University of Bamberg

Evaluating Cloud-native Software Architectures with Clounaq

Robin Lichtenthäler, Guido Wirtz
Distributed Systems Group, University of Bamberg

Motivation: Cloud-native

Automated deployment and **infrastructure automation**, towards the use of cloud-native services.
- Pahl2018

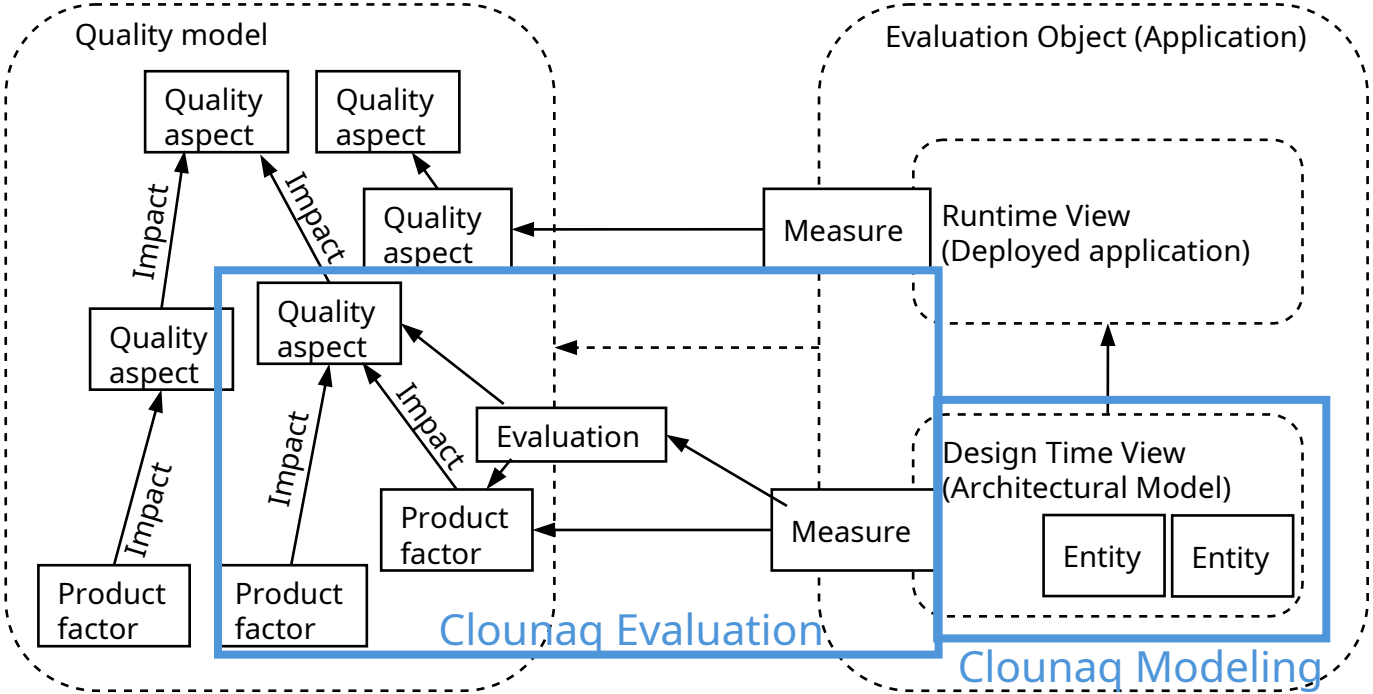
Cloud-native applications are designed so that **upgrade and test occur seamlessly** without disrupting production.
- Gannon2017

Cloud-native applications are designed with a clear separation among **stateless and stateful** services.
- Wurster2020

A cloud-native application (CNA) is a distributed, elastic and horizontal scalable system composed of (micro)services […] These services are **built around business capabilities** and independently deployable by fully automated deployment machinery.
- Kratzke2017

Service-to-service **communications** in cloud-native application are API-based.
- Wurster2020

Clounaq (Cloud-native architectural quality)

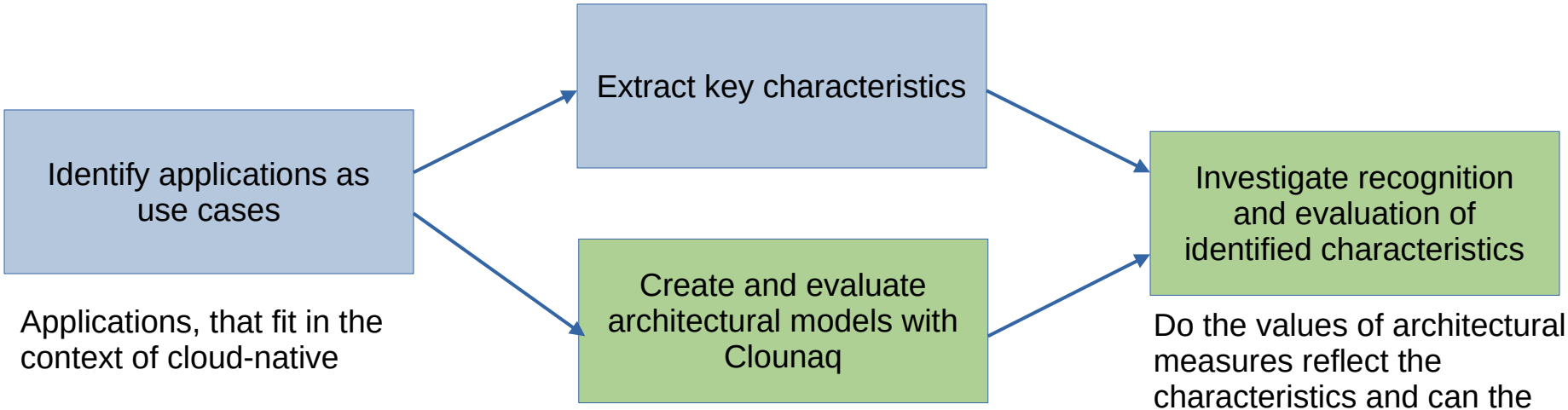


Research Question

How well can the Clounaq approach be used for evaluating software architectures according to cloud-native characteristics?

Used approach

Which architectural style, which patterns, which technologies were used?



Applications, that fit in the context of cloud-native

Do the values of architectural measures reflect the characteristics and can the evaluation results be used?

Evaluated Use Cases

TeaStore

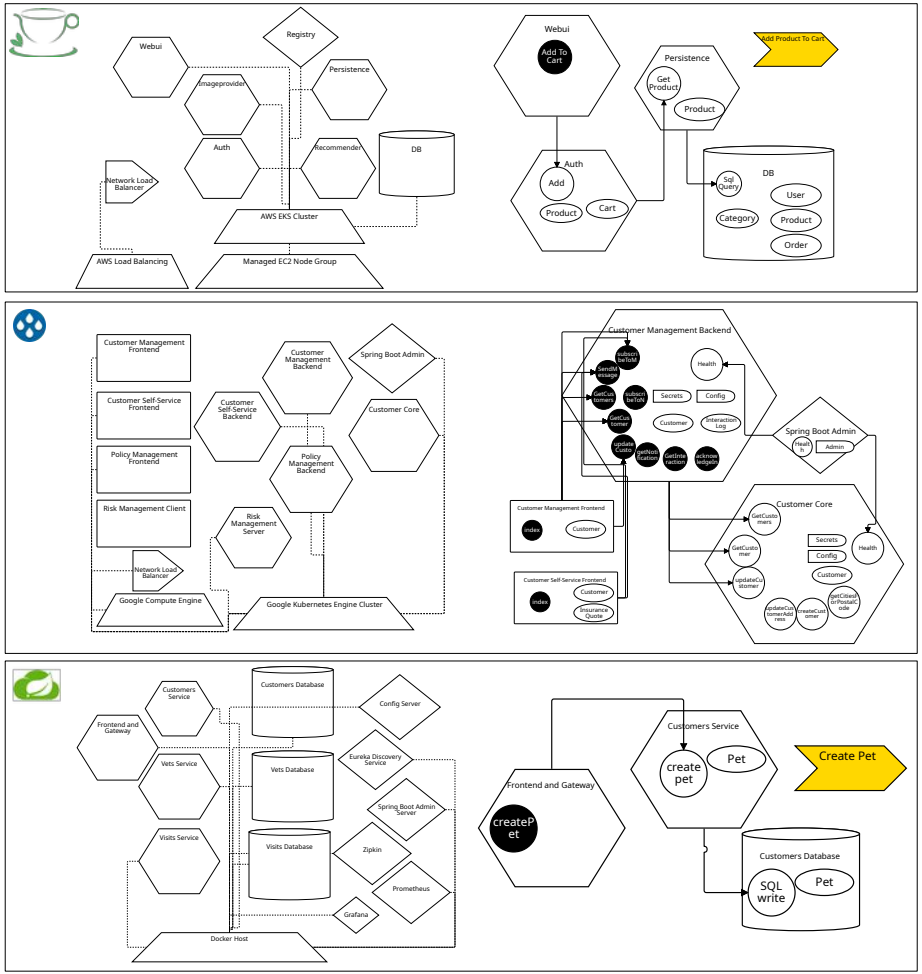
https://github.com/DescartesResearch/TeaStore

LakesideMutual

https://github.com/Microservice-API-Patterns/LakesideMutual

Pet Clinic Cloud

https://github.com/spring-petclinic/spring-petclinic-cloud



Live-Demo: https://clounaq.de

Discussion of the results (extract)



LakesideMutual is the only application with asynchronous messaging	Asynchronous communication (→ Loose coupling → Modularity)	none	low	none
	Degree of asynchronous communication	0	0.065	0
	Asynchronous communication utilization	0	0.22	0
TeaStore stores configuration in the Environment (K8s). Only PetClinic uses an explicit configuration backing service	Configuration Management (→ Adaptability)	moderate	neutral	high
	Isolated Configuration	high	none	high
	Configuration Externalization	1	0	1
	Configuration stored in specialized services	none	none	high
	Configuration stored in config service	0	0	1

Discussion of the results (extract)



Only LakesideMutual uses no explicit caching at all.	Vertical data replication (→ Replication →Time-Behavior)	low	none	moderate
	Ratio of cached data aggregates	0.307	0	0.5

Registry services in TeaStore and Petclinic, K8s-based in LakesideMutual	Addressing abstraction (→ Modifiability)	high	high	moderate
	Service Discovery Usage	1	0.98	0.634

- Overall, the key characteristics of the applications are covered
- Certain aspects are out of scope (such as detailed endpoint characteristics considering parameters or message content)
- Reported values are on the level of the systems as a whole, but consideration of other entities is also possible

Limitations

- Ease of Use and User-specific customization of the Clounaq tool
- Modeling Architectures is a manual process
- Evaluation approach needs further validation
- Additional applications with differing characteristics need to be considered

Conclusion

- Breadth of cloud-native covered with a structured approach
- Evaluation on different layers possible
  (→ based on the different modeling entities)
- Architecture Evaluation only serves as an informational basis; Decisions about changes to the application architecture need to be made application-specific

Thank you for your attention!

References

Repository with the software architecture models: https://github.com/r0light/clounaq-evaluation
(These models can be imported in Clounaq)

Literature:

- Gannon, Dennis / Barga, Roger / Sundaresan, Neel: Cloud-Native Applications, 2017, IEEE Cloud Computing , Vol. 4, No. 5, Institute of Electrical and Electronics Engineers (IEEE), p. 16-21
- Kratzke, Nane / Quint, Peter-Christian: Understanding Cloud-native Applications after 10 Years of Cloud Computing - A Systematic Mapping Study, 2017, Journal of Systems and Software, Journal of Systems and Software , Vol. 126, p. 1-16
- Pahl, Claus / Jamshidi, Pooyan / Zimmermann, Olaf: Architectural Principles for Cloud Software, 2018, ACM Transactions on Internet Technology , Vol. 18, No. 2, Association for Computing Machinery (ACM), p. 1-23
- Wurster, Michael / Breitenbücher, Uwe / Brogi, Antonio / Leymann, Frank / Soldani, Jacopo: Cloud-native Deployability: An Analysis of Required Features of Deployment Technologies to Deploy Arbitrary Cloud-native Applications, 2020, Proceedings of the 10th International Conference on Cloud Computing and Services Science, SCITEPRESS