SQAaaS platform as the quality gate for Digital Twins

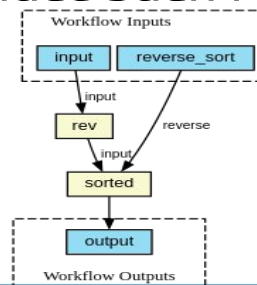Pablo Orviz

orviz@ifca.unican.es
IFCA-CSIC

SQAaaS: the QA building block for Digital Twins

↘ **SQAaaS** features:
- Programmatic QA assessment (RESTful API) and awarding of digital research content
  - ○ SQA characteristics on code and (web) services (custom)
  - ○ FAIR principles on data (RDA maturity model)
- Adhoc composition of CI/CD pipelines

..besides providing clients (web portal, CLI) to streamline usage

↘ Digital Twins orchestrate and operate **computational end-to-end data analysis workflows**
- Workflow steps consume/execute/produce *such research content*:
  - ○ Input and output data
 - ○ Software/code and/or services

SQAaaS role in Digital Twin projects (EC-funded)

- SQAaaS fits in as the **QA building block during the development and operation phases of a Digital Twin**
  - QA+FAIR assessment of individual digital assets
  - Integration with Workflow Management Systems (WfMSs)
  - Integration with Scientific Data Management technologies
- SQAaaS is already an architectural component within projects developing Digital Twin solutions:
  - InterTwin: developing an engine (DTE) for building, validating and deploy multidisciplinary digital twins
  - DT-GEO: building a complex prototype digital twin on geophysical extremes

DT-〰-GEO

interTwin

Role of SQAaaS in Digital Twin projects: **DT-GEO**
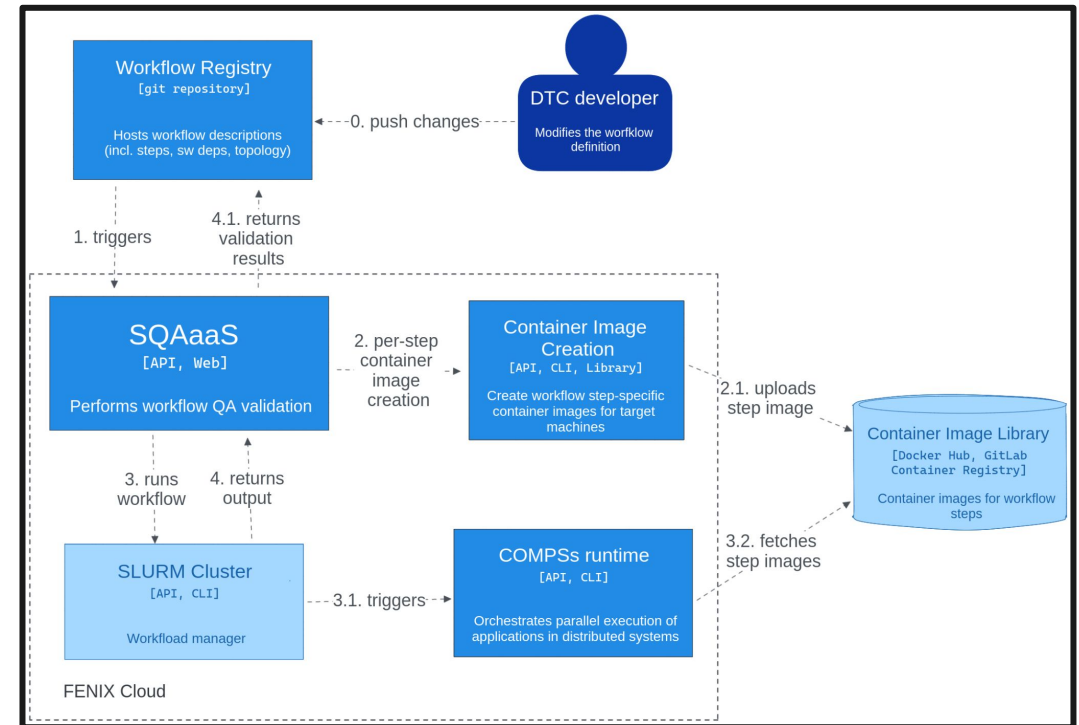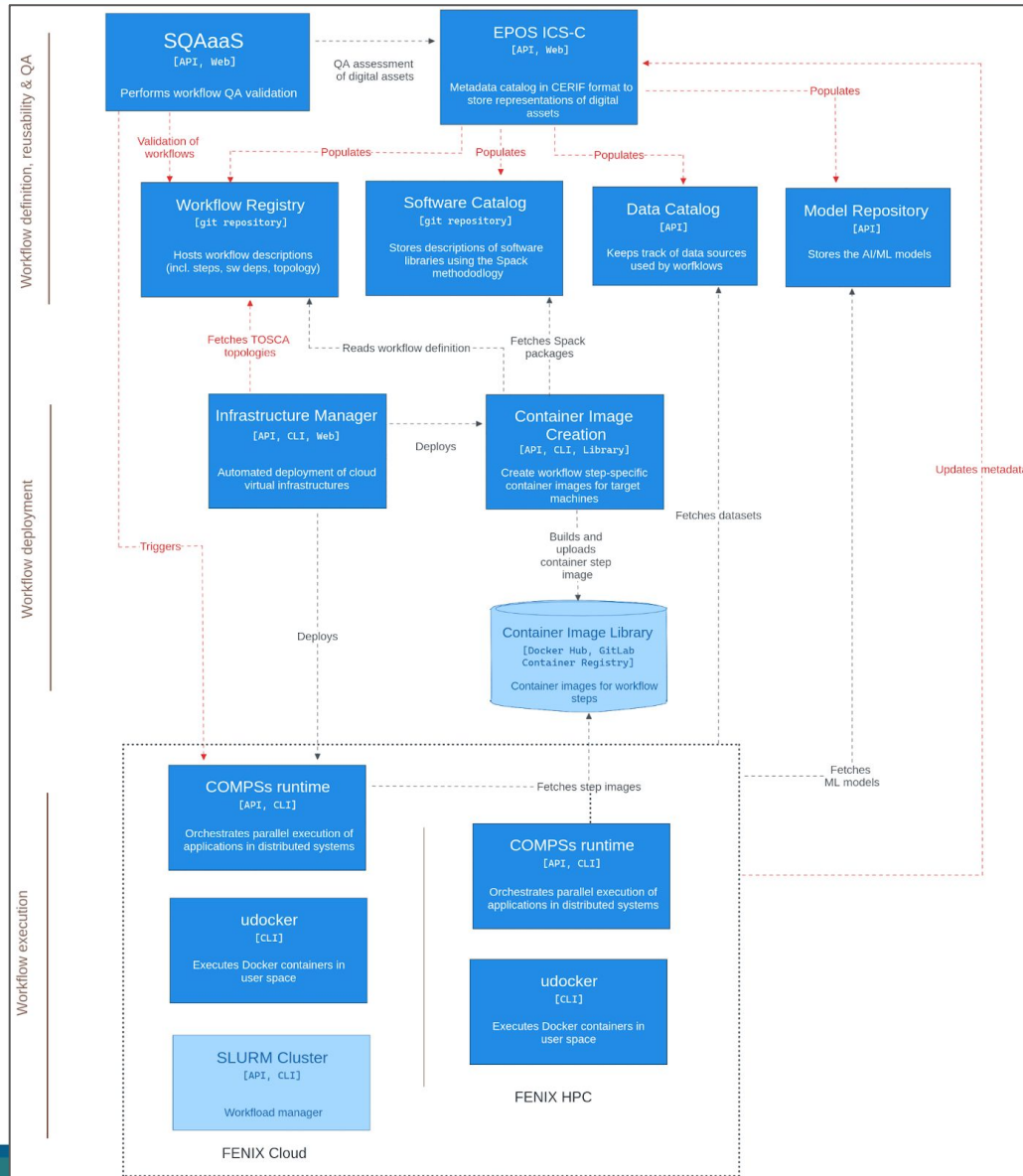
Role of SQAaaS in Digital Twin projects: InterTwin
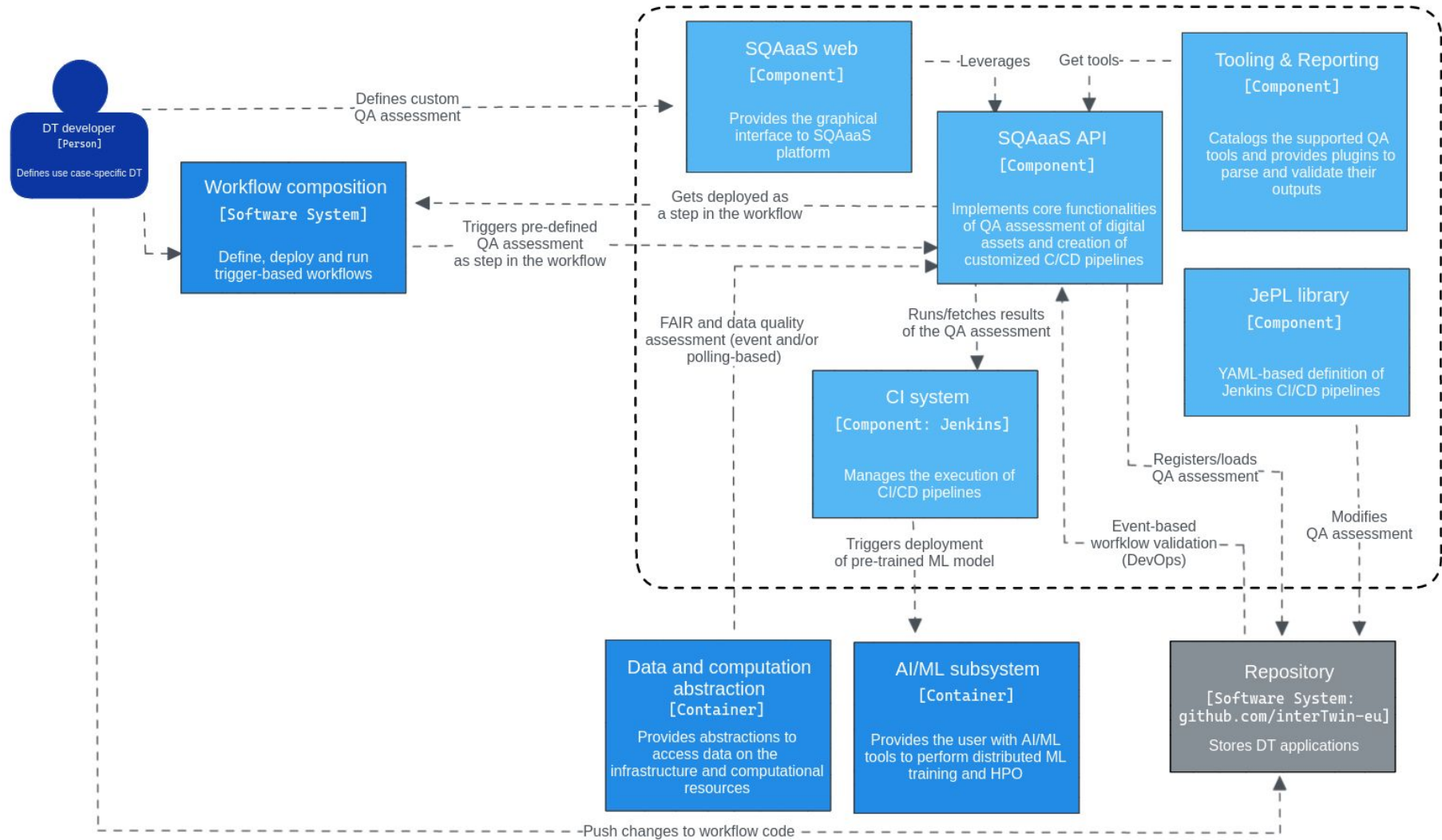
Integrations & features: **Code hosting platforms**

- **Suitable for**: QA work done at development time, in the platform where the code is maintained
- **Target/s**: source code (incl. workflow code)
- **Automation:** reacts to events triggered by the hosting platform (push, change request, releases, etc.)
- **Platform/s**: GitHub and GitLab

GitHub   GitLab

Features:
#1. Leverages already existent CI technologies: **GitHub Actions & GitLab CI**
#2. **Reporting summary** for each build
#3. **Dynamic badging**, updated as SQAaaS assessment progresses

Integrations & features: **Code hosting platforms**

Feature #1: Leverages already existent CI technologies: **GitHub Actions & GitLab CI**

Marketplace / Actions / SQAaaS source code assessment

GitHub Action

SQAaaS source code assessment

🏷 2.4.1 [Latest version]

`SQAaaS | source code` `building` `License GPLv3` `REUSE compliant` `code style black`

GitHub

```
uses: eosc-synergy/sqaaas-step-action@v1
with:
  name: model_validation_step
  tool: pytest
  test-path: ./tests
```

A 2nd action (sqaaas-step-action) defines a series of steps to be executed as part of the assessment in the SQAaaS platform

Baseline evaluation

```
uses: eosc-synergy/sqaaas-assessment-action@v2
```

Extended evaluation

```
uses: eosc-synergy/sqaaas-assessment-action@v2
with:
  qc_uni_steps: model_validation_step
```

SQAaaS tutorial at Ibergrid 2024 conference @ Mon 28

Integrations & features: **Code hosting platforms**

Feature #1: Leverages already existent CI technologies: **GitHub Actions & GitLab CI**

GitLab

SQAaaS CI Component ⊕
Collection of pipeline templates

Following the approach of GitHub, a 2nd CI component has been implemented to define an additional step in the SQAaaS assessment

SQAaaS tutorial at Ibergrid 2024 conference @ Mon 28

Baseline evaluation

```
include:
  - component: gitlab.com/dtgeo/sqa-assessment-ci-template/SQA.gitlab-ci@1.0.0
```

Extended evaluation

```
variables:
  INPUT_NAME: pycompss-step
  INPUT_TOOL: commands
  INPUT_COMMANDS: echo 'Here we will execute PyCOMPSs'
  INPUT_QC_UNI_STEPS: pycompss-step

stages:
  - SQA

include:
  - component: $CI_SERVER_FQDN/dtgeo/metadata/sqa-assessment-ci-template/SQA-step.gitlab-ci@1.0.0
  - component: gitlab.com/dtgeo/sqa-assessment-ci-template/SQA.gitlab-ci@1.0.0
```

Integrations & features: **Code hosting platforms**

Feature #2: **Reporting summary** for each build

- Displayed in GitHub Actions' job view
- Provides link to full report in SQAaS platform

Job that triggers SQAaS platform summary

SQAaS summary ♨

Result	Assertion	Subcriterion ID	Criterion ID
✔	Source code uses Git for version control	QC.Acc01	QC.Acc
✔	A README file is present in the code repository	QC.Doc06.1	QC.Doc
✔	A CODE_OF_CONDUCT file is present in the code repository	QC.Doc06.3	QC.Doc
✔	A CONTRIBUTING file is present in the code repository	QC.Doc06.2	QC.Doc
✔	Documentation resides in the same repository as code	QC.Doc01.1	QC.Doc
🔴	Docs are not fully compliant with markdownlint standard	QC.Doc02.X	QC.Doc
✔	An Open Source license found in the code repository: GPL-3.0	QC.Lic01	QC.Lic
✔	LICENSE file is visible at the root path of the code repository: LICENSE	QC.Lic01.1	QC.Lic
✔	License GPL-3.0 is approved by the Open Source Initiative	QC.Lic02	QC.Lic
✔	License GPL-3.0 is listed under the Open Source Initiative popular category	QC.Lic02.1	QC.Lic
✔	JSON files are compliant with jsonlint standard	QC.Sty01	QC.Sty
✔	The code repository uses tags for releasing new software versions	QC.Ver01.0	QC.Ver
✔	Latest release tag 2.9.1 is SemVer compliant	QC.Ver01	QC.Ver
✔	All release tags are SemVer compliant	QC.Ver02	QC.Ver
☐	No matching files found for language *CodeMeta* in repository searching by extensions or filenames No matching files found for language *Citation File Format* in repository searching by extensions or filenames	QC.Met01	QC.Met
🔴	No matching files found for language *Python* in repository searching by extensions or filenames No matching files found for language *Go* in repository searching by extensions or filenames	QC.Sec02	QC.Sec

◇ View full report at https://raw.githubusercontent.com/eosc-synergy/sqaaas-api-spec.assess.sqaaas/main/.report/assessment_output.json
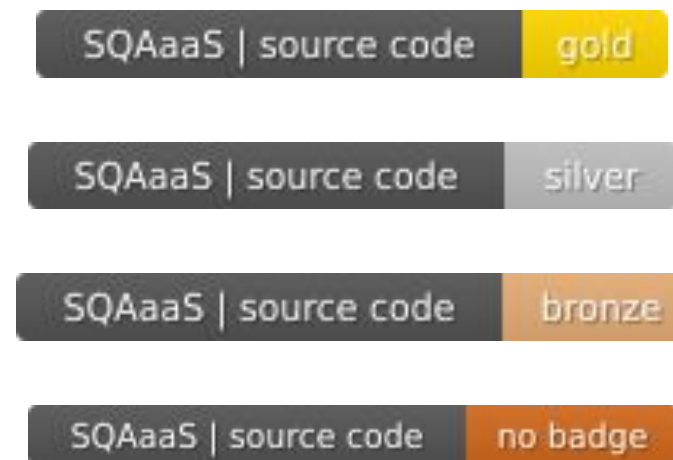
Job summary generated at run-time

Integrations & features: **Code hosting platforms**

Feature #3: **Dynamic badging** (powered by shields.io), updated as the SQAaaS assessment progresses

Intermediate status
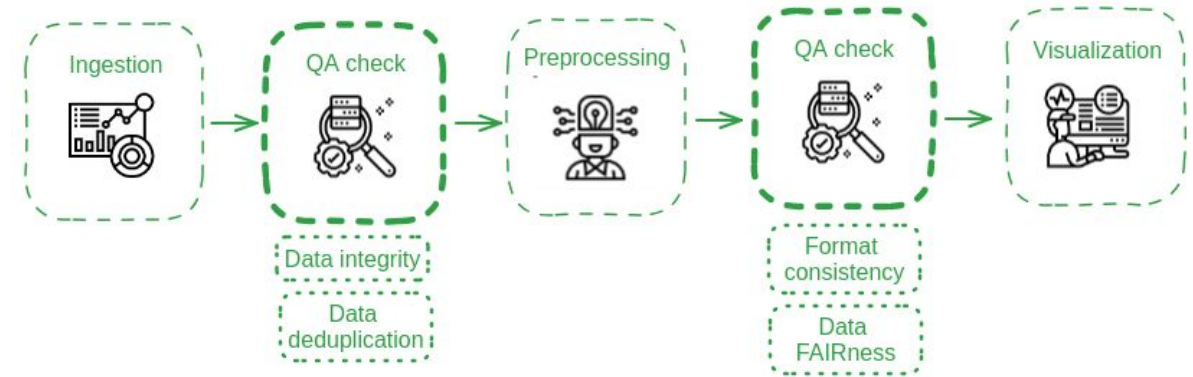

Final status

More on SQAaaS badging
docs.sqaaas.eosc-synergy.eu

Further Integrations & features (**in the making**)

- **Workflow Management Systems (WfMSs)**
  - **Suitable for**: individual QA work during the execution of a workflow
  - **Target/s**: components of a workflow
  - **Automation:** not through events, coded in the workflow specification
  - **Tool/s**: PyOphidia & CWL (InterTwin)

- **Data Management Systems**
  - **Suitable for**:
    - QA work on data (FAIR, quality dimensions)
    - React to CRUD operations on data repositories/lakes
  - **Target/s**: data
  - **Automation:** notifications sent by the data management system on CRUD operations
  - **Tool/s (potential)**: Rucio (InterTwin)

Use cases: **InterTwin's Climate Change (PyOphidia+SQAaaS)**

↘ Main integration leveraging SQAaaS GitHub actions:

1. **Validation of (PyOphidia) workflow description** document
2. **Test trained models statically** with small test set

↘ Further potential integrations (adhoc support into Workflow and Data Management Systems):**Stronger scientific validation of the model** with statistically relevant (larger) test set

- **Data quality and ML model FAIRness** (to be better explored)
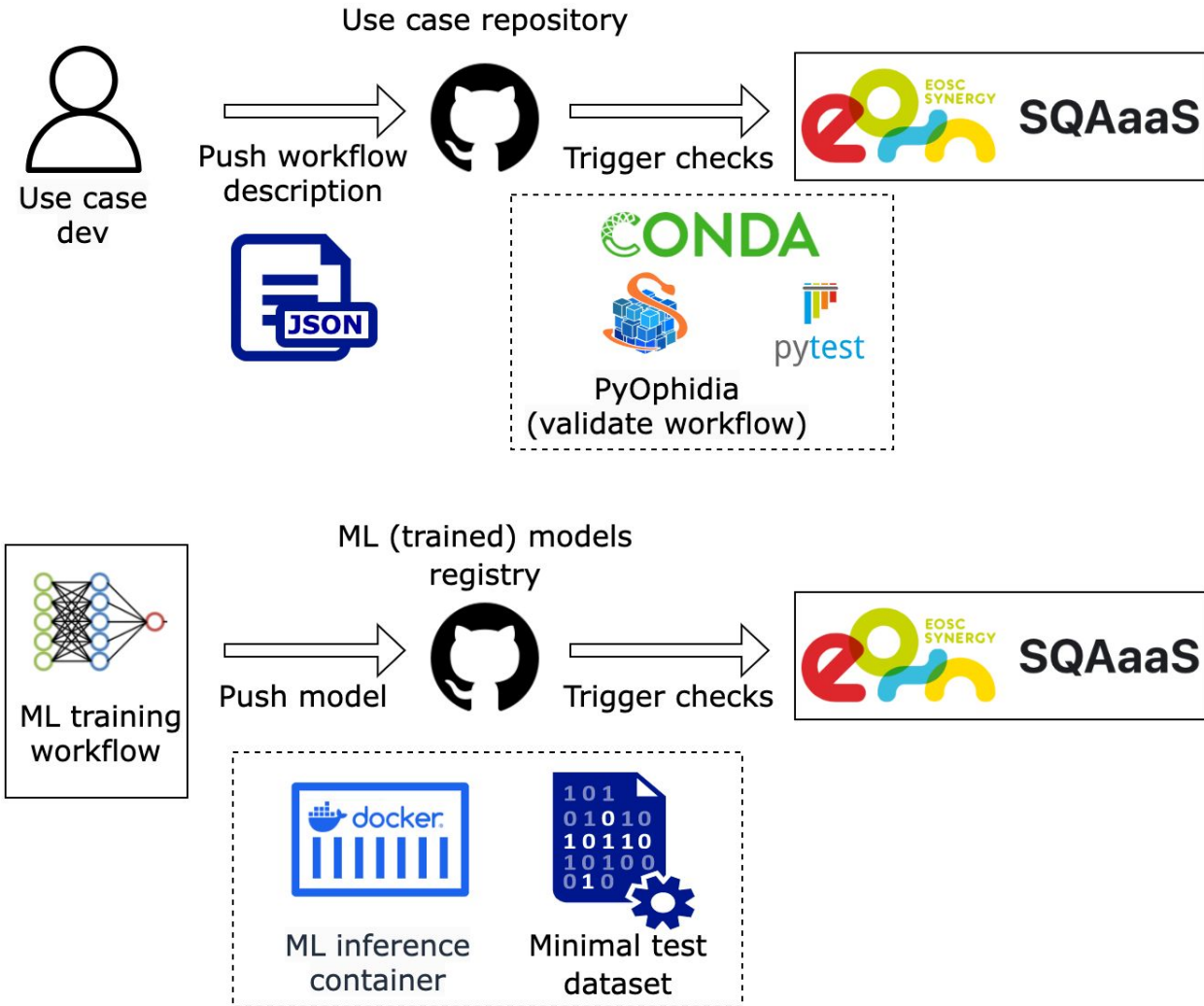
Use cases: **DTC development in DT-GEO**

workflow-registry

Pushes workflow code

Triggers GitLab CI pipeline

CI/CD

#1 Builds Docker images → image_creation

#2 Triggers SQAaaS assessment

EOSC SYNERGY | SQAaaS

#3 Triggers workflow execution

Jenkins + slurm workload manager + COMPSs
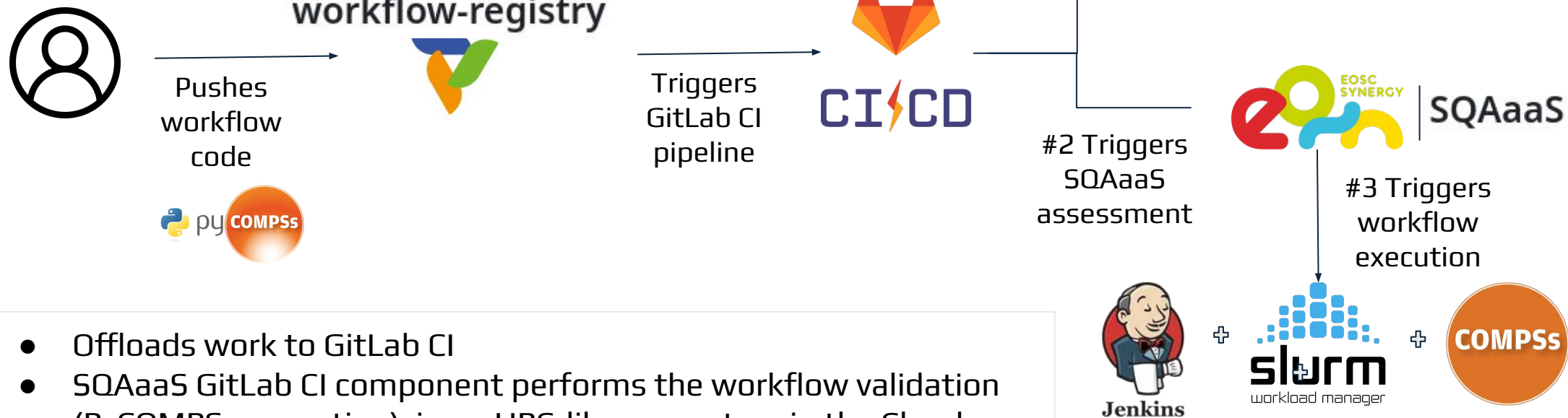
- Offloads work to GitLab CI
- SQAaaS GitLab CI component performs the workflow validation (PyCOMPSs execution) in an HPC-like ecosystem in the Cloud

Potential integrations with:
- PyCOMPSs WfMS: define QA work within workflow steps (e.g. data FAIRness)
- Software and data registries: react to changes

Further uses of SQAaaS platform: **QA tracker**

interTwin

Target	**Platform services, DTE core, DTE thematic modules, DT applications**
Frequency	**Upon product release**
Means	**Manual**
Reporting	**Tracked through JIRA issues (release-based)**
Managed by	**Software Release Management team (WP3, Mario & Joao from LIP bearing the burden)**

DT-GEO
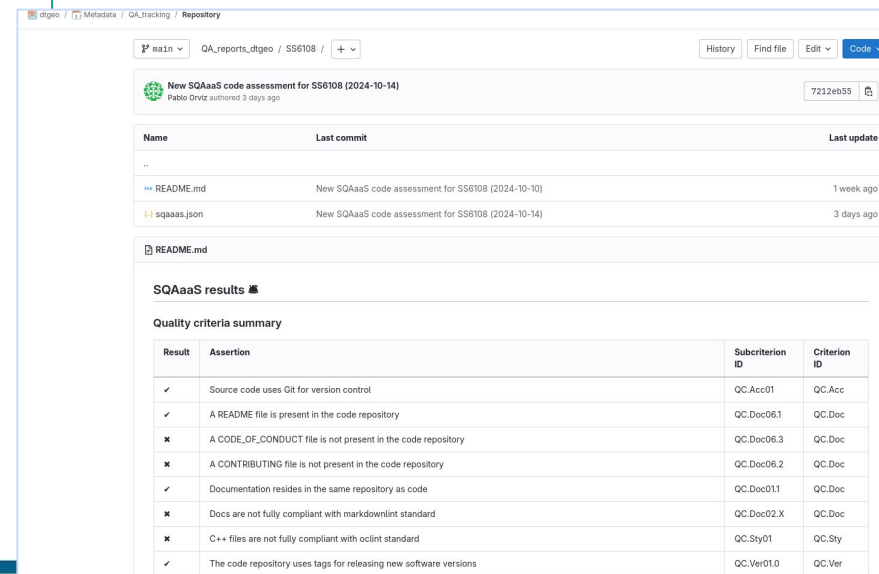
Target	**DT applications**
Frequency	**Periodically / cron-based**
Means	**Automated / Pipeline schedules / SQAaaS GitLab CI component**
Reporting	**Git-powered, summaries in READMEs**
Managed by	**Digital asset quality tracking (WP4's T4.2)**

Q&A

Thanks for your attention

orviz@ifca.unican.es
IFCA-CSIC