RADON: RATIONAL DECOMPOSITION AND ORCHESTRATION FOR SERVERLESS COMPUTING

G.Casale[1], M. Artač[2], W.-J. van den Heuvel[3], A. van Hoorn[4], P. Jakovits[5], F. Leymann[4], M. Long[6], V. Papanikolaou[7], D. Presenza[8], A. Russo[1], S.N. Srirama[5], D.A. Tamburri[3], M. Wurster[4], L. Zhu[1].

[1]Imperial College London, UK; [2]XLAB, Slovenia; [3]Jheronimus Academy of Data Science, the Netherlands; [4]Universität Stuttgart, Germany; [5]Tartu Ülikool, Estonia; [6]Praqma, Norway; [7]Athens Technology Center, Greece; [8]Engineering Ing. Informatica, Italy.

MOTIVATION

Serverless function-as-a-service (FaaS) offerings, enable developers to virtualize the internal logic of an application, **simplifying management** of cloud native applications and allowing **cost savings** through billing and scaling at the level of individual function calls. Platforms such as AWS Lambda, Microsoft Azure Functions, Google Cloud Functions are therefore rapidly shifting the attention of software vendors to the problem of developing cloud applications that can use FaaS platforms.

VISION

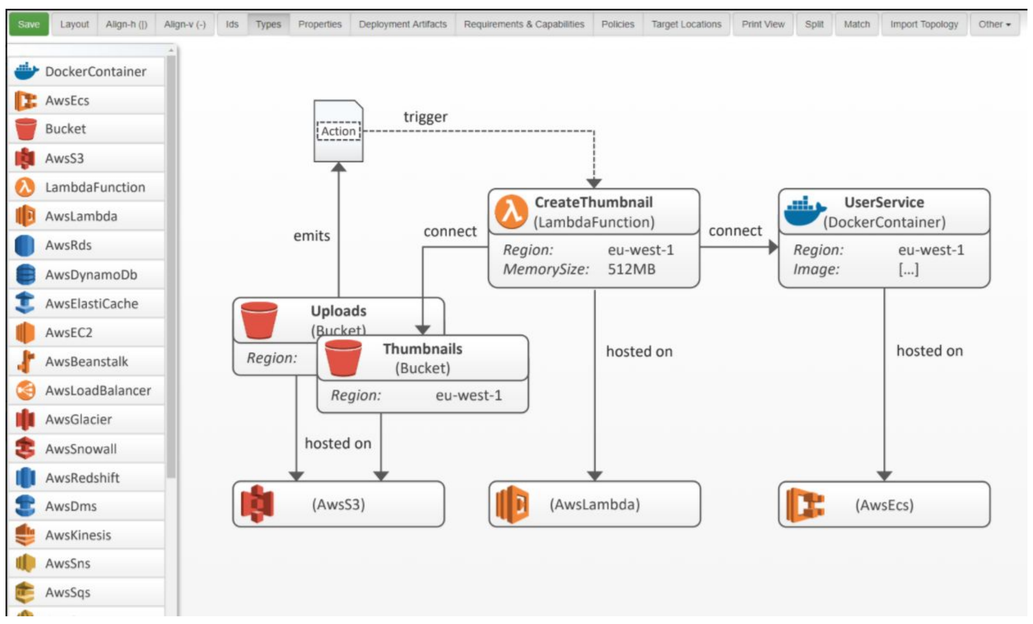
Cloud applications are increasingly finer-grained, consisting of independently deployable microservices, event-driven processing based on FaaS, all relaying on containerization technologies. To support this vision, RADON research agenda proposes to develop an advanced DevOps framework to prototype, design, deploy, test, verify and evolve cloud applications building on:

- **Serverless functions**, defining events, triggers and actions (handling functions);
- **Microservices** implementing the business logic in a highly-decomposed architecture;
- **Data pipelines** to manage the lifecycle of data, e.g., as in event-driven data transfers and cross-cloud data synchronizations.

CHALLENGES

- **Modelling for serverless FaaS.** As of today, serverless FaaS is not properly supported by existing modelling formalisms and tools, for example it is not currently possible to describe within Infrastructure-as-Code models (e.g., TOSCA) the dependencies of a FaaS-based application. FIGURE 1 conceptualizes an implementation of a tool that supports this capability.

FIGURE 1



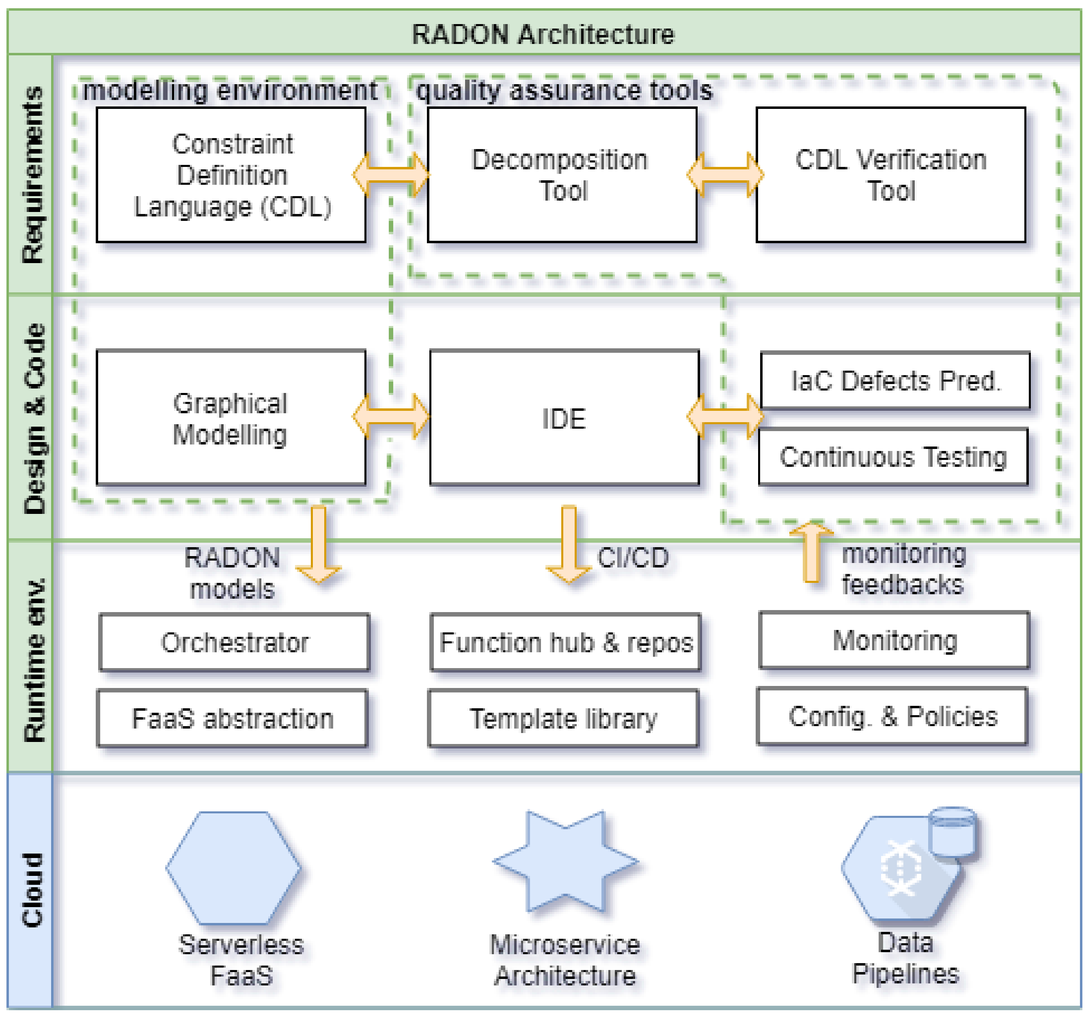
- **Decomposition trade-offs.** Questions such as *"Should my application functionality be implemented using serverless or microservices?"* or *"What is the optimal size for a service?"* are difficult to answer for the average developer, effectively increasing the risk of poor design decisions.
- **Runtime operations and delivery.** The combination of model-driven orchestration, CI/CD and Infrastructure-as-Code, although powerful, requires to define a large set of scripts and blueprints for instantiating and integrating the technologies at play in the deployment environment.
- **Avoiding FaaS lock-in.** As provocatively asserted by a popular technology website (The Register): "Lambda and serverless is one of the worst forms of proprietary lock-in we've ever seen in the history of humanity". Since FaaS technologies are expected to disrupt the way developers code their applications, they also pose the risk of proprietary lock-in in commercial offerings.

RESEARCH AGENDA

The RADON research agenda envisions a framework to tackle the above challenges and consisting of a modelling environment, a runtime environment, and a coding environment (IDE). The framework will be coupled with a **DevOps methodology** to coordinate the development and release of the applications, and with quality assurance tools to validate a particular implementation. FIGURE 2 gives an holistic view of the framework and its composing tools, a few highlights are as follows:

- A **graphical modelling environment** to visually describe FaaS-based architectures within Infrastructure-as-Code models such as TOSCA.
- A **constraint definition language** (CDL), together with a verification tool, to annotate and verify expressed requirements on the graphical TOSCA-based models;
- **Quality assurance tools** to decompose an architecture into functions and microservices, test them after prototyping, and identify Infrastructure-as-Code bugs and defects;
- **Runtime environment** for orchestration, multi-cloud FaaS instantiation, function hubs, and monitoring, among others.

FIGURE 2



LEARNING MORE

Check out `http://radon-h2020.eu` to learn more about the RADON vision and our progress in making the above vision into a concrete open source framework.

ACKNOWLEDGEMENT

This work has been partially funded by the European Commission grant H2020-825040 (RADON).