: 8th International Conference on Cloud Computing and Services Science. pp. 201–212 (2018)
41. Caballer, M., Zala, S., García, Á.L., Moltó, G., Fernández, P.O., Velten, M.: Orchestrating complex application architectures in heterogeneous clouds. Journal of Grid Computing **16**(1), 3–18 (2017)
42. Calcaterra, D., Cartelli, V., Di Modica, G., Tomarchio, O.: Combining TOSCA and BPMN to enable automated cloud service provisioning. In: 7th International Conference on Cloud Computing and Services Science. pp. 159–168 (2017)
43. Calcaterra, D., Cartelli, V., Modica, G.D., Tomarchio, O.: A framework for the orchestration and provision of cloud services based on TOSCA and BPMN. In: 7th International Conference on Cloud Computing and Service Science. pp. 262–285 (2017)
44. Calcaterra, D., Cartelli, V., Modica, G.D., Tomarchio, O.: Exploiting BPMN features to design a fault-aware TOSCA orchestrator. In: 8th International Conference on Cloud Computing and Services Science. pp. 533–540 (2018)
45. Cardoso, J., Binz, T., Breitenbücher, U., Kopp, O., Leymann, F.: Cloud computing automation: Integrating USDL and TOSCA. In: 25th International Conference on Advanced Information Systems Engineering. pp. 1–16 (2013)
46. Carrasco, J., Cubo, J., Durn, F., Pimentel, E.: Bidimensional cross-cloud management with TOSCA and Brooklyn. In: 9th IEEE International Conference on Cloud Computing. pp. 951–955 (2016)
47. Carrasco, J., Cubo, J., Pimentel, E.: Towards a flexible deployment of multi-cloud applications based on TOSCA and CAMP. In: Advances in Service-Oriented and Cloud Computing – Workshops of ESOCC 2014. pp. 278–286 (2014)
48. Chappell, C.: Deploying virtual network functions: The complementary roles of TOSCA & NETCONF/YANG. Heavy Reading White Paper, http://www.tail-f.com/wordpress/wp-content/uploads/2015/02/HR-Cisco-ALU-TOSCA-YANG-WP-2-17-15.pdf (2015)
49. Chareonsuk, W., Vatanawood, W.: Formal verification of cloud orchestration design with TOSCA and BPEL. In: 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (2016)

50. 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: 7th International Conference on Cloud Computing and Services Science. pp. 330–339 (2017)
51. Demont, C., Breitenbücher, U., Kopp, O., Leymann, F., Wettinger, J.: Towards integrating TOSCA and ITIL. In: 5th Central-European Workshop on Services and their Composition. pp. 28–31 (2013)
52. Di Martino, B., Cretella, G., Esposito, A.: Defining cloud services workflow: A comparison between TOSCA and OpenStack Hot. In: 9th International Conference on Complex, Intelligent, and Software Intensive Systems. pp. 541–546 (2015)
53. Di Martino, B., Cretella, G., Esposito, A.: Research initiatives and emerging standards. In: Cloud Portability and Interoperability: Issues and Current Trends, pp. 93–121. Springer International Publishing, Cham (2015)
54. Di Martino, B., Cretella, G., Esposito, A.: A comparison between TOSCA and OpenStack HOT through cloud patterns composition. International Journal of Grid and Utility Computing **8**(4), 299–311 (2017)
55. Endres, C., Breitenbücher, U., Leymann, F., Wettinger, J.: Anything to topology – A method and system architecture to topologize technology-specific application deployment artifacts. In: 7th International Conference on Cloud Computing and Services Science. pp. 180–190 (2017)
56. Glaser, F.: Domain model optimized deployment and execution of cloud applications with TOSCA. In: 9th International Conference on System Analysis and Modeling. pp. 68–83 (2016)
57. Glaser, F., Erbel, J., Grabowski, J.: Model driven cloud orchestration by combining TOSCA and OCCI. In: 7th International Conference on Cloud Computing and Services Science. pp. 644–650 (2017)
58. Görlach, K., Leymann, F.: Dynamic service provisioning for the cloud. In: 2012 IEEE Ninth International Conference on Services Computing. pp. 555–561 (2012)
59. Guerriero, M., Tajfar, S., Tamburri, D.A., Nitto, E.D.: Towards a model-driven design tool for big data architectures. In: Proceedings of the 2nd International Workshop on BIG Data Software Engineering. pp. 37–43 (2016)
60. Gusev, M., Kostoska, M., Ristov, S.: Cloud P-TOSCA porting of N-tier applications. In: 22nd Telecommunications Forum. pp. 935–938 (2014)
61. Han, R., Ghanem, M., Guo, Y.: Elastic-TOSCA: Supporting elasticity of cloud application in TOSCA. In: CLOUD COMPUTING 2013 – 4th International Conference on Cloud Computing, GRIDs, and Virtualization. pp. 93–100 (2013)
62. Harrer, S., Lenhard, J., Wirtz, G., Van Lessen, T.: Towards uniform BPEL engine management in the cloud. In: Informatik 2014. pp. 259–270 (2014)
63. Haupt, F., Leymann, F., Nowak, A., Wagner, S.: Lego4TOSCA: Composable building blocks for cloud applications. In: IEEE 7th International Conference on Cloud Computing. pp. 160–167 (2014)
64. Healy, P., Meyer, S., Morrison, J., Lynn, T., Paya, A., Marinescu, D.: Bid-centric cloud service provisioning. In: IEEE 13th International Symposium on Parallel and Distributed Computing. pp. 73–81 (2014)
65. Hirmer, P., Breitenbücher, U., Binz, T., Leymann, F.: Automatic topology completion of TOSCA-based cloud applications. In: Informatik 2014. pp. 247–258 (2014)
66. Hirmer, P., Mitschang, B.: TOSCA4Mashups: enhanced method for on-demand data mashup provisioning. Computer Science – Research and Development **32**(3-4), 291–300 (2017)
67. Hung, Y.M., Chien, S.C., Chunghwa, Y.Y.: Orchestration of NFV virtual applications based on TOSCA data models. In: 19th Asia-Pacific Network Operations and Management Symposium. pp. 219–222 (2017)
68. Ivanovska, B., Ristov, S., Kostoska, M., Gusev, M.: Using the P-TOSCA model for energy efficient cloud. In: 38th International Convention on Information and Communication Technology, Electronics and Microelectronics. pp. 245–249 (2015)
69. Katsaros, G., Menzel, M., Lenk, A., Rake-Revelant, J., Skipp, R., Eberhardt, J.: Cloud application portability with TOSCA, Chef and OpenStack. In: 2014 IEEE International Conference on Cloud Engineering. pp. 295–302 (2014)

70. Kehrer, S., Blochinger, W.: AUTOGENIC: automated generation of self-configuring microservices. In: 8th International Conference on Cloud Computing and Services Science. pp. 35–46 (2018)
71. Kehrer, S., Blochinger, W.: TOSCA-based container orchestration on Mesos. Computer Science – Research and Development **33**(3-4), 305–316 (2018)
72. Képes, K., Breitenbücher, U., Leymann, F.: The SePaDe system: Packaging entire XaaS layers for automatically deploying and managing applications. In: 7th International Conference on Cloud Computing and Services Science. pp. 626–635 (2017)
73. Komarek, A., Pavlik, J., Sobeslav, V.: Hybrid system orchestration with TOSCA and Salt. Journal of Engineering and Applied Sciences **12**(9), 2396–2401 (2017)
74. Kopp, O., Binz, T., Breitenbücher, U., Leymann, F.: BPMN4TOSCA: A domain-specific language to model management plans for composite applications. In: Business Process Model and Notation – 4th International Workshop. pp. 38–52 (2012)
75. Kopp, O., Binz, T., Breitenbücher, U., Leymann, F.: Winery – a modeling tool for TOSCA-based cloud applications. In: 11th International Conference on Service-Oriented Computing. pp. 700–704 (2013)
76. Kopp, O., Binz, T., Breitenbücher, U., Leymann, F., Michelbach, T.: A domain-specific modeling tool to model management plans for composite applications. In: 7th Central European Workshop on Services and their Composition. pp. 51–54 (2015)
77. Kostoska, M., Chorbev, I., Gusev, M.: Creating portable TOSCA archive for iKnow University Management System. In: Federated Conference on Computer Science and Information Systems. pp. 761–768 (2014)
78. Kostoska, M., Donevski, A., Gusev, M., Ristov, S.: Porting an n-tier application on cloud using P-TOSCA: A case study. In: 38th International Convention on Information and Communication Technology, Electronics and Microelectronics. pp. 281–285 (2015)
79. Kostoska, M., Gusev, M., Ristov, S.: A new cloud services portability platform. In: 24th DAAAM International Symposium on Intelligent Manufacturing and Automation. vol. 69, pp. 1268–1275 (2014)
80. Leymann, F.: Linked compute units and linked experiments: Using topology and orchestration technology for flexible support of scientific applications. In: Software Service and Application Engineering. pp. 71–80 (2012)
81. Li, F., Vögler, M., Claeßens, M., Dustdar, S.: Towards automated IoT application deployment by a cloud-based approach. In: IEEE 6th International Conference on Service-Oriented Computing and Applications. pp. 61–68 (2013)
82. Lipton, P.: Escaping vendor lock-in with TOSCA, an emerging cloud standard for portability. In: CA Technology Exchange 4. pp. 49–55 (2013)
83. Lipton, P., Palma, D., Rutkowski, M., Tamburri, D.A.: TOSCA solves big problems in the cloud and beyond! IEEE Cloud Computing **5**(2), 37–47 (2018)
84. Mann, Z.Á.: Resource optimization across the cloud stack. IEEE Transactions on Parallel and Distributed Systems **29**(1), 169–182 (2018)
85. Mann, Z.Á., Metzger, A.: Optimized cloud deployment of multi-tenant software considering data protection concerns. In: 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. pp. 609–618 (2017)
86. Mann, Z.Á., Szabó, M.: Which is the best algorithm for virtual machine placement optimization? Concurrency and Computation: Practice and Experience **29**(10), e4083 (2017)
87. Modica, G.D., Tomarchio, O., Calcaterra, D., Cartelli, V.: Implementation of a fault aware cloud service provisioning framework. In: 6th IEEE International Conference on Future Internet of Things and Cloud. pp. 9–16 (2018)
88. Molto, G., Caballer, M., Perez, A., Alfonso, C., Blanquer, I.: Coherent application delivery on hybrid distributed computing infrastructures of virtual machines and Docker containers. In: 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing. pp. 486–490 (2017)
89. Nowak, A., Binz, T., Fehling, C., Kopp, O., Leymann, F., Wagner, S.: Pattern-driven green adaptation of process-based applications and their runtime infrastructure. Computing **94**(6), 463–487 (2012)
90. OASIS: Web Services Business Process Execution Language Version 2.0 (Apr 2007), OASIS Standard

91. OASIS: Topology and Orchestration Specification for Cloud Applications Version 1.0. (Nov 2013), OASIS Standard, http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html
92. OMG: Business Process Model and Notation (BPMN) Version 2.0. (Jan 2011), OMG Document Number: formal/2011-01-03
93. Palma, D., Rutkowski, M., Spatzier, T.: TOSCA Simple Profile in YAML Version 1.0 (Dec 2016), OASIS Standard, http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/TOSCA-Simple-Profile-YAML-v1.0.html
94. Panarello, A., Breitenbücher, U., Leymann, F., Puliafito, A., Zimmermann, M.: Automating the deployment of multi-cloud applications in federated cloud environments. In: 10th EAI International Conference on Performance Evaluation Methodologies and Tools. pp. 194–201 (2016)
95. Qanbari, S., Li, F., Dustdar, S.: Toward portable cloud manufacturing services. IEEE Internet Computing **18**(6), 77–80 (2014)
96. Qanbari, S., Sebto, V., Dustdar, S.: Cloud Resources-Events-Agents model: Towards TOSCA-based applications. In: 3rd European Conference on Service-Oriented and Cloud Computing. pp. 160–170 (2014)
97. Qanbari, S., Zadeh, S., Vedaei, S., Dustdar, S.: Cloudman: A platform for portable cloud manufacturing services. In: IEEE International Conference on Big Data. pp. 1006–1014 (2014)
98. Qasha, R., Cala, J., Watson, P.: Towards automated workflow deployment in the cloud using TOSCA. In: 8th IEEE International Conference on Cloud Computing. pp. 1037–1040 (2015)
99. Qasha, R., Cala, J., Watson, P.: Dynamic deployment of scientific workflows in the cloud using container virtualization. In: 8th IEEE International Conference on Cloud Computing Technology and Science. pp. 269–276 (2016)
100. Razaq, A., Tianfield, H., Barrie, P., Yue, H.: Service broker based on cloud service description language. In: 15th International Symposium on Parallel and Distributed Computing. pp. 196–201 (2016)
101. Russo, E., Costa, G., Armando, A.: Scenario design and validation for next generation cyber ranges. In: 17th IEEE International Symposium on Network Computing and Applications. pp. 1–4 (2018)
102. Saatkamp, K., Breitenbücher, U., Kopp, O., Leymann, F.: Topology splitting and matching for multi-cloud deployments. In: 7th International Conference on Cloud Computing and Services Science. pp. 247–258 (2017)
103. Saatkamp, K., Breitenbücher, U., Leymann, F., Wurster, M.: Generic driver injection for automated IoT application deployments. In: 19th International Conference on Information Integration and Web-based Applications & Services. pp. 320–329 (2017)
104. Saatkamp, K., Breitenbücher, U., Képes, K., Leymann, F., Zimmermann, M.: OpenTOSCA Injector: Vertical and horizontal topology model injection. In: Service-Oriented Computing – ICSOC 2017 Workshops. pp. 379–383 (2017)
105. Salsano, S., Lombardo, F., Pisa, C., Greto, P., Blefari-Melazzi, N.: RDCL 3D, a model agnostic web framework for the design and composition of NFV services. In: 3rd IEEE Conference on Network Function Virtualization and Software Defined Networks. pp. 216–222 (2017)
106. Sampaio, A., Rolim, T., Mendona, N., Cunha, M.: An approach for evaluating cloud application topologies based on TOSCA. In: 9th IEEE International Conference on Cloud Computing. pp. 407–414 (2016)
107. Sebrechts, M., Johns, C., van Seghbroeck, G., Wauters, T., Volckaert, B., Turck, F.D.: Beyond generic lifecycles: Reusable modeling of custom-fit management workflows for cloud applications. In: 11th IEEE International Conference on Cloud Computing. pp. 326–333 (2018)
108. 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 **33**(3-4), 317–327 (2018)
109. Silva, G., Rose, L., Calinescu, R.: Cloud DSL: A language for supporting cloud portability by describing cloud entities. In: 2nd International Workshop on Model-Driven Engineering on and for the Cloud. pp. 36–45 (2014)

110. 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)
111. Steimle, F., Wieland, M., Mitschang, B., Wagner, S., Leymann, F.: Extended provisioning, security and analysis techniques for the ECHO health data management system. Computing **99**(2), 183–201 (2017)
112. Sundararajan, P., Durairajan, S.: Portable service management deployment over cloud platforms to support production workloads. In: IEEE International Conference on Cloud Computing in Emerging Markets (2013)
113. Sungur, C., Kopp, O., Leymann, F.: Supporting informal processes. In: 6th Central-European Workshop on Services and their Composition. pp. 49–56 (2014)
114. Tricomi, G., Panarello, A., Merlino, G., Longo, F., Bruneo, D., Puliafito, A.: Orchestrated multi-cloud application deployment in OpenStack with TOSCA. In: 2017 IEEE International Conference on Smart Computing (2017)
115. Tsigkanos, C., Kehrer, T.: On formalizing and identifying patterns in cloud workload specifications. In: 13th Working IEEE/IFIP Conference on Software Architecture. pp. 262–267 (2016)
116. Ulbricht, S., Amme, W., Heinze, T., Moser, S., Wehle, H.D.: Portable green cloud services. In: 4th International Conference on Cloud Computing and Services Science. pp. 53–59 (2014)
117. Vetter, A.: Detecting operator errors in cloud maintenance operations. In: 8th IEEE International Conference on Cloud Computing Technology and Science. pp. 639–644 (2016)
118. Vukojevic-Haupt, K., Karastoyanova, D., Leymann, F.: On-demand provisioning of infrastructure, middleware and services for simulation workflows. In: 2013 IEEE 6th International Conference on Service-Oriented Computing and Applications. pp. 91–98 (2013)
119. 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 (2013)
120. Waizenegger, T., Wieland, M., Binz, T., Breitenbücher, U., Leymann, F.: Towards a policy-framework for the deployment and management of cloud services. In: 7th International Conference on Emerging Security Information, Systems and Technologies. pp. 14–18 (2013)
121. Weerasiri, D., Barukh, M.C., Benatallah, B., Sheng, Q.Z., Ranjan, R.: A taxonomy and survey of cloud resource orchestration techniques. ACM Comput. Surv. **50**(2), 26:1–26:41 (2017)
122. Wendland, F., Banse, C.: Enhancing NFV orchestration with security policies. In: Proceedings of the 13th International Conference on Availability, Reliability and Security. pp. 45:1–45:6 (2018)
123. 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: 3rd International Conference on Cloud Computing and Services Science. pp. 437–446 (2013)
124. Wettinger, J., Binz, T., Breitenbücher, U., Kopp, O., Leymann, F.: Streamlining cloud management automation by unifying the invocation of scripts and services based on TOSCA. International Journal of Organizational and Collective Intelligence **4**(2), 45–63 (2014)
125. 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: 4th International Conference on Cloud Computing and Services Science. pp. 559–568 (2014)
126. Wettinger, J., Breitenbücher, U., Kopp, O., Leymann, F.: Streamlining DevOps automation for cloud applications using TOSCA as standardized metamodel. Future Generation Computer Systems **56**, 317–332 (2016)
127. Wettinger, J., Breitenbücher, U., Leymann, F.: Standards-based DevOps automation and integration using TOSCA. In: 7th IEEE/ACM International Conference on Utility and Cloud Computing. pp. 59–68 (2014)

128. Wurster, M., Breitenbücher, U., Falkenthal, M., Leymann, F.: Developing, deploying, and operating twelve-factor applications with TOSCA. In: 19th International Conference on Information Integration and Web-based Applications & Services. pp. 519–525. ACM (2017)
129. Wurster, M., Breitenbücher, U., Kopp, O., Leymann, F.: Modeling and automated execution of application deployment tests. In: 22nd IEEE International Enterprise Distributed Object Computing Conference, EDOC 2018, Stockholm, Sweden, October 16-19, 2018. pp. 171–180 (2018)
130. Yongsiriwit, K., Sellami, M., Gaaloul, W.: A semantic framework supporting cloud resource descriptions interoperability. In: 9th IEEE International Conference on Cloud Computing. pp. 585–592 (2016)
131. Yoshida, H., Ogata, K., Futatsugi, K.: Formalization and verification of declarative cloud orchestration. In: 17th International Conference on Formal Engineering Methods. pp. 33–49 (2015)
132. Zimmermann, M., Breitenbücher, U., Leymann, F.: A TOSCA-based programming model for interacting components of automatically deployed cloud and IoT applications. In: 19th International Conference on Enterprise Information Systems. pp. 121–131 (2017)
133. Zimmermann, M., Breitenbücher, U., Falkenthal, M., Leymann, F., Saatkamp, K.: Standards-based function shipping – how to use TOSCA for shipping and executing data analytics software in remote manufacturing environments. In: 21st IEEE International Enterprise Distributed Object Computing Conference. pp. 50–60 (2017)
134. Zimmermann, M., Breitenbücher, U., Leymann, F.: A method and programming model for developing interacting cloud applications based on the TOSCA standard. In: 19th International Conference on Enterprise Information Systems. pp. 265–290 (2017)