university of groningen / faculty of science and engineering

A New Approach to the Lifecycle Management of Cloud-based Applications

Vasilios Andrikopoulos
Software Engineering and Architecture Group
Johann Bernoulli Institute for Mathematics and Computer Science

Problem Context & Scope

The adoption of cloud computing by organizations of all sizes and types in the recent years has created multiple opportunities and challenges for the development of software to be used in this environment. In one way or another, cloud computing offerings and associated technologies are currently being used by the majority of software-intensive enterprises. The availability of multiple options with respect to the deployment and delivery models available, in conjunction with the plethora of offerings by cloud providers like Amazon Web Services, Microsoft Azure, and Google Compute Platform, and software solutions for the deployment of private clouds such as the ones by VMware and OpenStack, create both **opportunities and challenges for software developers**.

At the same time, in the recent years the discourse on the best practices and principles of software development, at least in the industry, has been affected significantly by the introduction of **two movements** that have a co-dependence relation with cloud computing. The first one is the use of **DevOps technologies and processes** in order to bridge the gap between development and operations of software. Software containerization as made popular by Docker allows for each architectural component to be developed, deployed, managed, and updated in its own software stack, following a lifecycle that is loosely coupled with the overall system evolution. This principle is made even more prominent in the second of the movements relevant to the discussion, i.e. **microservices**. In [Andrikopoulos, 2017], the author observes that *these developments have significantly evolved the way that software is developed, deployed, and managed over time, and as such our methods and techniques for engineering software, especially with respect to how we architect it, have to be adapted accordingly*.

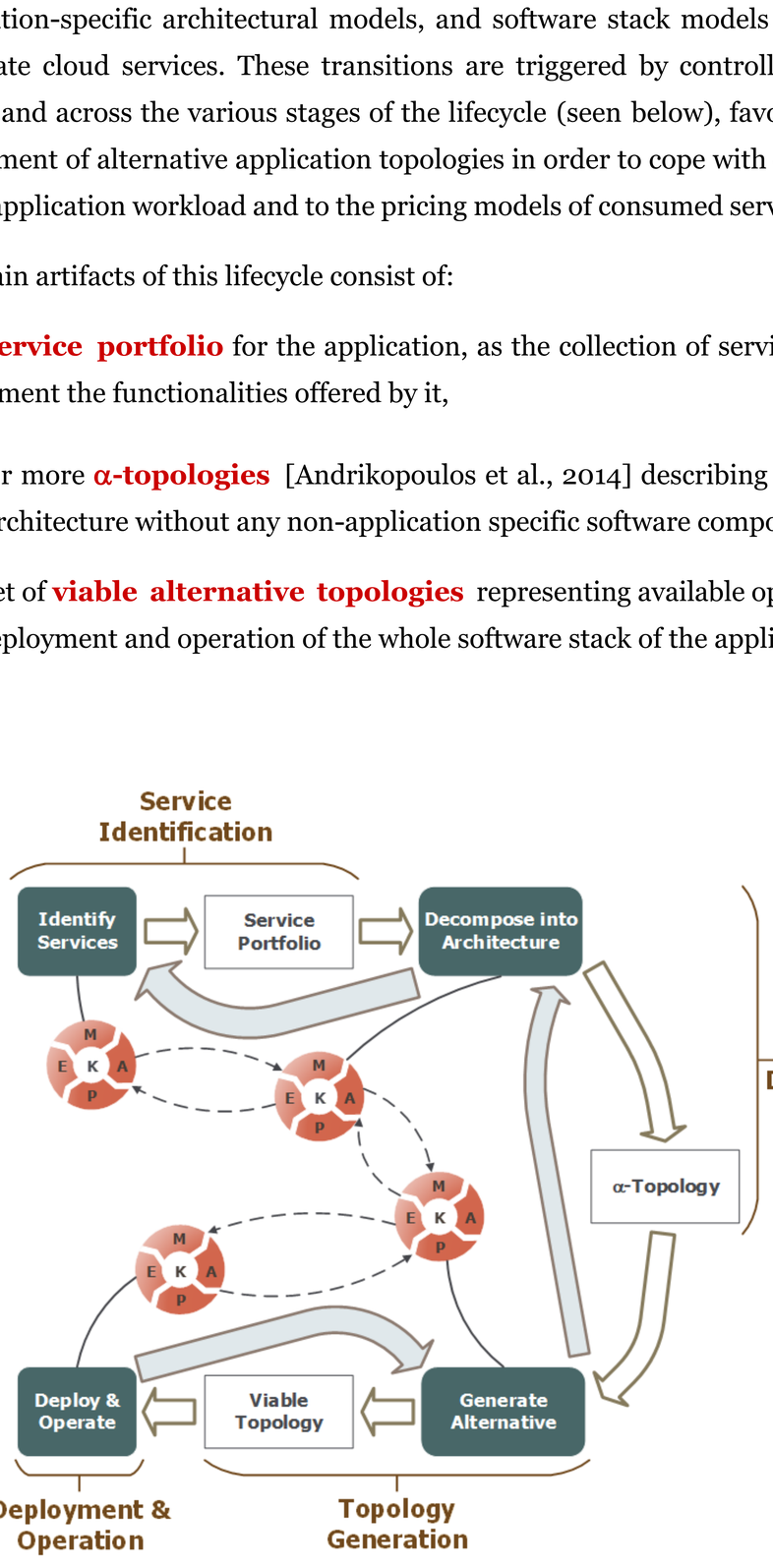
Contribution

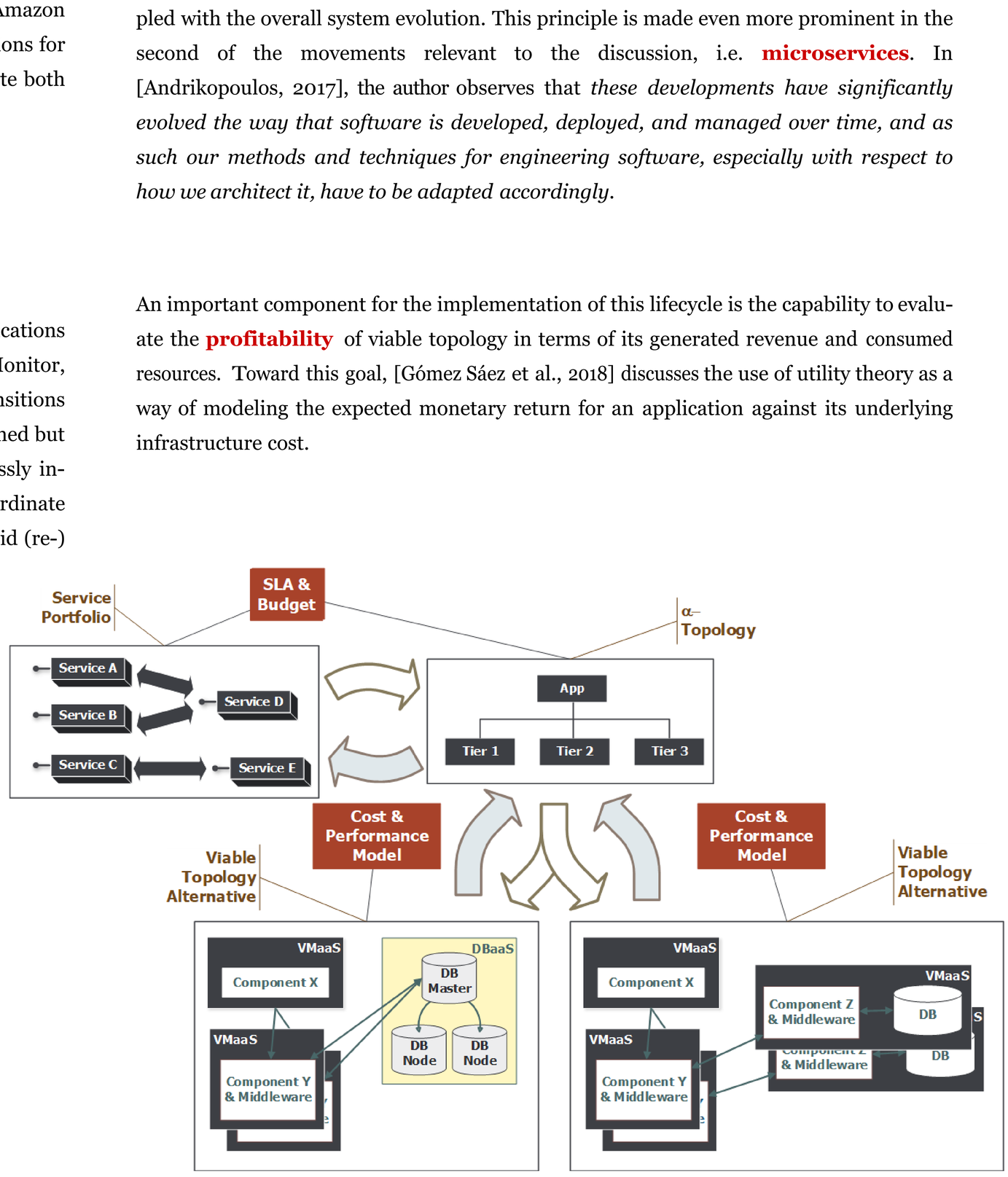
For this purpose, [Andrikopoulos, 2017] proposes a lifecycle for cloud-based applications that is defined as a series of control loops, following a **hierarchical MAPE-K** (Monitor, Analyze, Plan, Execute and Knowledge, respectively) model. These loops define transitions between sets of application functionalities encapsulated as services, abstractly defined but application-specific architectural models, and software stack models that seamlessly incorporate cloud services. These transitions are triggered by controllers that coordinate within and across the various stages of the lifecycle (seen below), favoring the rapid (re-) deployment of alternative application topologies in order to cope with changes to the application workload and to the pricing models of consumed services.

The main artifacts of this lifecycle consist of:

- The **service portfolio** for the application, as the collection of services that implement the functionalities offered by it,
- One or more **α-topologies** [Andrikopoulos et al., 2014] describing the system architecture without any non-application specific software components,
- The set of **viable alternative topologies** representing available options of the deployment and operation of the whole software stack of the application.

An important component for the implementation of this lifecycle is the capability to evaluate the **profitability** of viable topology in terms of its generated revenue and consumed resources. Toward this goal, [Gómez Sáez et al., 2018] discusses the use of utility theory as a way of modeling the expected monetary return for an application against its underlying infrastructure cost.





Ongoing & Future Work

The main action items with respect to current and future work are summarized as:

- **Instrument** the MAPE-K controller across the different abstraction levels, starting from the lower one (i.e. viable alternative topologies)
- **Experiment** with different architectures and loads from different application domains to collect a core set of rules for the Knowledge Base
- **Evaluate** the overall approach in a series of case and field studies

References

Andrikopoulos, V. (2017). **Engineering Cloud-based Applications: Towards an Application Lifecycle**. In Joint Pre-Proceedings of the Workshops Associated with ESOCC 2017. Available online: https://www.duo.uio.no/bitstream/handle/10852/58955/preproceedings.pdf

Gómez Sáez, S., Andrikopoulos, V., Bitsaki, M., Leymann, F., and van Hoorn, A. (2018). **Utility-based decision making for migrating cloud-based applications**. ACM Transaction of Internet Technologies, 18(2):1–22.

Andrikopoulos, V., Gómez Sáez, S., Leymann, F., Wettinger, J. (2014). **Optimal Distribution of applications in the Cloud**. In Proceedings of the 26th Conference on Advanced Information Systems Engineering (CAiSE 2014), Springer, pp.75-90.

Contact

Vasilios Andrikopoulos
Email: v.andrikopoulos@rug.nl
Twitter: v_andrikopoulos
Address: Nijenborgh 9, 9747 AG Gronginen
Web: https://vandriko.github.io