



RADON Webinar III

RADON Data Pipelines



Agenda



- Introduction
 - Data Pipelines
 - RADON framework
 - OASIS TOSCA
- RADON data pipeline models
- Tutorial demo

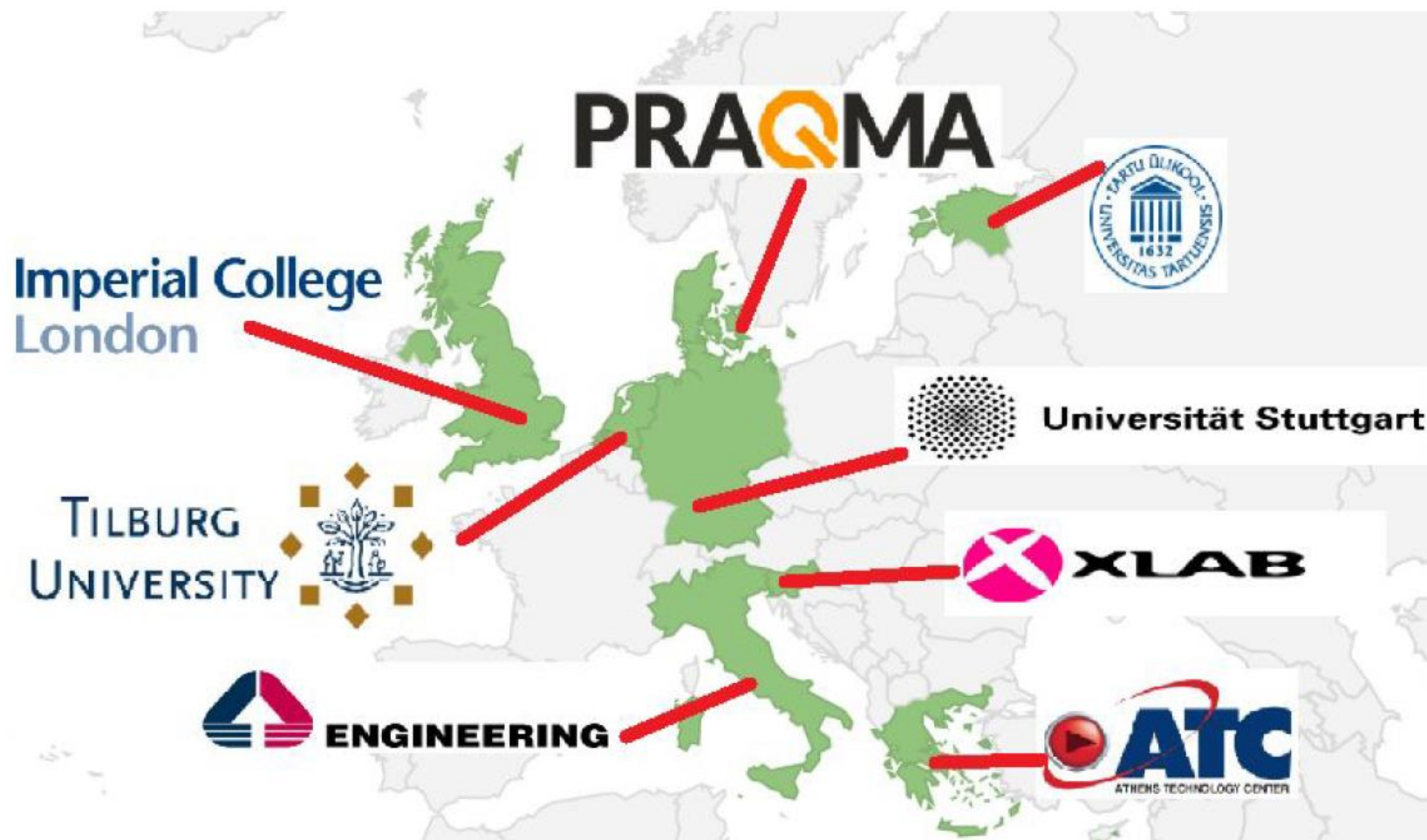


RADON

Rational Decomposition and Orchestration for Serverless Computing

Value proposition:

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

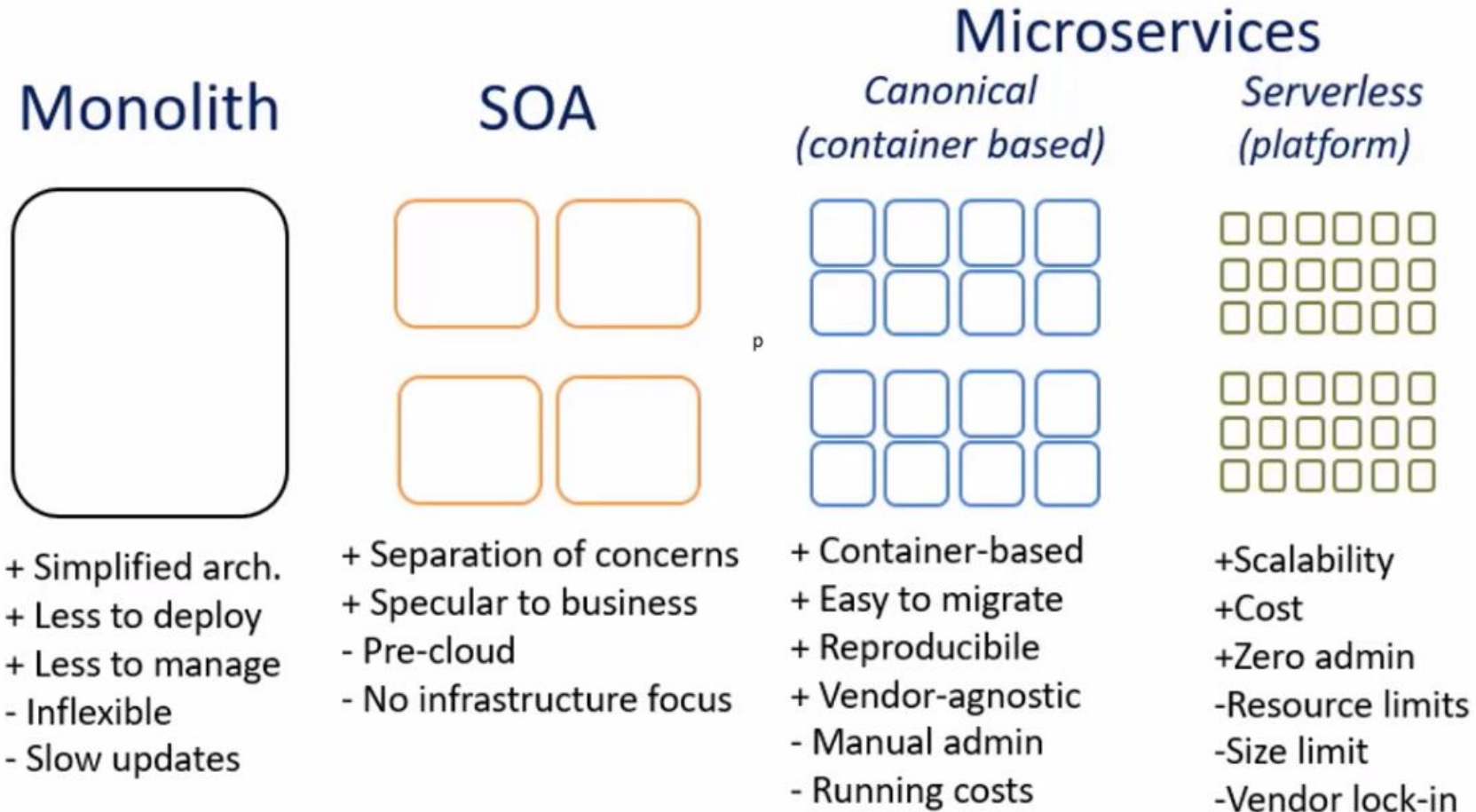


RADON consortium

5



From Monoliths to Serverless





RADON

Intro: TOSCA

TOSCA



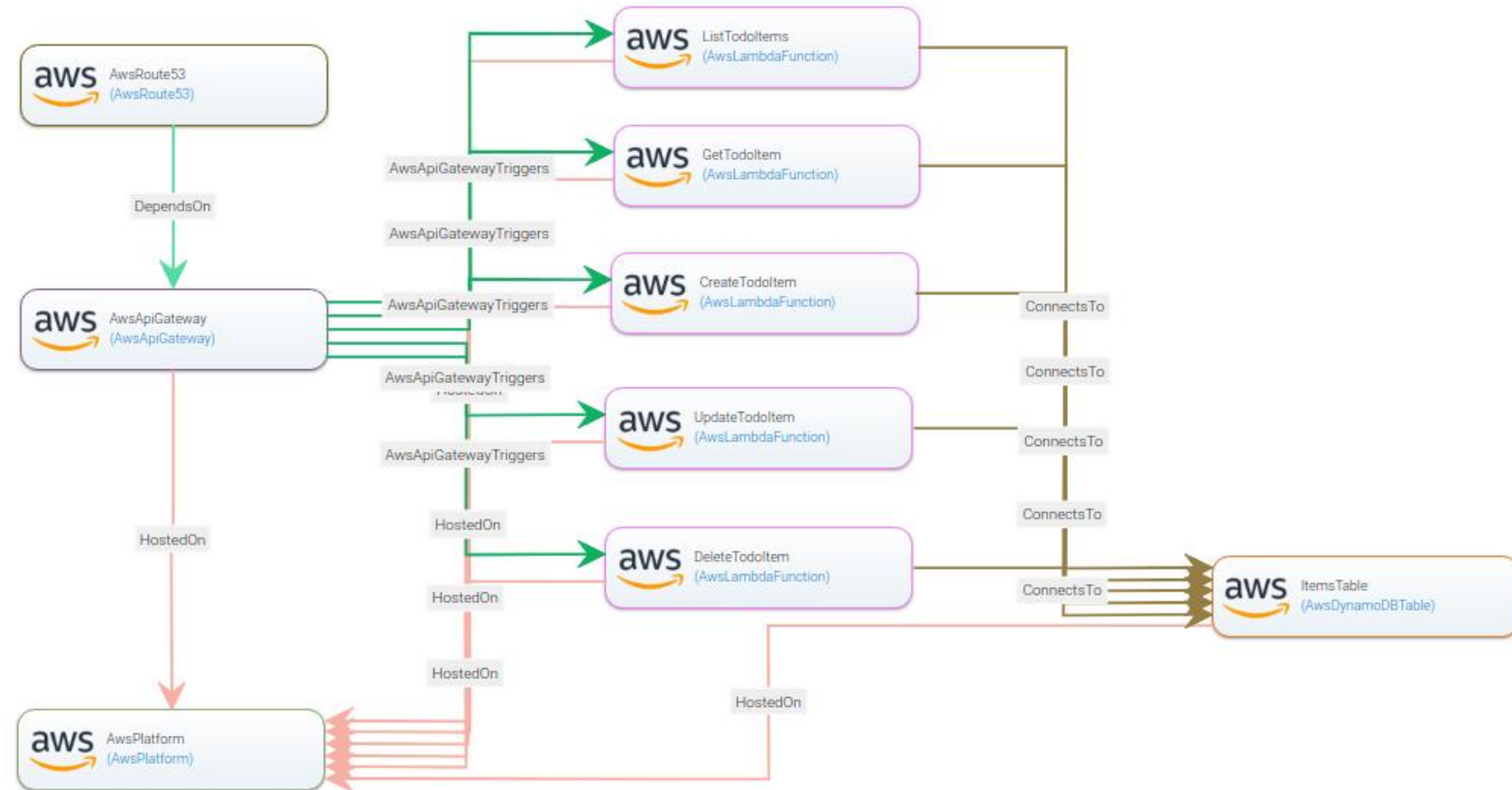
- Topology and Orchestration Specification for Cloud Applications (TOSCA)
- Standard for automating the deployment and management of cloud applications
- Started by major companies, such as IBM.
- Major goals:
 - Automation of Deployment and Management
 - Portability
 - Interoperability
 - Vendor-neutral ecosystem

TOSCA Example

```

1  tosca_definitions_version: tosca_simple_yaml_1_3
2
3  metadata:
4    targetNamespace: "radon.blueprints"
5  topology_template:
6    node_templates:
7      AwsLambdaFunction_1:
8        type: radon.nodes.aws.AwsLambdaFunction
9        metadata:
10         x: "894"
11         y: "207"
12         displayName: "GetTodoItem"
13        properties:
14         handler: "get.handler"
15         memory: 128
16         name: "get-item"
17         runtime: "nodejs12.x"
18         alias: "dev"
19         statement_id: "get-stmt"
20         zip_file: { get_artifact: [ SELF, get_item ] }
21         timeout: 300
22         env_vars:
23           TODOS_TABLE: "items"
24        requirements:
25         - endpoint:

```



For more info on TOSCA, check previous webinars in the **RADON H2020** [Youtube channel](#)



RADON

Intro: Data Pipelines

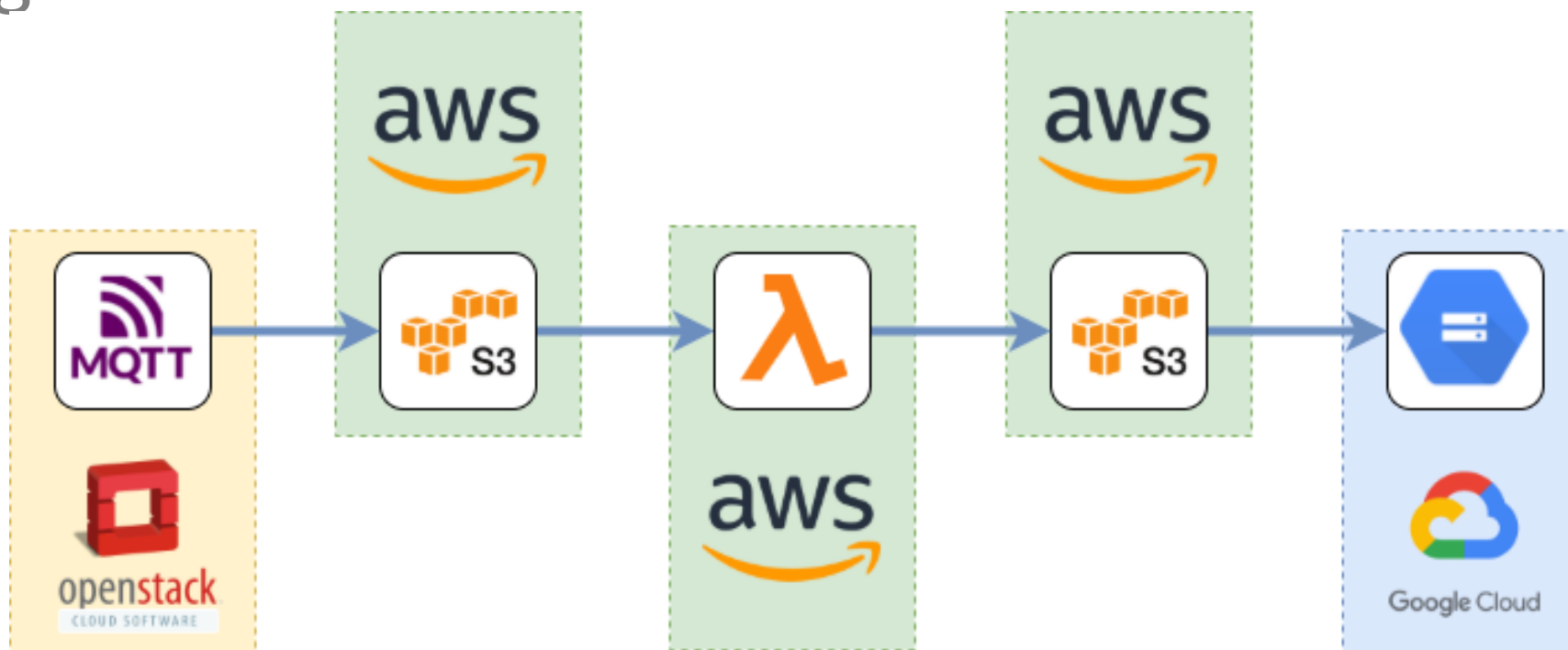
Data Pipelines



- Services for automating the data migration, processing and storage across multi-cloud environments
- Support distributed FaaS applications
 - Where functions are deployed on-premise, HPC or on different cloud providers
- **Goal:** Re-useable data pipeline services that can be composed into more complex data management pipelines

Example data pipeline

Real-time data migration from Openstack to AWS to Google Cloud

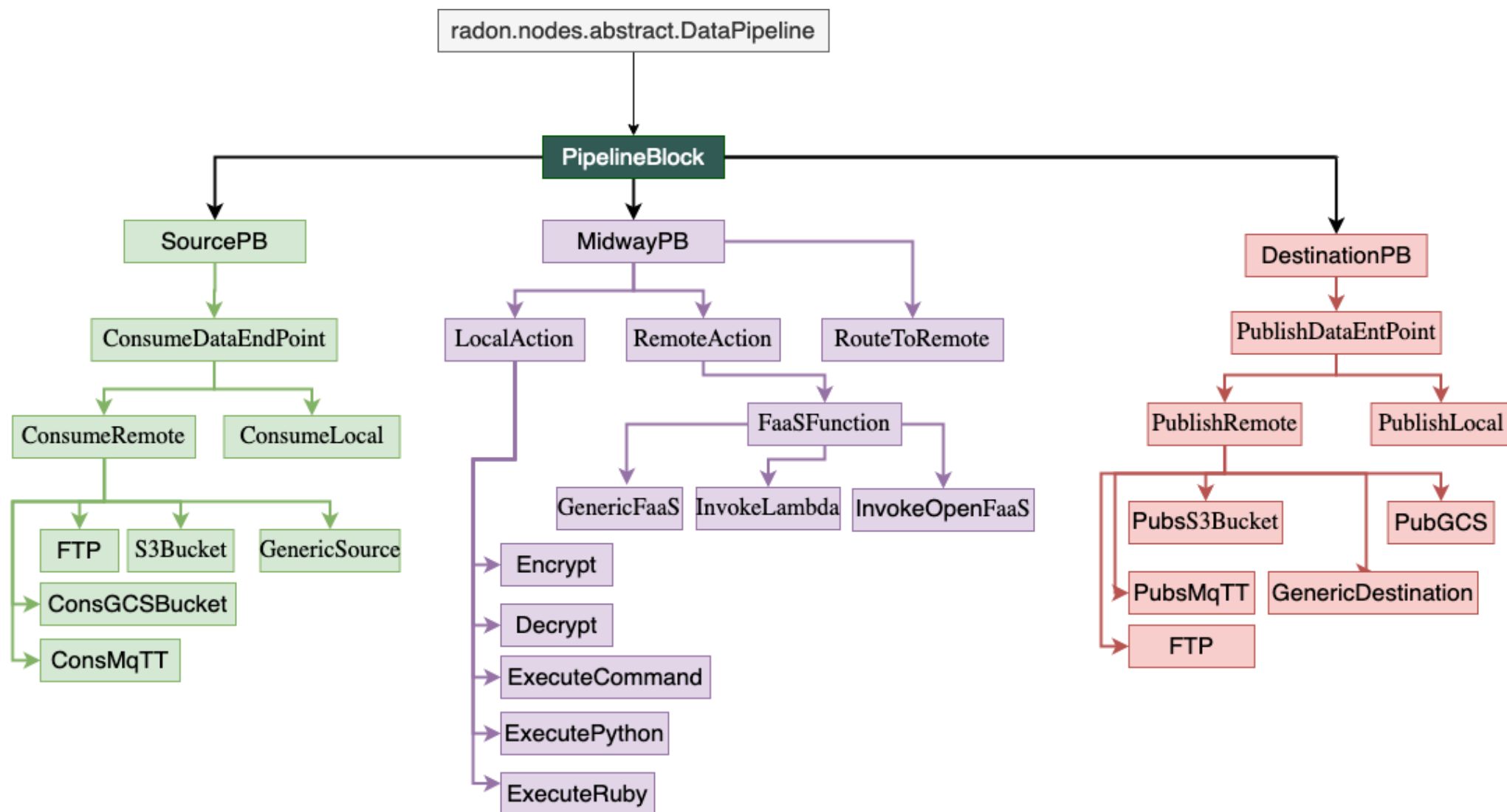


RADON data pipelines



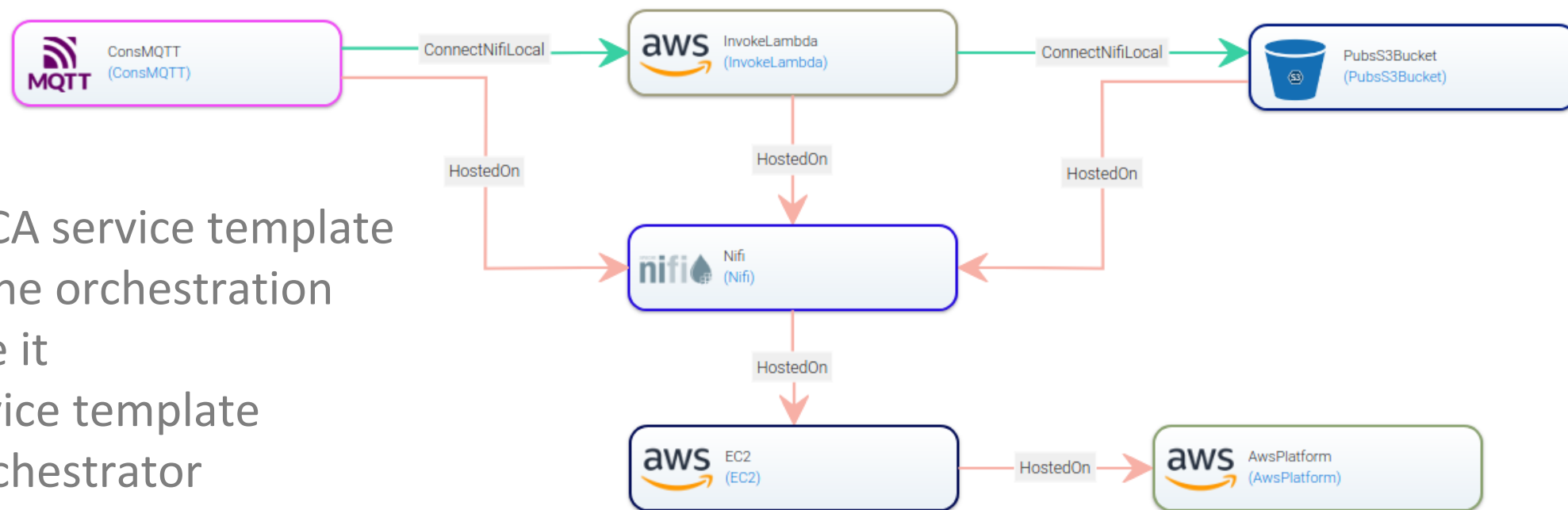
- Extend TOSCA with support for modelling data migration and management across multi-cloud environments
- Provide a set of reusable node types which can be composed into data pipelines
- Data automation is implemented by:
 - Apache NiFi
 - AWS Data pipeline service
- Ansible is used to install, configure and start the underlying services and cloud resources

RADON data pipeline models



Using RADON tools

- Access RADON web IDE
- Use **RADON GMT** to design the Data pipeline
- Define where it is deployed: in AWS EC2 VM, OpenStack VM or as a Docker container



- Download TOSCA service template
- Use Data pipeline orchestration plugin to validate it
- Deploy the service template using RADON Orchestrator

RADON IDE

The screenshot displays the RADON IDE interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The left sidebar shows the EXPLORER view with a tree of projects and files. The main editor area displays the README.md file for the RADON project. The right sidebar shows the MY WORKSPACE: WORKSPACE view with a list of user runtimes and plugins.

EXPLORER

- PROJECTS
 - .radonCTT
 - radon-particles M
 - .github
 - artifacttypes M
 - capabilitytypes
 - datatypes
 - docs
 - grouptypes
 - interfacetypes
 - nodetypes
 - policytypes
 - relationshipatypes
 - servicetemplates
 - .editorconfig
 - .gitattributes
 - .gitignore
 - gmt-docker-comp...
 - Jenkinsfile
 - LICENSE
 - README.md

README.md

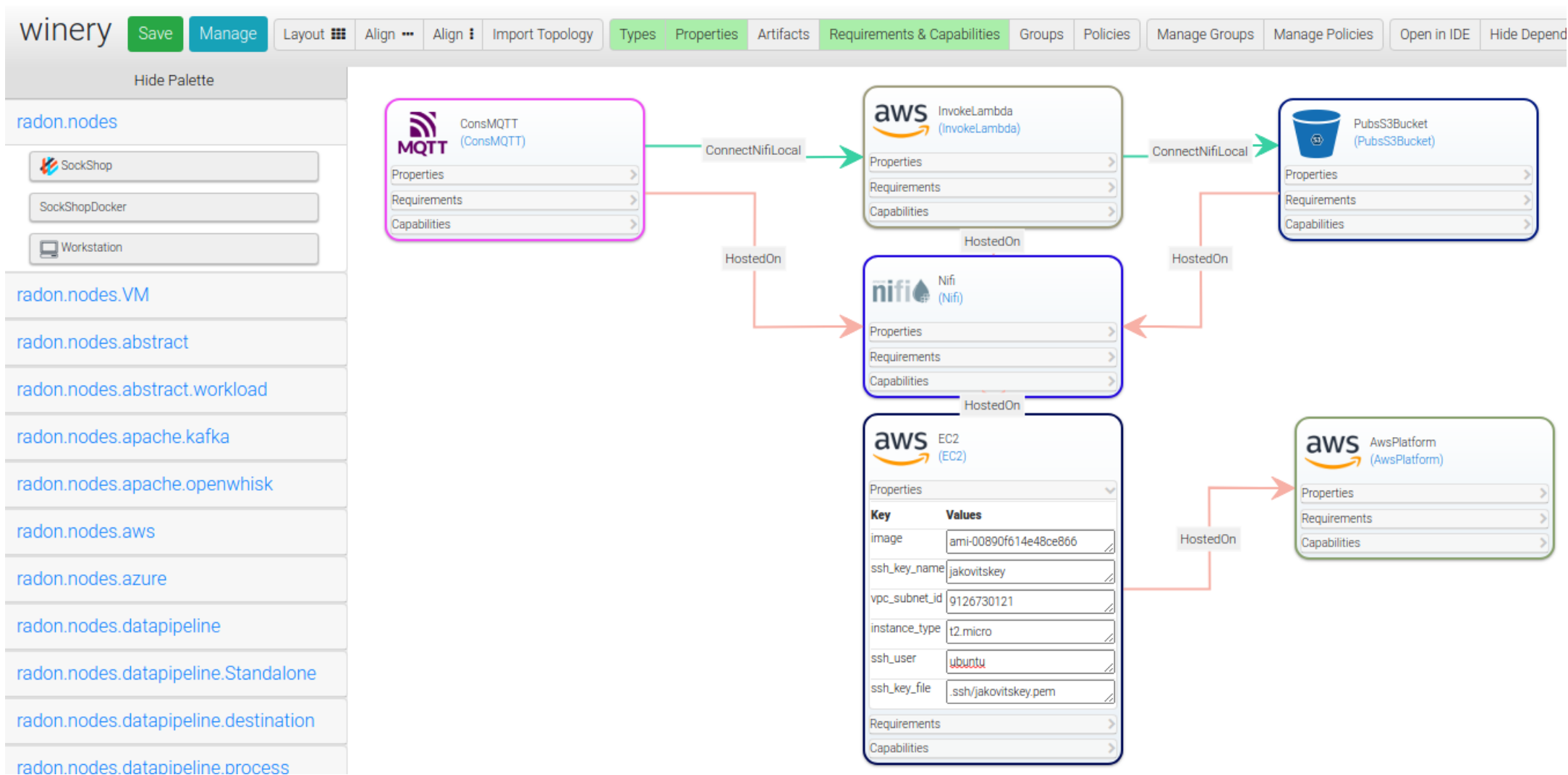
```

1  # RADON Particles
2
3  > TOSCA definitions repository for the [RADON project](http://radon-h2020.eu)
4
5  [![License](https://img.shields.io/badge/License-Apache%202.0-blue.svg)](https://
6  opensource.org/licenses/Apache-2.0)
7  [![RADON on Twitter](https://img.shields.io/twitter/url/https/twitter.com/RADON_2020?
8  label=RADON%20on%20Twitter&style=social)](https://twitter.com/RADON_2020)
9
10 The RADON Particles repository contains TOSCA blueprints, reusable definitions and
11 extensions to deploy and manage RADON applications.
12
13 It provides reusable TOSCA types of application runtimes, computing resources, and FaaS
14 platforms in the form of abstract as well as deployable modeling entities.
15
16 The repository also comprises RADON's FaaS abstraction layer that provides several TOSCA
17 definitions to deploy a particular FaaS application component to different cloud
18 providers.
19
20 ---
21
22 Node types in this public repository are in a certain state of development, indicated by
23 the following badges:
24
25 *  - initially published or
26   currently under development
27
28 *  - current version working
29   under certain conditions
  
```

MY WORKSPACE: WORKSPACE

- User Runtimes
 - datapp-deploy/datapp
 - New terminal
 - radon-datapp
 - winery-deployment/winery
 - New terminal
 - radon-gmt
 - ctt-deployment/ctt
 - New terminal
 - radon-ctt
 - vt-deployment/vt
 - New terminal
 - radon-vt
- Plugins

RADON GMT: Winery



RADON orchestrator: xOpera

. SaaS version

. CLI version

The screenshot shows the xOpera SaaS interface. The sidebar on the left contains the following navigation items: SaaS, Dashboard, Secrets, Workspaces (highlighted), Triggers and Events, Help, Need help?, Auth, and Logout. The main content area is titled 'Workspaces' and features an 'Add workspace' button. Below this is a table of Workspaces:

Workspace ID	Name	Secret IDs	Project IDs	Ownership
1	tes	n/a	1, 18	Owner
7	Demonstration-xopera	n/a	16, 17, 19	Owner
8	DemoWorkspace	2	20, 21	Owner
9	NewTestingWorkspace	2	22, 23, 24, 25	Owner
10	TheNewWorkspace	2	26, 27	Owner
11	TestingCredentials	3	28	Owner
12	SeEnWorkspace	2	227, 238, 176	Owner

A modal is open for Workspace 8, showing a table of Projects:

Project ID	Name
20	ThumbGen
21	ThumbGen12

The modal also includes buttons for 'Add project' and 'Manage project' for each project.

```
(.venv) ubuntu@pelletest88:~/opera$ opera deploy -r pelleazu
The resume deploy option might have unexpected consequences c
Do you want to continue? (Y/n): y
[Worker_0] Deploying AzurePlatform_0_0
[Worker_0]   Executing create on AzurePlatform_0_0
[Worker_0] Deployment of AzurePlatform_0_0 complete
[Worker_0] Deploying AzureResourceGroup_0_0
[Worker_0]   Executing create on AzureResourceGroup_0_0
[Worker_0] Deployment of AzureResourceGroup_0_0 complete
[Worker_0] Deploying AzureStorageAccount_0_0
[Worker_0]   Executing create on AzureStorageAccount_0_0
[Worker_0] Deployment of AzureStorageAccount_0_0 complete
[Worker_0] Deploying AzureHttpTriggeredFunction_0_0
[Worker_0]   Executing create on AzureHttpTriggeredFunction_0_0
[Worker_0] Deployment of AzureHttpTriggeredFunction_0_0 complete
```

Additional information



- TOSCA open source nodetypes: <https://github.com/radon-h2020/radon-particles/>
- Data Pipeline documentation: <https://datapipeline-plugin.readthedocs.io>
- RADON videos: <https://www.youtube.com/channel/UCgoXX6JZ6bDqTxVBRm4KWnQ>
- RADON github: <https://github.com/radon-h2020/>
- RADON website: <https://radon-h2020.eu/>



RADON Webinar III

Demo

