EUCAIM
CANCER IMAGE EUROPE

Project title: European Federation for Cancer Images

Project acronym: EUCAIM

Grant Agreement: 101100633

Call identifier: DIGITAL-2022-CLOUD-AI-02

D*6.1*: Design of the architecture and APIs and modules' specification of the federated analysis infrastructure

Partner(s): UB; BSC

Author(s): Josep Ll. Gelpí, Socayna Jouide, Laia Codó, Salvador Capella, Charles Hernandez-Ferrer

Date of delivery: 22/12/2023

Version: 1

Table of contents

Executive summary 3

Introduction and motivation 3

Overall strategy 4

Software screen survey 5

Content requested 5

Initial results 6

Annotation Hackathons 7

Federated processing architecture 7

Technical Showrooms 7

Demonstration experiments 7

Global design 7

High-level architecture 8

Requirements 9

Analysis Platform API definition 10

Envisioned Components and Technologies 11

Development Roadmap 12

Current status 12

Short term goals (2024 Q1-Q2) 12

Mid term goals (2024 Q3-Q4) 13

Executive summary

The EUCAIM project endeavours to establish a robust European infrastructure for the storage and analysis of cancer image data. This initiative is built upon two key foundations: the accumulation of a substantial image dataset and the creation of a secure analytical infrastructure. The project's Work Package 6 (WP6) focuses on developing a federated analysis infrastructure, enabling analysis, including AI training and inference, while preserving data on their original sites under specific regulatory frameworks.

The Federated Processing (FP) infrastructure operates within the broader EUCAIM ecosystem, relying on core services like Authentication and Authorization, Data and Tools Catalogues, and Negotiation for data access. This federated approach ensures that data remain at their source, adhering to specific regulations. The FP infrastructure addresses the challenge of analysing decentralised data by facilitating analysis tools to operate in a "federated/distributed" mode, transmitting only the analysis results to a central site.

To inform the development of the FP infrastructure, the project conducted a comprehensive survey of existing software modules, annotation hackathons, technical showrooms, and demonstration experiments. The survey revealed the current landscape of available software, with a focus on licensing, maturity levels, and capabilities. Technical showrooms provided insights into platforms and tools for distributed and federated data analysis, leading to the identification of key requirements and challenges.

The proposed FP architecture involves core components such as an Analysis Platform, Message Broker, Orchestrator, and relevant Software. The system aims to orchestrate tasks efficiently across distributed nodes while ensuring secure communication and adherence to defined hardware and software requirements.

The project's development roadmap outlines short-term goals, including the completion of orchestrator development, additional technical demonstrations, and the release of Data APIs at multiple sites. Mid-term goals involve the completion of the layered infrastructure, evaluating efficiency in real data sites, adapting privacy-preserving protocols, and deploying the EUCAIM toolbox.

Introduction and motivation

The EUCAIM project aims to build a large European-based infrastructure for the storage and analysis of cancer image data. This objective relies on two main pillars: i) the collection of a large dataset of images and ii) the development of a secure infrastructure for analysis of such images. Due to the sensitive nature of medical images, we must assume that a significant part of the EUCAIM dataset will be subjected to restriction on the distribution and even on the analysis itself. For this reason, the analysis framework developed for the EUCAIM infrastructure should be designed in a way that transmission and access restrictions can be honoured following a “privacy by design” paradigm. To this end, EUCAIM’s WP6 has been conceived to develop such analysis infrastructure based on a federated approach. This will allow to perform the envisioned analysis including AI training and inference while keeping data in their original location, under the specific regulations applying on each data site. Although some of the data available on EUCAIM will be centralised (e.g the Atlas of Cancer Images), the best suited approach that will serve both scenarios is a federated approach. D6.1 will cover the overall description of the infrastructure and its components and the outline of the process followed and expected roadmap.

The main assumptions taken to define the Federated Processing infrastructure are:

1. This infrastructure will work as part of the overall EUCAIM infrastructure and relies on a series of core services (developed at WP4). These services include Authentication and Authorization, Data and tools Catalogues, Negotiation for data access.
2. The infrastructure will not decide on the internal organisation of data sites, or specific computational power (except for the minimum resources defined on the enrollment procedure outlined on D2.1 - Onboarding invitation package). Instead, we will assume a uniform data access procedure and relay on data sites to provide the necessary interfaces.

3. The infrastructure will not address data authorization by itself, relying on the core services either by assuming no restrictions, or by passing the necessary authorization credentials to data sites.
4. Both tools for federated processing infrastructure and tools serving other aspects like data management will configure the project's tools Marketplace, that will include the necessary metadata for tools characterization, including their deployment strategies.
5. Analysis tools will work in "federated/distributed" mode, i.e. they should perform the required task without requiring moving any original data outside of the site. Only analysis results will be transmitted to a central site.
6. Likewise, analysis tools will not require establishing incoming network connection to data sites. In the case, tools do require such connections, the necessary site-dependent arrangements could be made (e.g. VPNs), but these are not considered here.

The building of the FP infrastructure is running closely and with full alignment with the development of core services (WP4) and data infrastructure (WP5). The initial design outlined here, may eventually evolve as the different components of such are consolidated.

Overall strategy

Following the initial assumptions indicated above, we have performed a series of actions to further define the software components and platforms to be included in the design. The main activities follow and are developed further in this document.

1. **Software screen survey**. EUCAIM is designed as an infrastructure and should be largely based on existing software modules, to assure a high TRL level from the initial stages. We have performed a global survey among EUCAIM technical partners about the software modules available and their state of development, and availability.
2. **Annotation Hackathons.** Workshops devoted to collecting the necessary metadata from the available tools following the recommendation and standards of the ELIXIR infrastructure.
3. **Technical Showrooms.** Technical workshops to understand further the tools and platforms to be involved in the infrastructure, and their potential fit in the ensemble of offered functionality.
4. **Demonstration experiments.** A series of computational experiments performed with selected platforms solving a specific data analysis challenge. The experiments are done in a real distributed scenario including both computational and data sites. One of the main outcomes of the experiments is the understanding of technical issues arising of the deployment of such platforms in a distributed network.

Software screen survey

This first section describes the technical survey initiated at the moment the project started in order to create an initial picture of the software ecosystem within EUCAIM as well as to depict which software was meant to be involved in WP4, WP5 and WP6.

Content requested

In order to report a software in the survey, 20 questions should be answered, some are closed questions and some are open questions.

Table 1. Information gathered for each software contributing to EUCAIM.

#	Question	Type[1]	Description
1	Name	O	Name of the software
2	Contributor	O	Who contributes the software to ECUAIM. Also, who is reporting the software.
3	Involvement	C	the involvement of the *Contributor* to the software `Options: developer, contributor, power user, user`
4	Segment	C	At which segment the software is placed. `Options: Intrastr. platform, FL platform, Analytical software, ML models`
5	Licence	O	Licence of the software if it has it (otherwise N/A).
6	Open source	C	Is the software open source? `Options: Yes, No`
7	Free software	C	Is the software open source? If so, the web page and the source code were asked. `Options: Yes, No`
8	Has a dataset?	C	Does the software include a dataset? If so, the URL to the source of the dataset (doi, github…) was asked. `Options: Yes, No`
9	Maturity level	C	The “Technology Readiness Level” of the software according to the contributor `Options: The 9 TRLs`
10	Data security	C	Is data security taken into account by the software? `Options: security-by-default, 3th party dependent, depends on upper-layer, not taken into account`
11	Sensitive data?	C	Is the software capable of dealing with sensitive data in a secure fashion? `Options: Yes, No`
12	Certification	C	Has the software been developed under a specific certification? Or has the software been awarded with a certification? If so, which? `Options: Yes, No`
13	DPIA up to date	C	Has the software been evaluated under DPIA? `Options: Yes, No`
14	Technical requirements	C	Does the software require intensive CPU usage? Does the software require intensive GPU usage? Does the software have any other technical requirements? `Options: Yes, No`

[1] O: Open question (free text); C: Close question (fixed options were provided as a selection-box).

15	Internet access	C	Does the software require access to the Internet for its expected proper function? `Options: Yes, No`
16	Running mode	C	Whis is the running mode of the software? `Options: Interactive, Batch, Both interactive and batch, Other`
17	Input required	O	Short description of the expected input of the software.
18	Output provide	O	Short description of the expected output of the software
19	What library(es) do you use?	O	List of (3th party) libraries used by the software that have to be installed before the software can be used.
20	What aggregation strategy do you implement/require?	O	If the software is a platform for federated learning, description of the aggregators that can be used.

Initial results

On November 30th, 34 softwares were reported. Figure 1 shows the results obtained from the technical screening, summarising the answers to questions 3 (subfigure A), 4 (subfigure B), 5 (subfigure C), 6 and 7 (subfigure F), 11 (subfigure E), 14 (subfigure H), and 16 (subfigure D).

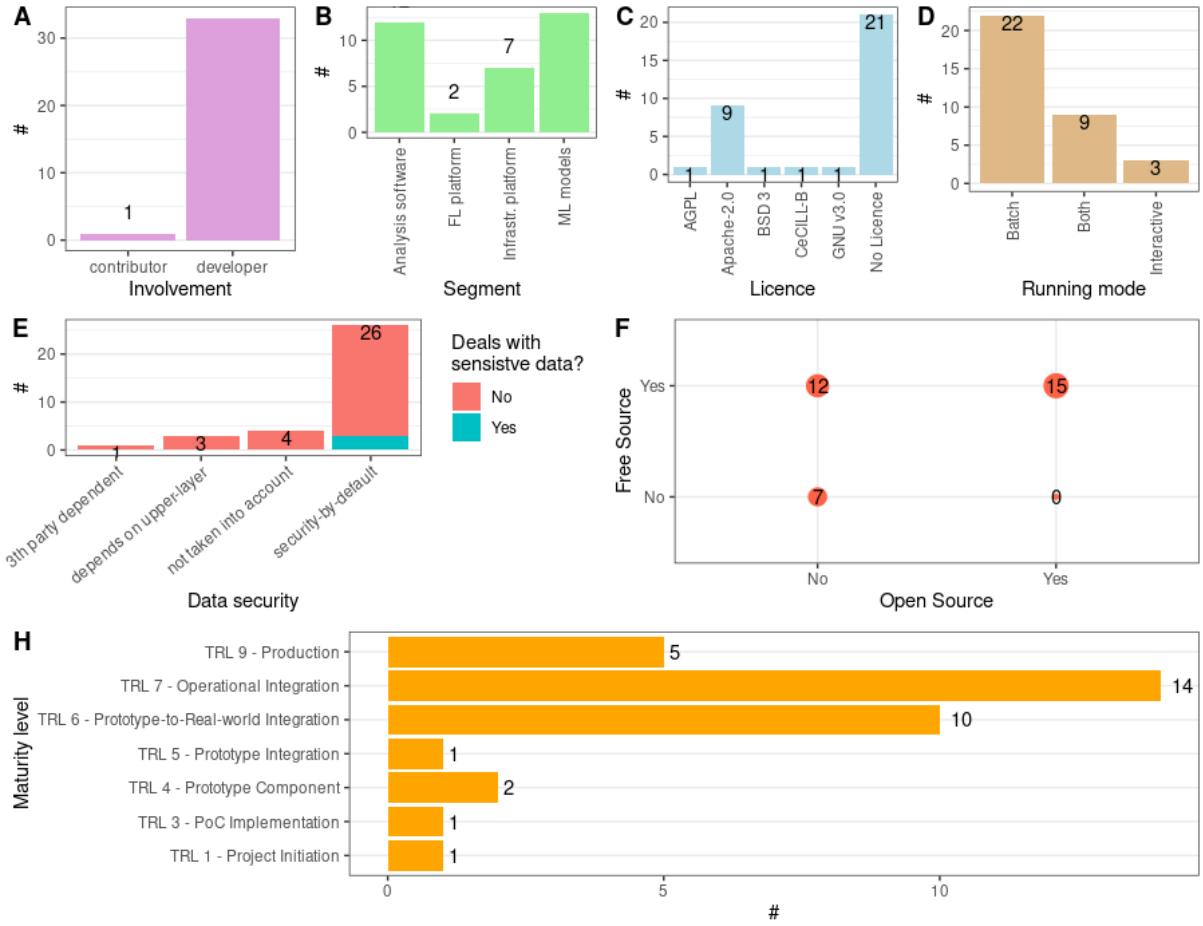

Figure 1. Results of the technical screening on November 20th.

On this initial step we identified that 61.67% of the reported software lacks a licence (Figure 1-C). From all, 35.29% were assigned to “analysis software”, 5.88% as “federated learning platform”, 20.59% as “infrastructure platform”, and 38.24% as “machine learning models” (Figure 1-B). According to the maturity level of registered software, 11.71% are under TRL 4 stage, 32.35% are on prototyping (TRL 5 or TRL 6) and 55.58% are at TRL 7 or over. See MS9 for more details.

Annotation Hackathons

To assure the proper description and registration of the software tools and modules to be used in the infrastructure, it was decided to follow the guidelines of the ELIXIR Tools Platform, regarding best practices for software development and management. In particular, the proposal included the registration of EUCAIM software in the bio.tools ELIXIR registry (https://bio.tools) and the Biocontainers packaging facility to provide software containers. This aimed to enable public referencing of software packages and facilitate future workflow design. Additionally, the use of the ELIXIR tools platform infrastructure will allow the inclusion of EUCAIM tools in other ELIXIR products like OpenEBench (https://openebench.bsc.es) software quality monitoring services.

To facilitate the dissemination of these recommendation among EICAUM tool providers, we plan to host a series of content·a·thon's with a dual objective: i) boost visibility for the software screen survey (outlined in the preceding section); ii) train and encourage the registration on the bio.tools portal. At present, EUCAIM collection is already created in the bio.tools portal and 8 software packages are already fully registered.

Federated processing architecture

Technical Showrooms

The first Technical Showroom was focused on software and platforms for distributed and federated data analysis. The session was held at the facilities of the Barcelona Supercomputing Centre on July 29th, 2023.

8 EUCAIM WP6 partners participated in the first of the Technical Showrooms (ITI, UPV, INRIA, Owkin, HULAFE, Qibim, CNRS, DKFZ). The software displayed on this initial session were Aitana, Harmonization, IM (EC2), OSCAR, Fed-BioMed, Substra, VIP, MITK, Kaapana.ia. Additionally several models and software for local analysis and data management were also presented as by-products of some of the federated and distributed software.

The main conclusions of the session were:

- EUCAIM needs a uniform way to package the software so they can be distributed across any sort of nodes, but especially when software to be used on newly installed data nodes.
- Federated processing within EUCAIM needs a way to orchestrate the software so the different modules of federated/distributed platforms are raised properly, software for local execution is raised on the right data nodes, and results are collected accordingly.

A second session, focused on platforms for distributed analysis, will be held next January 15th (M13).

Demonstration experiments

As the main outcome of the first Technical Showroom, and as part of the milestone MM1, a demonstrator including the three selected federated learning platforms (Fed-BioMed, Substra, and Flower) was designed. The objective for each platform was to showcase the training of two models—one for clinical data and one for image data. The outcome of the demonstration comprised a Demonstration video and a descriptive document. Currently, a second demonstrator centred on platforms for distributed analysis is being prepared.

Global design

In this section, we will define and discuss federated processing, which is a method of processing data that involves multiple entities collaborating on a task without sharing their data directly. This is achieved by using a decentralised architecture that allows each entity to maintain control and ownership over its data while still contributing to the task at hand.

High-level architecture

The high-level architecture of the proposed federated processing system involves a series of core services that coordinate the collaboration between the entities. These core services provide access to the remaining elements like user authentication and user authorization.

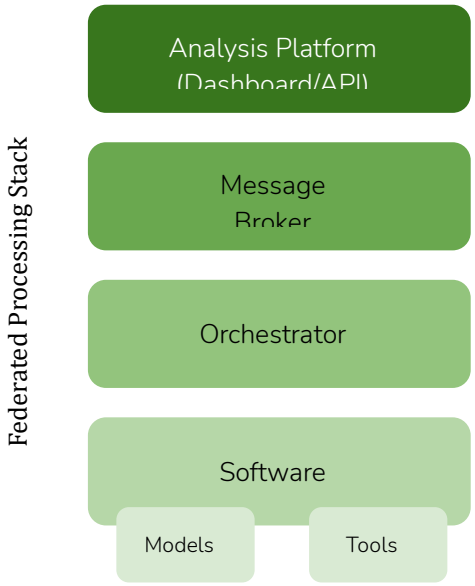

Figure 2. Layers of the distributed and federated processing architecture

Figure 2 depicts the layers of the architecture used to implement the federated processing platform. Using a federated learning use-case as example, the following components are involved in the flow to perform an experiment:

- **Analysis Platform** (Dashboard and API). User interface where the user triggers an experiment and retrieves the results. The API can be called from the platform's dashboard, its command line tools, or from the EUCAIM general dashboard. The user has to be already authenticated and the datasets used for the experiment authorised.
- **Message broker.** Currently based on RabbitMQ. It is the component that centralizes the scheduling of the different tasks in the process. The message broker accumulates the execution requests made by the Analysis platform. Data nodes will access their specific queues on the message broker to obtain the assigned tasks and eventually convey the analysis results back to the Analysis Platform. The use of a centralized broker allows the infrastructure to perform distributed activities without the need of incoming communication at the data sites.
- **Orchestrator**. A daemon is set up in each node that will connect to the message broker to obtain the assigned tasks and initiate any software execution required. The daemon knows the resources available where it is instantiated and should be able to interact with the site specific execution infrastructure (docker, kubernetes, batch queues, ...). The orchestrator assumes that the required software is already instantiated in the site.
- **Software (models/tools)**: These are the software pieces initiated by the daemon. It is expected that the tools will be already available in software containers, or can be installed on-the-fly using for instance a git repository as source.

Requirements

Functional requirements

1. Service should provide OIDC-compliant APIs and with appropriate documentation (ideally machine readable) for inter-module communication.
2. Service should be able to interoperate with other components of the platforms, i.e. main dashboard, data query and access services, etc.
3. The federated processing platform should be able to automatically start the process at the local sites.
4. Service should allow for user identification through the EUCAIM AAI system.
5. Service should provide the capability to load and choose a specific model/analysis
6. Service should provide an interface for the user to start and monitor the process.
7. Service should support a secure communication system between the different nodes.

Non functional requirements

1. Hardware: GPUs that support CUDA and have at least 12GBs of VRAM and 32GBs RAM, recommended: GPUs with more 24GBs VRAM, 64GBs RAM. (to be revised according to EUCAIM on -boarding conditions)
2. Allowed outgoing network connection to be able to connect to EUCAIM core services and Federated processing management.
3. Allowed software installation (docker or singularity containers).
4. Allowed (internal) access to site specific Data API(s).

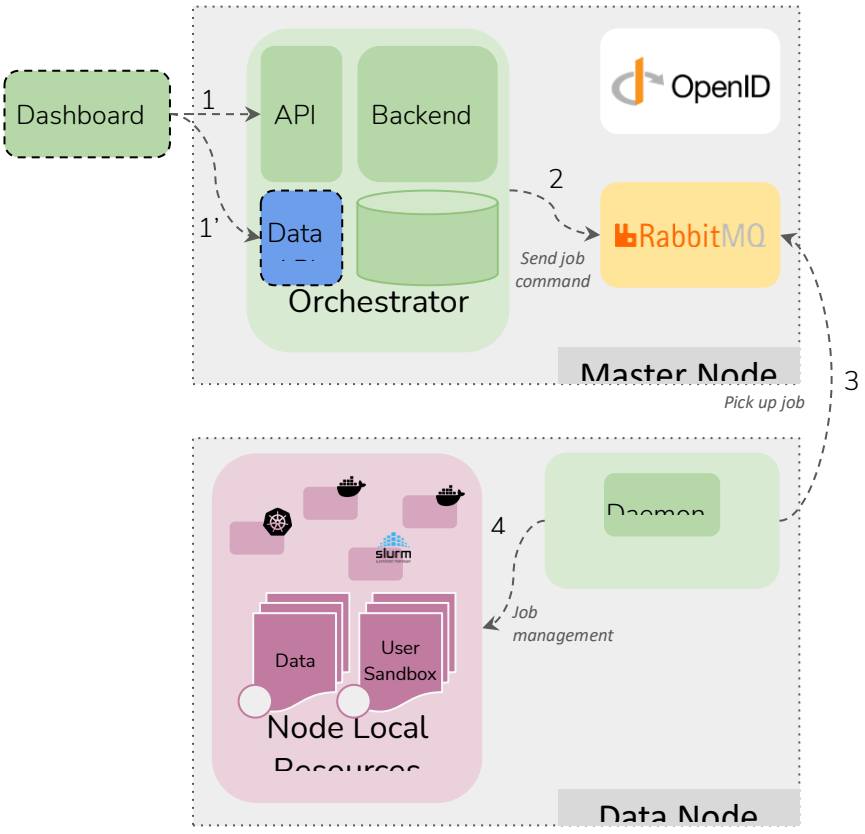

Figure 3. Components of the distributed and federated processing architecture

In Figure 3, the detailed process of the federated processing system is illustrated. Components within or outside the system are indicated with dashed lines. Users, accessing either the federated processing' dashboard or the central dashboard, can choose datasets and tools for experiments (figure 3, step1). Tools configuration (i.e. available parameters, associated container/s, computational resources requirements, etc.) are sourced from the Database, which in turn, is linked to the central EUCAIM tools registry of approved software. Using a site's specific Data API (step 1'), users will request data materialisation. After materialisation, job configurations are sent to the message broker (step 2), queuing for nodes with relevant data. Orchestrators at nodes retrieve this information (step 3) and execute the jobs, triggering the necessary containers and data volumes (step 4) by means of a local executor. The current prototype implements Docker as container runtime, although others managers could be included to orchestrate local resources, like Kubernetes or scheduling queue systems (i.e. SLURM). Results are then made available through the Data APIfor user/Dashboard retrieval. A more in-detail display of the communication between components can be seen in Figure 4.

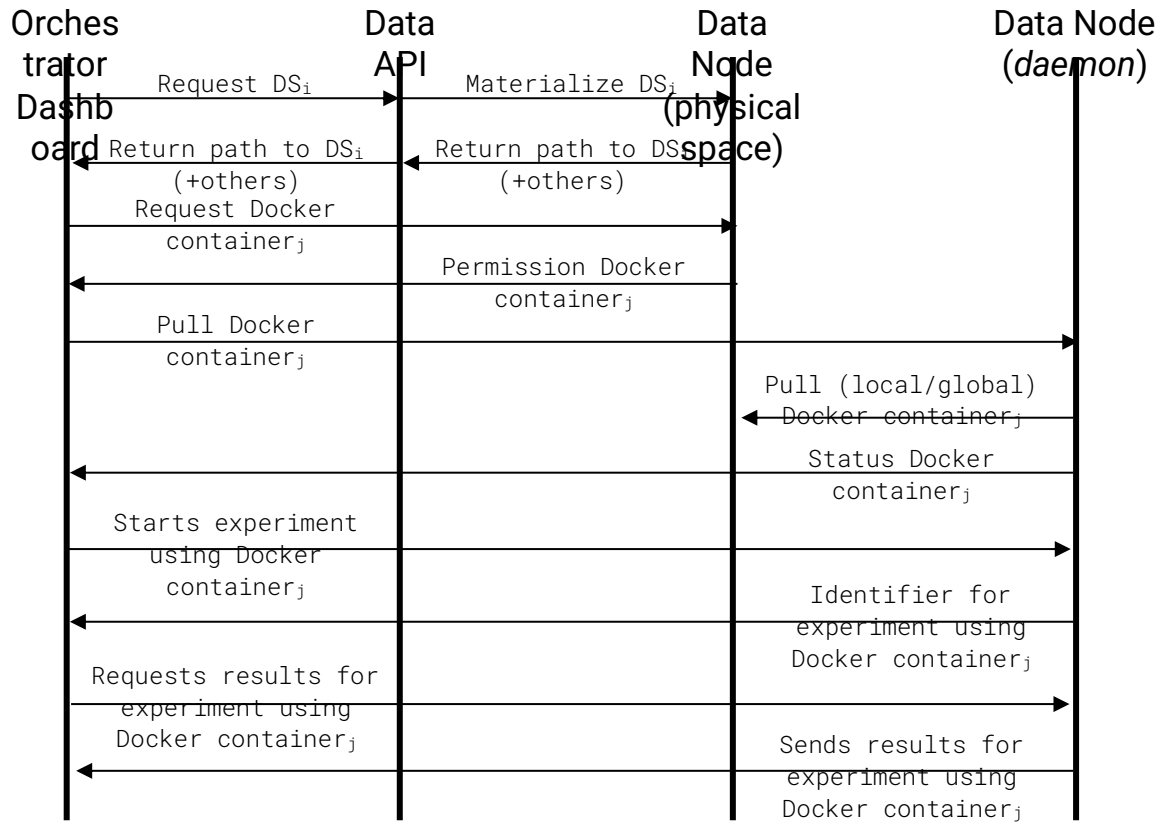

Figure 4. Full use case of designing a federated processing experiment (dockerized software).

After the user selects datasets and the processing software on the dashboard, it triggers the data API to materialise the chosen datasets. The data API then materialises the datasets in each data node, capturing POSIX-like paths to them, and defines the temporary working space, and the result storage location. These paths are relayed back to the dashboard. Using the paths, the dashboard requests an execution (typically running a Docker container or an alternative, such as a Kubernetes pod or submitting a queue job, depending on the specific infrastructure available) for the corresponding software from the daemon on each involved data node. Leveraging the physical paths and the 'Docker container,' the dashboard generates experiment metadata for each data node and initiates the experiment by dispatching it to each daemon. The daemon returns the running experiment's ID, allowing the dashboard to later query its status and retrieve the results.

Analysis Platform API definition

The API is designed to manage and execute tasks. It provides endpoints for interacting with tools, tasks, and performing health checks. The API supports authentication through OAuth2 and allows triggering tools, listing tools and tasks, and performing health checks on the system. The API communicates with a

database to store tools, tasks and hosts information along with the logging. The API will include these endpoints as a starting point (see the documentation of the present prototype here):

- Authentication:
  - POST /token: Get an access token using OAuth2 password credentials.
- Tools and Tasks:
  - GET /tools: List of available tools.
  - GET /tools/{tool_id}: Get information about a specific tool, which in turn, is pointing to the list of tasks to be executed remotely.
  - GET /tools/tasks: List of available tasks, which include all the details to dynamically build the execution command at each one of the sites with the local variables.
  - GET /tools/job/{tool_name}: Trigger a specific tool execution.
- Hosts:
  - GET /hosts/: List of federated active hosts
  - GET /hosts/{host_id}: Get information about a specific host, its role, in the federation, location, accessibility, resources, etc.
  - GET /hosts/health/: Perform a basic health check on the system.
- Documentation:
  - GET /docs: Returns the OpenAPI JSON document for API documentation.
- Data Operations:
  - PUT /tasks/{task_id}/input: Send input data for a specific task.
  - GET /tasks/{task_id}/output: Retrieve output data for a specific task.
  - GET /data/files/: user's available files as a list of URIs
  - GET /data/file/{file_id}: returns the file entry, defining its location, metadata, etc.
  - GET /data/file/{file_id}/access/: Returns a URI used to fetch the bytes of the file (computational nodes only)
  - PUT /data/file/{file_path}: Creates a new file entry

Envisioned Components and Technologies

The precise definition of software components are still in process, but at the design time it is reasonable to envision a series of components, with an initial selection of software. Following the development of the different activities (technical showrooms and demonstrations), the list of components will be revised.

- **ELIXIR Tools platform**
  - Link: ELIXIR Tools platform
  - Description: The ELIXIR Tools platform serves as a centralised resource for accessing and discovering tools in the life sciences, promoting collaboration and efficiency in research. Main components to leverage in the development will be the bio.tools software registry, the Biocontainers packaging facility and the OpenEBench benchmarking and tools monitoring system.
- **Message broker (RabbitMQ)**
  - Link: RabbitMQ

    - Description: RabbitMQ is an open-source message broker that facilitates communication between distributed systems, ensuring reliable and scalable data exchange. The Broker will centralize the task scheduling for the distributed processing.

- **openVRE (dashboard)**
    - Link: openVRE
    - Description: Open Virtual Research Environment (openVRE) will be used to build the Analysis dashboard, alone or included in the main EUCAIM dashboard. It will provide a user interface able to handle all operations included in the federated processing.
- **Building Blocks (packaging)**
    - Link: BioBB
    - Description: The Building Blocks Strategy for packaging involves a systematic approach to packaging software, breaking it down into modular components to enhance maintainability and reusability.
- **Docker/singularity containers for software deployment**
    - Docker, Singularity, Podman containers offer lightweight and portable solutions for packaging and deploying software, enabling consistency, and easy deployment across different computing environments.
- **OIDC-based AAI**
    - An Authentication and Authorization Infrastructure (AAI based on OpenID Connect (OIDC) and SAML, provides secure and standardised user authentication, enhancing the overall security of systems and applications. Fed. processing infrastructure will use OICD to accept authentication and authorization credential from EUCAM AAI service
- **Federated Learning Platforms and Analysis tools**
    - Following the software survey indicated above, a comprehensive toolbox including both Federated or local execution tools and platforms will be collected. As the initial demonstrators, Subtra, Fed-BioMed and Flower (using USB implementation) are being tested as Federated learning platforms.

Development Roadmap

Current status

At the present stage of the development, working implementations of Flower-UB, and Fed-BioMed have been deployed at three data sites (UB, BSC and FORTH). The installation of the third platform that has a more complex deployment procedure is under process. Flower-UB already follows the overall architecture depicted above. The definition of the Analysis Dashboard API is completed (see above) and an initial prototype is being tested.

Short term goals (2024 Q1-Q2)

- Complete the initial orchestrator development for all FL platforms
- Hold a 2nd Technical Showronn and a demonstration exercise to include federated analysis and distributed execution platforms (Showroom scheduled for 15th Jan 2024). This exercise will involve already European computational infrastructures like EGI, in preparation for T6.5.

- Perform additional iterations for the content-a-thon for software annotation and registration. Work together with bio.tools and EDAM to assure optimal coverage of EUCAIM tools and data.
- Eventually adapt or extend the design to cover for the new added scenarios extracted form such demonstration exercise
- Initial release of Data APIs in at least two data sites (work is ongoing with the Chameleon data site at UPV)
- Complete a survey on the technical requirements and capabilities of the data sites.
- Complete the initial release of EUCAIM toolbox (in collaboration with WP4 and WP5)

Mid term goals (2024 Q3-Q4)

- Complete the layered infrastructure including specific requirements of the participating sites
- Evaluate the efficiency of the federated processing architecture in real data sites, with eventual communication restrictions.
- Adapt privacy preserving protocols (from T6.3) to the architecture design.
- Develop the EUCAIM project space on openEBench.
- Deploy EUCAIM toolbox on data sites and computational nodes.