



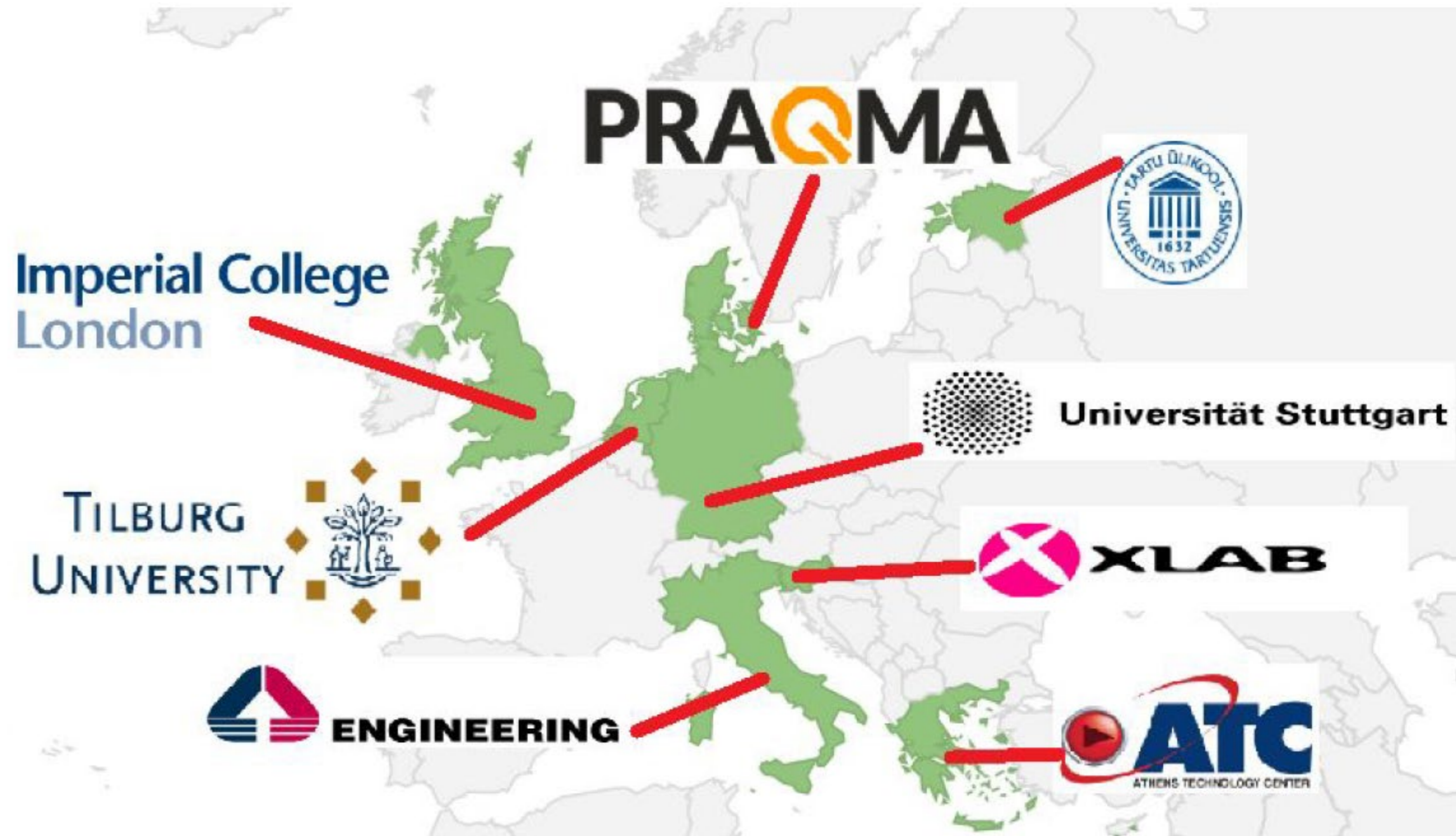
RADON

Advanced Engineering of Microservices and Serverless Applications:
The RADON approach

RADON Consortium



- ICT-16-2018: Software Technologies
- 30 months project (Jan 2019 - Jun 2021) – 8 organizations



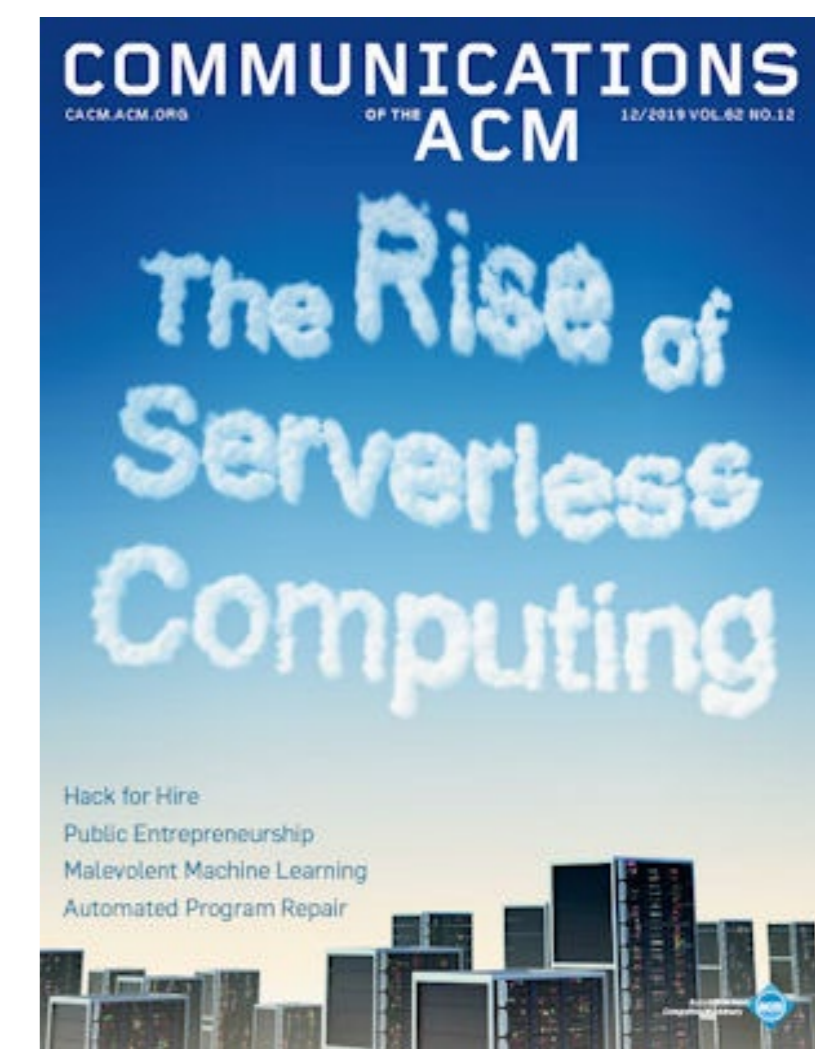
Serverless Function-as-a-Service

- FaaS: function calls served from the cloud, event-driven paradigm
- Quick prototyping and demonstration without infrastructure management issues

- A way to reduce costs
 - Fine-grained billing
 - Automated deallocation

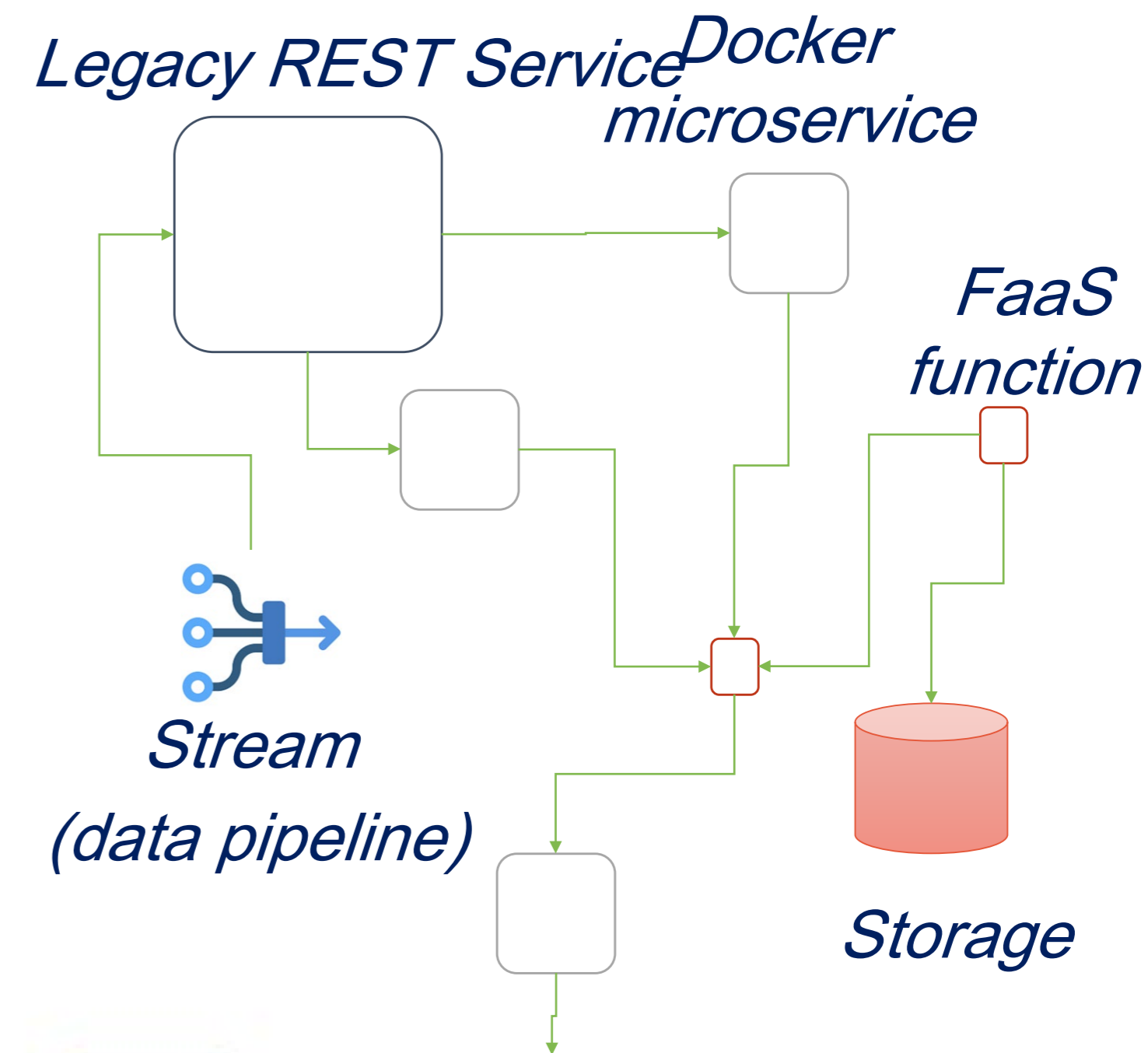
```
com.example.lambda.demo.Hello Lambda Console
Skip uploading function code since no local change is found...
Invoking function...
===== FUNCTION OUTPUT =====
"Hello, AWS Lambda!"
===== FUNCTION LOG OUTPUT =====
START RequestId: 5287a47b-baa9-11e7-b87a-c1cfa64acbad Version: $LATEST
Input: AWS LambdaEND RequestId: 5287a47b-baa9-11e7-b87a-c1cfa64acbad
REPORT RequestId: 5287a47b-baa9-11e7-b87a-c1cfa64acbad Duration: 37.27 ms Billed Duration: 100 ms Memory S
```

- Natural to combine with microservices-based architectures
 - Fine-grained software architecture
 - Automated autoscaling
 - Flexibility and responsiveness
 - High-degree of reuse of platform services



Some challenges

- How to **deploy and update** in continuous, DevOps fashion, hybrid serverless-based applications?
- How to choose an **optimal deployment** configuration respecting QoS?
- How to **debug** infrastructure code?
- How to best do **testing** and **monitor** outcomes?



Amazon Lambda



Google Cloud Functions

RADON framework overview

Value proposition:

Offer an open source **DevOps framework** to help the EU software industry adopting **serverless FaaS** without vendor lock-in

Tools at advanced state.

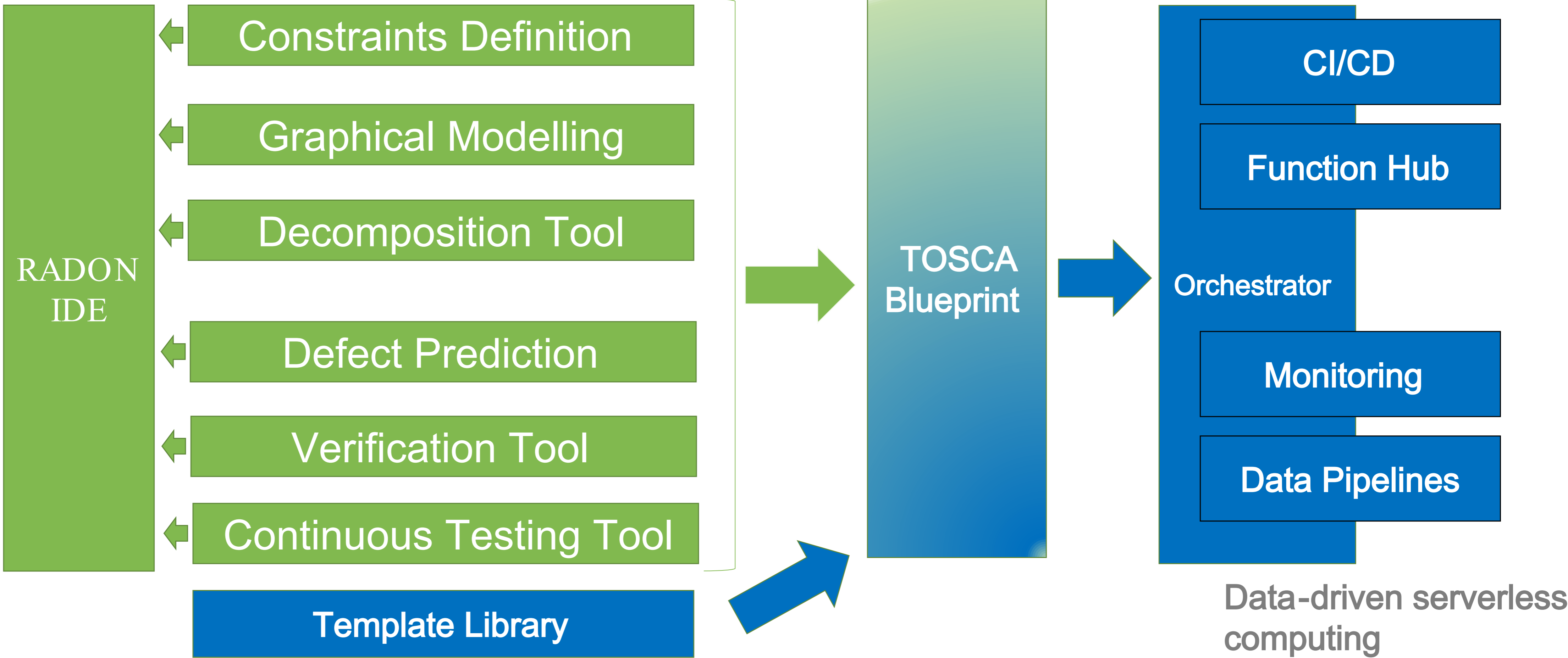
Open source releases.

The screenshot shows the GitHub profile for the RADON Consortium. At the top, the profile name 'RADON Consortium' is displayed with a location 'EU' and website 'http://radon-h2020.eu'. Below this is a navigation bar with tabs for 'Repositories 30', 'Packages', 'People 23', 'Teams', 'Projects 4', and 'Settings'. A search bar is present with filters for 'Type: All' and 'Language: All'. The main content area lists several repositories:

- radon-particles**: TOSCA definitions repository for the RADON project. License: Apache-2.0. 8 forks, 4 stars, 10 issues, 1 pull request. Updated 12 hours ago.
- radon-csars**: 1 fork, 0 stars, 0 issues, 0 pull requests. Updated 21 hours ago.
- radon-plugin-registry**: A repository to store the plug-in meta information and packages required to publish VS Code extensions in an Eclipse Che plug-in registry. License: Apache-2.0. 0 forks, 0 stars, 0 issues, 0 pull requests. Updated 22 hours ago.
- demo-radon-plugins**: Examples for RADON. License: Apache-2.0. 0 forks, 0 stars, 0 issues, 0 pull requests. Updated 22 hours ago.

On the right side, there are three widgets: 'Top languages' (Python, TypeScript, Shell, MATLAB, CSS), 'Most used topics' (iac, metrics), and 'People' (23 members).

RADON framework overview



Graphical Modeling Tool

winery Save Manage Layout Align Import Topology Types

Hide Palette

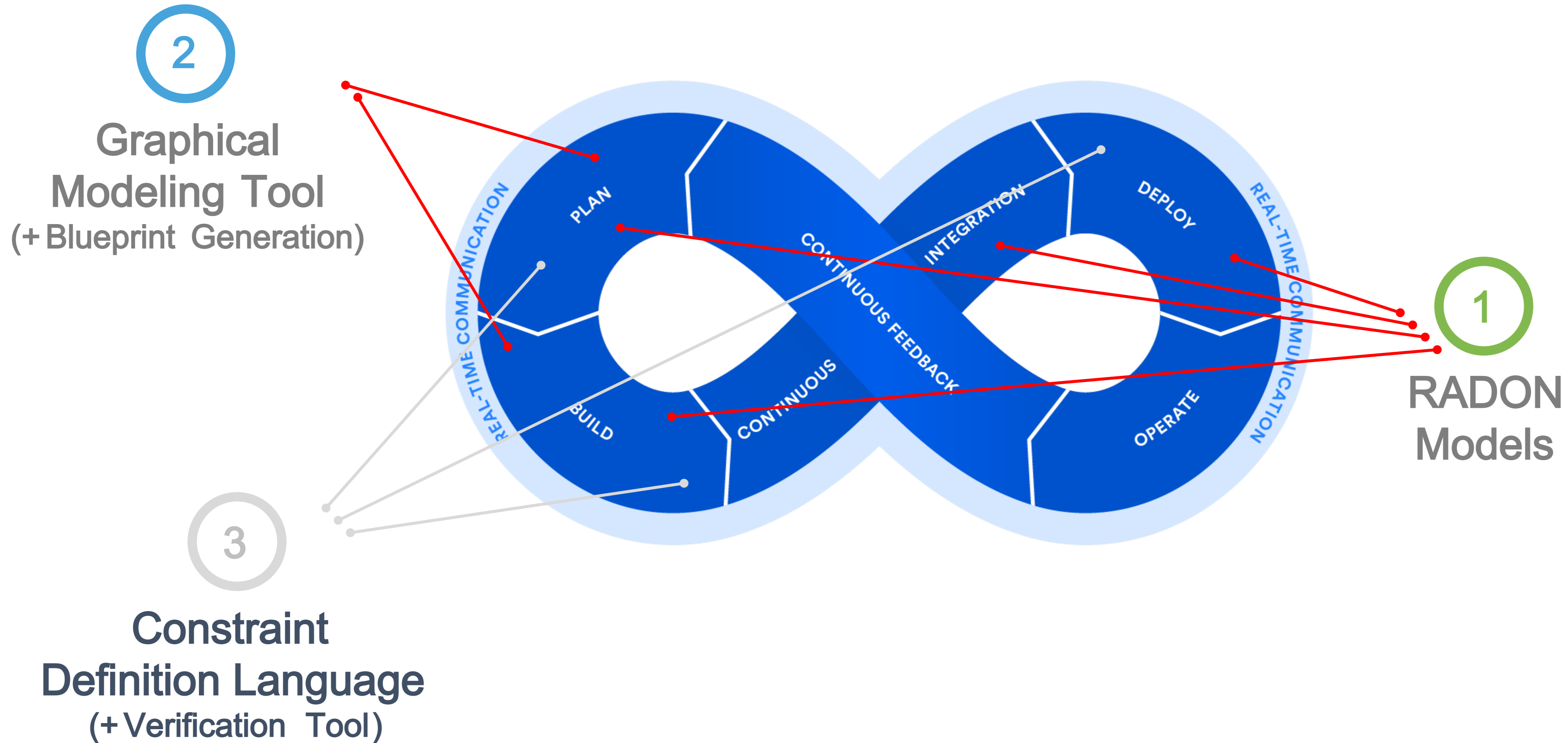
- radon.nodes.abstract
- radon.nodes.apache.kafka
- radon.nodes.apache.nifi
- radon.nodes.apache.openwhisk
- radon.nodes.aws
 - aws AwsLambdaFunction
 - aws AwsPlatform
 - aws AwsS3Bucket
- radon.nodes.azure
- radon.nodes.docker
- radon.nodes.google
- radon.nodes.java
- radon.nodes.mongodb
- radon.nodes.mysql
- radon.nodes.nodejs
- radon.nodes.openfaas

winery Save Manage Layout Align Import Topology Ids Types Properties Deployment Artifacts Requirements & Capabilities

Determine Stateful Components Determine Freezable Components Clean Freezable Components Refine Pattern

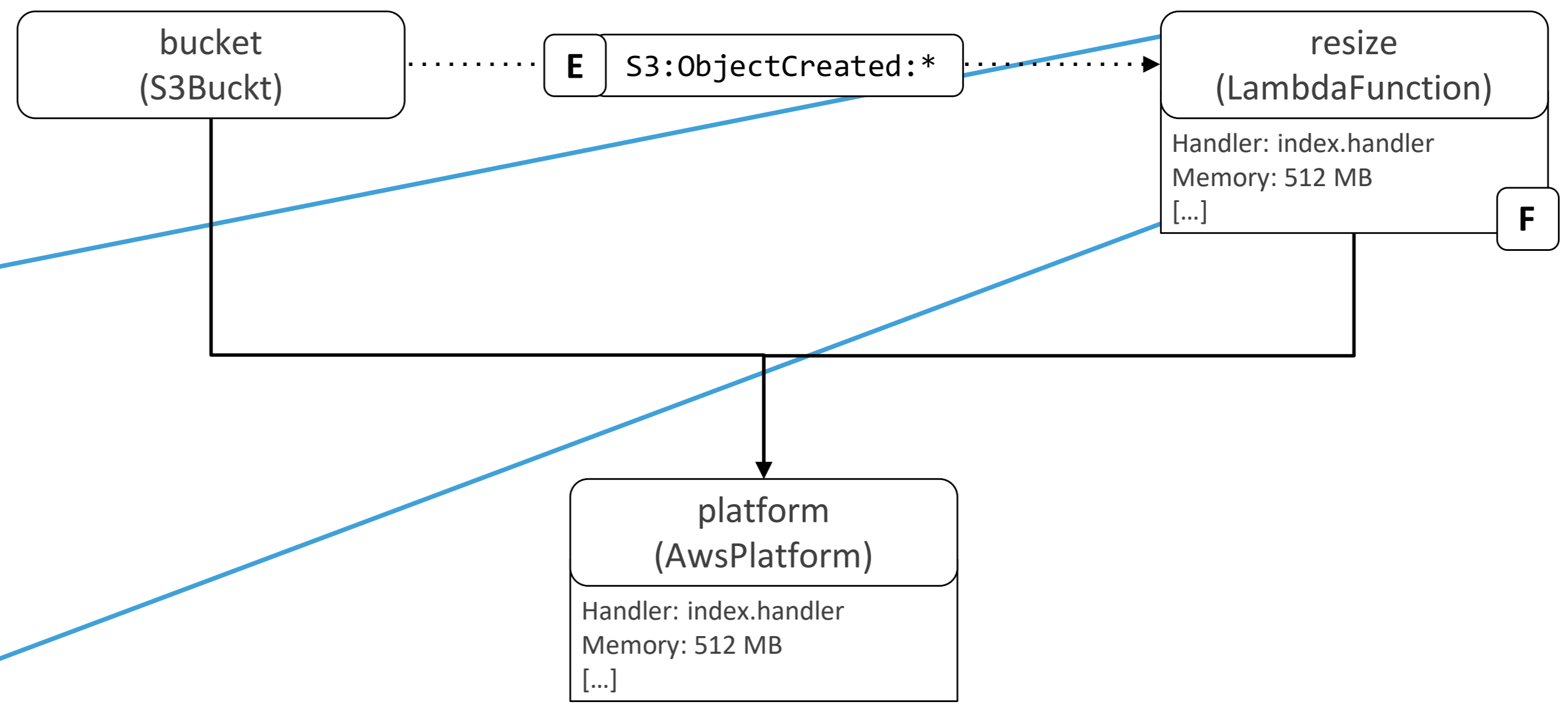
RADON: modelling for serverless FaaS

- Easy-to-use composition of functions, microservices, storage, VMs, ...
- Reuse modular element to assemble complex applications



RADON Models

```
tosca_definitions_version: tosca_simple_yaml_1_3
topology_template:
  node_templates:
    platform:
      type: radon.nodes.aws.AwsPlatform
      properties:
        # omitted for brevity
    resize:
      type: radon.nodes.aws.LambdaFunction
      properties:
        handler: index.handler
        memory: 512
        # ...
      artifacts:
        deployment_package:
          file: thumbnail.zip
          type: radon.artifacts.archive.Zip
      requirements:
        - host: platform
    bucket:
      type: radon.nodes.aws.S3Bucket
      requirements:
        - host: platform
        - invoker:
            node: resize
            relationship: trigger
```



Models automatically deployable
using the RADON orchestrator.

Quality Guardrails in RADON



- 1. Create model
- 2. Run verification
e.g., GDPR constraints

Verification Tool

[All constraints verified]

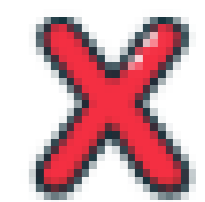
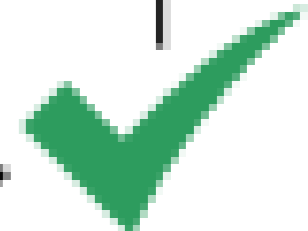
- 1. Extract *product* metrics
(e.g. # lines of code)
- 2. Extract *delta* metrics
(between two successive releases)
- 3. Extract *process* metrics
(e.g., # modifications to the file in a release)
- 4. Run detection

Defect Prediction Tool

- 1. Create tests
- 2. Run tests

Continuous Testing Tool

DEPLOY



Application Source Code

```
int div(a, b):  
return a/b
```

Possible division by zero

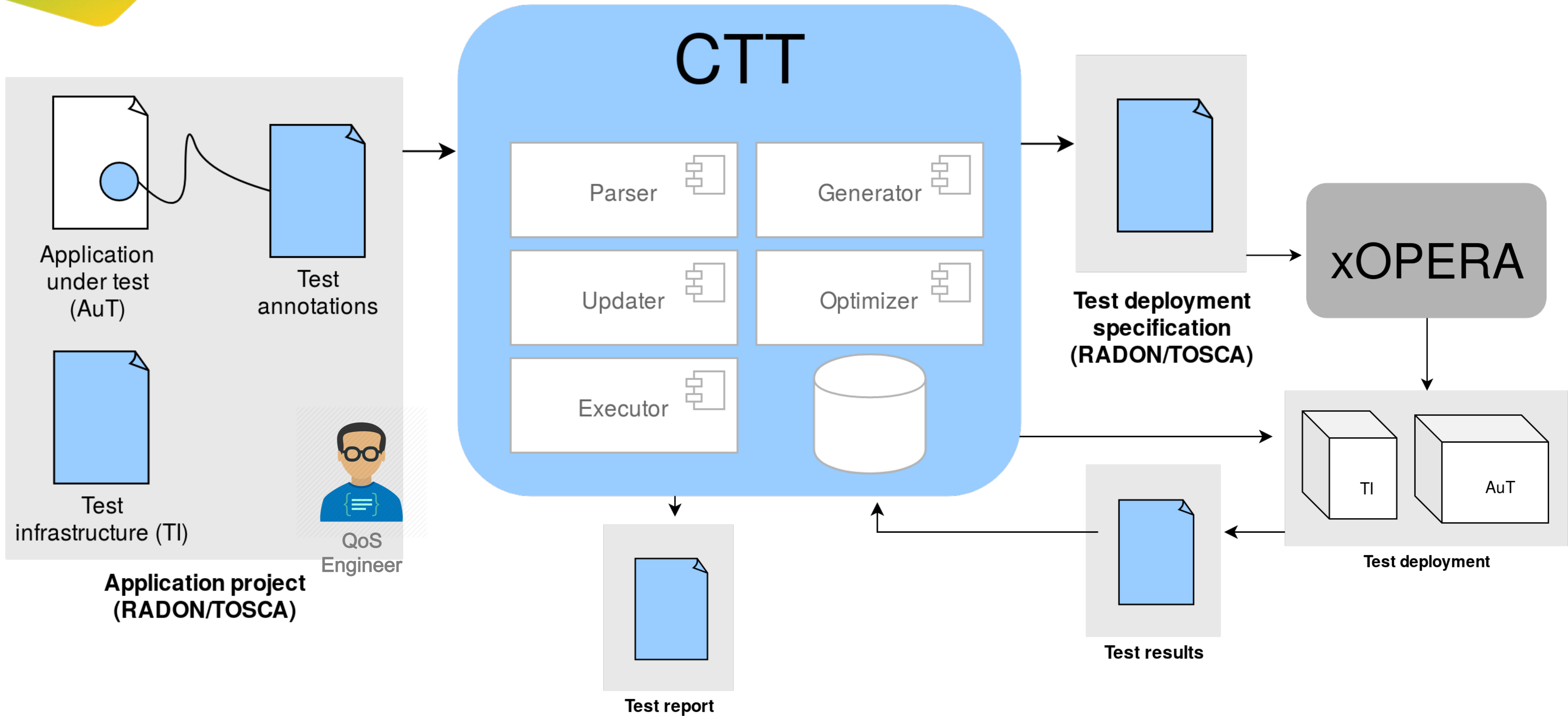
```
- name: "foo"  
include: es-template.yml  
when: es_templates  
when: es_templates | bool
```

Infrastructure Code

this makes the application behave wrongly



Continuous Testing

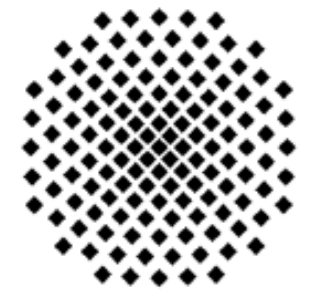


Imperial College
London



RADON

PRAQMA



Universität Stuttgart



CONTACT US



<http://radon-h2020.eu/>

<https://github.com/radon-h2020/>

twitter.com/RADON_2020

[linkedin.com/company/radon-2020/](https://www.linkedin.com/company/radon-2020/)

Demo

The image shows a YouTube video player interface. At the top left is the YouTube logo with 'DE' next to it. To the right is a search bar with the text 'Search' and a magnifying glass icon. The video player itself has a dark blue background. On the left side of the video frame, the RADON logo (a colorful geometric shape) is at the top, followed by the text 'RADON' in large white letters. Below that is 'Unlocking the benefits of serverless FaaS' in white, and 'for the European software industry' in a smaller, lighter blue font. At the bottom left of the video frame, it says '- Graphical Modelling'. On the right side of the video frame, there is a 3D illustration of a man in a suit standing on a glowing blue cube, surrounded by various digital devices like laptops and tablets, all connected by glowing lines and dots. A large play button icon is centered over the video frame. At the bottom of the video player, there is a control bar with play, next, volume, and progress (0:00 / 10:30) icons. To the right of the control bar are icons for closed captions, settings, and full screen. Below the video player, the video title 'RADON 2020 - Graphical Modelling' is displayed, followed by '20 views • Apr 6, 2020'. At the bottom right, there are icons for likes (0), dislikes (0), share, save, and a more options menu.

Demo

The image shows a YouTube video player interface. At the top left, there is a menu icon and the YouTube logo with 'DE' next to it. To the right is a search bar with the text 'Search' and a magnifying glass icon. The main video area features a dark blue background with the RADON logo (a colorful geometric shape) at the top left. Below the logo, the text reads 'RADON' in large white letters, followed by 'Unlocking the benefits of serverless FaaS' in white, and 'for the European software industry' in a smaller, lighter blue font. At the bottom left of the video area, it says '- Decomposition'. The video player includes a large play button in the center and a progress bar at the bottom with a play, pause, and volume icon, and a timestamp of '0:00 / 8:03'. On the right side of the progress bar, there are icons for closed captions, settings, a video thumbnail, a full screen icon, and a share icon. The video content itself is a 3D illustration of a man in a suit standing on a glowing blue cube, surrounded by various digital devices like laptops and tablets, all connected by glowing lines and dots, representing a serverless architecture.

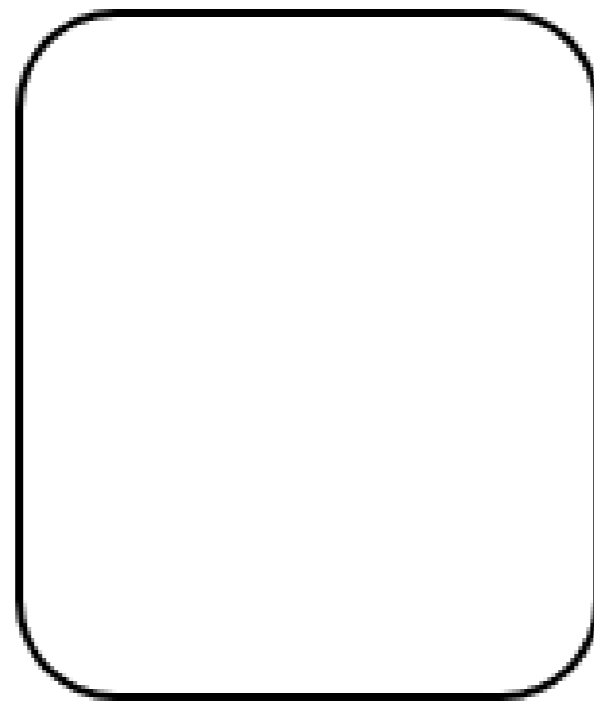
Demo

The image shows a YouTube video player interface. At the top left is the YouTube logo with 'DE' next to it. To the right is a search bar with the word 'Search' and a magnifying glass icon. The video content area features a dark blue background with the RADON logo (a colorful geometric shape) in the top left. Below the logo, the text reads 'RADON' in large white letters, followed by 'Unlocking the benefits of serverless FaaS' in white, and 'for the European software industry' in a smaller, light blue font. At the bottom left of the video area, it says '- Continuous Testing'. The video player includes a play button in the center, a progress bar at the bottom left showing '0:00 / 6:25', and control icons (CC, settings, full screen, etc.) at the bottom right. The video content itself is a 3D illustration of a man in a suit standing on a server rack, surrounded by laptops and data visualizations.

RADON: optimization & decomposition trade -offs

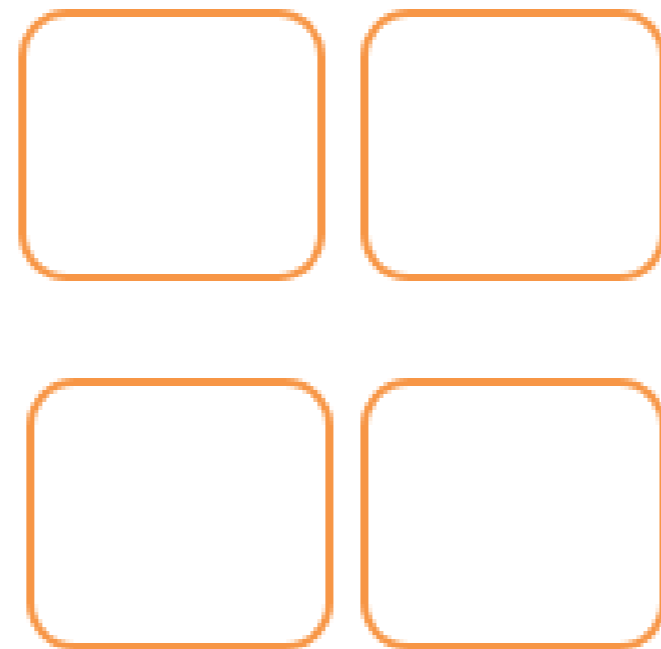
- What is the optimal size for a service taking into account for **constraints**?
- How do we converge through development cycles towards an **optimal architecture**?
- How to model and predict QoS?

Monolith



- + Simplified arch.
- + Less to deploy
- + Less to manage
- Inflexible
- Slow updates

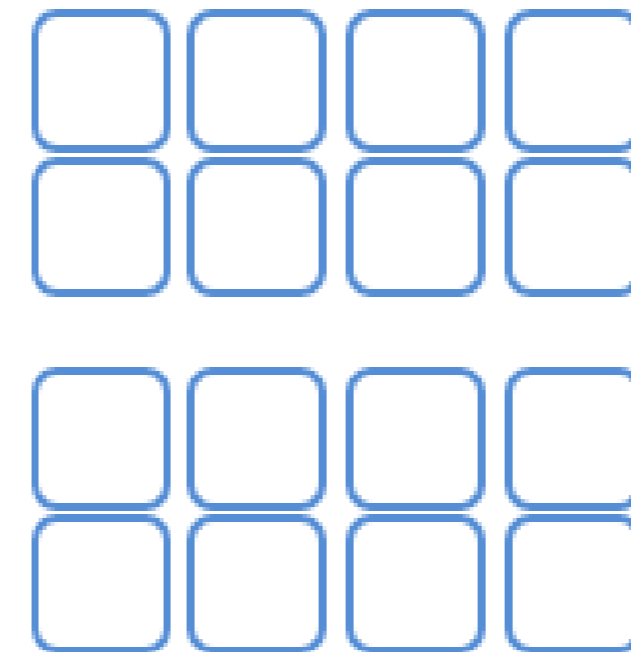
SOA



- + Separation of concerns
- + Specular to business
- Pre-cloud
- No infrastructure focus

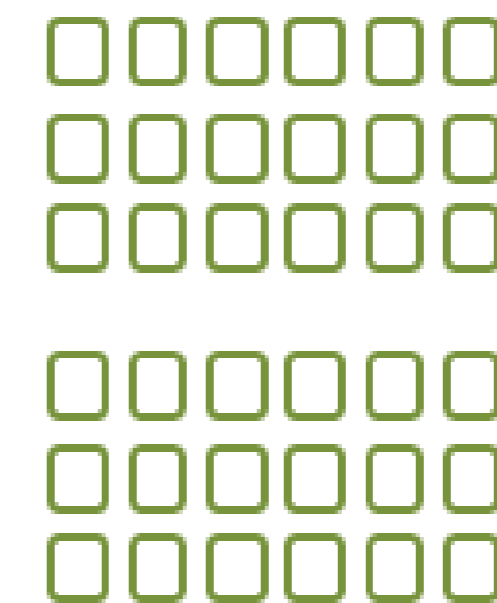
Microservices

Canonical (container based)



- + Container-based
- + Easy to migrate
- + Reproducible
- + Vendor-agnostic
- Manual admin
- Running costs

Serverless FaaS (platform)



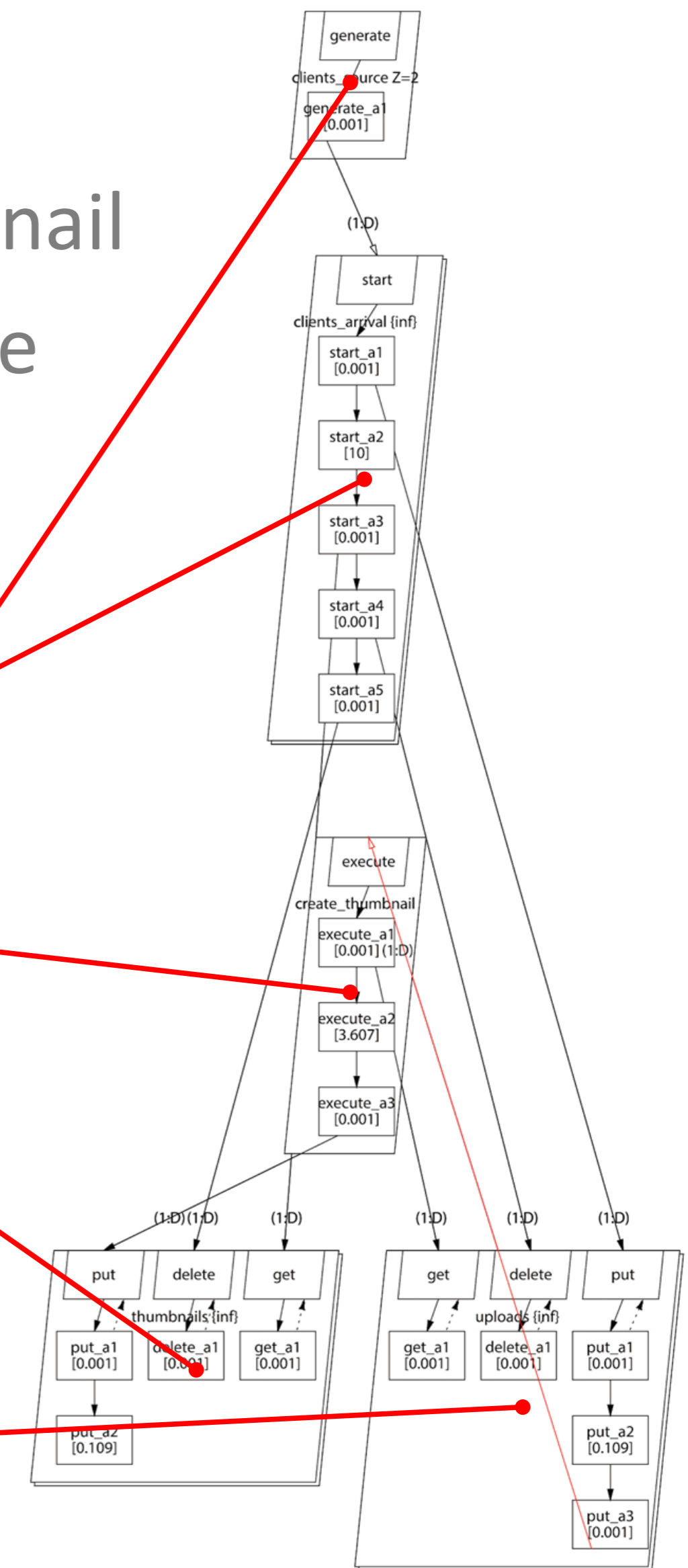
- + Scalability
- + Cost
- + Zero admin
- Resource limits
- Size limit
- Vendor lock-in

Decomposition & Optimization

- Optimization problem:
 - **type:** non-linear integer programming (NLIP);
 - **variables:** memory and concurrency;
 - **constraints:** average response time less than 2.5 sec
- Performance modeling:
 - **benchmarking:** service demand estimation
 - **formalism:** layered queueing networks (LQNs)

- LQN for the thumbnail generation example (simplified)

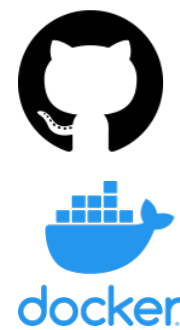
clients
(open workload)
create_thumbnail
(Lambda function)
thumbnails
(S3 bucket)
uploads
(S3 bucket)



T3.3: Continuous Testing

Results after Y1

- Design of continuous testing workflow, tool architecture, and integration;
- Support for modeling and executing selected test types (focus: performance tests)
- Initial research contributions on:
 - Tailored testing (MASCOTS 19);
 - Regression testing of microservices (accepted for ICPE 20);
- Application to RADON examples (SockShop and Thumbnail);
- Started interaction with use cases (ATC, PRQ) and tools (monitoring, CI/CD);
- Prototypes being made available as open-source:
 - CTT server: <https://github.com/radon-h2020/radon-ctt>
 - CTT agent: <https://github.com/radon-h2020/radon-ctt-agent>
 - <https://hub.docker.com/r/ustctt/>



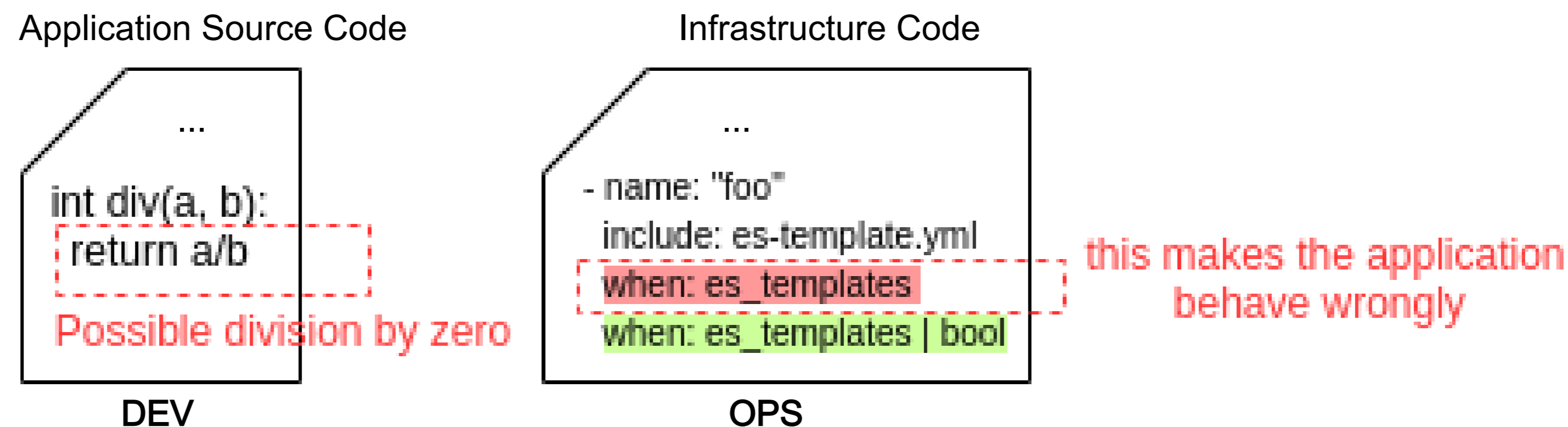
T3.4: Defect Prediction Tool

Why?

“Infrastructure-as-code (IaC) ⇒ managing and provisioning compute datacenters through machine-readable definition files”

Cit. TOSCA Simple Profile Yaml v1.3, CSD2

- As any other source code artifact, IaC files may contain defects that can preclude their correct functioning and operations;



- The tool is intended for detecting defect-prone IaC blueprints at the end of a release cycle;
- Defect-Prediction SoTA from Dev. sourcecode is well-established in the use of MachineLearning techniques:
 - Scripts prone to contain imperfections or deficiencies cause them not to meet their requirements or specifications;
 - Metrics identify such qualities, so that smells or bug-proneness can be detected and possibly repaired;

Continuous Testing Tool (CTT)

- Functionalities grouped into 3 usage scenarios:
 - i. Test case definition
 - ii. Test execution
 - iii. Test maintenance
- CTT modules
 - i. Microservices/FaaS
 - ii. Data pipelines
- Usage:
 - i. Standalone tool (open-source)
 - ii. Invocation via the RADON IDE or CI/CD

