



RADON

RADON: DevOps for Serverless Computing

Speaker & Project coordinator:

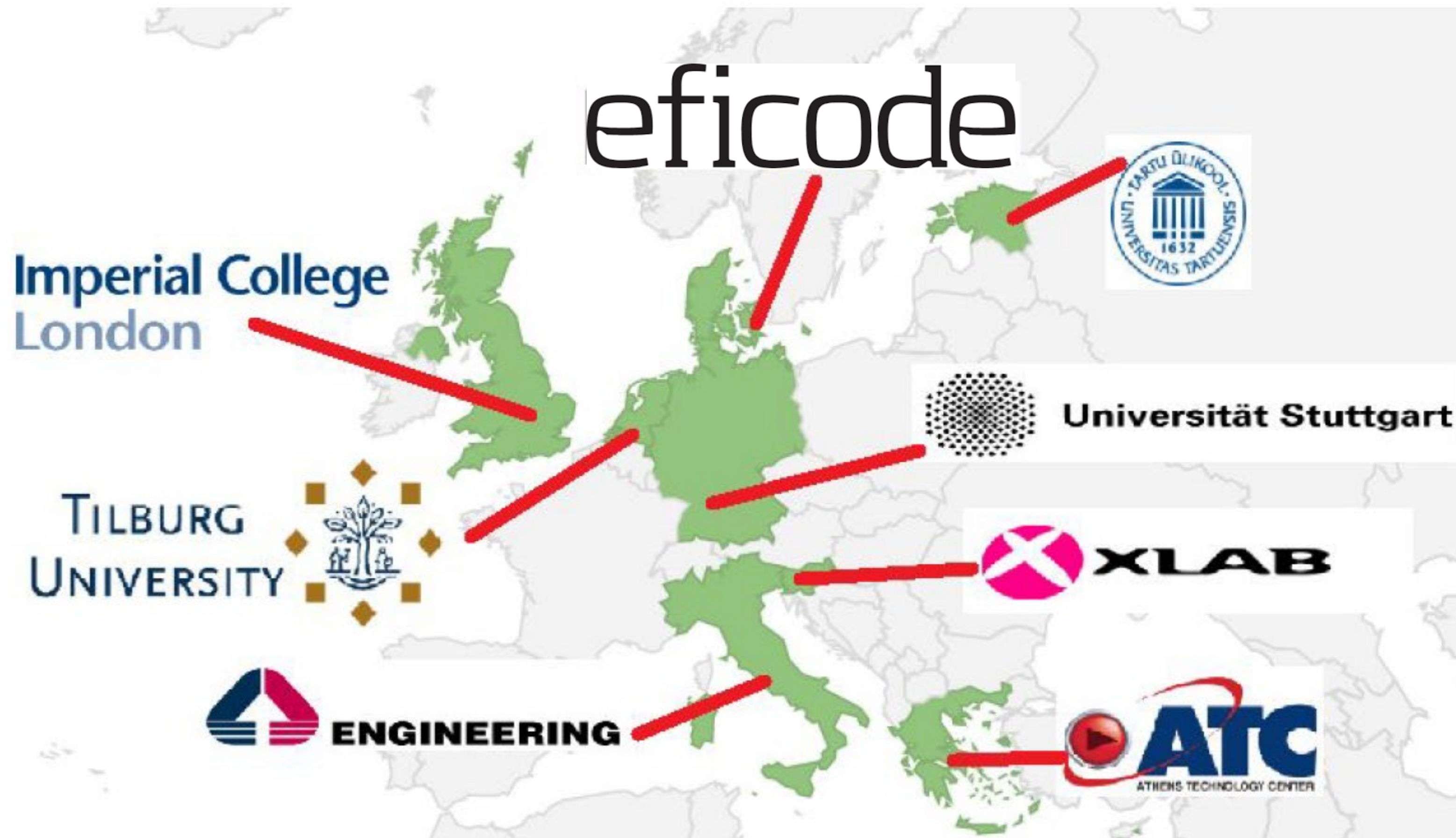
Giuliano Casale

Imperial College London

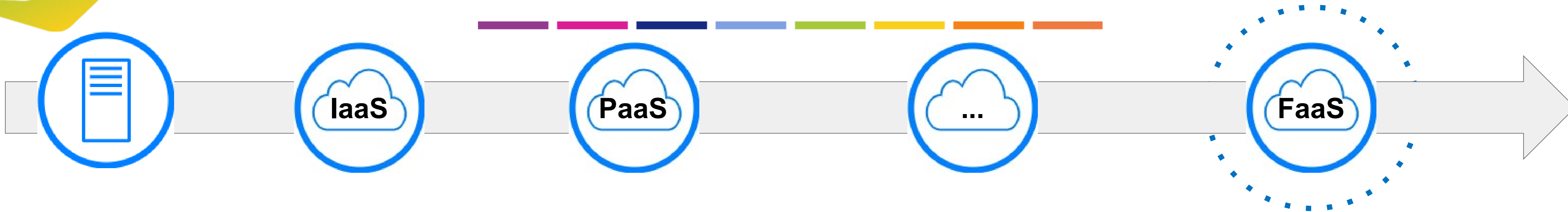
RADON Consortium



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



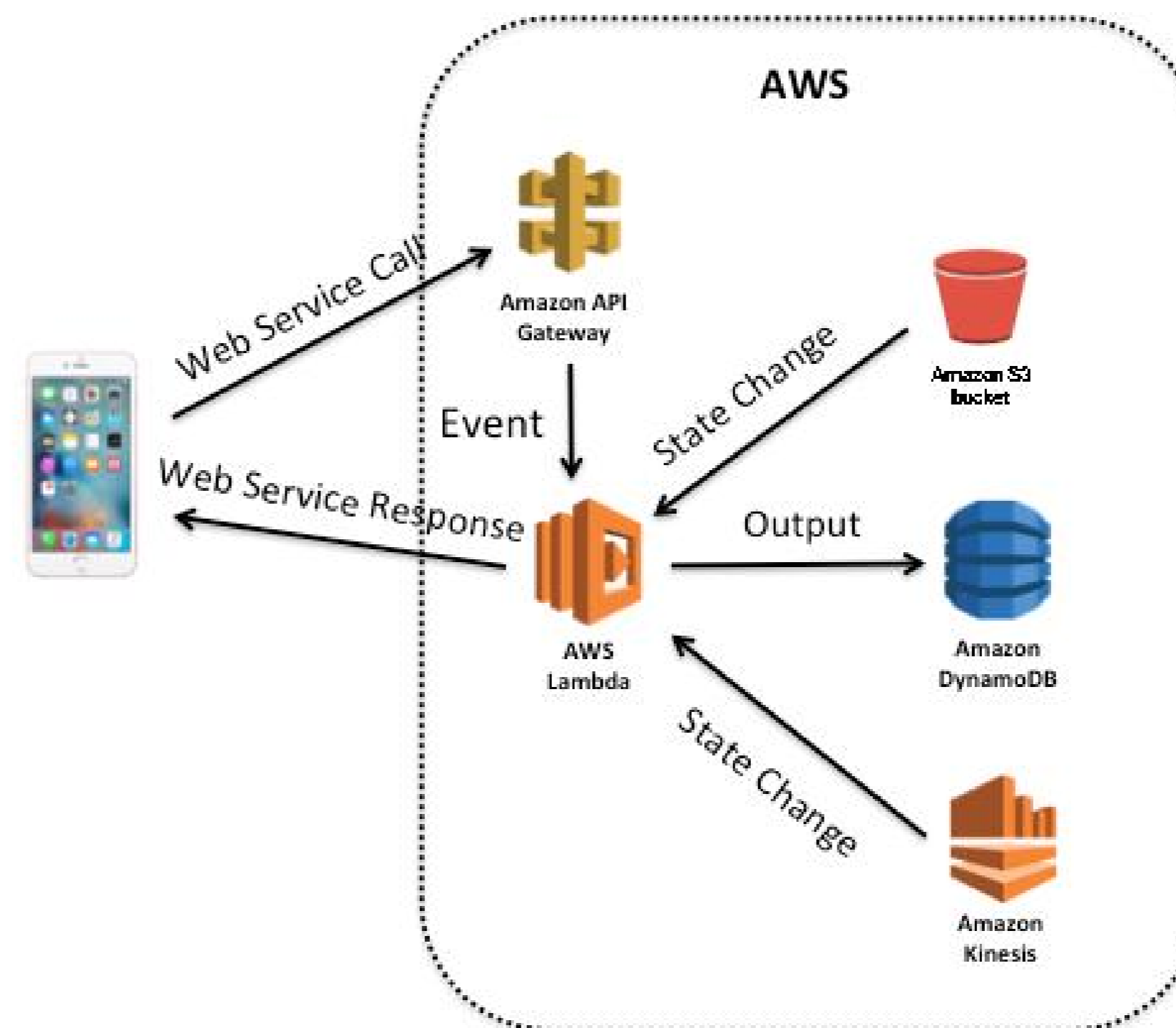
The Rise of Serverless FaaS



Serverless Function-as-a-Service (FaaS)

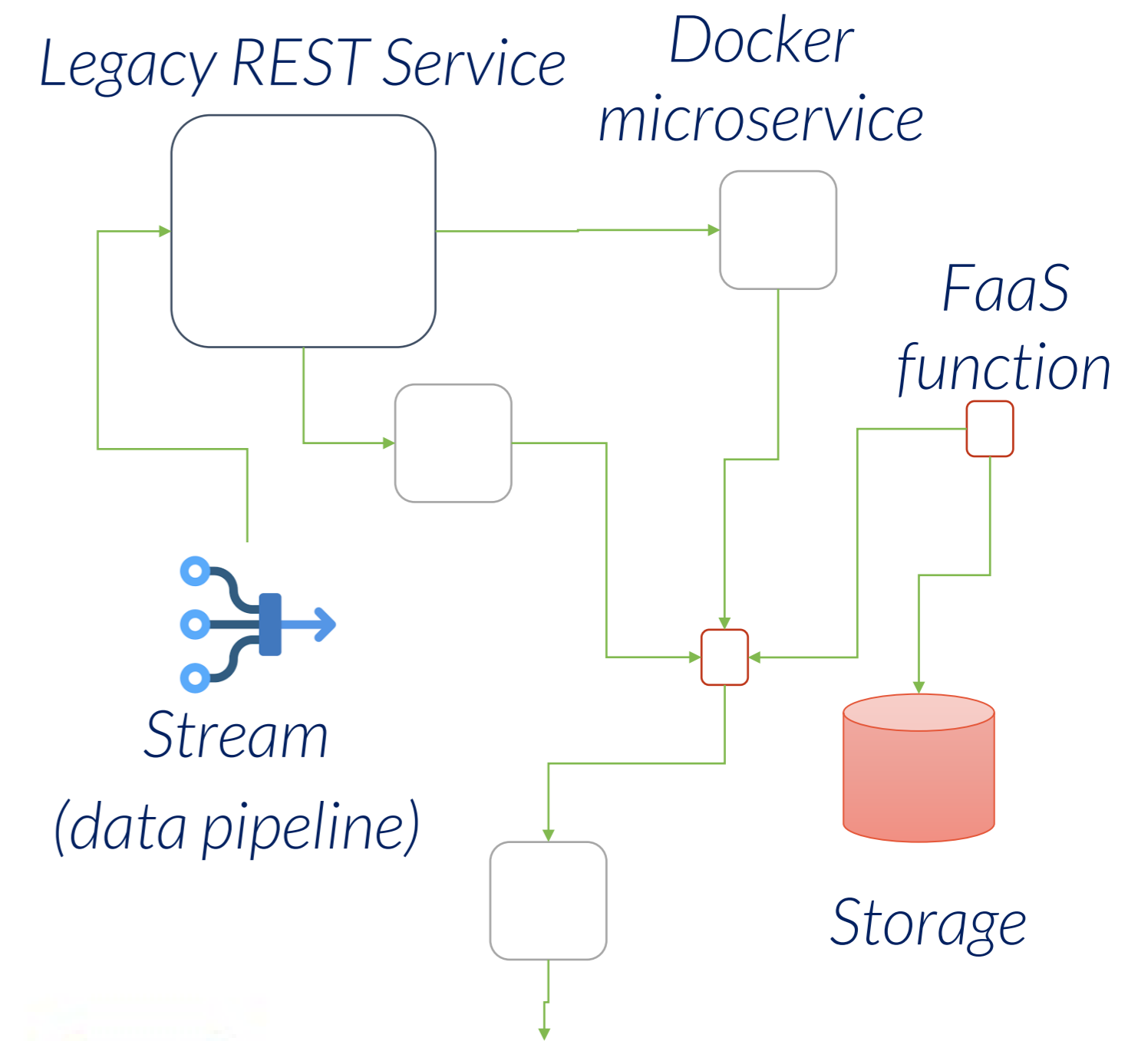
- Fine-grained functions hosted in the cloud and fully managed by the provider
- Cost-savings in event-driven workloads (e.g., IoT)
- Strong synergy with microservices
- Resource decoupling
 - stateless functions
 - state persisted via storage
 - state change can produce events

Serverless/FaaS Architecture



Challenges in DevOps for Serverless

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



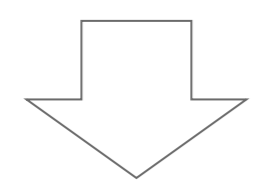
Benefits for the end user



A DevOps framework to help developing FaaS-based products without code & data lock-in

Embrace a technology shift

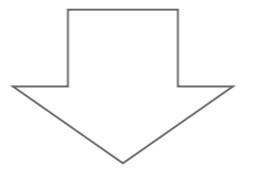
DevOps methods and tools to automatically build & deploy microservices & FaaS



Reduce the learning curve

Low cost and low risk

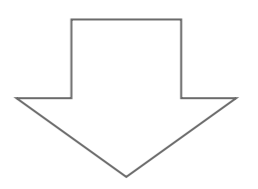
Multi-cloud serverless FaaS
Move/synch data between clouds
Compare and test alternatives and their costs



Select best vendor yet avoiding code & data lock-in risks

Embrace portability across business use cases

Hw/platform independent
TOSCA model facilitate reuse and customization
Multi-tenant orchestration



Model-based approach

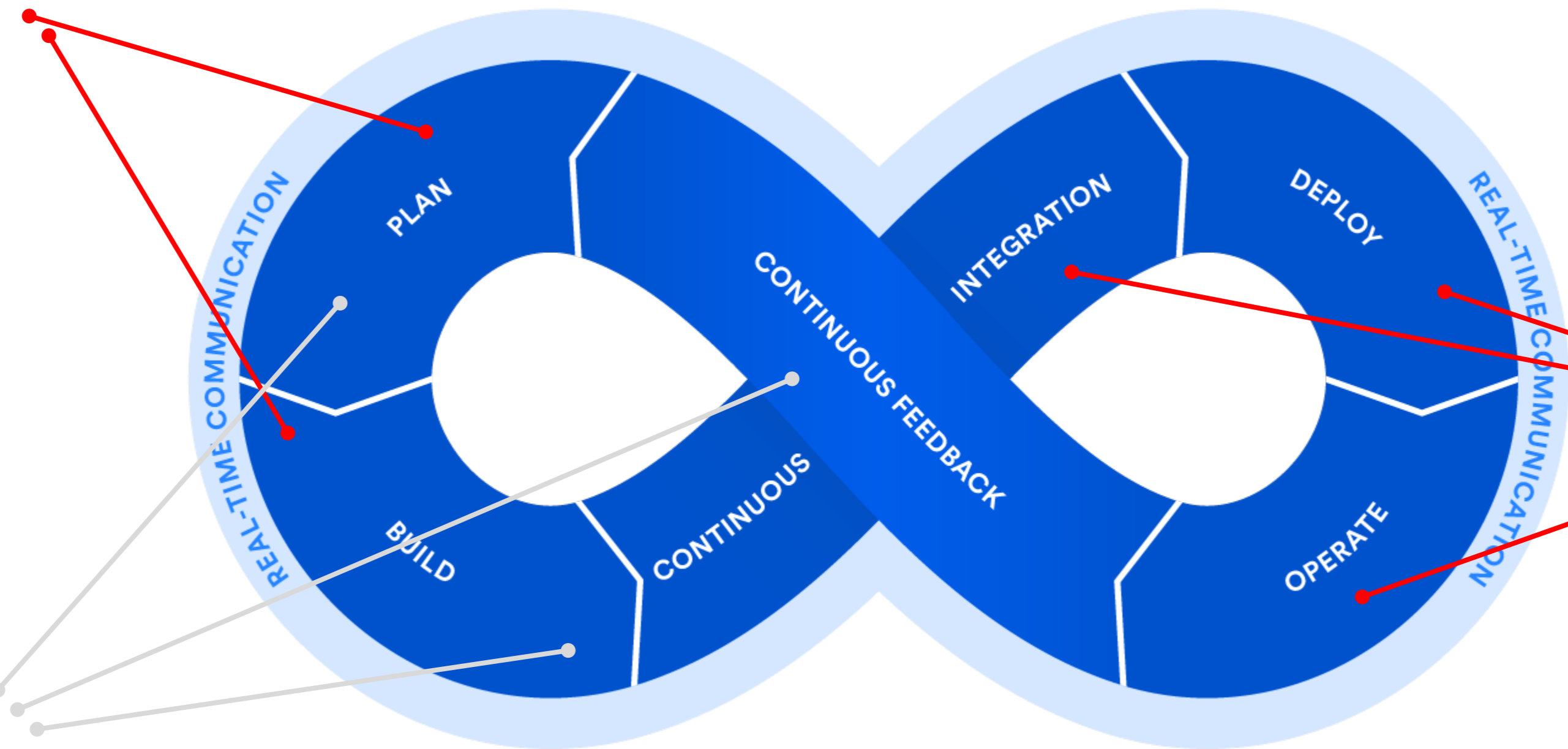
RADON: modelling for serverless FaaS



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

1

Graphical Modeling



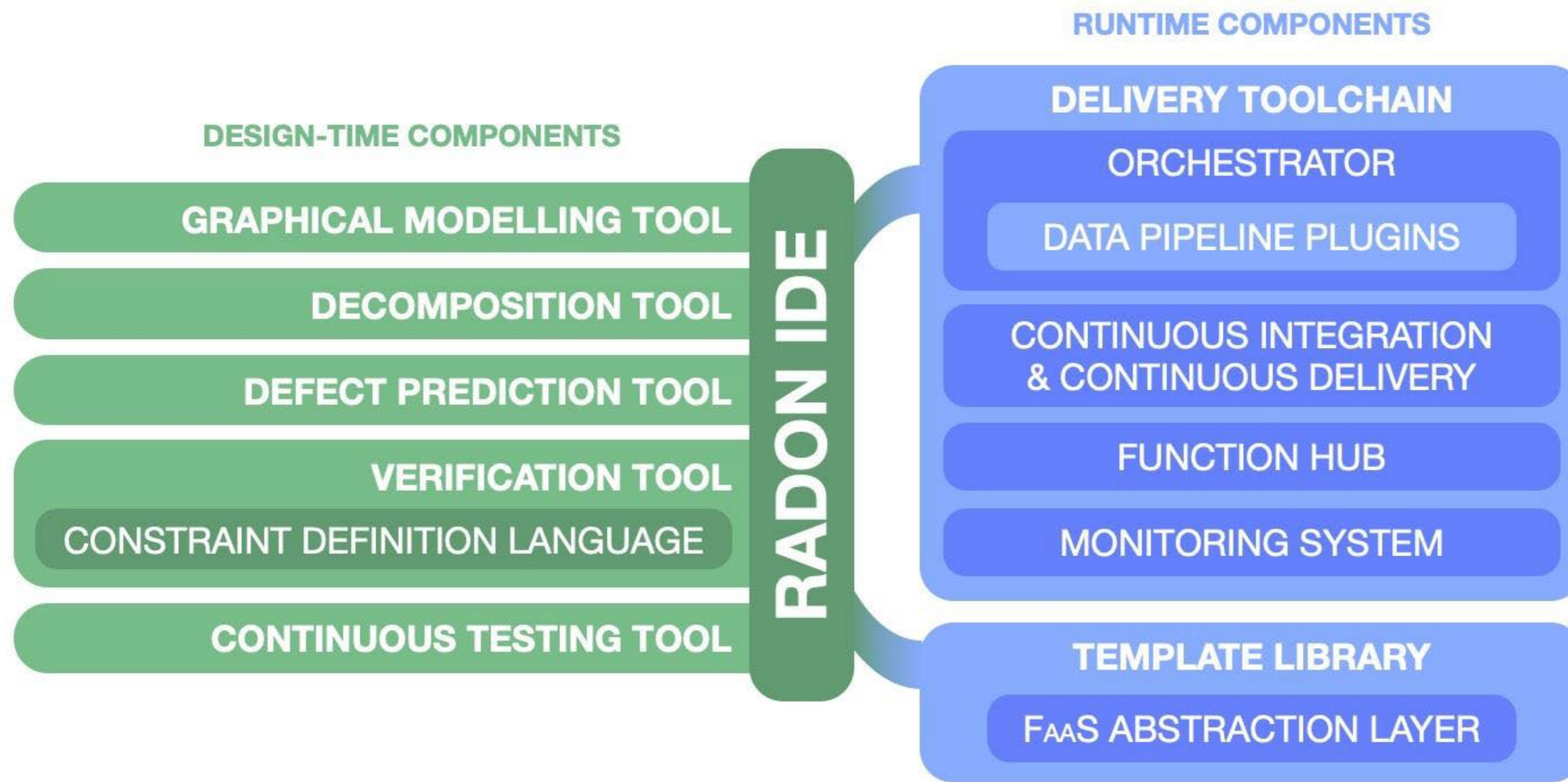
2

Model-based Delivery

3

Quality Engineering

RADON Framework



- Technical objectives:
 - A single **multi-user environment** to access all RADON artifacts;
 - A **front-end of the RADON methodology** enabling users to invoke RADON tools;

RADON IDE / Eclipse Che

The screenshot shows the Eclipse Che dashboard at the URL `che-che.217.172.12.178.nip.io/dashboard/#/getstarted?tab=getStarted`. The dashboard features a sidebar on the left with navigation options: Workspaces, Get Started, Stacks, Factories, and Administration. The main content area displays a grid of development stacks. The 'RADON Workspace' card, which includes the RADON logo and the text 'RADON Workspace' and 'RADON Stack', is highlighted with a red border. Other visible stacks include NodeJS Express Web Application, Java with Spring Boot and MySQL, Java Gradle, Java Vert.x, NodeJS React Web Application, Apache Camel K, Java Spring Boot, Python Django, PHP Laravel with MySQL, Quarkus Tools, and NodeJS Web Application based on Yarn. The browser's address bar and tabs are visible at the top, and the system tray at the bottom shows the user 'Damian Damian Ta...' and the Eclipse Che version '7.14.1'.

RADON IDE Integration



RADON Kubernetes components

The screenshot displays the RADON IDE interface with three main components highlighted by green arrows:

- RADON Plugins:** A context menu is open over a terminal window titled '>RADON'. The menu items are:
 - RADON: Open Help Page
 - RADON: Open Monitoring page
 - RADON: Show Deployment Status
 - RadonCTT: Execute test configuration
- RADON Workspace:** A red arrow points to the 'Template library set REST API endpoint' option in the file explorer's context menu.
- RADON Kubernetes components:** A green arrow points to the 'Plugins' folder in the workspace view on the right side of the IDE.

RADON Graphical Modeling

The image displays two side-by-side screenshots of the 'winery' graphical modeling tool. The left screenshot shows the 'Hide Palette' view, and the right screenshot shows the 'Open Palette' view.

Left Screenshot (Hide Palette):

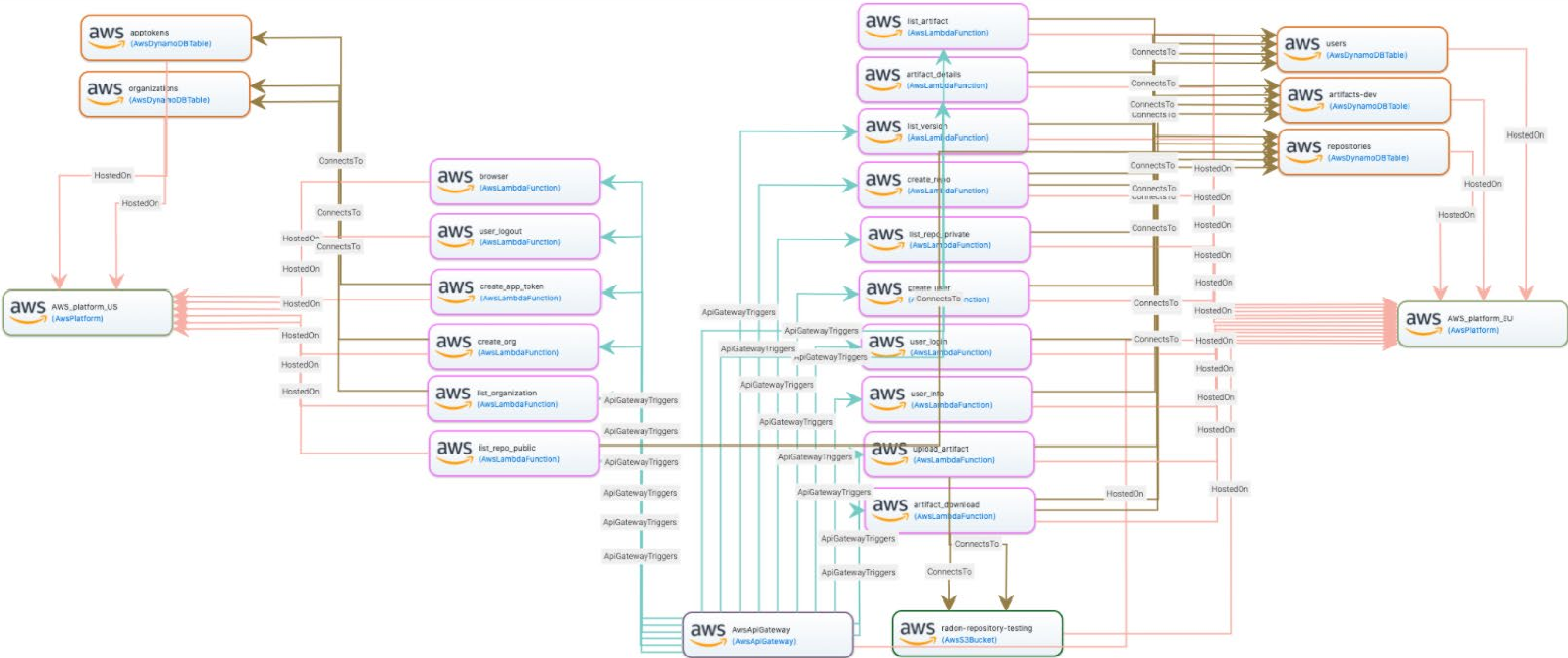
- Toolbar:** Save, Manage, Layout, Align, Import Topology, Types, Properties.
- Component Palette:** Lists various nodes such as `radon.nodes.abstract`, `radon.nodes.apache.kafka`, `radon.nodes.aws`, `radon.nodes.azure`, `radon.nodes.docker`, `radon.nodes.google`, `radon.nodes.java`, `radon.nodes.mongodb`, `radon.nodes.mysql`, `radon.nodes.nodejs`, and `radon.nodes.openfaas`.
- Component Details:** Shows an `aws AwsLambdaFunction (AwsLambdaFunction)` component with a `HostedOn` relationship pointing to an `aws AwsPlatform (AwsPlatform)` component. The `AwsPlatform` component has a table of properties:

Key	Values
<code>access_key_id</code>	TBD
<code>secret_access_key</code>	TBD
<code>name</code>	Edit your value here.
<code>region</code>	eu-west-1

Right Screenshot (Open Palette):

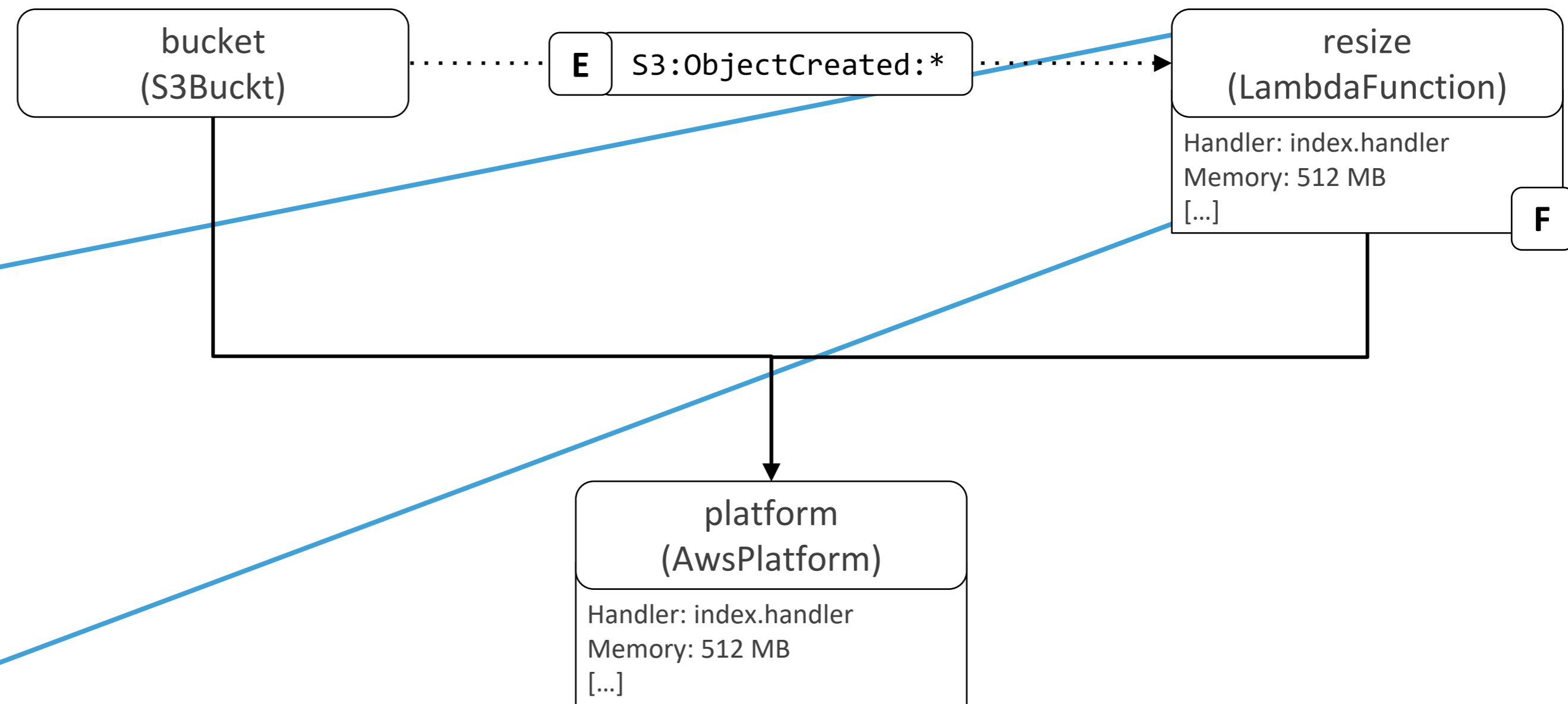
- Toolbar:** Save, Manage, Layout, Align, Import Topology, Ids, Types, Properties, Deployment Artifacts, Requirements & Capabilities.
- Component Palette:** Includes buttons for `Determine Stateful Components`, `Determine Freezable Components`, `Clean Freezable Components`, and `Refine Pattern`.
- Graph:** A dependency graph showing relationships between components:
 - `S3Bucket_1.0.0-w1-wip1 (S3Bucket_1.0.0-w1-wip1)` triggers `LambdaFunction_1.0.0-w1-wip1 (LambdaFunction_1.0.0-w1-wip1)` via `Triggers_1.0.0-w1-wip1`.
 - `S3Bucket_1.0.0-w1-wip1` is hosted on `AwsPlatform_1.0.0-w1-wip1 (AwsPlatform_1.0.0-w1-wip1)` via `HostedOn_1.0.0-w1-wip1`.
 - `LambdaFunction_1.0.0-w1-wip1` connects to `AwsPlatform_1.0.0-w1-wip1` via `ConnectsTo_w1-wip1`.
 - `LambdaFunction_1.0.0-w1-wip1` is hosted on `AwsPlatform_1.0.0-w1-wip1` via `HostedOn_1.0.0-w1-wip1`.

RADON Graphical Modeling



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
```

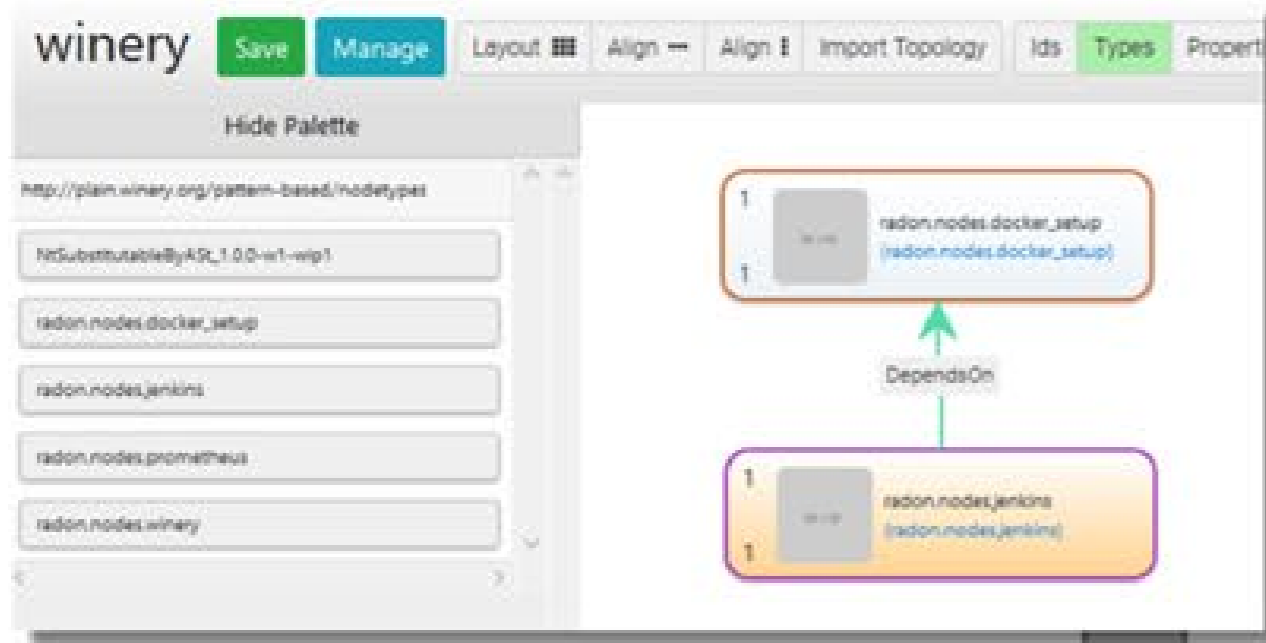
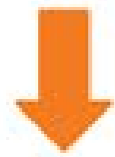


Models **automatically deployable** using the RADON orchestrator.
Compatible with OASIS TOSCA standard.

RADON Delivery toolchain

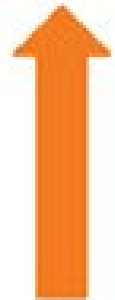


Start here



Graphical tool

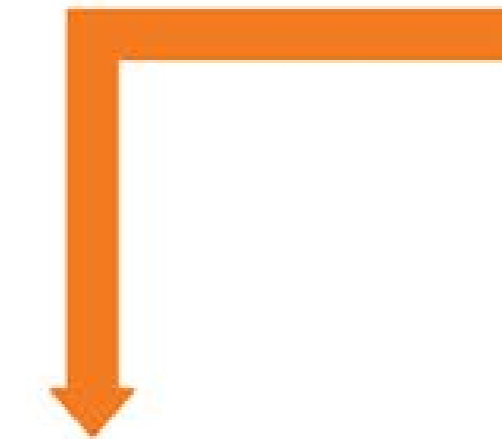
Template library




RADON Data Pipeline



Orchestrator
xOpera



Monitoring

Prometheus



Quality Guardrails in RADON



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

Verification Tool

[All constraints verified]

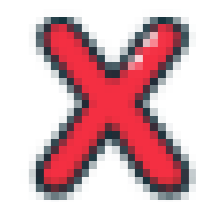
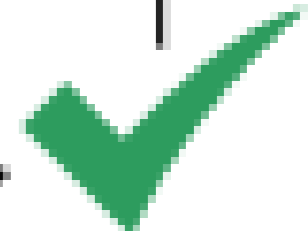
- 1. Extract *product* metrics
(e.g. # lines of code)
- 2. Extract *delta* metrics
(between two successive releases)
- 3. Extract *process* metrics
(e.g., # modifications to the file in a release)
- 4. Run detection

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

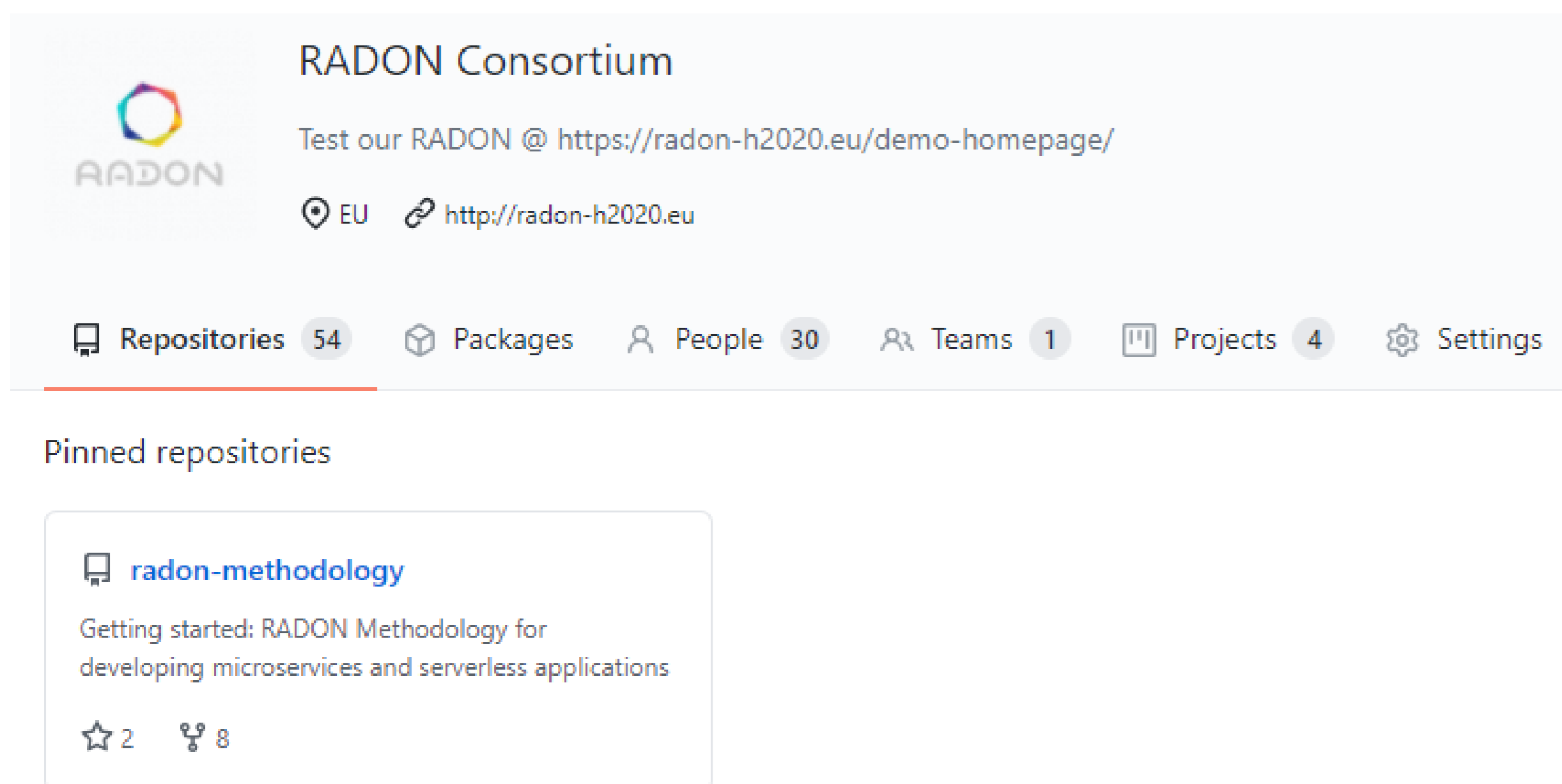
```
- name: "foo"  
include: es-template.yml  
when: es_templates  
when: es_templates | bool
```

Infrastructure Code

this makes the application behave wrongly



More about RADON



RADON Consortium

Test our RADON @ <https://radon-h2020.eu/demo-homepage/>

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

Repositories 54 Packages People 30 Teams 1 Projects 4 Settings

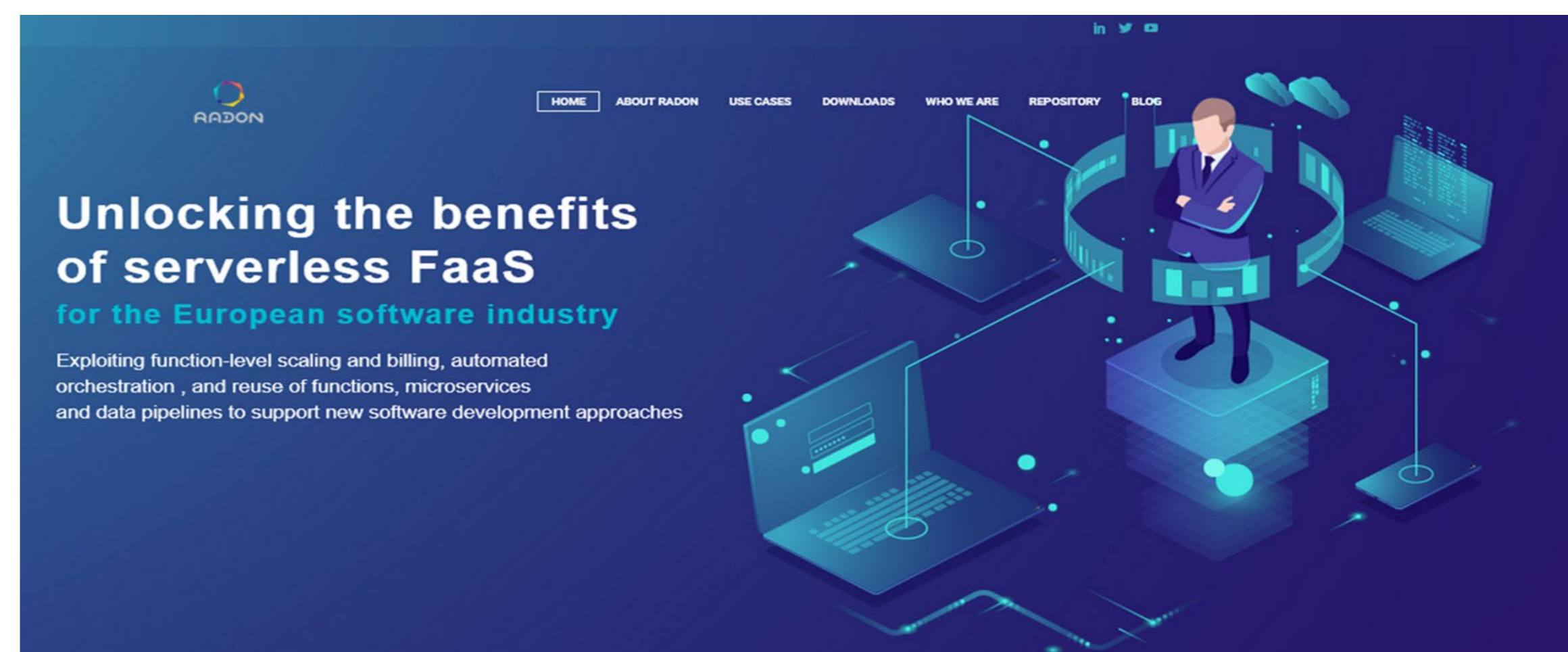
Pinned repositories

radon-methodology

Getting started: RADON Methodology for developing microservices and serverless applications

☆ 2 🍴 8

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



Unlocking the benefits of serverless FaaS for the European software industry

Exploiting function-level scaling and billing, automated orchestration, and reuse of functions, microservices and data pipelines to support new software development approaches

Navigation: HOME ABOUT RADON USE CASES DOWNLOADS WHO WE ARE REPOSITORY BLOG

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

cf. YouTube videos

Imperial College
London



RADON

eficode



CONTACT US



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

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

twitter.com/RADON_2020
linkedin.com/company/radon-2020/