Institute of Architecture of Application Systems

Sustaining the Musical Competitions Database: a TOSCA-based Approach to Application Preservation in the Digital Humanities

Claes Neuefeind[1], Philip Schildkamp[1], Brigitte Mathiak[1], Aleksander Marčić[1], Frank Hentschel[1], Lukas Harzenetter[2], Uwe Breitenbücher[2], Johanna Barzen[2], Frank Leymann[2]

[1] University of Cologne, 50923 Cologne, Germany
[2] University of Stuttgart, 70569 Stuttgart, Germany

{c.neuefeind,philip.schildkamp,bmathiak,amarcic2,fhentsch}@uni-koeln.de
{lukas.harzenetter,uwe.breitenbuecher,johanna.barzen,frank.leymann}@iaas.uni-stuttgart.de

BibTeX:

```
@inproceedings {Neuefeind2019_MusicalCompetitionsDatabase,
    author    = {Claes Neuefeind and Philip Schildkamp and Brigitte Mathiak and
                 Aleksander Mar{\v{c}}i{'c} and Frank Hentschel and Lukas Harzenetter
                 and Johanna Barzen and Uwe Breitenb{\"u}cher and Frank Leymann},
    title     = {{Sustaining the Musical Competitions Database: A TOSCA-based
                  Approach to Application Preservation in the Digital Humanities}},
    booktitle = {Book of Abstracts, DH 2019},
    publisher = {ADHO},
    year      = 2019,
}
```


Universität Stuttgart
Germany

Sustaining the Musical Competitions Database: a TOSCA-based Approach to Application Preservation in the Digital Humanities

Claes Neuefeind, Philip Schildkamp, Brigitte Mathiak, Aleksander Marčić, Frank Hentschel, Lukas Harzenetter, Uwe Breitenbücher, Johanna Barzen, Frank Leymann

1 University of Cologne, 50923 Cologne, Germany
2 University of Stuttgart, 70569 Stuttgart, Germany

{c.neuefeind,philip.schildkamp,bmathiak,amarcic2,fhentsch}@uni-koeln.de
{lukas.harzenetter,uwe.breitenbuecher,johanna.barzen,frank.leymann}@iaas.uni-stuttgart.de

Keywords: Living Systems, Sustainability, Research Software, Musicology, TOSCA-Standard

1 Introduction

Within the Digital Humanities (DH), research applications such as databases, digital editions, interactive visualizations, and virtual research environments play a central role in securing and presenting research results [16]. Often, such *living systems* [15] are the actual bearers of information content, thus representing the added value of the scientific output [16]. However, within the DH a great number of smaller, highly heterogeneous software solutions are produced, which all are subject to the problem of *software aging* [14]. Against this background, institutions like the Data Center for the Humanities at the University of Cologne (DCH, http://dch.uni-koeln.de) face the challenge of preserving an unknown, potentially unlimited number of research software systems to assure their availability on a permanent basis. While there are well-established methods of preserving primary research data, e.g. in existing data repositories and archives, living systems are part of a constantly changing digital ecosystem and must regularly adapt to it, e.g. they need (security) updates. However, due to their steadily increasing number and their heterogeneity (both technologically and methodologically), permanent maintenance, support and provisioning of such living systems is a major technical, organizational, and thereby ultimately a financial challenge.
This contribution presents an approach to the preservation of web-based research applications in the DH, based on the *Topology Orchestration Specification for Cloud Applications* (TOSCA) [11, 12, 13]. TOSCA is an OASIS standard for modeling, provisioning, and managing cloud applications in a standardized and provider-independent way. In the following, we focus on an exemplary use case, namely the *Musical Competitions Database*, to describe the main concepts of our approach.

2 The Musical Competitions Database

The DFG-funded project *Musical Competitions between 1820 and 1870* is conducted by the Department of Musicology at the University of Cologne in cooperation with the Cologne Center for eHumanities (CCeH). The aim of the project is to gather comprehensive information about music related competitions from 1820 to 1870 [6]. Data is extracted by musicologists from music-related journals and stored as JSON files in a document-oriented database (CouchDB). Access to the data is given through a web application written in JavaScript/React (http://musical-competitions.uni-koeln.de). Further, ElasticSearch is used to provide advanced options for querying/filtering and analysis of the data. At the time of writing, the database features information on approximately 1300 musical competitions, 1000 corporations and 3100 persons related to those competitions. The Musical Competitions Database contains and presents a unique data set relevant to the musicology community. To allow for reproducibility in the sense of good

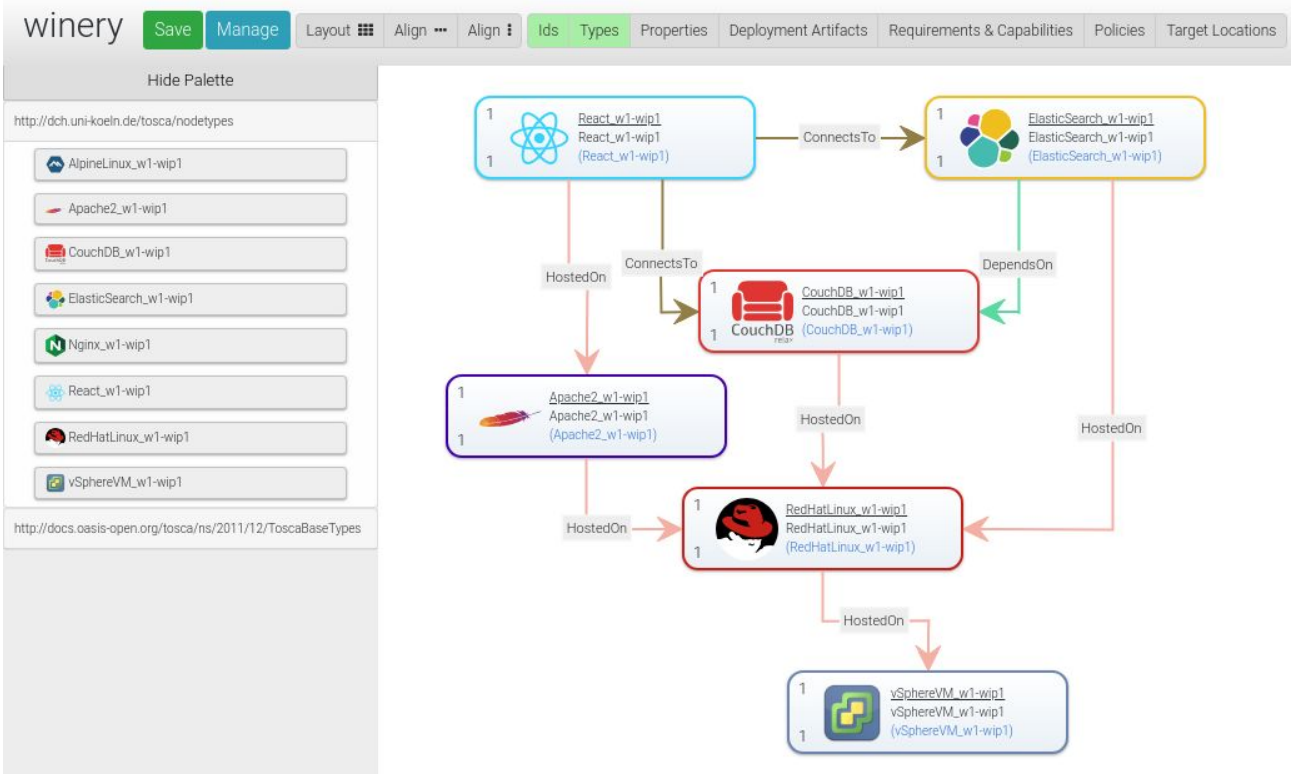
scientific practice, a sustainability strategy to keep this data accessible on a permanent basis must include the web application itself, because the separation and archiving of the primary data alone would inevitably lead to a loss of functionality (and thus information).

3 TOSCA and OpenTOSCA

Technological basis of our approach is the OASIS standard TOSCA [11, 12, 13]. TOSCA allows for a portable description of IT systems to automate their provisioning and management. In TOSCA, a cloud application or service [9] is modeled as a *Service Template*. Inside a Service Template, the *Topology Template* describes the service's topology as a directed multigraph, consisting of *Node Templates* and *Relationship Templates* that specify the edges between the nodes. Thus, this enables to describe arbitrary deployments in the form of *declarative deployment models* [5]. Underneath, TOSCA employs a type system defining common properties and attributes in *Node Types* and *Relationship Types*, respectively. To automatically deploy, provision and manage the modeled service, TOSCA defines a self-contained archive called *Cloud Service Archive* (CSAR) which contains the Service Template, including all Node Types and Relationship Types, as well as all required software artifacts, scripts, and binaries needed for provisioning. Moreover, imperative management plans can be added to CSARs, which enables the implementation of arbitrary kinds of management functionality in an automatically executable manner. These plans can be implemented using standardized workflow languages such as BPEL or BPMN, or domain-specific modeling extensions such as BPMN4TOSCA [7]. Any TOSCA runtime environment can consume such a CSAR to automatically deploy and instantiate the enclosed application [2].
In a series of research projects, the Institute for Architecture of Application Systems (IAAS, http://iaas.uni-stuttgart.de) at the University of Stuttgart has developed the OpenTOSCA ecosystem, an open source implementation for the TOSCA standard. OpenTOSCA includes (i) the graphical modeling tool *Winery* for the creation of TOSCA-based application models [8], (ii) the runtime environment *OpenTOSCA container* for automated provisioning and management of the modeled applications [1], and (iii) the self-service portal *Vinothek* [4], which lists all applications installed in the OpenTOSCA container and serves as a graphical user interface. The TOSCA standard is generally suitable for assuring the digital sustainability of research results, as research applications, which are packaged in CSARs, can be executed years later by a TOSCA-compliant runtime environment [3].

4 A TOSCA Model for the Musical Competitions Database

In the following, we describe an application model for the above use case to exemplify some of the basic concepts of (Open)TOSCA. As stated above, the implementation of the web application is still ongoing work at the time of writing, the same applies to the respective model. The Musical Competitions Database is made up of a CouchDB, ElasticSearch and a JavaScript frontend. The resulting TOSCA-compliant topology model is depicted in figure 1.



Screenshot of OpenTOSCA's modeling tool *Winery*, showing the topology of the use case application. On the left side, the available components are listed.

The topology in figure 1 consists of a Javascript/React web application that is hosted on a Apache2 web server that itself is hosted on a Red Hat Linux system running as a vSphere virtual machine. The application connects to a CouchDB database and to ElasticSearch, which are both hosted on the Linux virtual machine. Therefore, six different node types, namely *React*, *Apache2*, *CouchDB*, *ElasticSearch*, *RedHatLinux* and *vSphereVM*, as well as the *HostedOn*, the *ConnectsTo* and the *DependsOn* relationship types must be available. A TOSCA Service Template describing this application will contain the six node templates and three relationship templates modeling the relationship between the nodes – where each template is an instance of the respective type definition. The resulting service template can then be packed in a CSAR which may be instantiated by any TOSCA runtime or to be archived in a repository.

5 Summary and Outlook

The previously discussed concepts emerged from *SustainLife* [10], a DFG-funded joint project of the DCH Cologne and the IAAS Stuttgart. The overall objective is to develop generic solutions for standards-based operation and maintenance of DH-applications and to implement them in a way that they find practical application in humanities data centers like the DCH.
As TOSCA depends on a generic type system enabling the reuse of recurring components like web servers, operating systems, or messaging middleware, we work towards providing a set of typical system components, e.g. a component library. By modeling use cases like the one described above, we can identify the components with the highest synergetic effects and model them within TOSCA. Examples for components, which were already identified in further use cases [10] are Java runtime environments, the Spring framework and several types of databases like MySQL, mongoDB and eXist-db. In addition, reusable Service Templates reflecting typical software stacks are developed. For this reason, we analyze use cases for typical application structures and create corresponding models. For example, a common pattern for web applications is the so-called LAMP-stack, composed of a Linux operating system, an Apache2 web server, a MySQL/MariaDB

database and a PHP/Perl/Python interpreter. These components and templates can be reused to model further applications and are intended to simplify future application development and maintenance using TOSCA and the OpenTOSCA ecosystem.
Beyond that, a number of further extensions of the OpenTOSCA ecosystem are in the scope of the SustainLife project. For example, applications that are archived in CSARs need to be deployable several years after their development. Therefore, approaches to *freeze* and *defrost* whole applications, so they can be reinstated in the same state as they were decommissioned, are part of our research. This also includes the possibility to version TOSCA models, because living systems are subject to constant changes. Another desideratum is that there is currently no possibility to update a service's components. If a component must be exchanged because of security issues or deprecation, the CSAR may no longer be deployable. We therefore work on additional management functionalities which provide standardized operating and maintenance solutions, e.g., applying (security) updates or software patches.
With our approach we expect to reduce maintenance costs significantly and will evaluate this expectation on the basis of selected use cases. Findings and best practices are prepared in a way that they can be transferred to partners and are communicated to the scientific community through workshops and publications. Thus, with this contribution, we want to trigger a discussion about the applicability of methods and technologies of professional cloud deployment and provisioning strategies to problems of long-term availability of research software in the DH-community.

Acknowledgements

This work is partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation). Project title: "SustainLife – Erhalt lebender, digitaler Systeme für die Geisteswissenschaften" (see http://gepris.dfg.de/gepris/projekt/379522012).

References

1. T. Binz, U. Breitenbücher, F. Haupt, O. Kopp, F. Leymann, A. Nowak, S. Wagner. "OpenTOSCA - A Runtime for TOSCA-based Cloud Applications". In: ICSOC, 2013, S. 692-695, 2013.
2. U. Breitenbücher, T. Binz, K. Képes, O. Kopp, F. Leymann, J. Wettinger. "Combining Declarative and Imperative Cloud Application Provisioning based on TOSCA", IC2E, S. 87–96, 2014.
3. U. Breitenbücher, J. Barzen, M. Falkenthal, F. Leymann. "Digitale Nachhaltigkeit in den Geisteswissenschaften durch TOSCA: Nutzung eines standardbasierten Open-Source Öko-systems". Konferenzabstracts DHd 2017: Digitale Nachhaltigkeit, S. 235-237, 2017.
4. U. Breitenbücher, T. Binz, O. Kopp und F. Leymann. "Vinothek - A Self-Service Portal for TOSCA". In: ZEUS 2014, p. 69-72, 2014.
5. C. Endres, U. Breitenbücher, M. Falkenthal, O. Kopp, F. Leymann, J. Wettinger. "Declarative vs. Imperative: Two Modeling Patterns for the Automated Deployment of Applications". In Proceedings of the 9th International Conference on Pervasive Patterns and Applications (PATTERNS), 2017, pp. 22-27.
6. F. Hentschel. "Institutionalisierung des ästhetischen Werturteils: Musikalische Preisausschreiben im 19. Jahrhundert". In: Archiv für Musikwissenschaft 69 (2012), pp. 110-121, 2012.
7. O. Kopp, T. Binz, U. Breitenbücher, F. Leymann. „BPMN4TOSCA: A domain-specific language to model management plans for composite applications". In International Workshop on Business Process Modeling Notation, pp. 38-52. Springer, Berlin, Heidelberg, 2012.
8. O. Kopp, T. Binz, U. Breitenbücher, F. Leymann. "Winery – A Modeling Tool for

TOSCA-based Cloud Applications”. In: ICSOC, 2013, S. 700-704, 2013.
9. F. Leymann, U. Breitenbücher, S. Wagner, J. Wettinger. “Native Cloud Applications: Why Monolithic Virtualization Is Not Their Foundation”. Cloud Computing and Services Science, Springer, pp. 16-40, 2017.
10. C. Neuefeind, L. Harzenetter, P. Schildkamp, U. Breitenbücher, B. Mathiak, J. Barzen, F. Leymann. "The SustainLife Project – Living Systems in Digital Humanities". In: Proceedings of the 12th Advanced Summer School on Service-Oriented Computing, 2018 (IBM Research Report RC25681), pp.101-112, 2018
11. OASIS: “Topology and Orchestration Specification for Cloud Applications Version 1.0”, 2013.
12. OASIS. “Topology and Orchestration Specification for Cloud Applications (TOSCA) Primer Version 1.0”. 2013.
13. OASIS: “TOSCA Simple Profile in YAML”, Version 1.0, 2015.
14. D. L. Parnas. “Software Aging”. In: Proceedings of the 16th International Conference on Software Engineering (ICSE 1994). IEEE, May 1994, pp. 279-287, 1994.
15. P. Sahle, S. Kronenwett. “Jenseits der Daten: Überlegungen zu Datenzentren für die Geisteswissenschaften am Beispiel des Kölner Data Center for the Humanities”. In: LIBREAS. Library Ideas 23, pp. 76-96, 2013.
16. U. Wuttke, C. Engelhardt, S. Buddenbohm. “Angebotsgenese für ein geisteswissenschaftliches Forschungsdatenzentrum”. In: Zeitschrift für digitale Geisteswissenschaften, 2016.