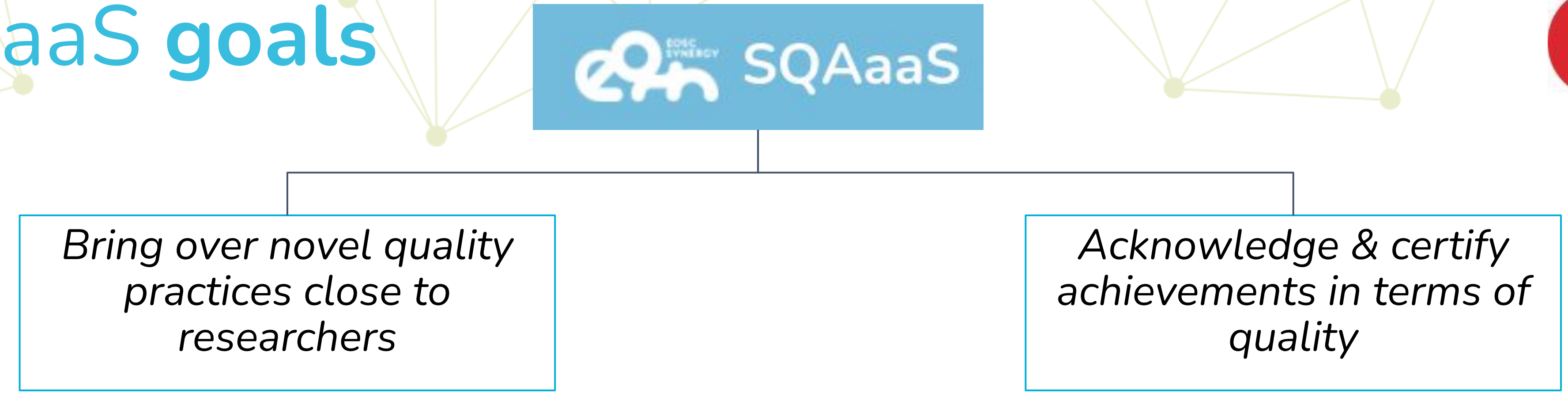
The SQAaaS platform

Pablo Orviz (IFCA, CSIC) - on behalf of EOSC-Synergy WP3

EOSC-SYNERGY receives funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 857647

SQAaaS goals

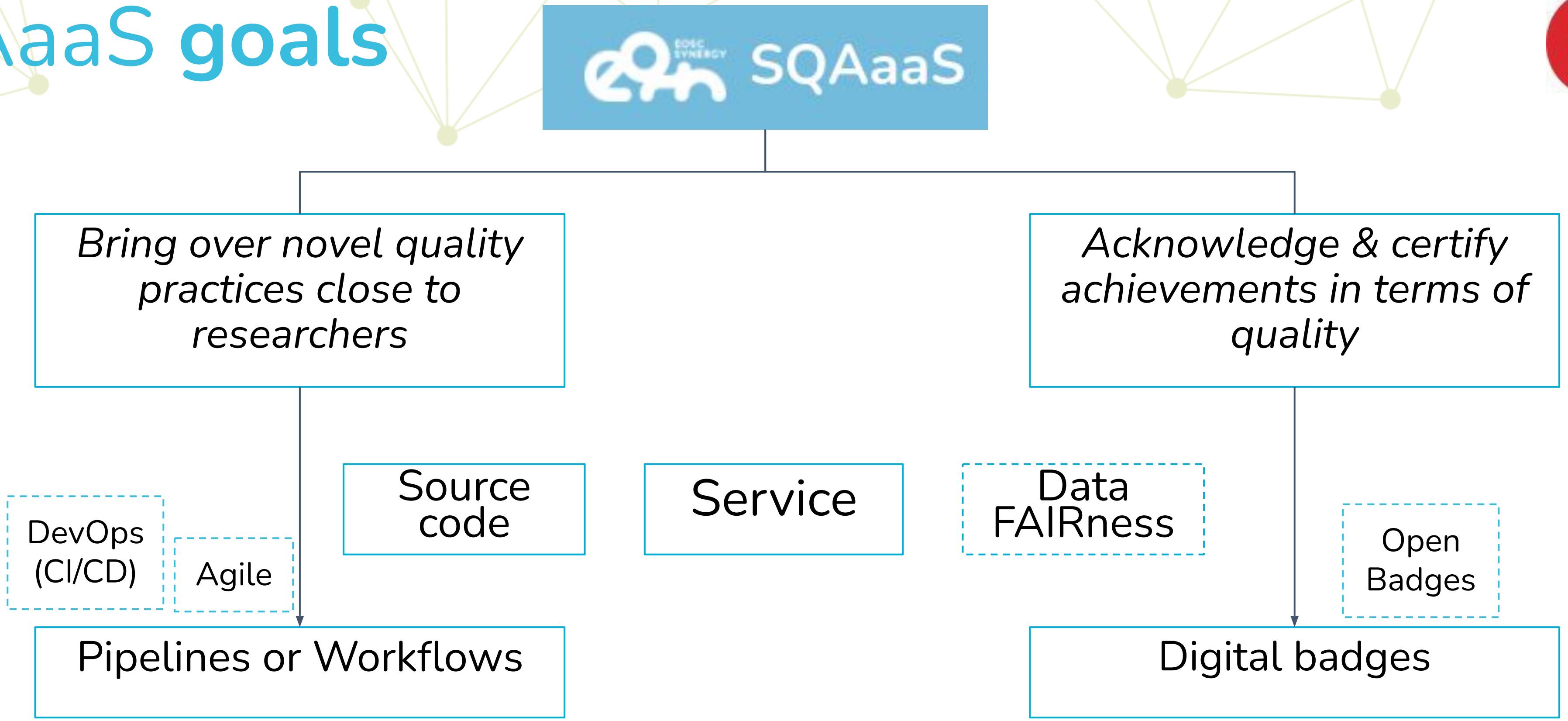

SQAaaS
Bring over novel quality practices close to researchers
Acknowledge & certify achievements in terms of quality

Source code

Service

Data FAIRness

SQAaaS **goals**



SQAaaS

Bring over novel quality practices close to researchers

Acknowledge & certify achievements in terms of quality

DevOps (CI/CD)

Agile

Source code

Service

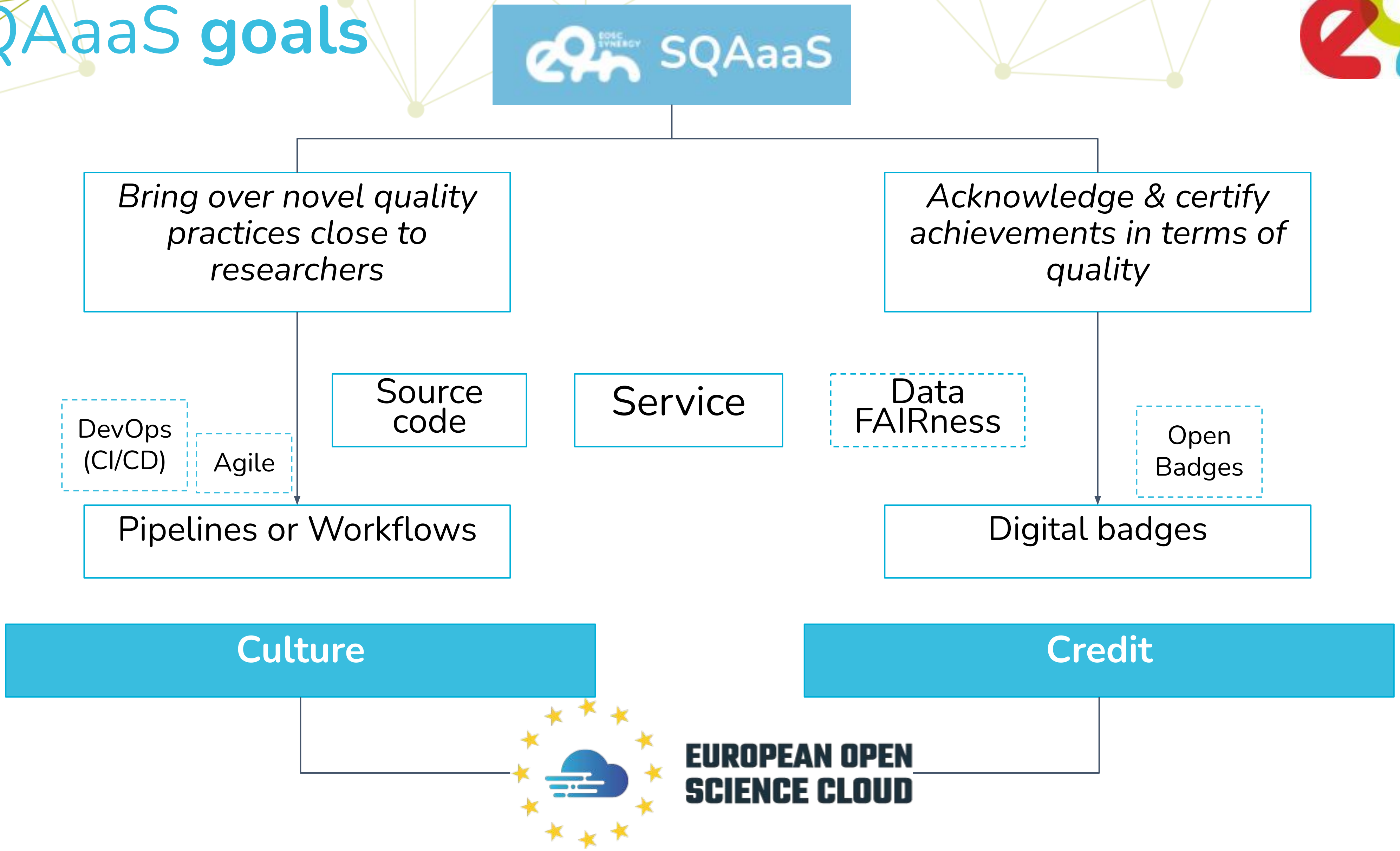
Data FAIRness

Open Badges

Pipelines or Workflows

Digital badges

SQAaaS **goals**



SQAaaS

Bring over novel quality practices close to researchers

Acknowledge & certify achievements in terms of quality

DevOps (CI/CD)

Agile

Source code

Service

Data FAIRness

Open Badges

Pipelines or Workflows

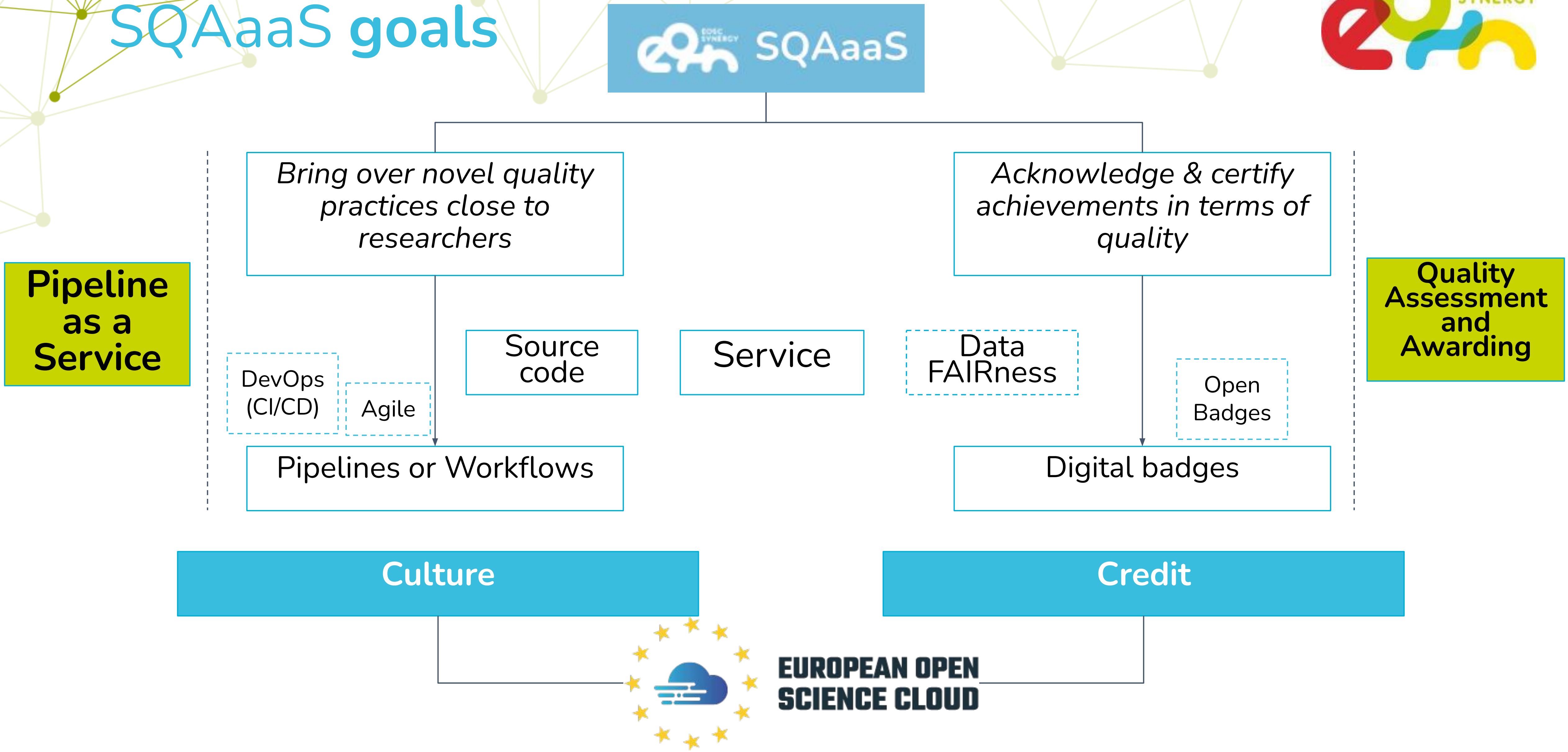
Digital badges

Culture

Credit

EUROPEAN OPEN SCIENCE CLOUD

SQAaaS **goals**



SQAaaS

Pipeline as a Service

Bring over novel quality practices close to researchers

DevOps (CI/CD)

Agile

Pipelines or Workflows

Source code

Service

Data FAIRness

Acknowledge & certify achievements in terms of quality

Open Badges

Digital badges

Quality Assessment and Awarding

Culture

Credit

EUROPEAN OPEN SCIENCE CLOUD

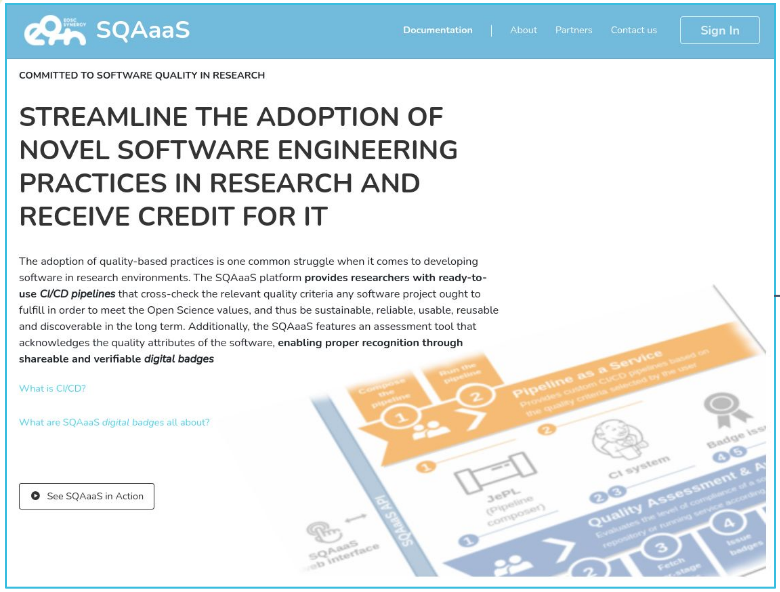
Accessing SQAaaS platform

https://sqaaas.eosc-synergy.eu

EUROPEAN OPEN SCIENCE CLOUD

EGI-Checkin

Straightforward to give it a try, open for every researcher



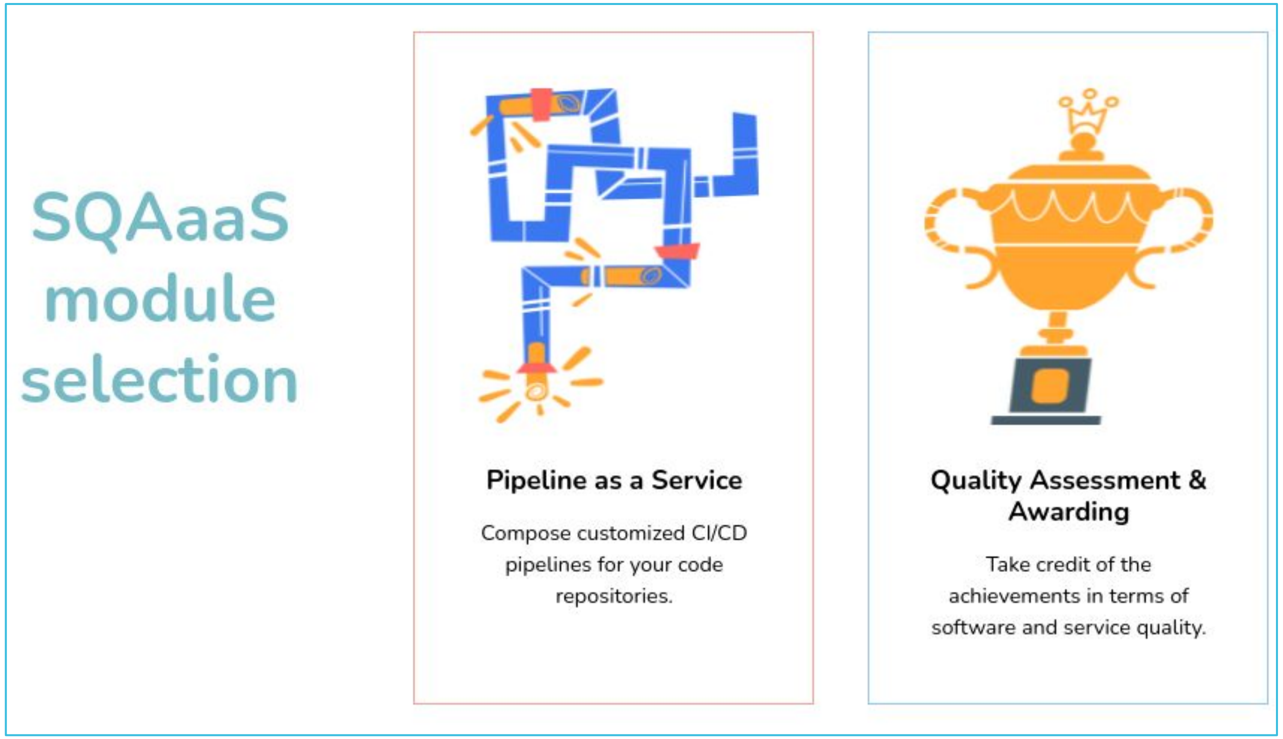
COMMITTED TO SOFTWARE QUALITY IN RESEARCH

STREAMLINE THE ADOPTION OF NOVEL SOFTWARE ENGINEERING PRACTICES IN RESEARCH AND RECEIVE CREDIT FOR IT

The adoption of quality-based practices is one common struggle when it comes to developing software in research environments. The SQAaaS platform **provides researchers with ready-to-use *CI/CD pipelines*** that cross-check the relevant quality criteria any software project ought to fulfill in order to meet the Open Science values, and thus be sustainable, reliable, usable, reusable and discoverable in the long term. Additionally, the SQAaaS features an assessment tool that acknowledges the quality attributes of the software, **enabling proper recognition through shareable and verifiable *digital badges***

What is CI/CD?

What are SQAaaS *digital badges* all about?



SQAaaS module selection

Pipeline as a Service

Compose customized CI/CD pipelines for your code repositories.

Quality Assessment & Awarding

Take credit of the achievements in terms of software and service quality.

SQAaaS: **Pipeline as a Service**

The Pipeline as a Service

1 **Define the Repositories**

Code repository settings
- GIT repo URL
- GIT repo Branch

Credentials
- Private code repos
- Private container registries

Environment variables
- Bash-like style

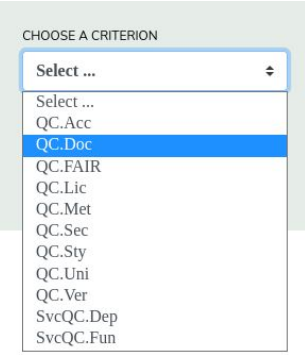
2 **Add the Quality Criteria**

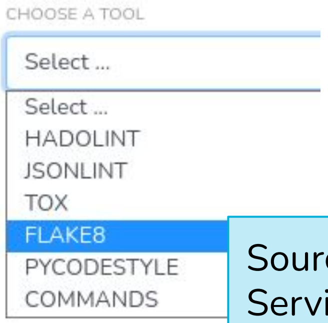
Select the Quality Criterion

Select the Tool

Customize the operational environment (containers)
- Dockerfile
- Pull from registry

Code & Service QA baselines





Source code: SAST tools, style & unit, ...
Services: IM, EC3, k8s (staging)
Data FAIRness: FAIR-EVA & F-uji

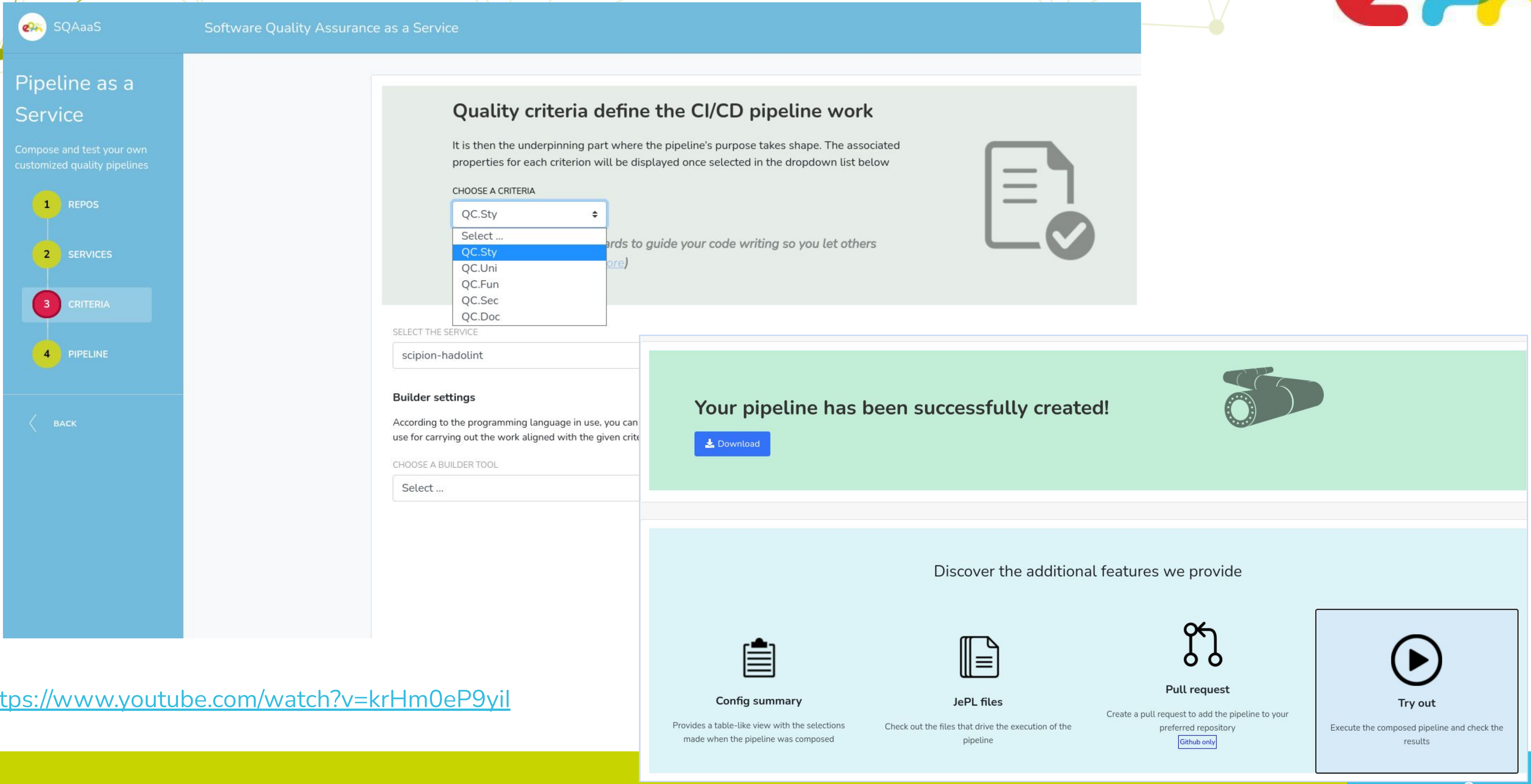
Features

Pipeline composition
JePL files

Pipeline sharing
- Download & push to your code repo
- PRs only for GitHub (GitHub)

Pipeline testing
Jenkins

The Pipeline as a Service



SQAaaS

Software Quality Assurance as a Service

Pipeline as a Service

Compose and test your own customized quality pipelines

Quality criteria define the CI/CD pipeline work

It is then the underpinning part where the pipeline's purpose takes shape. The associated properties for each criterion will be displayed once selected in the dropdown list below

CHOOSE A CRITERIA

QC.Sty

ards to guide your code writing so you let others

ore)

SELECT THE SERVICE

scipion-hadolint

Builder settings

According to the programming language in use, you can use for carrying out the work aligned with the given crite

CHOOSE A BUILDER TOOL

Select ...

Your pipeline has been successfully created!

Discover the additional features we provide

Config summary

Provides a table-like view with the selections made when the pipeline was composed

JePL files

Check out the files that drive the execution of the pipeline

Pull request

Create a pull request to add the pipeline to your preferred repository

Try out

Execute the composed pipeline and check the results

https://www.youtube.com/watch?v=krHm0eP9yiI

The JePL library

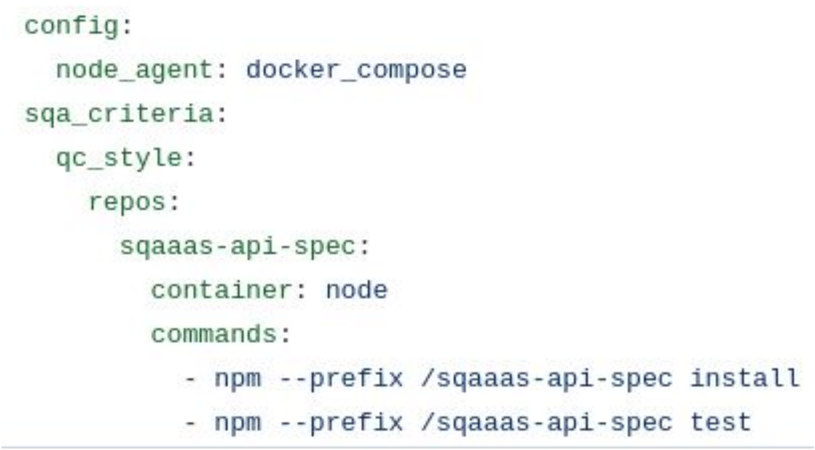
Pipeline customization
JePL

https://github.com/indigo-dc/jenkins-pipeline-library

SQAaaS does not provide a graphical way to modify existing pipelines

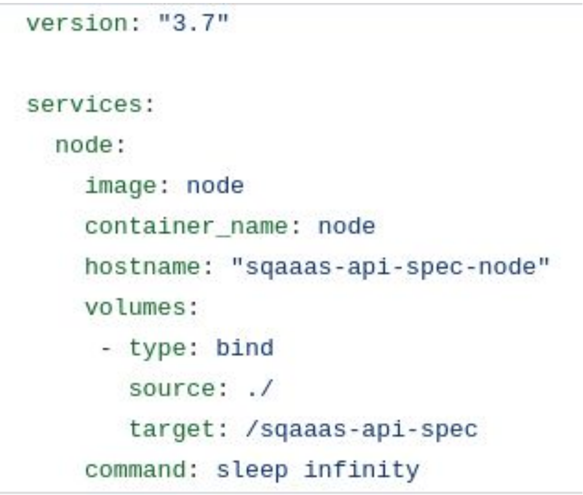
Fortunately, JePL simplifies the configuration of Jenkins pipelines

config.yml



```
config:
  node_agent: docker_compose
sqa_criteria:
  qc_style:
    repos:
      sqaaas-api-spec:
        container: node
        commands:
          - npm --prefix /sqaaas-api-spec install
          - npm --prefix /sqaaas-api-spec test
```

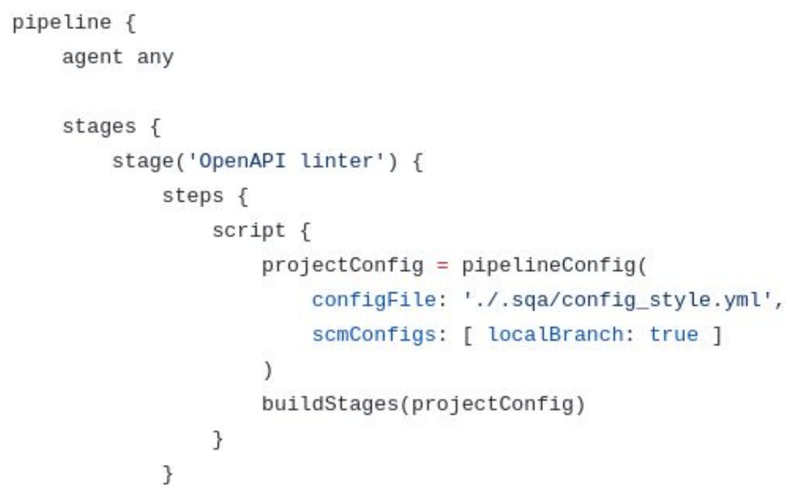
docker-compose..yml



```
version: "3.7"

services:
  node:
    image: node
    container_name: node
    hostname: "sqaaas-api-spec-node"
    volumes:
     - type: bind
       source: ./
       target: /sqaaas-api-spec
    command: sleep infinity
```

Jenkinsfile



```
pipeline {
    agent any

    stages {
        stage('OpenAPI linter') {
            steps {
                script {
                    projectConfig = pipelineConfig(
                        configFile: './.sqa/config_style.yml',
                        scmConfigs: [ localBranch: true ]
                    )
                    buildStages(projectConfig)
                }
            }
```

The JePL library

Pipeline customization

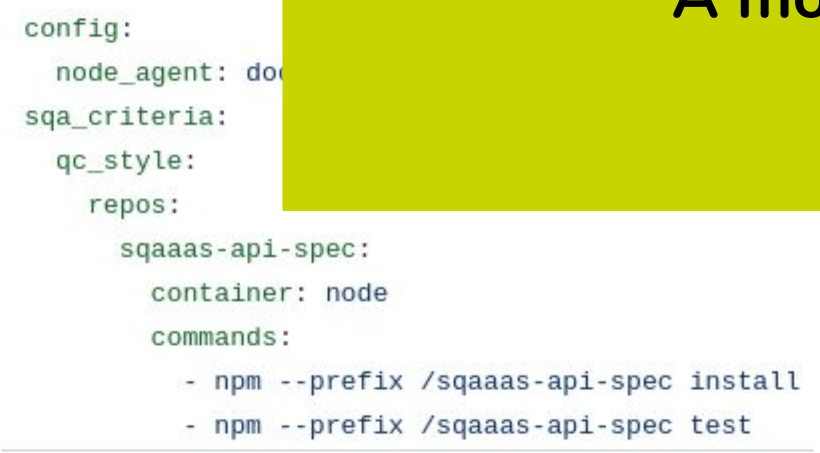
JePL

https://github.com/indigo-dc/jenkins-pipeline-library

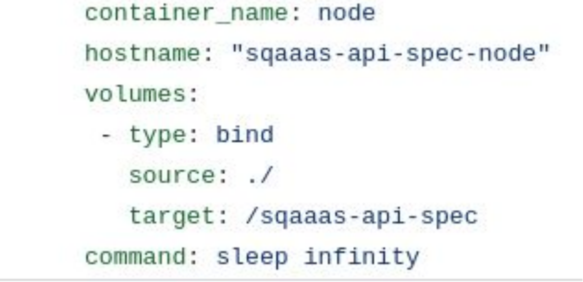
SQAaaS does not provide a graphical way to modify existing pipelines

Fortunately, JePL simplifies the configuration of Jenkins pipelines

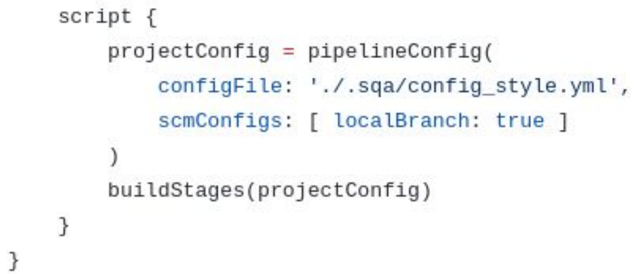
A more in-depth introduction to JePL in the next presentation



```
config:
  node_agent: do
sqa_criteria:
  qc_style:
    repos:
      sqaaas-api-spec:
        container: node
        commands:
          - npm --prefix /sqaaas-api-spec install
          - npm --prefix /sqaaas-api-spec test
```



```
container_name: node
hostname: "sqaaas-api-spec-node"
volumes:
 - type: bind
   source: ./
   target: /sqaaas-api-spec
command: sleep infinity
```



```
        script {
            projectConfig = pipelineConfig(
                configFile: './.sqa/config_style.yml',
                scmConfigs: [ localBranch: true ]
            )
            buildStages(projectConfig)
        }
    }
```

SQAaaS: **Quality Assessment & Awarding**

The Quality Assessment & Awarding

1 Trigger the Assessment

Source code

GIT code repo URL (& branch)

GIT doc repo URL (& branch)

Service (web)

GIT repo with deployment recipes (& branch)

Orchestration tool

Data

Persistent identifier

FAIRness tool

Matchmaking policies

2 Assessment results

Criteria breakdown

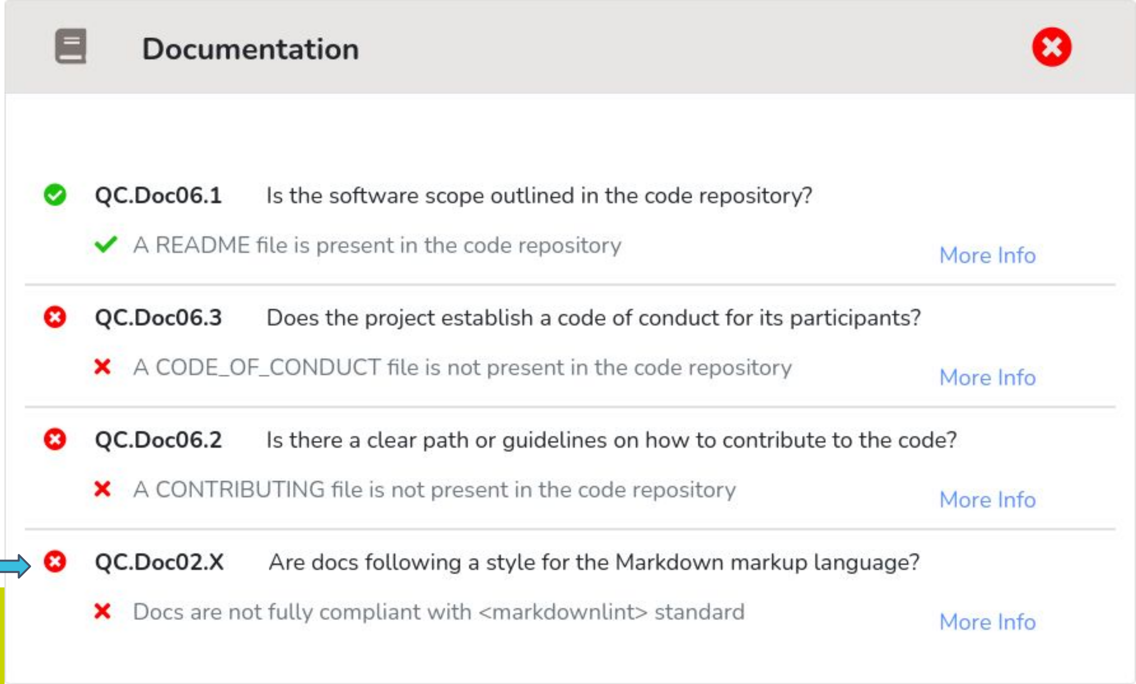
Subcriteria

ID

Evidence

Tools

Metrics



Documentation

✓ **QC.Doc06.1** Is the software scope outlined in the code repository?

✓ A README file is present in the code repository More Info

✗ **QC.Doc06.3** Does the project establish a code of conduct for its participants?

✗ A CODE_OF_CONDUCT file is not present in the code repository More Info

✗ **QC.Doc06.2** Is there a clear path or guidelines on how to contribute to the code?

✗ A CONTRIBUTING file is not present in the code repository More Info

✗ **QC.Doc02.X** Are docs following a style for the Markdown markup language?

✗ Docs are not fully compliant with <markdownlint> standard More Info

The Quality Assessment & Awarding

Outputs

Digital badges

Open Badges specification v2.0 → Metadata

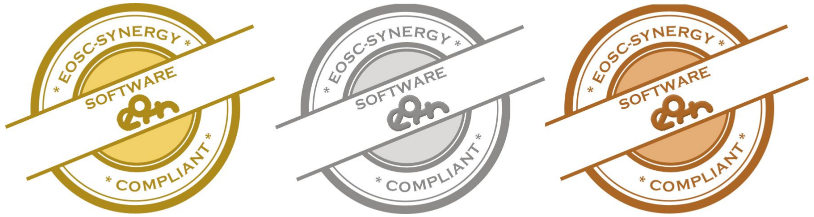
Quality report

Build logs

EOSC-Synergy issuer profiles @eu.badgr.com

EOSC-Synergy badging Quality Criteria↔ Badges mapping

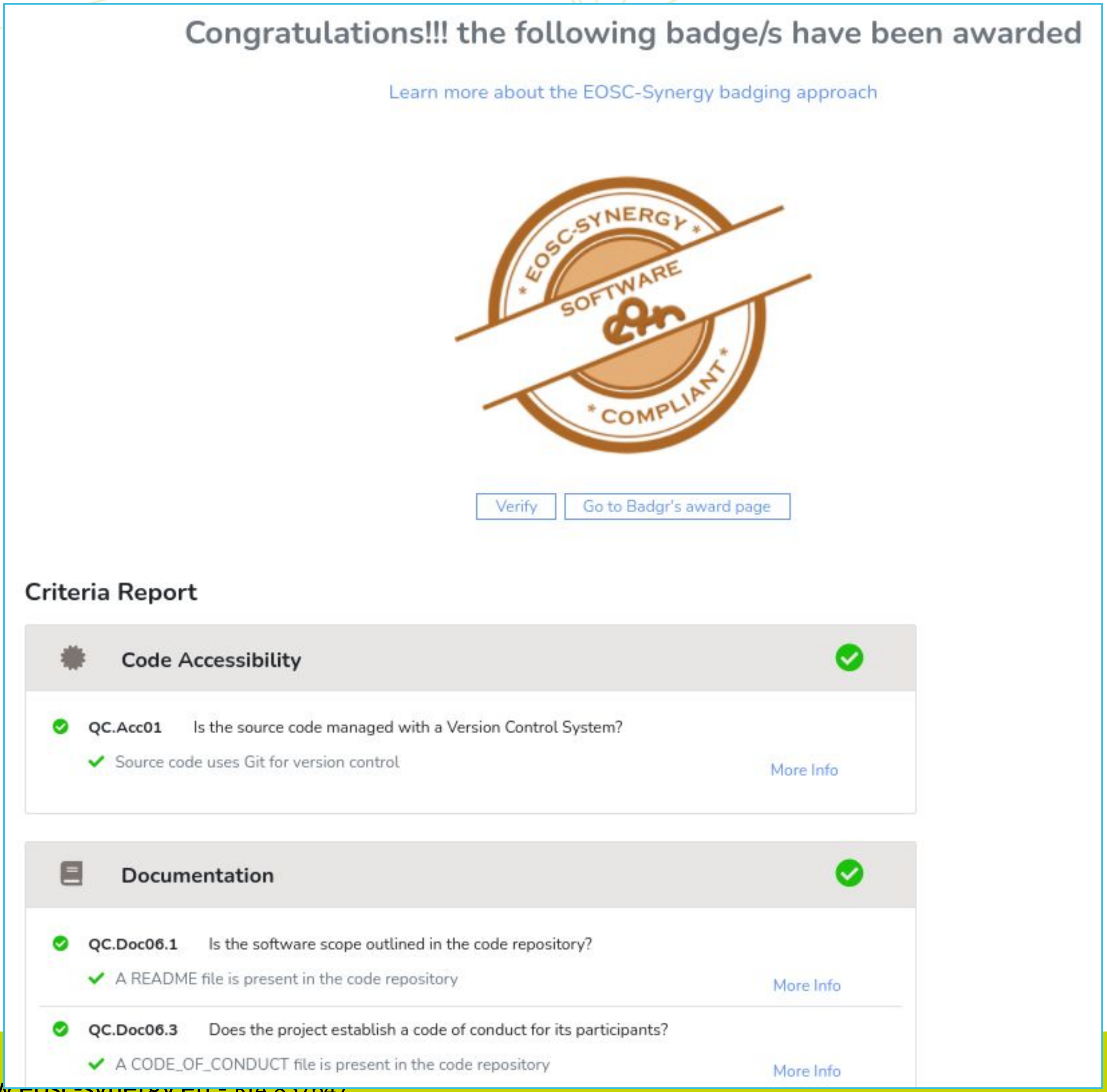
3 badge classes (bronze, silver, gold) for 3 categories (software, services and data)



Quality report

Fully-fledged report containing the detailed result of the subcriteria associated with each criterion → Highlights Quality Achievements & How to improve them

The Quality Assessment & Awarding



Congratulations!!! the following badge/s have been awarded

Learn more about the EOSC-Synergy badging approach

Criteria Report

Code Accessibility ✓

✓ **QC.Acc01** Is the source code managed with a Version Control System?

✓ Source code uses Git for version control

More Info

Documentation ✓

✓ **QC.Doc06.1** Is the software scope outlined in the code repository?

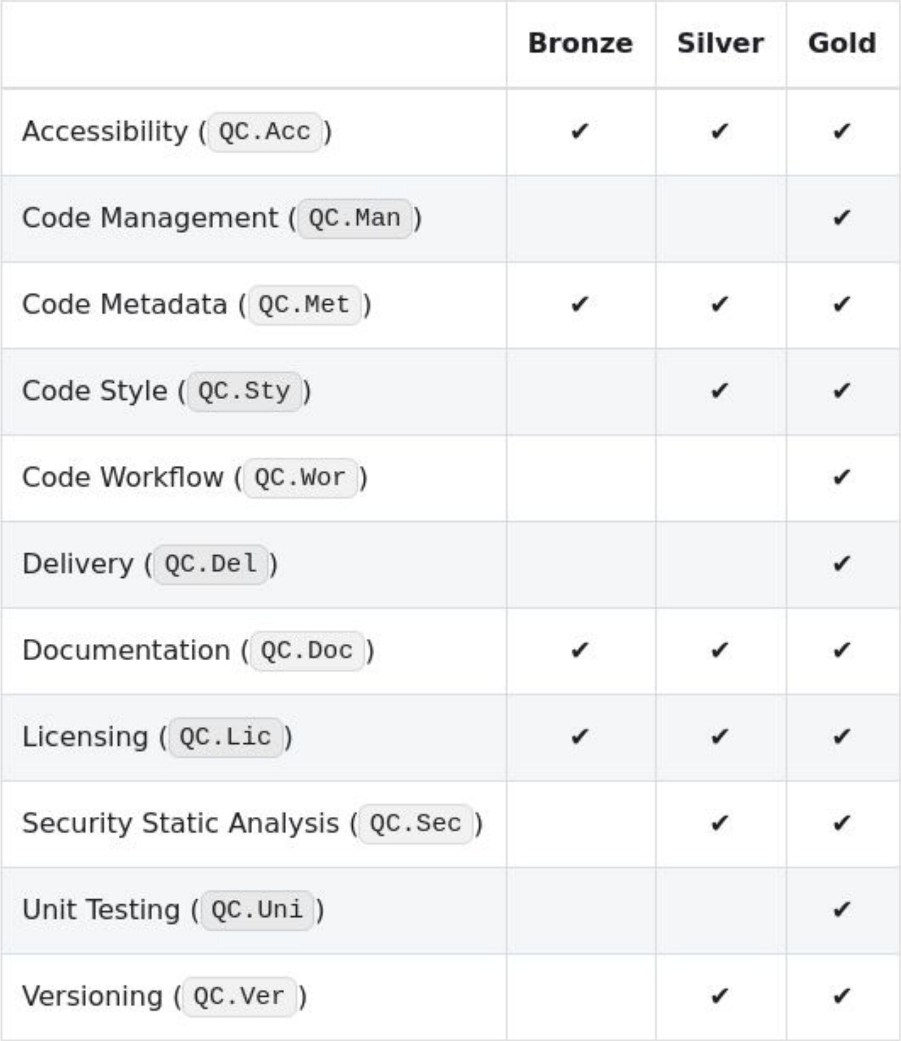
✓ A README file is present in the code repository

More Info

✓ **QC.Doc06.3** Does the project establish a code of conduct for its participants?

✓ A CODE_OF_CONDUCT file is present in the code repository

More Info



	Bronze	Silver	Gold
Accessibility (`QC.Acc`)	✔	✔	✔
Code Management (`QC.Man`)			✔
Code Metadata (`QC.Met`)	✔	✔	✔
Code Style (`QC.Sty`)		✔	✔
Code Workflow (`QC.Wor`)			✔
Delivery (`QC.Del`)			✔
Documentation (`QC.Doc`)	✔	✔	✔
Licensing (`QC.Lic`)	✔	✔	✔
Security Static Analysis (`QC.Sec`)		✔	✔
Unit Testing (`QC.Uni`)			✔
Versioning (`QC.Ver`)		✔	✔

The Quality Assessment & Awarding

SQAaaS Hackathon at Ibergrid 2022 (Mon 10th)

- ~20 participants
- Focus on source code (quality assessment)
- Diverse software projects (Python, Golang, Vue JS, Ruby, Java, ..)
- Takeaways from yesterday
  - Platform tested at a larger scale
    - *Hit limit imposed by GitHub API* (both from Jenkins & SQAaaS API)
    - SQAaaS API overloaded with requests (LB solution for hackathon-like exercises)
  - Dissemination (participants)
    - Became familiar with good practices on SQA (fixing issues identified by the SQA tools)
    - Many achieved bronze badges, even 2 golden badges (both cases started from no badge)

Towards QA in EOSC

Service validation

- Targeting EOSC, the SQAaaS platform can contribute on:
  - EOSC onboarding process
    - Objectively certify that a minimum TRL has been acquired
  - EOSC portal
    - Credit, recognition & dissemination of QA achievements through digital badges
- Thematic Services from EOSC-Synergy (integrated in EOSC portal):
  - Deploy the service, check that all required components are up & running
    - DONE for 3 TSs (2 more on the way through k8s support)
  - Perform functional tests to the deployed service
    - TESTING phase

Data FAIRness

- EOSC core requirement
- SQAaaS integrates FAIR-EVA (RDA) & F-UJI (FAIRsFAIR) tools
  - DONE for pipeline composition (SQAaaS' Pipeline as a Service)
  - TESTING the assessment (SQAaaS' QAA)

Highlights

- Build culture: bring quality practices for software development into the research ecosystem
- Give credit: Proof of concept of a **quality certification tool** for software (incl. services) and data
  - **Digital badges**
    - Containing **metadata** about the assessment process
    - **Shared & Verifiable** (Open Badges v2.0)
- **Used by thematic services** (9) from EOSC-Synergy (specific presentation in this session)
  - ALL being delivered with a minimum quality (bronze) for code
  - AT LEAST 4 will achieve minimum quality (bronze) for services
- Be sure to try it out https://sqaaas.eosc-synergy.eu/
  - Docs at https://docs.sqaaas.eosc-synergy.eu/

Gracias !

Obrigado !

Danke !

Dziękuję !

Vďaka !

Dekuji !

Bedankt !

Merci !

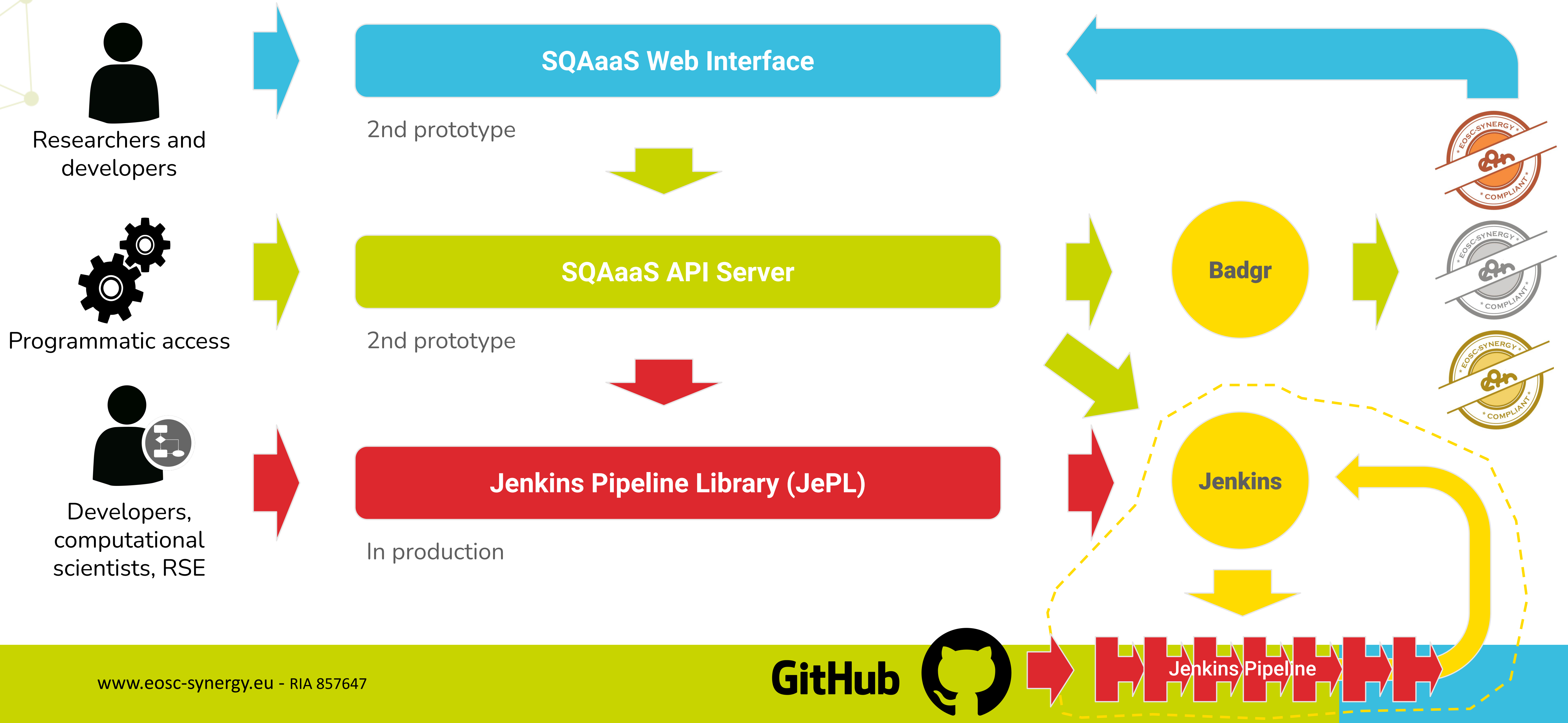
Thanks !

EOSC SYNERGY

THEMATIC SERVICES

Backup slides

SQAaaS: architecture and components



Researchers and developers

Programmatic access

Developers, computational scientists, RSE

SQAaaS Web Interface

2nd prototype

SQAaaS API Server

2nd prototype

Jenkins Pipeline Library (JePL)

In production

Badgr

Jenkins

SQAaaS breakdown: **the library (JePL)**

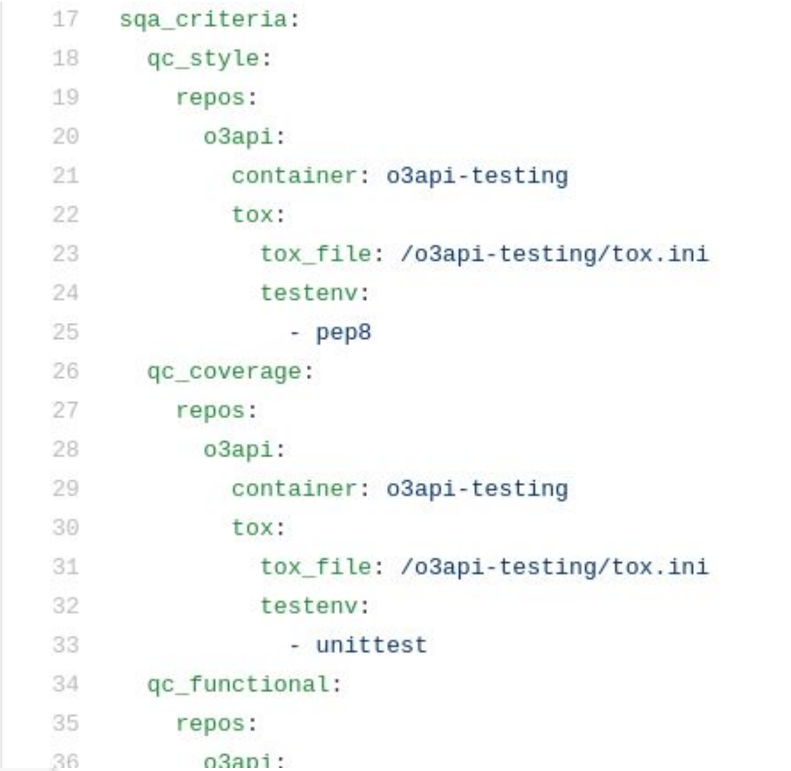
- SQAaaS core component

  ⇉ **Implements the previous quality criteria**

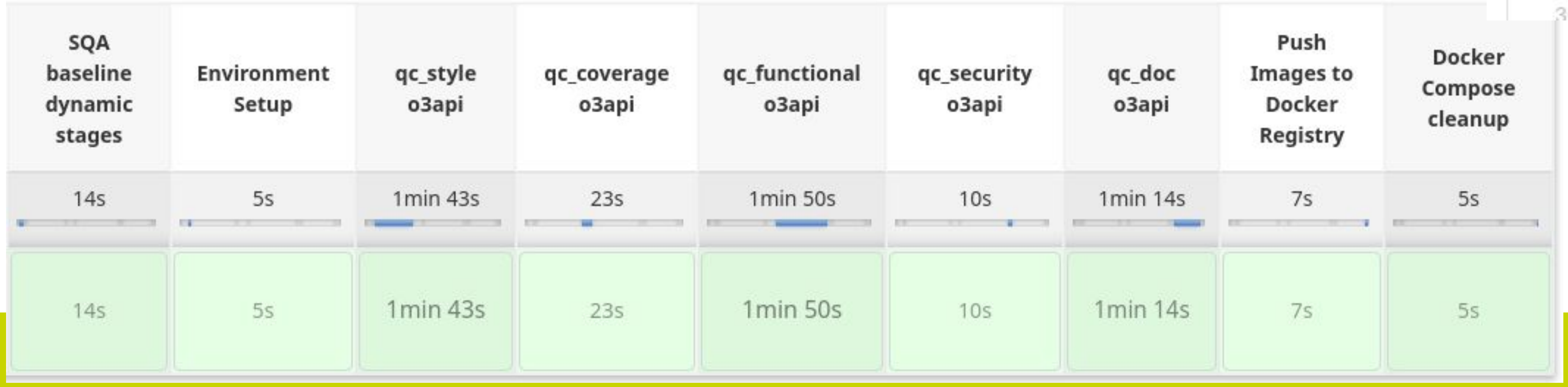
  - Criterion-driven YAML config file (pipeline stage)
- Technology dependent

  ⇉ Jenkins Pipeline as Code (PaC)

  ⇉ Docker Compose for service orchestration



```yaml
sqa_criteria:
  qc_style:
    repos:
      o3api:
        container: o3api-testing
        tox:
          tox_file: /o3api-testing/tox.ini
          testenv:
            - pep8
  qc_coverage:
    repos:
      o3api:
        container: o3api-testing
        tox:
          tox_file: /o3api-testing/tox.ini
          testenv:
            - unittest
  qc_functional:
    repos:
      o3api:
```

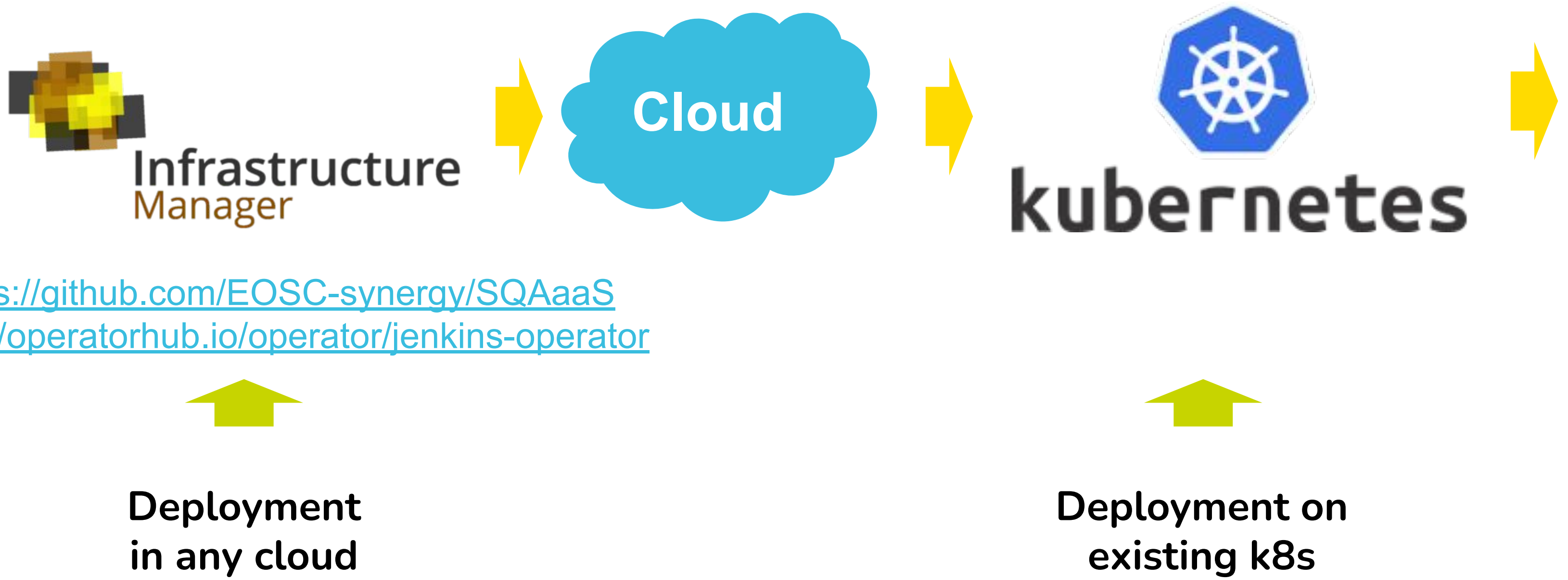


SQA baseline dynamic stages	Environment Setup	qc_style o3api	qc_coverage o3api	qc_functional o3api	qc_security o3api	qc_doc o3api	Push Images to Docker Registry	Docker Compose cleanup
14s	5s	1min 43s	23s	1min 50s	10s	1min 14s	7s	5s
14s	5s	1min 43s	23s	1min 50s	10s	1min 14s	7s	5s

Current release: 2.1.0
https://indigo-dc.github.io/jenkins-pipeline-library/

SQAaaS: automated deployment under development

- Automated deployment of the complete SQAaaS platform
  - Facilitates SQAaaS production deployment, testing & promotes adoption
  - Also important for closed / private environments

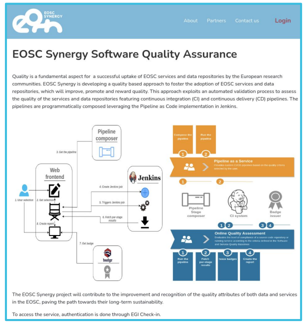


https://github.com/EOSC-synergy/SQAaaS
https://operatorhub.io/operator/jenkins-operator

Deployment in any cloud

Deployment on existing k8s

Jenkins



EOSC Synergy Software Quality Assurance

Quality is a fundamental aspect for a successful uptake of EOSC services and data repositories by the European research communities. EOSC Synergy is developing a quality based approach to foster the adoption of EOSC services and data repositories, which will improve, promote and reward quality. This approach exploits an automated validation process to assess the quality of the services and data repositories featuring continuous integration (CI) and continuous delivery (CD) pipelines. The pipelines are programmatically composed leveraging the Pipeline as Code implementation in Jenkins.

The EOSC Synergy project will contribute to the improvement and recognition of the quality attributes of both data and services in the EOSC, paving the path towards their long-term sustainability.

To access the service, authentication is done through EGI Check-in.

SQAaaS platform

User

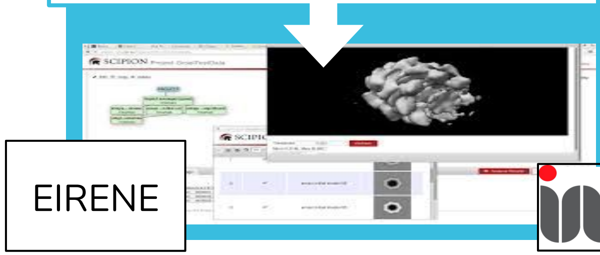
Thematic Services

EOSC SYNERGY

SCIPION

CNB CENTRO NACIONAL DE BIOTECNOLOGIA CSIC

CryoEM data processing for Structural Biology



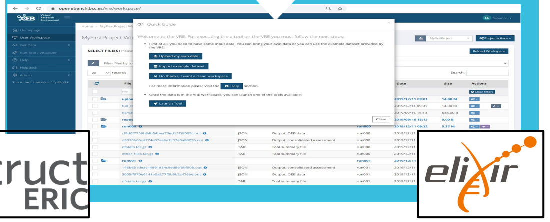
EIRENE

instruct ERIC

OpenEBench

BSC

ELIXIR benchmarking and technical monitoring platform



elixir

LAGO

Ciemat Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

Latin American Giant cosmic ray Observatory

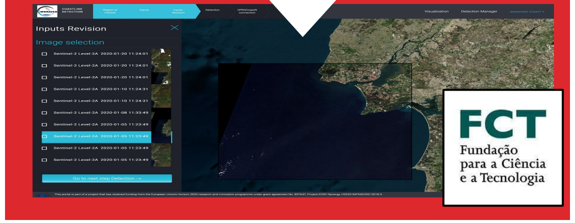
LAGO

WORSICA

LIP

LNEC LABORATÓRIO NACIONAL DE ENGENHARIA CIVIL

Water Monitoring Sentinel Cloud Platform

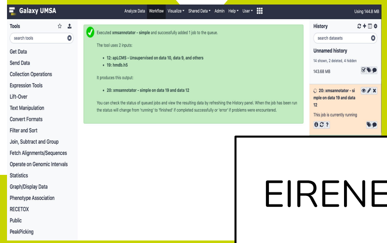


FCT Fundação para a Ciência e a Tecnologia

UMSA

cesnet

Untargeted Mass-Spectrometry Analysis



EIRENE

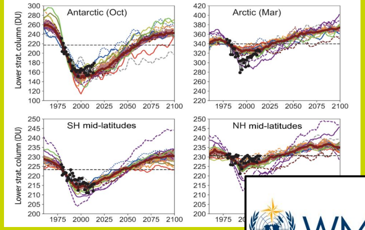
MSWSS

Water Supply Systems modeling and analysis

O3AS

KIT Karlsruher Institut für Technologie

Ozone Analysis Service

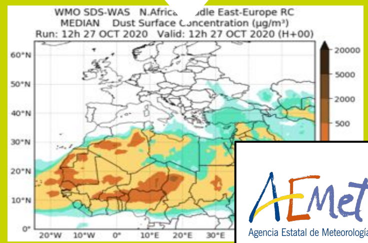


WMO

SDS-WAS

BSC

A Service related to mineral dust forecast


WMO SDS-WAS N.Africa ...dle East-Europe RC
MEDIAN Dust Surface Concentration (µg/m³)
Run: 12h 27 OCT 2020 Valid: 12h 27 OCT 2020 (H+00)
20000
5000
2000
500
60°N
50°N
40°N
30°N
20°N
10°N
0°
20°W 10°W 0° 10°E 20°E 30°E

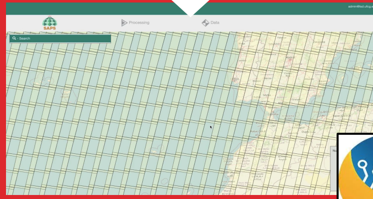
AEMet Agencia Estatal de Meteorología

SAPS

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Surface Energy Balance Automated Processing Service

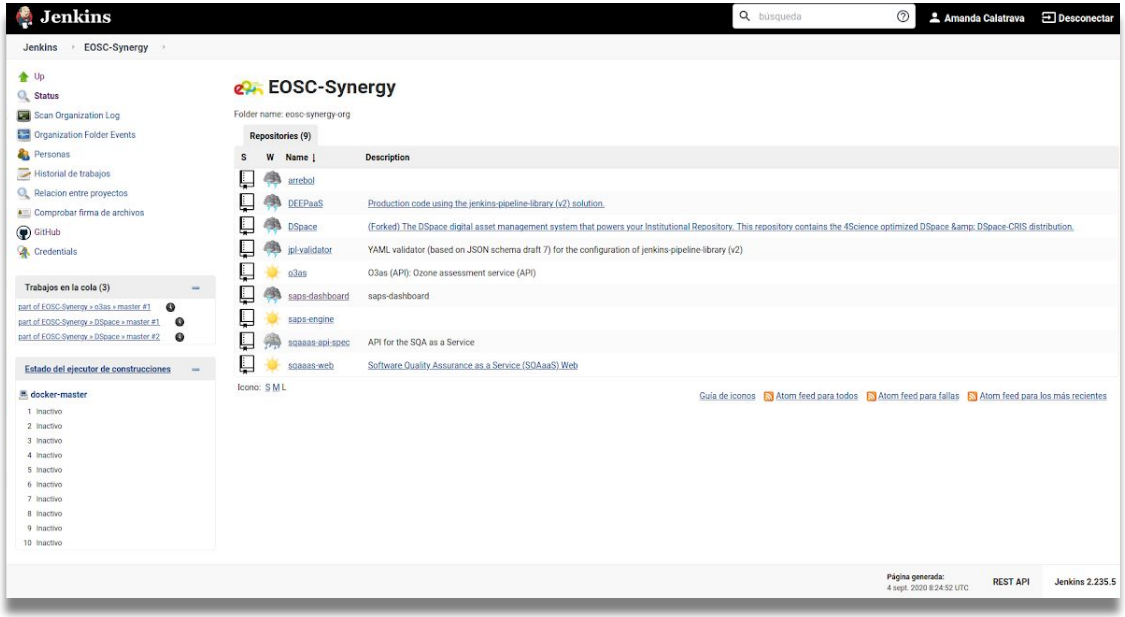
UFCG Universidade Federal de Campina Grande



MCTI

SQAaaS: Jenkins instance for SQAaaS

- EOSC-Synergy Jenkins instance
- Checks automatically the projects in EOSC Synergy Github organization:
  - https://github.com/EOSC-synergy
- Jenkins Operator deployment to create your own Jenkins instances available:
  - A user deployable Jenkins with required plugins is almost ready for those that need on-premises solution



EOSC-Synergy

Folder name: eosc-synergy-org

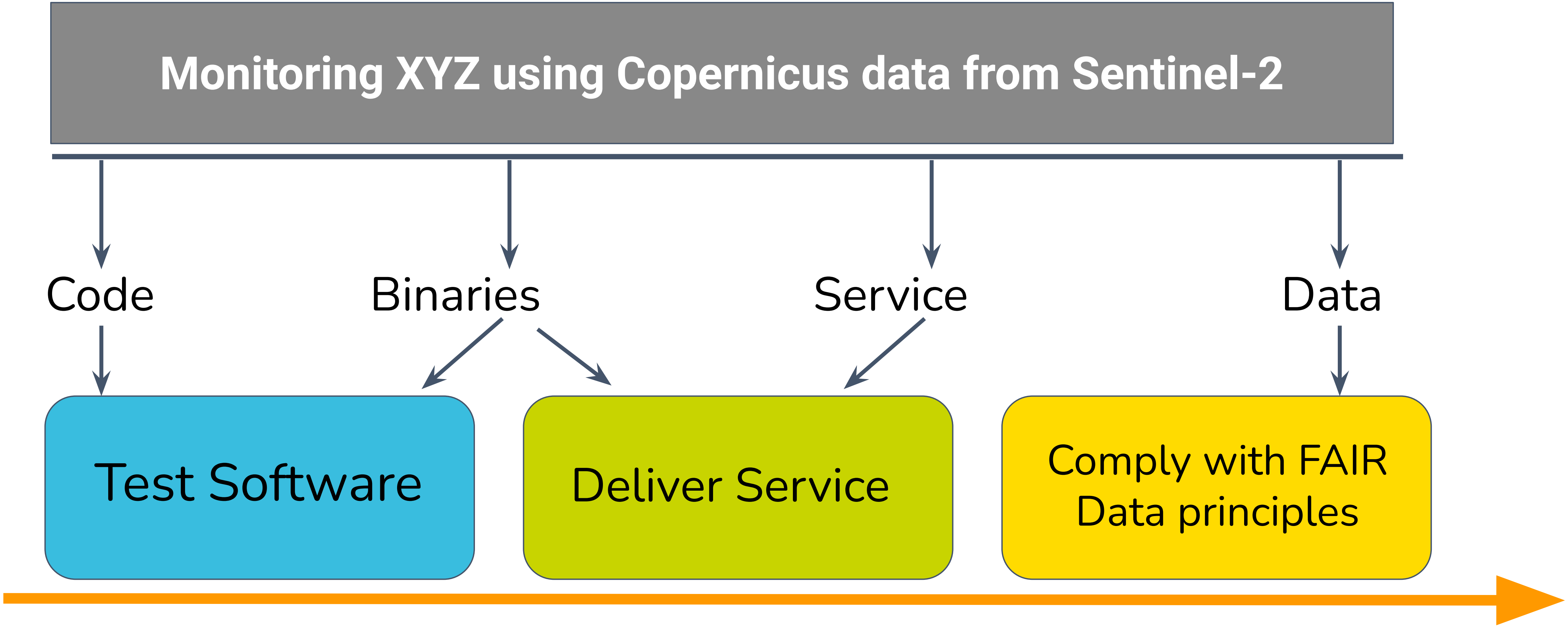
Repositories (9)

S	W	Name ↓	Description
	⛈	arrebol	
	⛈	DEEPaaS	Production code using the jenkins-pipeline-library (v2) solution.
	⛈	DSpace	(Forked) The DSpace digital asset management system that powers your Institutional Repository. This repository contains the 4Science optimized DSpace &amp; DSpace-CRIS distribution.
	⛈	jpl-validator	YAML validator (based on JSON schema draft 7) for the configuration of jenkins-pipeline-library (v2)
	☀	o3as	O3as (API): Ozone assessment service (API)
	⛈	saps-dashboard	saps-dashboard
	☀	saps-engine	
	🌧	sqaaas-api-spec	API for the SQA as a Service
	☀	sqaaas-web	Software Quality Assurance as a Service (SQAaaS) Web

Página generada:
4 sept. 2020 8:24:52 UTC

https://jenkins.eosc-synergy.eu/job/eosc-synergy-org/

Generic Case in Earth Observation: Monitoring “XYZ” using Copernicus Data



Monitoring XYZ using Copernicus data from Sentinel-2

Code

Binaries

Service

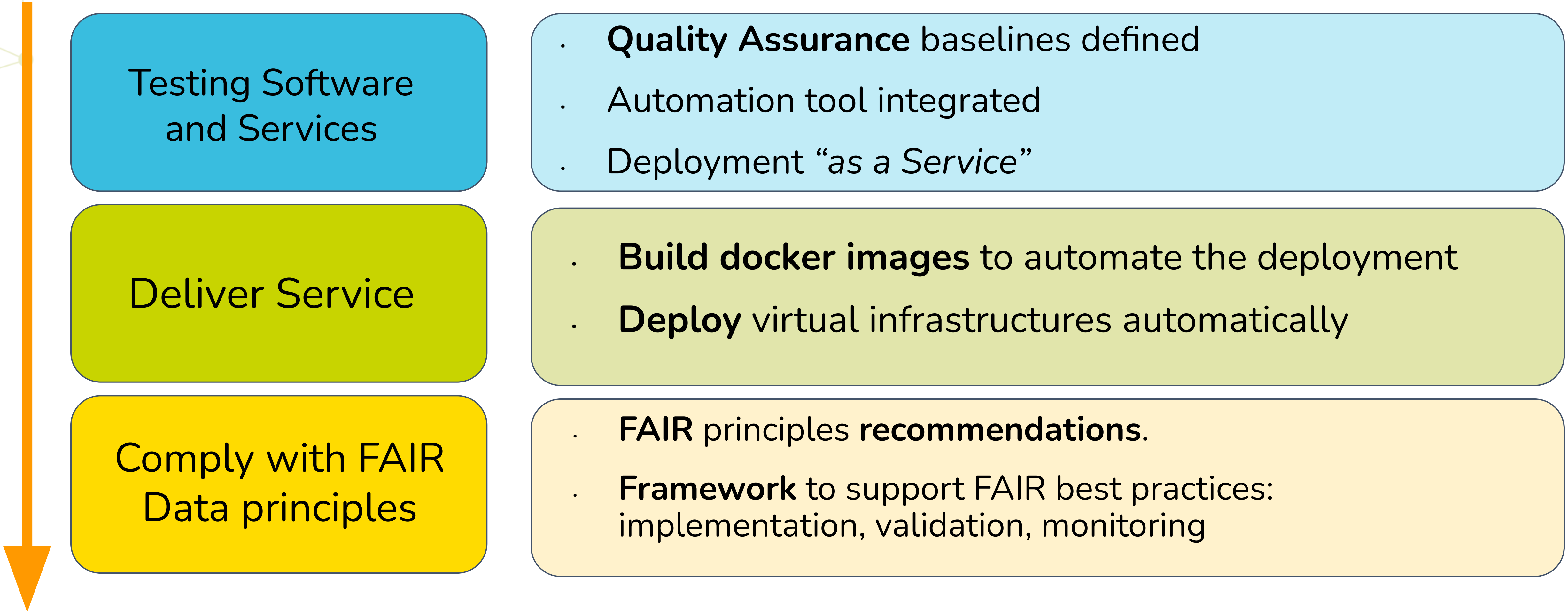
Data

Test Software

Deliver Service

Comply with FAIR Data principles

SQAaaS: Assessment of FAIR criteria



Testing Software and Services

- **Quality Assurance** baselines defined
- Automation tool integrated
- Deployment “*as a Service*”

Deliver Service

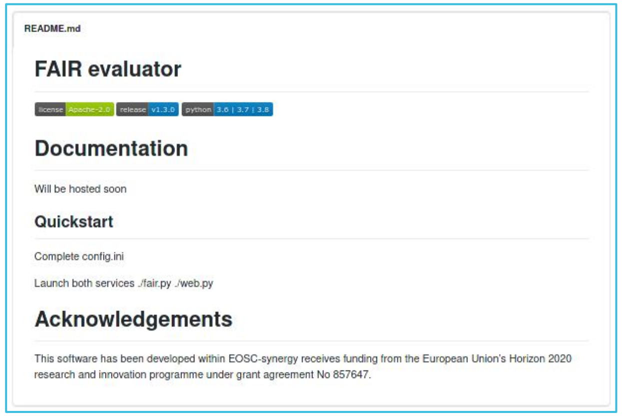
- **Build docker images** to automate the deployment
- **Deploy** virtual infrastructures automatically

Comply with FAIR Data principles

- **FAIR** principles **recommendations**.
- **Framework** to support FAIR best practices: implementation, validation, monitoring

Technical Framework: FAIR-Evaluator

- Open Source tool for evaluating FAIR digital objects
- Oriented to:
  - Researchers and repository administrators.
  - To get feedback on FAIR compliance level of research data.
  - For institutional/multidisciplinar repositories.
- Provides FAIR assessment based on RDA indicators



README.md

FAIR evaluator

Documentation

Will be hosted soon

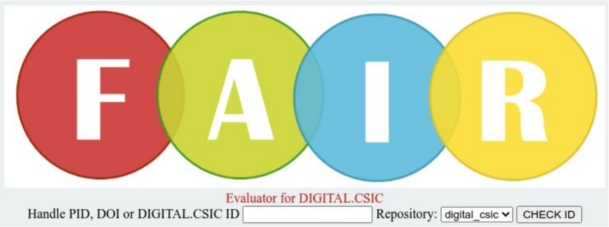
Quickstart

Complete config.ini

Launch both services ./fair.py ./web.py

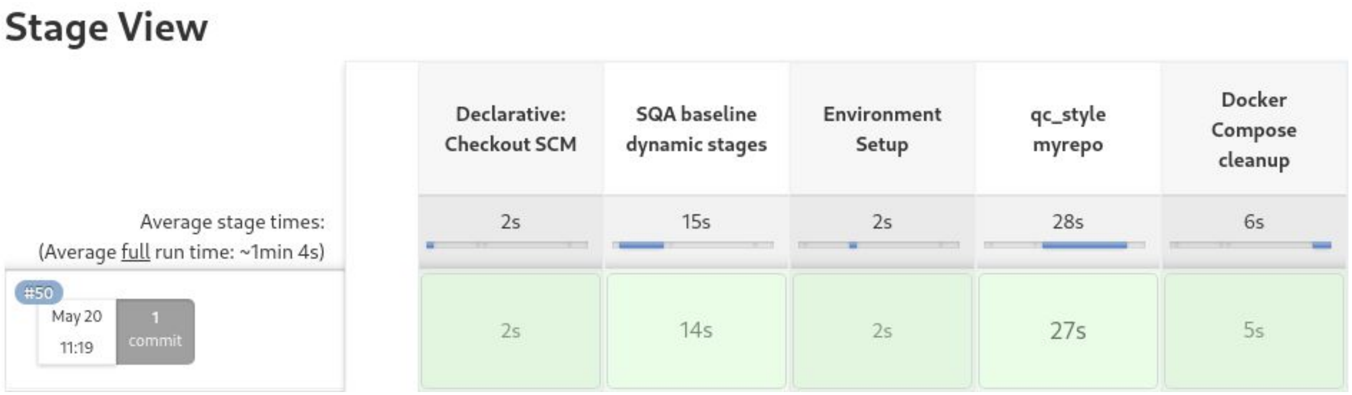
Acknowledgements

This software has been developed within EOSC-synergy receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857647.



Evaluator for DIGITAL.CSIC

Handle PID, DOI or DIGITAL.CSIC ID ____ Repository: digital_csic



Stage View

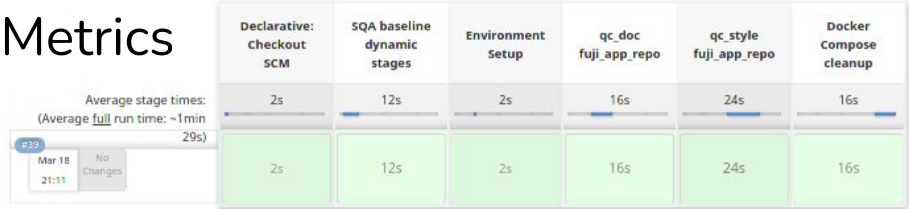
	Declarative: Checkout SCM	SQA baseline dynamic stages	Environment Setup	qc_style myrepo	Docker Compose cleanup
Average stage times: (Average full run time: ~1min 4s)	2s	15s	2s	28s	6s
#50 May 20 11:19 1 commit	2s	14s	2s	27s	5s

Leveraging pipelines for SQAaaS

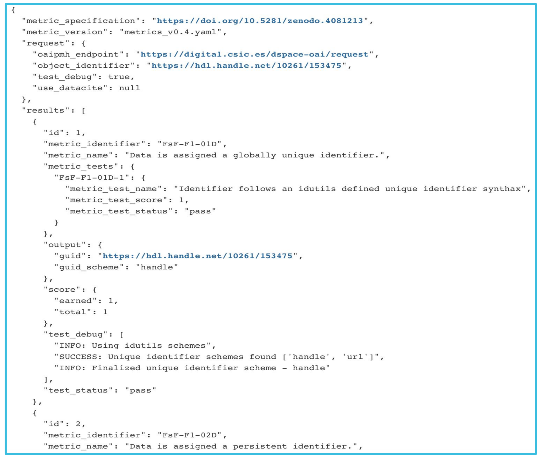
https://github.com/EOSC-synergy/FAIR_eva

Technical Framework: F-UJI metrics in SQAaaS

- F-UJI is a FAIR assessment REST web service developed in FAIRsFAIR
- Enables validation of FAIRsFAIR Data Object Assessment Metrics
- Comparing alignment of F-UJI and FAIR validator metrics
- Looking at common interfaces for metrics
- Integrating F-UJI in the SQAaaS platform



	Declarative: Checkout SCM	SQA baseline dynamic stages	Environment Setup	qc_doc fuji_app_repo	qc_style fuji_app_repo	Docker Compose cleanup
Average stage times: (Average full run time: ~1min 29s)	2s	12s	2s	16s	24s	16s
#39 Mar 18 21:11 No Changes	2s	12s	2s	16s	24s	16s



```
{
  "metric_specification": "https://doi.org/10.5281/zenodo.4081213",
  "metric_version": "metrics_v0.4.yaml",
  "request": {
    "oaipmh_endpoint": "https://digital.csic.es/dspace-oai/request",
    "object_identifier": "https://hdl.handle.net/10261/153475",
    "test_debug": true,
    "use_datacite": null
  },
  "results": [
    {
      "id": 1,
      "metric_identifier": "FsF-F1-01D",
      "metric_name": "Data is assigned a globally unique identifier.",
      "metric_tests": {
        "FsF-F1-01D-1": {
          "metric_test_name": "Identifier follows an idutils defined unique identifier synthax",
          "metric_test_score": 1,
          "metric_test_status": "pass"
        }
      },
      "output": {
        "guid": "https://hdl.handle.net/10261/153475",
        "guid_scheme": "handle"
      },
      "score": {
        "earned": 1,
        "total": 1
      },
      "test_debug": [
        "INFO: Using idutils schemes",
        "SUCCESS: Unique identifier schemes found ['handle', 'url']",
        "INFO: Finalized unique identifier scheme - handle"
      ],
      "test_status": "pass"
    },
    {
      "id": 2,
      "metric_identifier": "FsF-F1-02D",
      "metric_name": "Data is assigned a persistent identifier.",
```