Initial high-level architecture specification

Deliverable ID	D 3.2- Initial high-level architecture specification
Work Package Reference	WP1
Issue	1.0
Due Date of Deliverable	28/02/2023
Submission Date	22/03/2023
Dissemination Level	PU
Lead Partner	CSIC
Contributors	CSIC
Grant Agreement No	**101058593**
Call ID	**INFRA-2021-EOSC-01**

Funded by
the European Union

Prepared by	Reviewed by	Approved by
Álvaro López García	Pablo Orviz	Project Management Board
Lisana Berberi	José González Abad	
Valentin Kozlov	Mario David	
Marica Antonacci	Aida Palacio	
Amanda Calatrava	Marcin Plociennik	
Sergio Langarita		
Germán Moltó		

Issue	Date	Description	Author(s)
0.1	20/09/2022	Initial ToC following arc42 model	Álvaro López García
0.2	20/09/2022	Architecture draft following C4 model	Álvaro López García
0.3	01/01/2023	Initial deliverable structure	Álvaro López García
0.4	26/01/2023	Architecture updates	Álvaro López García
0.5	01/02/2023	Input from WP6	Lisana Berberi
0.6	06/02/2023	Updated architecture diagrams, including OSCAR	Sergio Langarita Amanda Calatrava Germán Moltó
0.7	10/02/2023	Updated architecture diagrams, including orchestrator	Marica Antonacci
0.8	20/02/2023	Dynamic views of the architecture	Álvaro López García Lisana Berberi Sergio Langarita
0.9	22/02/2023	Deployment views of the architecture	Álvaro López García
0.10	25/02/2023	Draft version to review	Álvaro López García
0.11	15/03/2023	Including AIoD feedback	Álvaro López García
1.0	20/03/2023	Final version	Álvaro López García

Table of contents

1. Introduction and goals

This document provides the initial architecture documentation of the AI4EOSC platform. This architecture has been elaborated with input from the D3.1 - State of the art landscaping and initial platform requirements specification deliverable and the "D6.1 Analysis of user applications, collection of requirements" [2] and will serve as the basis for the WP4 and WP5 implementation plans ("D4.1 - Implementation Plan" [3] and "D5.1 - Implementation Plan" [4] respectively).

The architecture described in this deliverable is an evolution of the DEEP-Hybrid-DataCloud architecture [5], whose platform is being evolved and enhanced by the AI4EOSC initiative. This architecture is expected to evolve as needed during the project development, and it will be reviewed in the "D3.3 - Final high-level architecture specification and catalogue of services" document [6]. The live version of the Architecture document is found in the AI4EOSC Confluence Page [7], under the High-level architecture specification [8] document.

Document structure and notation

The document is structured following the arc42 architecture template [9], a lightweight architecture description and documentation template aimed at delivering clear, understandable, simple and effective architecture documents.

The architecture notation and diagrams follow the C4 model [10], a lightweight approach to visualize and diagram software architectures. The C4 model is under discussion to be used in the EOSC Horizon Europe projects working group on Technology and to be adopted by the EOSC Task Force on Technical Interoperability of Data and Services. C4 provides an abstraction-first approach providing guidelines to ensure that architecture documents are understandable, regardless of the chosen notation. The model is based on a hierarchy of abstractions, each of them corresponding to a different level in the system: a **software system** is made up of one or more **containers** (applications and data stores, not to be confused with the term used for operating system level virtualization), each of which contains one or more **components**, which in turn are implemented by one or more **code elements**. In order to visualize this hierarchy, one must define a collection of **Context (i.e. systems interacting with each other)**, **Containers**, **Components**, and **Code** diagrams.

In the current AI4EOSC architecture diagrams we have developed the following main visualizations:

- **System Landscape diagram**: This diagram provides a set of related systems and how they interact with each other, showing dependencies with external entities and the user relations
- **System Context diagrams**: This diagram describes how a specific system fits in the overall landscape.
- **Container diagrams**: This diagram provides a view on how one system is built up on several containers.

Moreover, this document will also include some additional visualizations, in order to provide a better view of the architecture, the user interactions and the production deployment of the systems:

- **Dynamic diagram**: This will be used in the "Runtime view" section of this document, providing a view on how the different C4 containers work together in order to complete a user action.
- **Deployment diagram**: This diagram will be used in the "Deployment view" section of this document in order to provide a vision on how the software systems and its C4 containers are deployed and mapped to the underlying infrastructure.

The AI4EOSC C4 model diagrams are automatically generated using the Structurizr Domain Specific Language (DSL) [11], with an online version of the updated diagrams in the following links:

- Diagrams (online live version) [12].
- Code (DSL) [13].

Goals

The aim of the AI4EOSC "Artificial Intelligence for the European Open Science Cloud" is to deliver an enhanced set services for the development of Artificial Intelligence, Machine Learning and Deep Learning models and applications in the EOSC. The project is focused on enhancing existing services in the EOSC marketplace (i.e. the DEEP-Hybrid-DataCloud platform) and its services will make use of advanced features such as distributed, federated and split learning; provenance metadata; event-driven data processing services or provisioning of ML/AI services based on serverless computing.

The vision of the AI4EOSC project is to increase the service offer in the EU landscape by expanding the European Open Science Cloud (EOSC) ecosystem to support the effective utilization of state of the art AI techniques by the research community. AI4EOSC will strive also to provide an easy entry point for non technological users, resulting in an effective usage of pan-European distributed e-Infrastructures.

Requirements overview

The AI4EOSC requirements are being tracked in a specific document, "D6.1 Analysis of user applications, collection of requirements" [14], these requirements are driven by the needs of the target users and the objectives of the project. At the time of writing this document, the complete list of 44 requirements from the three use cases defined in the project proposal is available in [15].

Quality goals

The main quality goals, aligned with the model defined in the ISO/IEC 25010 standard, of the AI4EOSC architecture (not to be confused with project goals) are the following:

- **Operability**: The system is easy to operate and control by platform operators and resource providers.
- **Compatibility**: The system is compatible with existent technologies, standards and community best practices.
- **Functional Suitability**: The system fulfils the user and operator expectations, in terms of completeness, correctness and appropriateness.
- **Performance Efficiency**: The system only consumes the resources needed, providing appropriate performance for each case.

Stakeholders

The following table provides a brief overview of the key stakeholders and their expectations. By understanding the expectations of each stakeholder, it is possible to ensure the AI4EOSC platform meets the needs of its users and continues to evolve to meet their changing needs.

Stakeholder	Expectations
EOSC users	These stakeholders represent the primary users of the AI4EOSC platform, who expect to have access to a wide range of resources and tools for scientific research and innovation. They expect the platform to be user-friendly, secure, and reliable, providing seamless access to the resources they need.
Platform operators	The platform operators are responsible for ensuring the AI4EOSC platform meets the expectations of its users. They expect to have the necessary technical infrastructure and resources in place, and to be able to manage and maintain the platform effectively.
Resource providers	These stakeholders provide the resources that are used by AI4EOC users. They expect to have a clear understanding of their responsibilities and obligations, and to be able to contribute effectively to the AI4EOSC platform
Research Infrastructures and e-Infrastructure managers	These stakeholders are considered as potential users (at a high level) and integrators of the AI4EOSC platform. As such, they need to understand a high-level view of the infrastructure, the technological requirements and the provided functionality, with clear boundaries and interfaces.
AI-on-Demand platform and AI4Europe project	Facilitator of European research and innovation in AI, with the objective to support all solutions and tools that contribute to the ecosystem of excellence and the ecosystem of trust, which define the European Vision of AI.

AI-on-Demand connection

The AI-on-Demand (AIoD) is focused on coordinating and providing support for all the AI initiatives in Europe. AIoD is currently designing a high level architecture towards the delivery of the platform, defining requirements from all relevant stakeholders. In this context, the AI4EOSC project has reached the AI-on-Demand technical coordination in order to ensure that the architecture that is being implemented by AI4EOSC is aligned with the ongoing effort of the AIoD platform, identifying common areas where both projects can collaborate in order to define horizontal services that could be co-designed and co-developed. However, due to the different project deadlines, it has not been possible to close the collaboration at the time of writing this deliverable. Nevertheless, the current AI4EOSC architecture described in this document fits in the overall AIoD scope, and an initial identification of collaboration areas is undergoing, namely on model metadata, compute and training services and API definitions. In the short term, AI4EOSC aims to leverage the AIoD AI Assets Catalogue to build and publish the AI4EOSC Exchange assets. Further integration and alignment of activities is currently being done.

2. Constraints

The architecture described in this document is influenced by the following constraints, divided in technical, organisational at the EOSC level and conventions adopted by the consortium.

Constraint	Explanation
Technical	
DEEP-Hybrid-DataCloud legacy architecture	The AI4EOSC platform is enhancing the DEEP-Hybrid-DataCloud (DEEP-HDC) [16, 17] platform, which already has an existing user base, as it is being offered in production through the EOSC portal. The new architecture must take that fact into account, providing a seamless transition between the legacy system and its evolved version.
Data ingestion	Data can be stored on a system external to the platform.
Data locality	Data can be stored at a different location where the computing is available. AI4EOSC should take into account the cost of waiting for a task, or transferring the data somewhere else.
Component deprecation	Some of the deployed components are being marked as deprecated by the supporting organizations (e.g. Mesos).
External components	Dependency on external components for the platform operations must be minimized, in order to avoid failures if external systems are down.
AI-hype	Data Science, AI, ML and passing through a hype cycle that favors the sprouting of new components. Mature solutions, and those closer to the AI4EOSC partners, should be favored

Organizational (EOSC-level)	
Distributed systems and operations	Federated and distributed operations are a must in the EOSC and the pan-European e-Infrastructures ecosystem. The design, architecture and its underlying technologies must take that fact into account in order to choose the best solution in terms of distributed operations.
EOSC Interoperability	The EOSC Interoperability Guidelines, and the system-of-systems approach must be taken into account from the design phases. Compatibility with the EOSC core is mandatory.
Conventions	
Open Source	All developments should be based on Open Source components.
Open API specifications	All APIs developed within the project will follow the OpenAPI specifications, with clear interfaces defined upfront.

3. Context and scope

The following diagrams include the AI4EOSC System landscape (Figure 1), as well as the System Context diagrams for the individual AI4EOSC systems: AI4EOSC platform, AI as a Service, PaaS orchestration and provisioning and MLOps system (Figure 2, 3, 4 and 5 respectively). A common legend is provided in Figure 6.

Starting from the highest level of abstraction the AI4EOSC System landscape (Figure 1) depicts how the foreseen users (EOSC users/scientists, general users and PaaS Operators) interact with all the AI4EOSC systems (i.e. services) and the other EOSC systems. This diagram also shows the high level relationships for all the designed systems and the interactions between them. The following diagrams (Figures 2, 3, 4 and 5) show the context for each of the systems, focusing only on them and without including additional interactions.

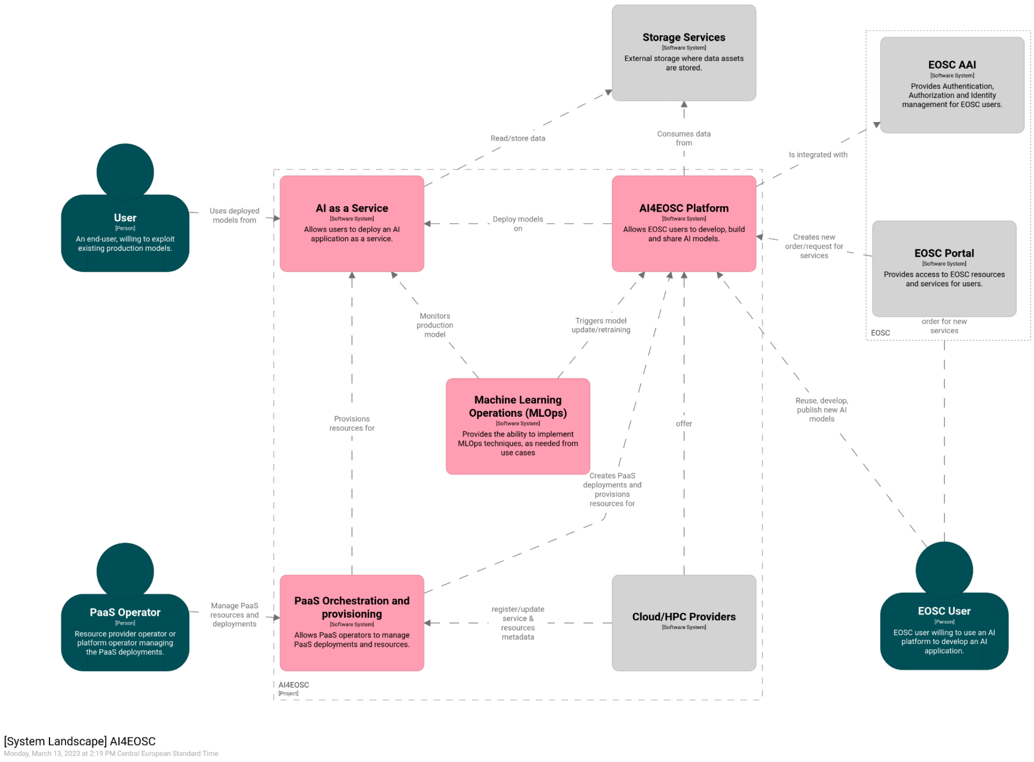

Figure 1: AI4EOSC system landscape [18].

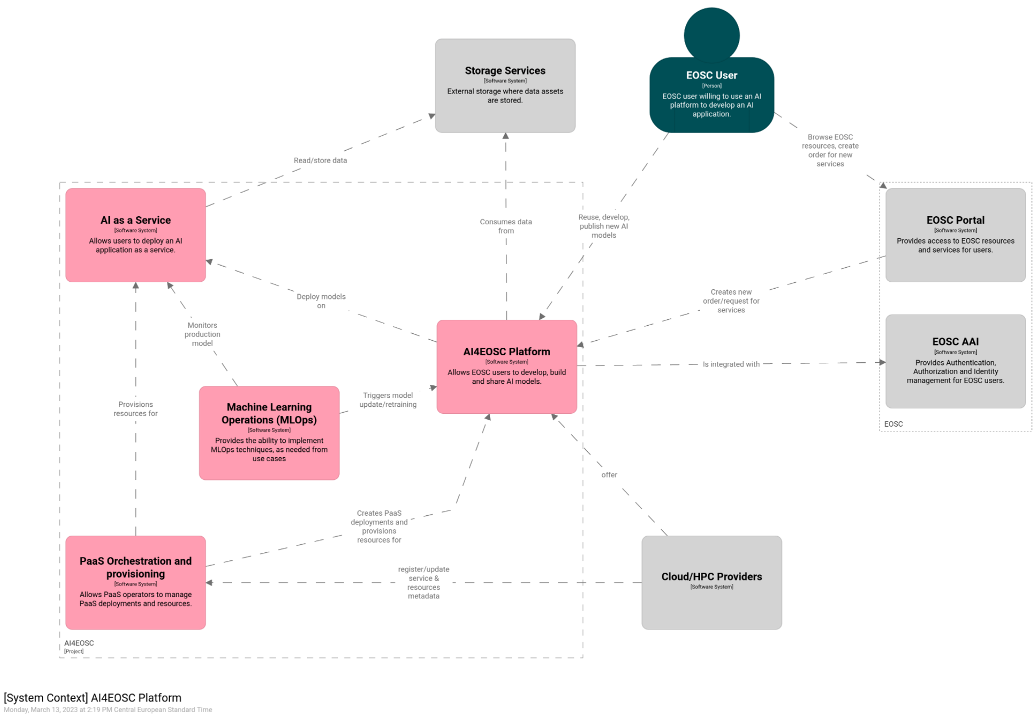

Figure 2: AI4EOSC platform system context [19].

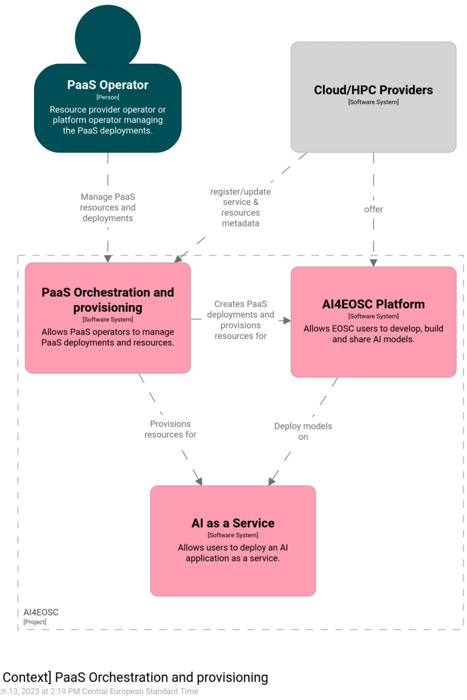

Figure 3: AI4EOSC PaaS Orchestration and Provisioning system context [20].

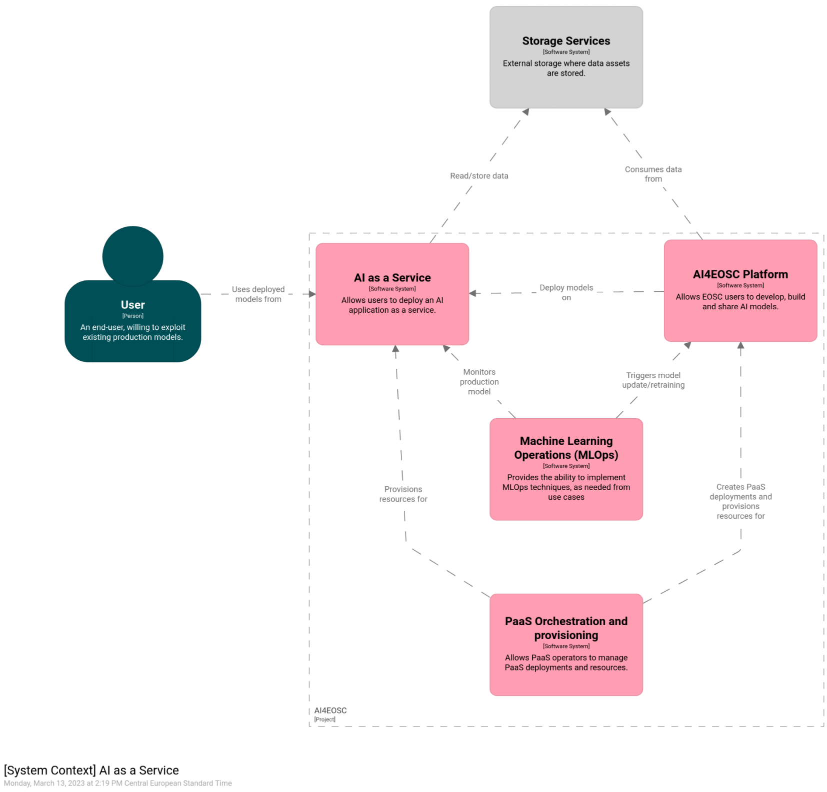

Figure 4: AI as a Service system context [21].

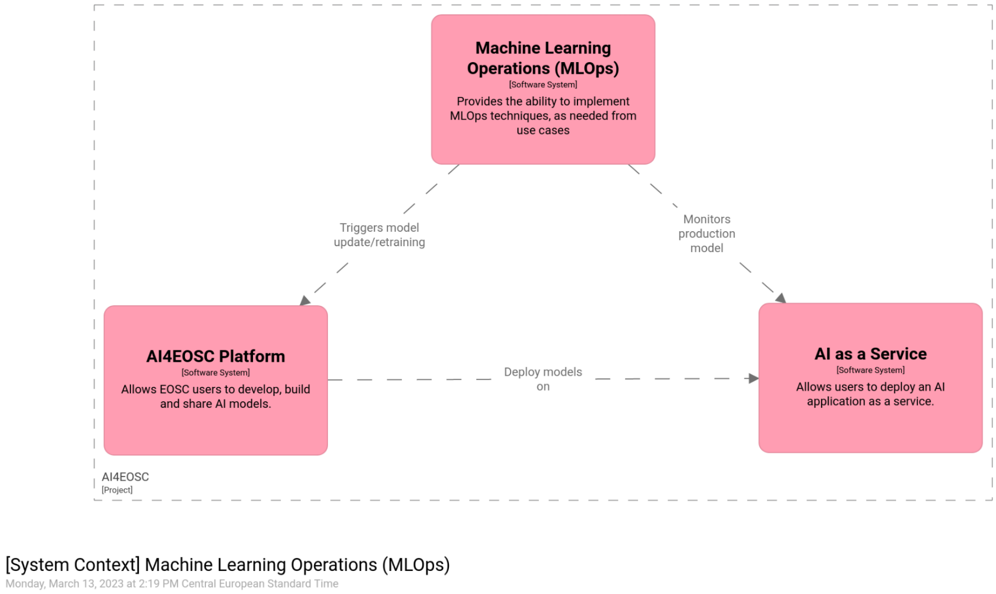

Figure 5: AI4EOSC MLOps system context [22].

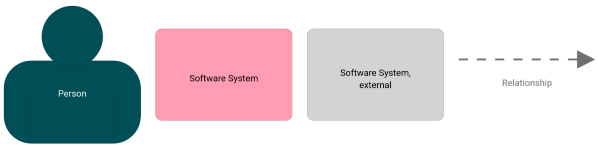

Figure 6: Legend for Figures 1-5.

4. Solution strategy

The following table relates the requirements, quality goals and constraints of the AI4EOSC platform with the architectural approaches and decisions adopted.

Requirement/Quality goal/Constraint	Decision
Key Requirements	
Provision/customise the required computing resources to optimise the ML/AI training process	Training complex machine learning models can be computationally intensive and time-consuming. In order to optimise the training process and achieve the best possible performance of the model, it is important to have the right amount of computing and storage resources and to tailor them to the specific needs of the training process. Designed three different components of the C4 model: Workload Management System, Training API, Compute Resources (offered from external

	Cloud/HPC providers as a software system) to fulfill this requirement.
Programmatically retrieve prediction information from Model Inference	Driven by the need to obtain insights and take action based on the predictions made by the model and to monitor its performance over time. Designed as an interaction between the OSCAR Manager component (in the C4 model part of "DEEP as a service" and the AI4EOSC Platform) [software system] to fulfill this requirement.
Provide interactive development environment to develop AI models while having access to the necessary computing resources and data sources	Enables data scientists, developers, and other users to interactively explore, experiment, and develop machine learning models. The IDE should be user-friendly and provide the necessary tools and features for the development and experimentation of machine learning models. Designed a separate component called "Interactive Development Environment" and added the respective relationships and interactions with other components, to fulfill this requirement.
Train an AI model efficiently	Driven by the need to reduce the time and resources required to develop and deploy effective models. One approach to achieving efficient training is to use parallel computing techniques, such as distributed training or GPU acceleration, which can significantly reduce the time required to train large models or get access to more resources to be able to train it. Grouped five relevant components to "AI4EOSC Training" with respective interactions to fulfill this requirement.
Deploy ML/AI model inference as a service	Deploy a trained model to a production environment and host the model on a cloud-based platform, using a containerization technology such as Docker, or deploying it on a server or cluster. Integrated OSCAR as a component (in the C4 model part of "DEEP as a service") to fulfill this requirement.
Try and combine different AI techniques (composite AI, Federated Learning)	Driven by the need to address complex and diverse problems that cannot be solved by a single AI technique. Designed the AI4Compose component as part of the DEEP as a service to provide service composition to fulfill this requirement.

Containerise the application(s)	Package and deploy an application in a consistent and reproducible way.
Enable secure access	Driven by the need to ensure that only authorised individuals or systems can access the AI system and the data it uses or produces. AuthN/Z based on GBAC using OpenID Connect.
Organise and track all training experiments	Developing an effective ML/AI model often involves multiple iterations, experiments, and adjustments to the model and its parameters. Keeping track of these experiments and their outcomes is important to understand what works and what doesn't, and to identify areas for improvement. Designed an experiment-centric dashboard, in order to keep track of all assets related with an experiment. Use state-of-the-art tools to keep track of data/model/code versions.
Publish and share results	Driven by the need to promote collaboration and knowledge sharing in the field of ML/AI. Designed the “AI4EOSC dashboard” container to fulfill this requirement.
Monitor the model and concept drift for the inference	Driven by the need to ensure the continued accuracy and effectiveness of ML/AI models in real-world settings. Designed the MLOps component to fulfill this requirement.
Preprocess and prepare input data sets to be used for training and evaluation	riven by the need to ensure the accuracy, effectiveness, and efficiency of ML/AI models. By properly formatting and cleaning the data, data scientists can reduce the likelihood of errors or biases in the model, and improve its ability to generalise to new data. Designed as an interaction between the “Storage system” and “AI4EOSC exchange” group to fulfil this requirement.
Provide metadata of a training dataset	The need to ensure the quality and consistency of data used in ML/AI models. Providing metadata can also help to ensure the reproducibility of ML/AI models by providing information on how the training data was collected

	and processed.
Goals	
Operability	Architecture follows the arc42 and C4 model, enhancing readability and understandability. System is based on well-known components with clear documentation. Specific guidelines, best practices and documents will be produced to ensure that the deployments are operable by resource providers. Platform will be provisioned through automation tools (i.e. Ansible) in order to ensure an easier deployment path.
Compatibility	Incorporate standards where appropriate (e.g. at the ML/AI model, at API level), adopting community standards when an open standard does not exist. Follow REST principles when designing APIs. Use OpenAPI Specifications.
Functional suitability	Involve users (through WP6) in co-design of architecture and AI4EOSC systems. Track user satisfaction with the design and evaluation of requirements being fulfilled according to user expectations. The AI4EOSC platform is decomposed into several systems, each of them providing different functionality for the users. This decomposition allows users to use each of the systems independently (for instance, the training platform can be used without the need of the orchestration and provisioning, as long as the operators are able to provide the resources needed by themselves).
Performance efficiency	Designed workload management system component taking into account the ability to easily track and monitor usage resource and consumption, avoiding the existence of idle resources. Provide interactive development environments for limited amount of time.
Constraints	
Component deprecation	In order to remove the DEEP legacy components that are not supported or deprecated upstream (like Apache Mesos), we have substituted them after an extensive state of the art survey, in order to replace them with alternative solutions.

DEEP legacy	The AI4EOSC platform will enhance the DEEP platform, being compatible at API level until a new major release is delivered.
Distributed systems	Utilization of a service mesh based on Hashicorp Nomad [23] and Consul [24] in order to provide seamless management of the distributed platform layer. A Service Oriented Architecture exposing well defined public facing APIs, allowing for easy changeability of components. Going down into each of the systems, as it will be seen in Section 5, the design follows a service oriented architecture, with a shared-nothing architecture with clear interfaces.
EOSC Interoperability	The architectural design has been done taking into account the EOSC interoperability guidelines existing at the moment.

5. Component view

The following diagrams include the AI4EOSC C4 Container view for the individual AI4EOSC systems: AI4EOSC platform, AI as a Service, PaaS orchestration and provisioning and MLOps system (Figure 7, 8, 9 and 10 respectively). A common legend is provided in Figure 11.

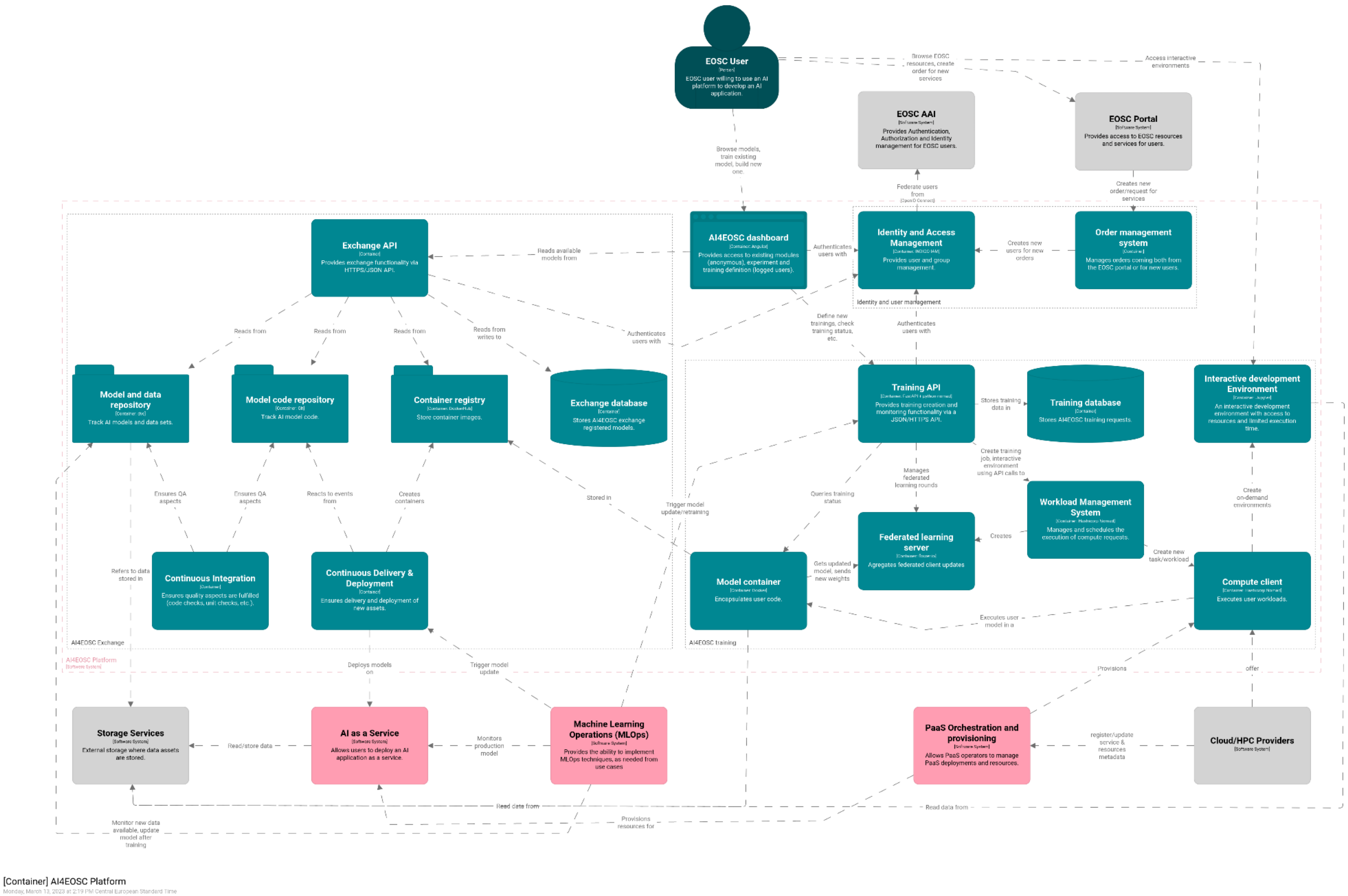

Figure 7: AI4EOSC Platform C4 Container view [25].

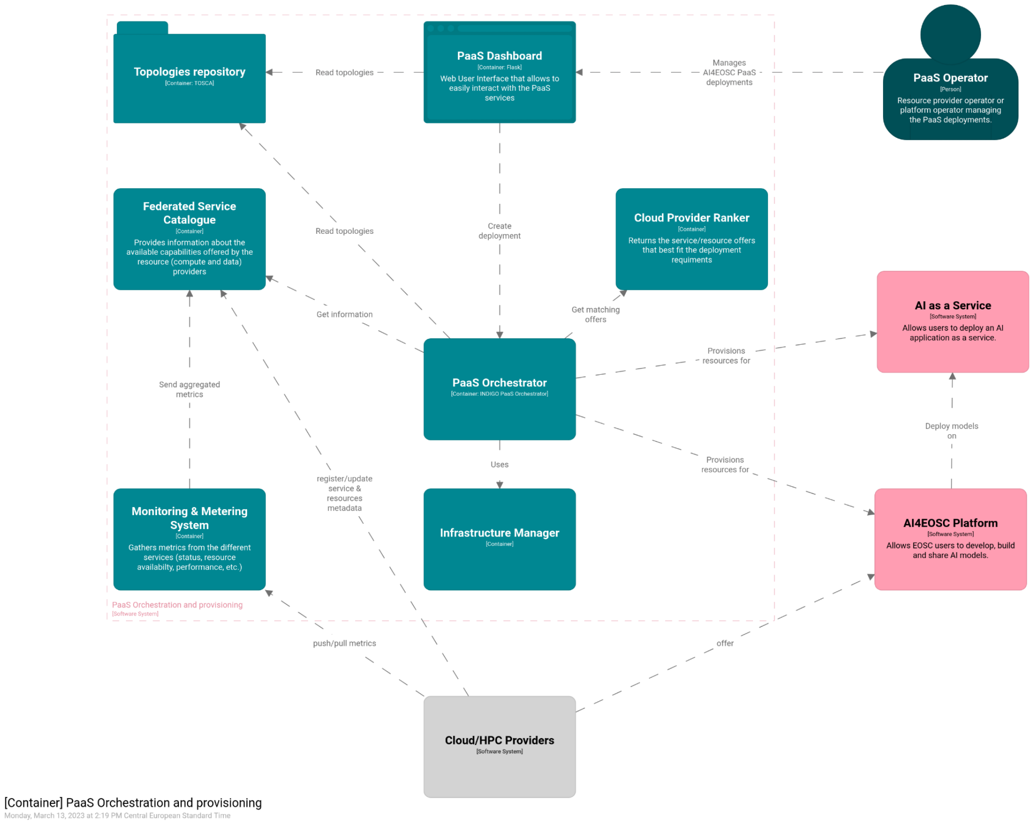

Figure 8: AI4EOSC PaaS Orchestration and provisioning C4 Container view [26].

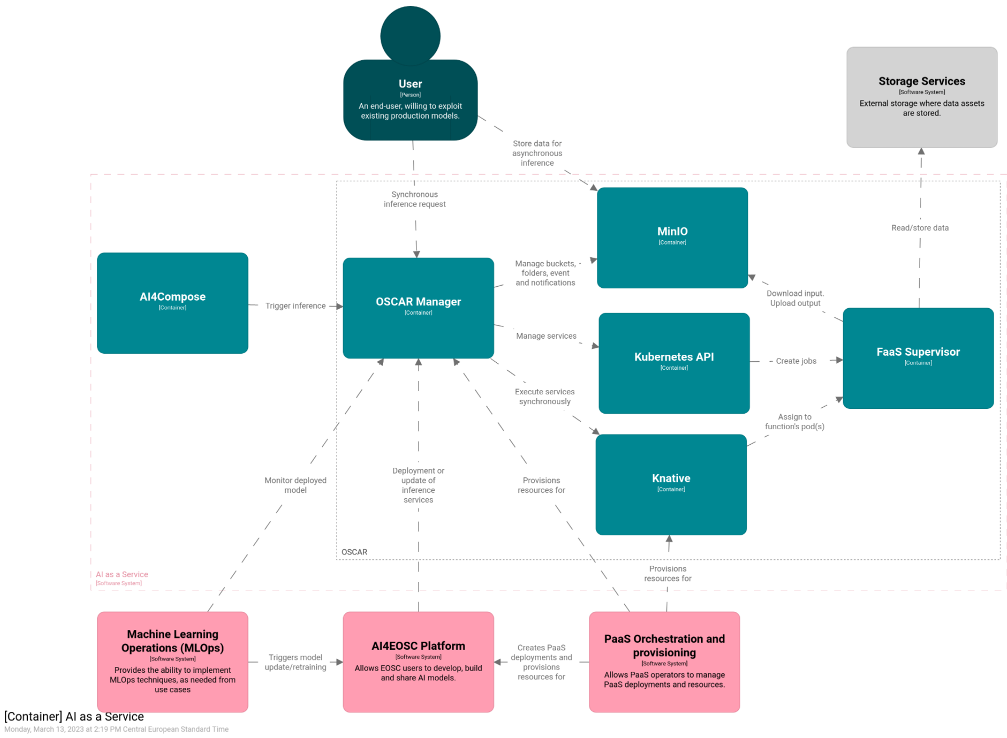

Figure 9: AI4EOSC DEEP as a Service C4 Container view [27].

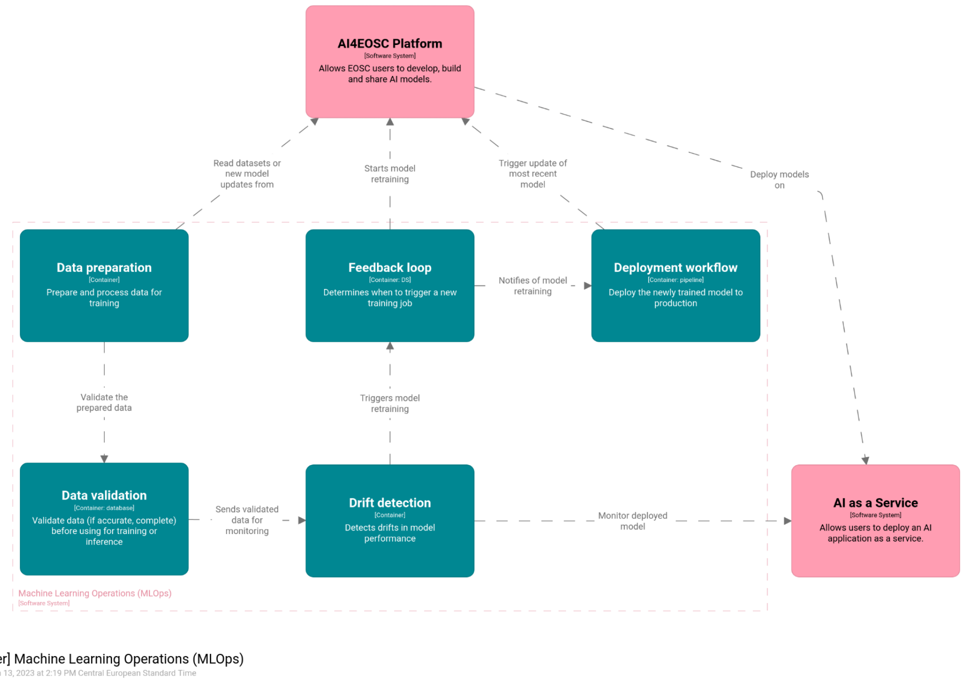

Figure 10: AI4EOSC MLOps C4 Container view [28].

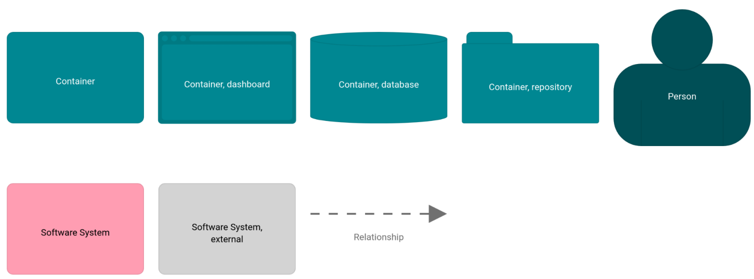

Figure 11: Legend for Figures 7-10.

6. RUNTIME VIEW

The following diagrams include the AI4EOSC runtime view for different user and system interactions. A legend is provided in Figure 16.

- Figure 12 shows the interaction for a user developing a new AI application.
- Figure 13 shows a user retraining an application.
- Figure 14 shows the deployment of a user application as a service.
- Figure 15 shows a federated learning scenario.

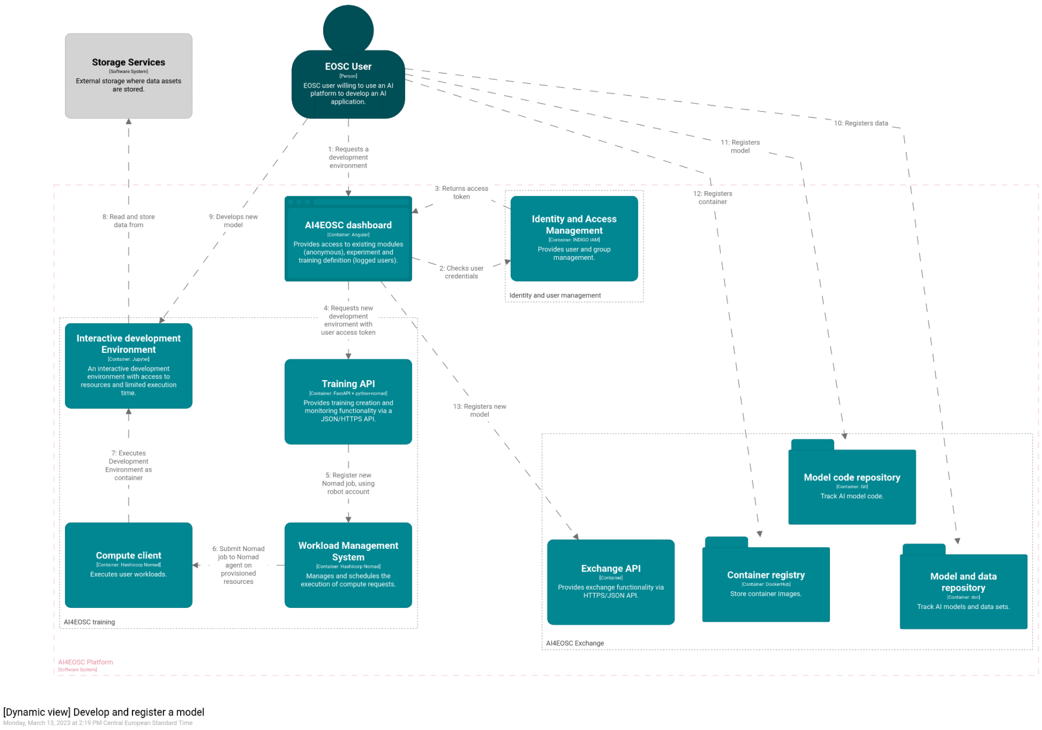

Figure 12: A user developing an application, interacting with the AI4EOSC platform [29].

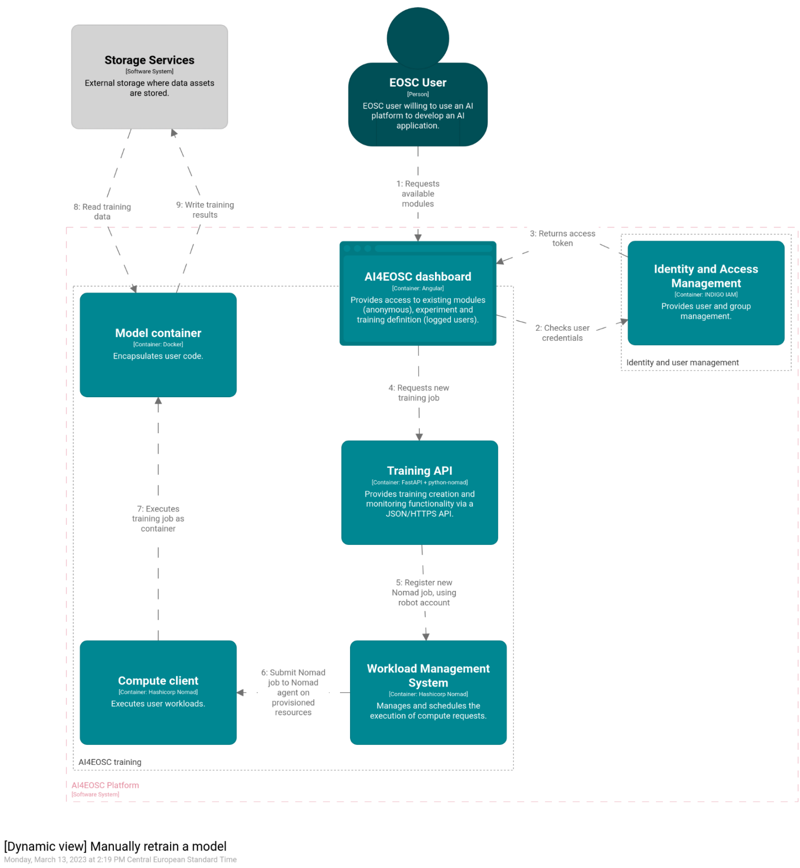

Figure 13: A user retraining an application, interacting with the AI4EOSC platform [30].

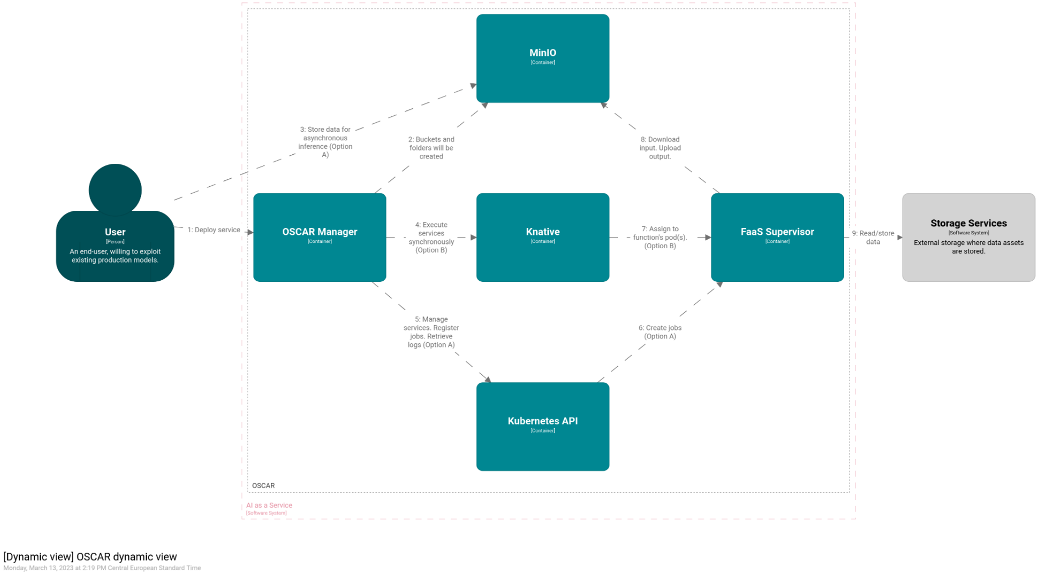

Figure 14: A user interacting with the OSCAR system to deploy a service [31].

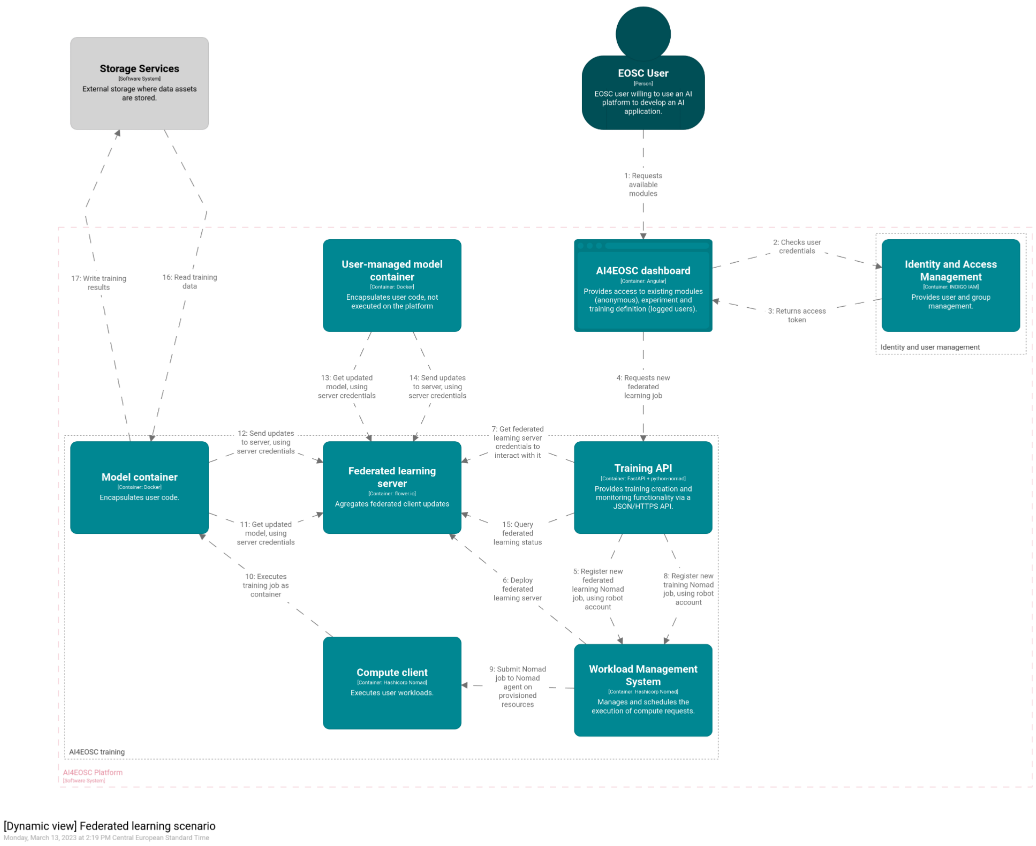

Figure 15: Federated learning scenario [32].

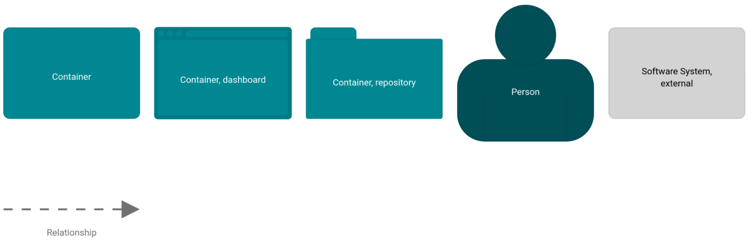

Figure 16: Legend for Figures 12-15.

7. Deployment view

Figure 17 shows an initial deployment view of the AI4EOSC main C4 Containers. Note that some services are duplicated (like the AI4 Control pane) so that redundancy can be achieved, but this is not mandatory.

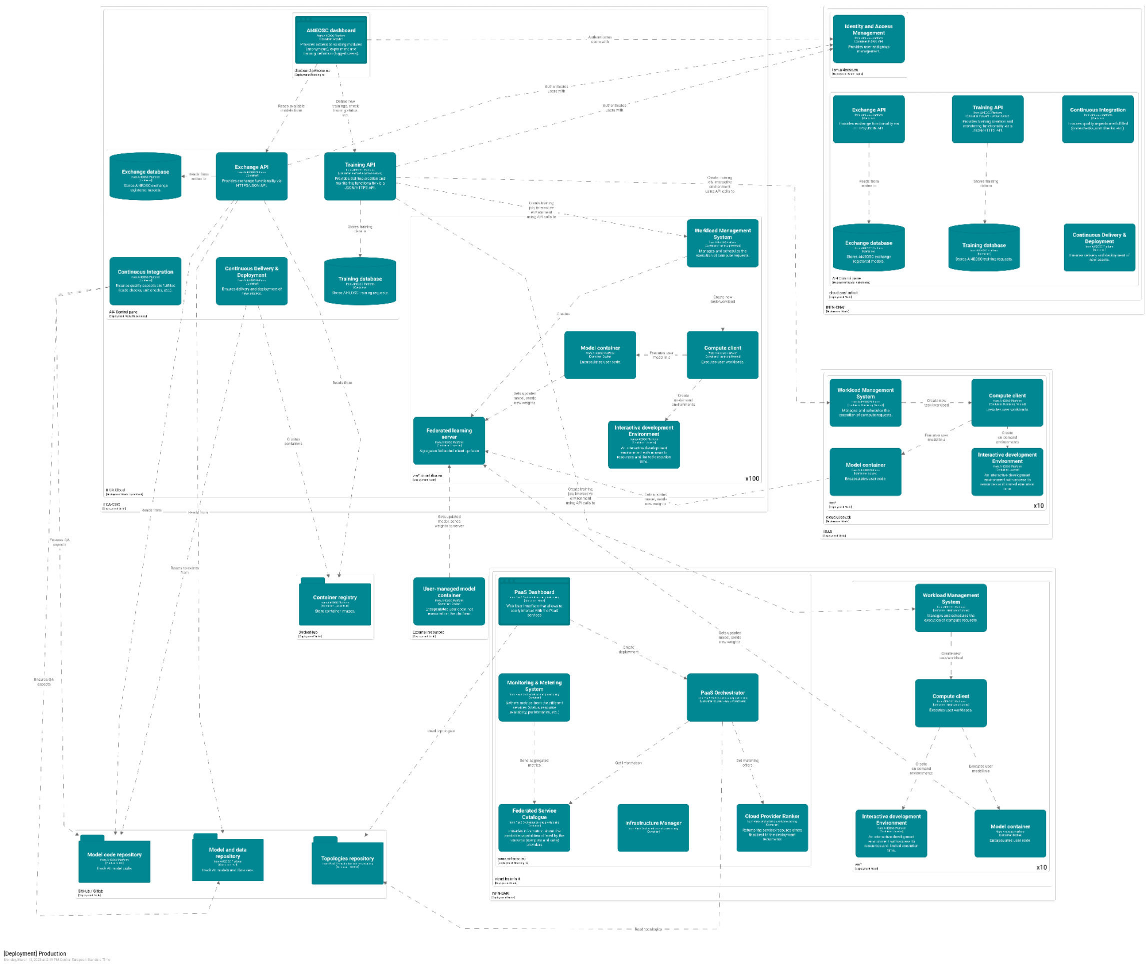

Figure 17: Deployment view of the AI4EOSC platform [33].

8. Architecture decisions

Important architecture decisions in the AI4EOSC project are recorded following the Architectural Decision Records (ADR) and are stored alongside the architecture repository. A summary of the most relevant decisions is included in what follows.

ID	Title	Context	Decision	Link
0002	Remove Alien4Cloud	A4C adoption limited	A4C has been removed from the architecture	https://github.com/AI4EOSC/ai4-architecture/blob/main/dsl/decisions/0002-remove-a4c-from-ai4eosc-architecture.md
0003	Decouple orchestrator from training system	PaaS orchestrator generic API used to submit ML/AI jobs	Provide specific training API, tailored to platform ends	https://github.com/AI4EOSC/ai4-architecture/blob/main/dsl/decisions/0003-decouple-orchestrator-from-training-system.md
0004	Merge exchange and dashboard into a single component	Two different endpoints, offering similar information	Homogenize into a single dashboard, anonymous views (marketplace) and authenticated views (training dashboard)	https://github.com/AI4EOSC/ai4-architecture/blob/main/dsl/decisions/0005-merge-exchange-and-dashboard-into-single-component.md
0006	Use OSCAR for the DEEP as a Service system	The current setup of the DEEP as a Service system is based on Apache Mesos and OpenWhisk. Mesos is a deprecated component.	OSCAR, a component managed by AI4EOSC consortium partners (i.e. UPV) will be used to provide a serverless approach for inference services	https://github.com/AI4EOSC/ai4-architecture/blob/main/dsl/decisions/0006-use-oscar-for-the-deep-as-a-service-system.md
0007	Replace legacy, unmaintained PaaS Components	Scheduling capabilities based on components that are not maintained anymore	Replace unmaintained **internal** orchestrator components with new technologies	https://github.com/AI4EOSC/ai4-architecture/blob/main/dsl/decisions/0007-replace-legacy-paas-components.md
0008	Utilize Nomad for the provisioning of the compute resources	Apache Mesos being used for platform layer is now deprecated	Substitute with Hashicorp Nomad and related tools (i.e. Consul Connect, etc.)	https://github.com/AI4EOSC/ai4-architecture/blob/main/dsl/decisions/0008-use-nomad-as-workload-management-system.md

9. Risks and technical debt

Identified risks, together with its occurrence likelihood, severity and the corresponding mitigation measures are monitored by WP1.

10. Abbreviations

AI	Artificial Intelligence
AIoD	AI-on-Demand Platform
API	Application Programming Interface
DEEP-HDC	DEEP-Hybrid-DataCloud
EOSC	European Open Science Cloud
DL	Deep Learning
CD	Continuous Deliver
CI	Continuous Integration
IaaS	Infrastructure as a Service
ML	Machine Learning
MLOps	Machine Learning Operations
PaaS	Platform as a Service
SaaS	Software as a Service

References

2. https://doi.org/10.5281/zenodo.7635452
3. https://doi.org/10.5281/zenodo.7649073
4. https://doi.org/10.5281/zenodo.1164244
5. https://digital.csic.es/handle/10261/221872
6. https://doi.org/10.5281/zenodo.7728886
7. https://confluence.ifca.es/display/AI4/AI4EOSC+Home
8. https://confluence.ifca.es/display/AI4/High-level+architecture+specification
9. https://arc42.org/
10. https://c4model.com/
11. https://github.com/structurizr
12. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1
13. https://github.com/AI4EOSC/ai4-architecture
14. https://doi.org/10.5281/zenodo.7635452
15. https://docs.google.com/spreadsheets/d/1wFe3YKH3hZTegTwaQR8rfFh-6dIFy6-G/edit#gid=224085165
16. https://deep-hybrid-datacloud.eu/
17. https://doi.org/10.1109/ACCESS.2020.2964386
18. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#system_view
19. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#ai4eosc_view
20. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#orchestration_view
21. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#deepaas_view
22. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#mlops_view
23. https://www.nomadproject.io/
24. https://www.consul.io/
25. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#ai4eosc_container_view
26. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#orchestration_container_view
27. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#deepaas_container_view
28. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#mlops_container_view
29. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#develop_view
30. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#manual_retrain_view
31. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#oscar_dynamic
32. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#federated_train_view
33. https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/diagrams#production_deployment