



DOME

(Distributed Open Marketplace for Europe)

D3.2 DOME Technical Infrastructure (v1)

Project full title

A Distributed Open Marketplace for Europe Cloud and Edge Services

Contract No.

101084071

Strategic Objective

DIGITAL-2021-CLOUD-AI-01-DS-MARKETPLACE-CLOUD

Project Document Number

DOME-D3.2

Project Document Date

29.12.2023

Deliverable Type and Security

OTHER, PUBLIC

Main editor

ENG

Contributors

DEMX, DigiteITS, EDGR, FF, FICODES, IN2, IONOS



Log Table

Version	Date	Change	Author/partner
v0.1	14/11/2023	First version of the ToC	Paolo Fabriani, Alfonso Pietropaolo (ENG)
v0.2	07/12/2023	Consolidated ToC	Paolo Fabriani, Alfonso Pietropaolo (ENG), Stefan Wiedemann (FF), Panagiotis Bouklis (EDGR)
v0.3	13/12/2023	Initial contributions to sections 1 Introduction, 2 Marketplace Overview, 3 Integration and Release Process and 4 First Major Release.	Paolo Fabriani (ENG), Ivan Ligotino (DEMX), Stefan Wiedemann (FF), Souvik Sengupta (IONOS)
v0.4	18/12/2023	Further contributions to section 3 on Jira and Agile approach; section 4 on Release Planning, live environment and release content.	Paolo Fabriani, Alfonso Pietropaolo (ENG)
v0.5	20/12/2023	Added the Executive Summary and Conclusions. Further contributions to section 2 Marketplace Overview and section 4.3 Release Content. Version ready for internal review.	Paolo Fabriani (ENG), Alejandro Nieto (DigitelITS), Jesus Ruiz (FF), Francisco de la Vega (FICODES), Oriol Canades (IN2)
v0.6	21/12/2023	Internal peer review	Panagiotis Bouklis (EDGR), Dennis Wendland (FF)
v0.7	22/12/2023	Minor updates after peer review	Paolo Fabriani (ENG), Dennis Wendland (FF), Francisco de la Vega (FICODES)
VFinal	29/12/2023	Final check and submission	Laura Pucci (ENG)

Table of Contents

1 Introduction	7
1.1 Executive Summary.....	7
1.2 Intended audience	8
1.3 Structure of the document	9
1.4 Related documents and resources	9
2 DOME Marketplace Overview	10
2.1 DOME high-level architecture	11
2.1.1 The Trust and IAM Framework.....	13
2.1.2 The Decentralised Persistence Layer.....	14
2.1.3 The DOME central marketplace	14
2.1.4 Value-added DOME service.....	14
3 Integration and release process	16
3.1 Version control and repositories	16
3.2 GitOps	17
3.3 Deployment Strategy	18
3.4 Integration pipeline overview	19
3.5 Support to release planning and monitoring.....	20
4 DOME Marketplace releases	23
4.1 Release Plan	23
4.2 First Major Release	24
4.3 Release Content.....	26
4.3.1 Credentials Config Service.....	28
4.3.2 Keycloak VC Provider	28
4.3.3 Kong PEP Plugin.....	29
4.3.4 DSBA PDP.....	30
4.3.5 Trusted Issuers List.....	31
4.3.6 VCVerifier.....	32
4.3.7 TMForum API.....	32
4.3.8 Scorpio Context Broker	34
4.3.9 Broker Adapter	34
4.3.10 Blockchain Connector	35



4.3.11 DLT Adapter.....	36
4.3.12 Business Ecosystem Logic Proxy.....	37
4.3.13 Business Ecosystem Charging Backend	38
5 Conclusions	40

List of figures

[Fig. 1: List of planned DOME services clustered according to...](#)

[Fig. 2: DOME High-level vision of DOME architecture,...](#)

[Fig. 3: DOME system view](#)

[Fig. 4: Exemplified schema of the “App of Apps...”](#)

[Fig. 5: Conceptual diagram for integrating DOME ecosystem’s components](#)

[Fig. 6: Releases overview in Jira](#)

[Fig. 7: Kanban boards overview](#)

[Fig. 8: Features included in the first DOME release Musala](#)

[Fig. 9: Live instance of the DOME central marketplace](#)



Acronyms

Acronym	Definition
BAE	Business API Ecosystem
CI/CD	Continuous Integration / Continuous Delivery
IaC	Infrastructure as Code
IAM	Identity and Access Management
JWT	JSON Web Token
NGSI	Next Generation Service Interfaces
NGSI-LD	Next Generation Service Interfaces - Linked Data
ODRL	Open Digital Rights Language
OIDC4VCI	OpenID Connect for Verifiable Credentials Issuance
OID4VP	OpenID Connect for Verifiable Presentations
PDP	Policy Decision Point
PEP	Policy Enforcement Point
VC	Verifiable Credential
VP	Verifiable Presentation

1 Introduction

Cloud computing is identified as a central piece of Europe's digital future, giving European businesses and public organisations the data processing technology required to support their digital transformation. The European Commission thereby stepped up its efforts to support cloud uptake in Europe as part of its strategy, notably with the pledge to facilitate "the set-up of a cloud services marketplace for EU users from the private and public sector". DOME will materialise the envisioned online marketplace, providing the means for accessing trusted services, notably cloud and edge services, building blocks deployed under the Common Services Platform and more generally any software and data processing services developed under EU programmes such as the Digital Europe Programme, Horizon 2020 or Horizon Europe Relying on Gaia-X concepts and open standards, DOME will serve as the final piece in the technical architecture that the Digital Europe Program is constructing to accelerate the growth and adoption of trusted Cloud and Edge services in Europe. It will serve as a central hub, fostering trustworthy connections between customers and service providers. DOME will take the form of a federated collection of marketplaces connected to a shared digital catalogue of cloud and edge services. Each of the federated marketplaces will be independent or connected to the offering of a given cloud providers which, in turn, can be classified as cloud IaaS providers or cloud platform providers (each of which provide a platform targeted to solve the integration of vertical data/application services from a given vertical domain, like smart cities or smart farming, or the integration of certain type of data/application services, e.g., AI services). DOME relies on the adoption of common open standards for the description of cloud and edge services and service offerings, as well as their access through a shared catalogue.

1.1 Executive Summary

DOME has been conceived as a federated collection of independent providers and marketplaces connected to a shared digital catalogue of cloud and edge services. Besides these marketplaces, DOME includes all the functionalities of a marketplace itself, including a marketplace portal through which cloud/edge service providers may publish their offerings and end customers can procure their products. The set of features planned to be included in the DOME marketplace are the result of a deep analysis of stakeholder requirements¹, a number of possible business models² and the conceptualisation study³ promoted by the EC.

In order to materialise such envisioned federation and central marketplace, the overall design of the DOME environment focuses on three major subsystems: a) the Trust and IAM framework enabling trusted operation without requiring a central entity intermediating in all interactions; b) a Decentralised Persistence Layer providing the necessary persistence and storage capabilities for marketplace catalogues and transactional data and c) the DOME central marketplace as a reference implementation of a federated marketplace with value-

¹ Project deliverable D2.1 "DOME requirements (v1)"

² Project deliverable D5.1 "DOME Business Model (First version)"

³ <https://digital-strategy.ec.europa.eu/en/library/building-european-cloud-marketplace-conceptualisation-study>



added features enhancing the quality and trust of the procurement process, and operated by DOME with a provider-neutral approach.

With the goal to achieve efficient and reliable software development, the project adopted a robust approach to integration and release, centred around Continuous Integration/Continuous Deployment (CI/CD) principles. For such a process to be effective, a number of supporting tools have been selected and adopted for development and release, but also for planning activities and monitoring them. This strategy accelerates development cycles, enhances collaboration, and delivers high-quality software to our users. Although shortly introduced here, the details and operational guidelines for such a process are further elaborated in Task T4.1 and will be documented at a later stage in deliverable D4.3.

Following the established guidelines and overall platform design, development and integration activity started earlier in the first project year. An initial release plan was drafted, spanning from M10 to M18 with almost monthly releases, with the goal to achieve an initial production-level deployment in the middle of the second project year to gather feedback from a wider user base than the consortium one.

This document accompanies the release of the first version of the DOME marketplace core technology. It consists of 13 components, integrated or developed within WP3, providing a preliminary realisation of the DOME reference architecture. It describes the main artefacts released and their functionalities, providing links to download and install instructions, licence and support contacts.

According to the project schedule, this is the first of three major releases due at the end of each project year, with a corresponding deliverable and associated project milestone:

- First release due at M12, documented in D3.2 (this document)
- Second release due at M24, documented in D3.5
- Third (final) Release due at M36, documented in D3.8

A live instance of the current release, available for showcasing purposes as well as to support providers and marketplace owners to get familiar with the environment and the technology. It is available at <https://dome-marketplace.org>

1.2 Intended audience

This document is meant to provide valuable information, guidance and references to:

- Owner of the DOME marketplace (a.k.a. DOME Operator) to gather all required software artefacts and documentation to initiate the deployment of the DOME software platform.
- DOME project partners, in particular federated marketplaces and providers, willing to deploy and run local instances of (part of) the DOME platform for development and testing purposes.
- Potential federated marketplaces and providers outside the consortium, wishing to gain knowledge on the open-source technical foundation of the DOME platform.

1.3 Structure of the document

This document is structured in three chapters:

- Chapter 2 DOME Marketplace Overview - recalls the main architecture of the DOME platform, including the functional role of each composing module and major interactions among them.
- Chapter 3 Integration and Release Process - provides an overview of the approach adopted for integration and release of the DOME platform, based on CI/CD principles and practices, their planning and monitoring.
- Chapter 4 DOME Marketplace first release - provides detailed information and references for each component introduced in chapter 2 that has been released as of December 2023.
- Chapter 5 Conclusions

1.4 Related documents and resources

Following is a list of valuable links to relevant documents and resources:

- [EU Digital Identity Wallet Architecture and Reference Framework](#)
- [DSBA Technology Convergence: Discussion Document](#)
- [Digital Signature Service - DSS](#)
- [DID ETSI Legal person Semantic Identifier Method Specification \(did:elsi\)](#)

Also, this document is related to other technical deliverables produced within the DOME project:

- Deliverable D1.2 “Technical Management Plan”
- Deliverable D2.1 “DOME requirements (v1)”
- Deliverable D3.1 “DOME Reference Architecture and Specifications (v1)”
- Deliverable D3.3 “DOME APIs (v1)”

2 DOME Marketplace Overview

The DOME environment has been designed (see deliverable D3.1) to take the form of a federated collection of marketplaces connected to a shared digital catalogue of cloud and edge services (which can be further classified as data services, application services and infrastructure services).

Each of the federated marketplaces in DOME will be a marketplace provided by an independent operator or a marketplace connected to the offerings of a given cloud / edge infrastructure service provider (IaaS or platform service providers).

Besides these marketplaces, DOME implements all the functionalities of a marketplace itself, including a marketplace portal through which cloud/edge service providers may register their product offerings and end customers can procure offered products.

DOME relies on the adoption of common open standards for the description of cloud and edge services and service offerings as well as their access through a shared catalogue. Also, DOME builds its technology on top of existing, production-grade technology brought to the project by consortium partners or available as open-source.

In line with the “Conceptualisation Study for Building a European Cloud Marketplace”⁴, the DOME federated marketplace ecosystem provides services for both customers and providers of edge and cloud services, in a trusted and secure business environment.

Guided by requirements elicited, analysed and organised in deliverable “D2.1 Requirements”, such an ecosystem has been characterised through a number of services that will be combined to specific business contexts supporting the journey of both providers and customers along the procurement process, and will be tailored to alternative configurations, offerings and prices/payments models.

The figure below presents a comprehensive listing of actual and planned DOME services, clustered according to their concern and differentiated as system services to be integrated or developed and supporting services to support and promote the marketplace to the community.

⁴ <https://digital-strategy.ec.europa.eu/en/library/building-european-cloud-marketplace-conceptualisation-study>

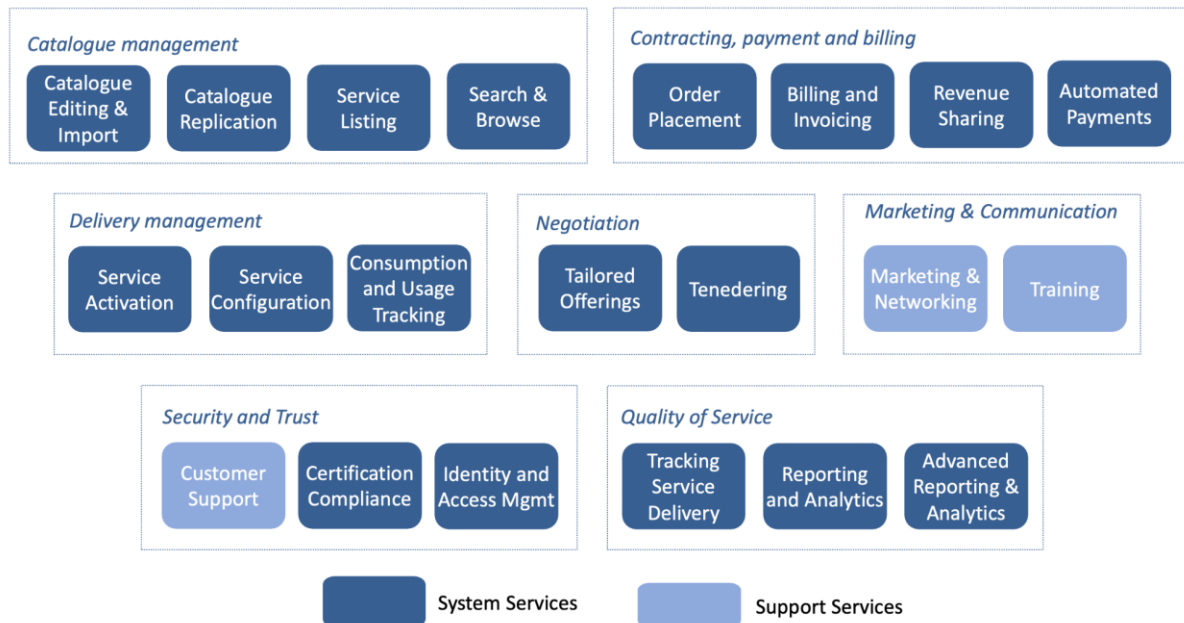


Fig. 1: List of planned DOME services clustered according to their concern

System services have been analysed and designed, to different degrees of detail and maturity (see deliverable D3.1, also including an early design of the Customer Support). Similarly, support services are being designed or - in some cases - are already being operated (see D2.2 and D2.3 regarding Marketing and Networking and future deliverable D4.3 regarding Training)

2.1 DOME high-level architecture

The high-level view of the DOME architecture is recalled below⁵, including major technical building blocks (blue and yellow boxes on the right) along with relevant stakeholders involved in marketplace processes.

⁵ A detailed discussion of all aspects of the DOME architecture has been presented in deliverable D3.1 "DOME Reference Architecture and Specifications (v1)". It is not the goal of this section to report any additional design choice; rather, it is intended to provide a short recap to ease the reading.

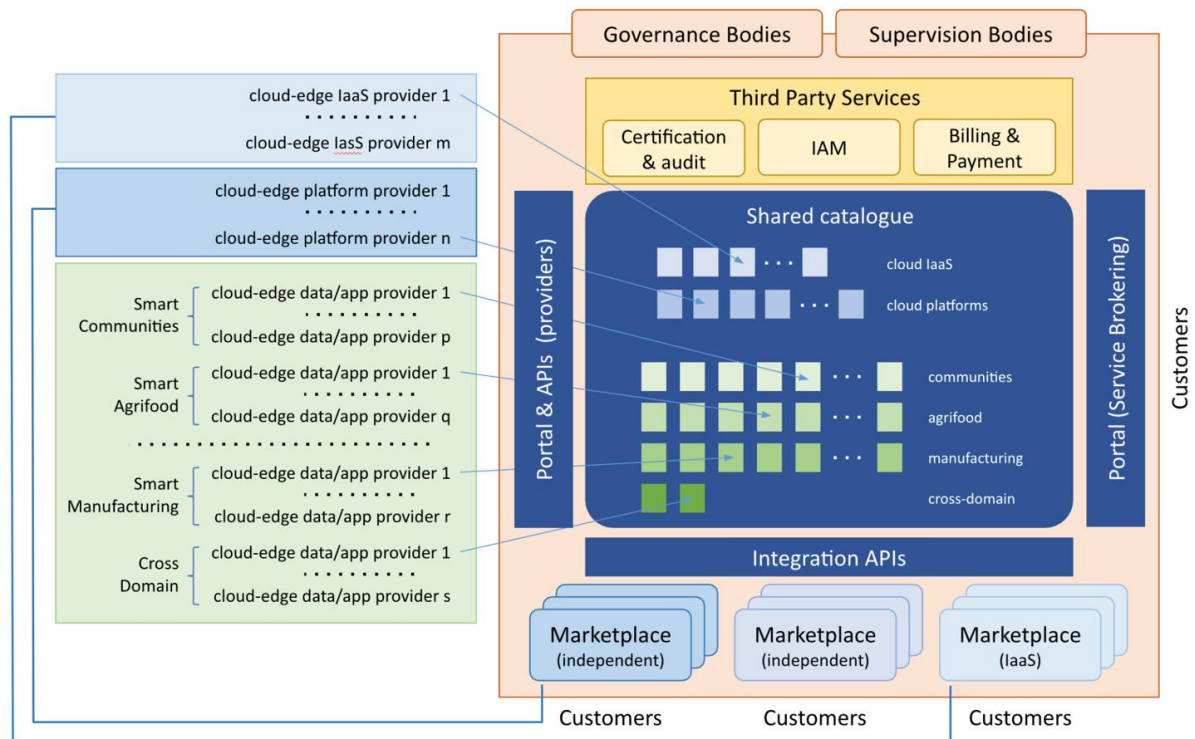


Fig. 2: DOME High-level vision of DOME architecture, operating model and roles

An alternative system view, with a more technical focus, is the following:

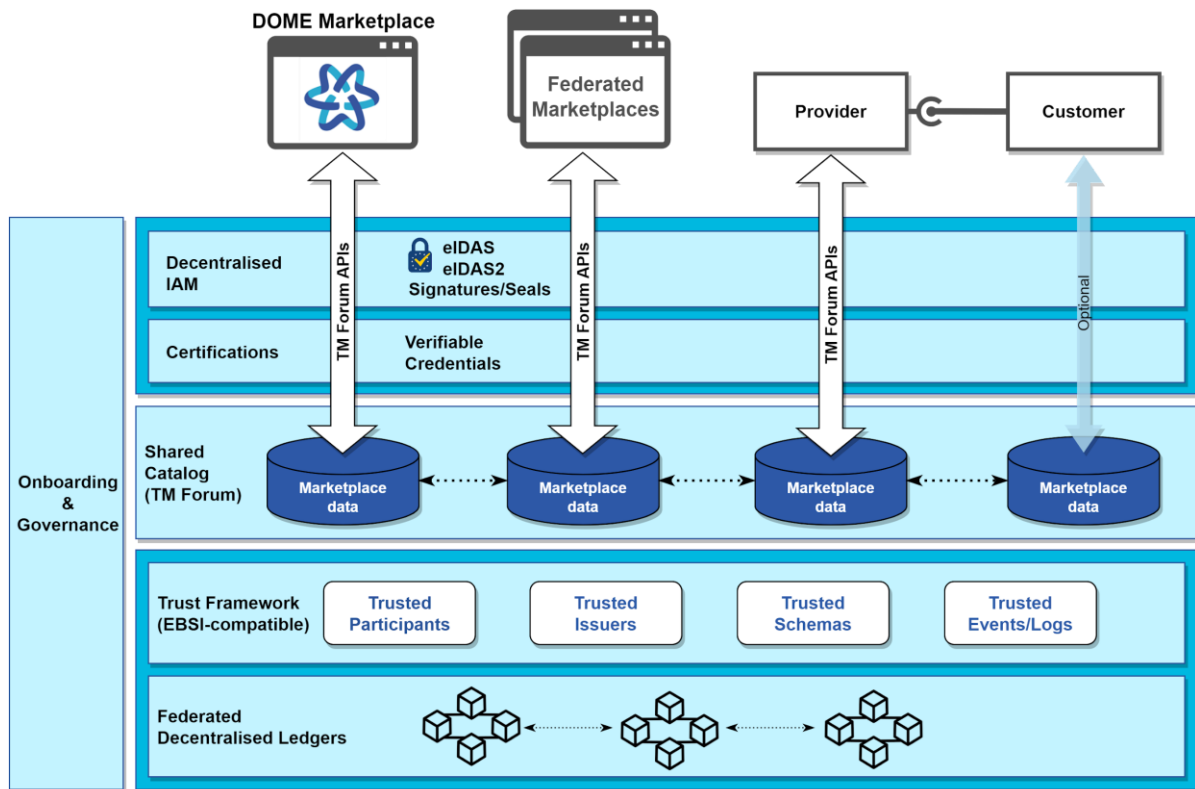


Fig. 3: DOME system view

Stepping from these overall views, three major subsystems have been identified to cope with corresponding enabling and business features, namely:

- The Trust and IAM Framework - to enable the trusted operation with the system without requiring a central entity intermediating in all interactions.
- Decentralised Persistence Layer - to provide the necessary persistence and storage capabilities for marketplace catalogues and transactional data.
- The DOME central marketplace and value-added services - a reference implementation of a federated marketplace, enriched with features enhancing the quality and trust of the procurement process and operated by DOME with a provider-neutral approach.

Within such wide subsystems, a number of components have been identified to be developed and/or integrated to deliver an initial, yet consistent, set of features for customers and providers, as most relevant stakeholders. The full listing of such components is provided in Section 4, later in this document,

2.1.1 The Trust and IAM Framework

The Trust framework ensures that the information published on DOME is trustful. It defines and enforces a set of rules that actors in the DOME ecosystem (data/app service providers, federated marketplace providers, end customers) agree to follow. By following them, all organisations can use their digital identities and characterise services in a consistent and



trustful manner. This lowers the barriers for organisations to complete transactions or share information with other organisations.

The IAM framework, on the other hand, enables actors in the DOME ecosystem to authenticate into DOME services (i.e., the DOME Portal, federated marketplace portals or TM Forum APIs implemented on top of the DOME Persistence Layer) and help to manage proper access to those services based on their profiles. This IAM framework relies on Verifiable Credentials/Verifiable Presentations and leverages the Trust framework to provide an efficient, scalable, and decentralised IAM that participants can use. The Trust and IAM framework implemented in DOME is not only for managing authentication and authorization in the interaction with the DOME services, but it is also available for data/app services which may use it for managing the interactions between them and their users.

2.1.2 The Decentralised Persistence Layer

The goal of the DOME Persistence Layer is to provide the necessary persistence and storage capabilities for the two conceptual components: the Shared Catalog, and the Transaction Ledger.

The Shared Catalog stores descriptions and specifications (Product Specification and Product Offerings) of cloud and edge services in the form of Verifiable Credentials and/or Verifiable Presentations, and other information for the DOME ecosystem consumption.

The Transaction Ledger securely records transactions, including products (the instances of the Product Specifications), Product Orders, and Product Usage.

2.1.3 The DOME central marketplace

On top of the foundation technologies introduced earlier, the DOME project realises an open standard-based, vendor-neutral reference framework to support federation of marketplaces.

Among those marketplaces, a notable instance is represented by the DOME marketplace itself, operated according to provider-neutral, inclusion and equality principles. The DOME central marketplace provides the needed backend and portal services for a) service providers to administer the description of their services and monitor their lifecycle as well as for b) end users to discover services and transition to marketplaces through which they will activate them for actual use, typically on environments setup on the cloud or edge by IaaS or Platform providers.

Features like user subscription, catalogue management, ordering, provisioning, usage tracking, billing and invoicing will be delivered as part of - or strictly related to - the DOME central marketplace.

2.1.4 Value-added DOME service

In addition to the core features delivered by the DOME central marketplace and the mechanisms enabling the federation of marketplaces, DOME will include additional services enhancing the quality and trust of the procurement process. In particular:

- In terms of certification of providers, marketplaces and published offerings, DOME will develop a formal process and methodological framework to verify the **compliance**



against reference standards; such a process will be supported by tools to automatically and continuously monitor the validity of the certificates and that security requirements are being fulfilled.

- A **payment subsystem**, including payment gateway connection handling, payment instrument saving and managing clearing and settlement (in case of the centralised payment gateway account management version), including the handling of bank transfers as well.
- **NLP-enabled search capabilities** with the goal of connecting customers with relevant and coherent services as quickly as possible. The subsystem will feature search algorithms specific to the customer's sector to provide results taking into account the customer's particularities. It also will exploit additional information (e.g., service ratings, click rates) to prioritise/rank search results.
- **Monitoring, reporting and analytics capabilities on business-level KPIs**. It is intended to be used by different DOME user classes, like the service providers and consumers, federated marketplaces operators and the DOME operator. It allows the creation of datasets out of different database data sources, which are either located on the DOME infrastructure or on the federated Marketplaces infrastructure, and their visualisations that can be combined into dashboards, according to the user's preferences.
- **Customer support capabilities** powered by an AI-based chatbot, a troubleticketing system and a knowledge base, with the goal of solving any kind of need the portal user may express in relation to the usage and usability of the DOME portal (thus including the DOME central marketplace but also value-added services mentioned here)

3 Integration and release process

In our pursuit of efficient and reliable software development, we have adopted a robust approach to integration and release, centred around Continuous Integration/Continuous Deployment (CI/CD) principles. This strategy enables us to accelerate development cycles, enhance collaboration, and deliver high-quality software to our users.

3.1 Version control and repositories

We choose GitHub⁶ as our primary version control and repository management platform. GitHub provides a centralised location for collaborative development, version tracking, and code review. Our approach involves the use of two types of repositories:

- **Application Code Repository:** One repository for each component of the DOME architecture, containing the source code of the application, structured into branches for development and production releases. Each team will be responsible for managing their own repositories, ensuring that they adhere to the overall standards and best practices established for DOME.
- **Infrastructure as Code (IaC) Repository:** A single shared repository that holds configuration files for defining and managing our deployment infrastructure. Any formal request of modification to the existing infrastructure will be evaluated and eventually approved by a GitOps maintenance team.

This is a common strategy that provides a clear separation of concerns and facilitates efficient management of both application code and infrastructure configurations.

A Pull Request (PR) is the formal way of suggesting modifications, additions, or deletions to a repository. It is a fundamental concept in version control systems, and it plays a crucial role in collaborative software development, particularly on complex platforms like DOME, as it allows collaborators to review and discuss the proposed changes before they are merged into the main branch.

Using PRs brings several benefits to our development process:

- **Code Quality:** Pull requests provide a structured process for reviewing code, which helps maintain high code quality standards.
- **Collaboration:** Team members can collaborate effectively, share knowledge, and catch errors early in the development process.
- **Visibility:** Pull requests provide visibility into ongoing development efforts, making it easy to track progress and understand the evolution of the codebase.
- **Security:** By integrating Continuous Integration (CI) checks and tests, pull requests contribute to a more secure development process by catching potential issues before they are merged.

⁶ <https://github.com/DOME-Marketplace>

3.2 GitOps

GitOps is an operational framework that takes DevOps best practices used for application development such as version control, collaboration, compliance, and CI/CD, and applies them to infrastructure automation. The core idea behind GitOps is to declare the desired state of the system, store it in a version control system, and use automation to bring the system into the desired state.

Key components and principles of GitOps include:

- **Declarative configuration:** The desired state of the entire system, including infrastructure and application configurations, is stored in Git repositories. This is typically done using declarative configuration files, such as YAML. This declarative approach is focused on the end goal (desired state) rather than the steps needed to achieve the required result.
- **Version Control:** All the declarative states are stored in the version-controlled system, which acts as the single source of truth. All changes to configurations, infrastructure, and application code are tracked and versioned in Git repositories, enabling users to identify infrastructure changes over time easily.
- **Automation:** Automation tools, often integrated with CI/CD pipelines, continuously monitor the Git repositories for changes. Upon detecting a change, these tools automatically apply the changes to the target environment, bringing it into the desired state.

The automation tool selected for DOME is ArgoCD. ArgoCD is an open-source declarative continuous delivery tool that helps automate the deployment and management of applications on Kubernetes clusters. It uses GitHub as the source of truth for defining and managing the desired state of the infrastructure.

ArgoCD doesn't enforce a specific way of structuring deployments or managing configurations. Instead, it supports various patterns and approaches, allowing users to choose the one that best fits their needs. In DOME we will adopt the “App of Apps” pattern. This pattern excels in scenarios where there is a need for centralized configuration management. This is particularly beneficial when dealing with complex platforms consisting of multiple applications and microservices by providing a high-level view of the entire system's desired state and allowing for efficient management of interdependent applications. Fig. 4 depicts an exemplified schema of the “App of Apps” pattern:

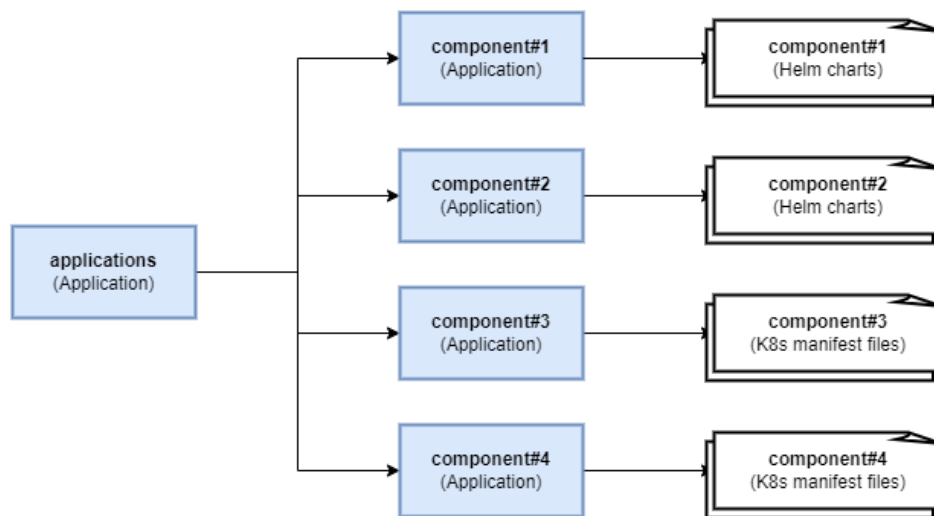


Fig. 4: Exemplified schema of the “App of Apps” pattern for DOME ecosystem’s components

ArgoCD seamlessly supports both Kubernetes manifest files and Helm charts, offering a declarative and GitOps-driven approach for continuous delivery to Kubernetes clusters. In the case of Helm charts, ArgoCD integrates with Helm repositories, allowing you to define applications pointing to Helm charts stored in GitHub or Helm repositories. ArgoCD supports versioning, enabling precise control over deployed chart versions. It also facilitates values overrides for Helm charts, enabling customization.

Helm charts are a key component in Kubernetes-based application deployment and management. Essentially, a Helm chart is a package of pre-configured Kubernetes resources, allowing for the streamlined installation, upgrading, and management of complex applications. With Helm charts, developers can define, version, and share their application's architecture and dependencies, facilitating consistent and reproducible deployments in Kubernetes clusters. Helm simplifies the packaging and deployment of applications, making it an instrumental tool for efficiently managing Kubernetes workloads.

3.3 Deployment Strategy

DOME deployment strategy involves the use of **two (2)** distinct environments:

- **Development/Integration environment:** Serves as a controlled testing ground for component integration within the DOME ecosystem
- **Production environment:** Where the latest version of the DOME ecosystem is pushed into live to the intended users.

In particular, the production environment will implement a blue-green approach. Blue-green deployment is an application release model that involves having two separate environments, one "blue" (currently live) and the other "green" (new version). In this approach, the live environment continues to handle production traffic while the new version is deployed and tested in the green environment. Once the green environment is deemed stable and ready for production, a switch is made to redirect traffic from the blue to the green environment.



The key benefit of a blue-green deployment is the ability to achieve near-zero downtime during the deployment process. If issues are identified in the green environment, the switch can be quickly reverted to the blue environment, ensuring a seamless rollback. This strategy also provides a controlled and risk-mitigated approach to introducing new versions, making it a popular choice for minimizing service disruptions and ensuring a smooth user experience during updates.

3.4 Integration pipeline overview

The integration pipeline serves as the backbone of the CI/CD workflow, connecting code repositories with testing environments and deployment mechanisms. Its primary objective is to enhance collaboration among development and operations teams by automating the integration and validation of code changes, thereby reducing the risk of errors and ensuring the consistent delivery of reliable software. Fig. 5 below depicts the overall approach to the integration for the DOME platform:

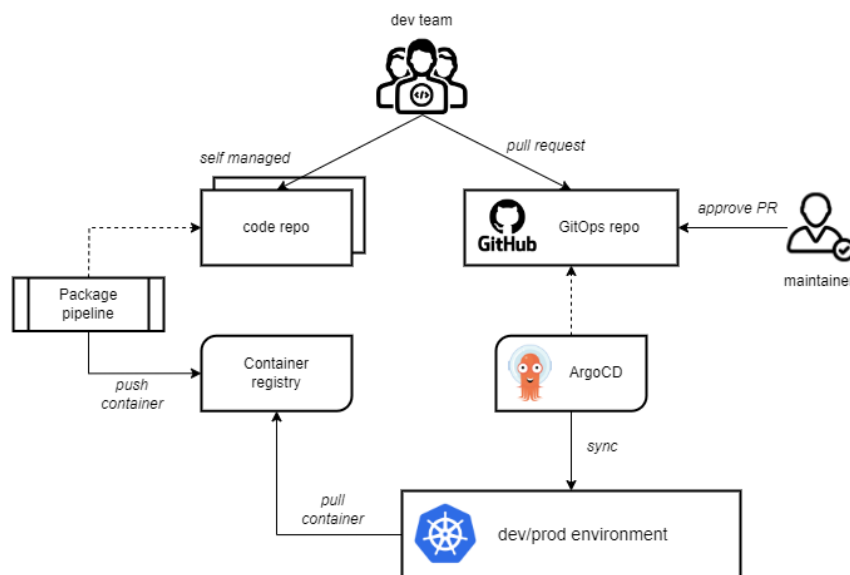


Fig. 5: Conceptual diagram for integrating DOME ecosystem's components

As previously mentioned, each team is responsible for the maintenance of their component, including source code and CI/CD pipelines. The process typically involves some or all of the following elements:

- **Source code management:** Development team manages its source code in a version control system. They create and maintain repositories for their components, ensuring that the codebase is organized, well-documented, and follows best practices.
- **Continuous Integration (CI) pipeline:** Teams should set up CI pipelines responsible for automating the build, test, and package phases of the development lifecycle. CI tools like Jenkins, GitLab CI, or GitHub Actions can be employed for this purpose.
- **Testing:** Automated tests, including unit tests, integration tests, and any other relevant testing suites, are integrated into the CI pipeline. The goal is to ensure that the code changes introduced by the team do not introduce regressions and meet quality standards.

- **Packaging:** Following successful testing, the CI pipeline should package the application into a Docker container. Teams create a Dockerfile specifying the application's dependencies, runtime environment, and other configurations. The resulting Docker image encapsulates the component for deployment.
- **Container Registry integration:** Teams configure the CI pipeline to push the Docker image to a container registry upon successful completion of the previous phases. Popular container registries like quay.io or Docker Hub are used for storing and versioning Docker images.
- **Deployment and orchestration:** Teams may include deployment configurations, such as Kubernetes manifests or Helm charts, within their repositories.

In the integration approach designed for DOME, the team intending to integrate its component needs to adhere to a structured set of steps:

1. **Fork the repository and create a new branch:** Team members begin by forking the main GitOps repository and creating a new branch for their specific integration work. This facilitates isolated development and changes.
2. **Add component manifest files:** The team incorporates their integration by adding either a Helm chart or plain Kubernetes manifests to a properly named folder under the designated directory. This directory serves as a centralized location for all integrations.
3. **Add Application to the “applications” folder:** The team places their specific ArgoCD application manifest in the “applications” folder.
4. **Create a Pull Request and wait for merge:** Upon completing the changes, the team initiates a pull request from their branch to the main one. This PR serves as a formal request for the integration to be reviewed and merged. Team members must wait for the review process to be completed.
5. **Automated deployment after merge:** Once the pull request is approved and merged into the main branch, the GitOps pipeline automatically triggers the deployment process. This involves synchronizing the desired state of the cluster with the changes introduced in the merged pull request. The deployment is executed based on the Helm chart or manifest configurations added to the repository.

By adopting this GitOps integration approach, DOME ensures a standardized and automated process for adding new components to the ecosystem.

3.5 Support to release planning and monitoring

The DOME project relies on **Jira**⁷ as a central platform for project and release management. DOME adopted a customised hybrid approach integrating **Scrum**⁸ and **Kanban**⁹ principles, fostering a flexible and adaptive structure. Unlike rigid sprint cycles, DOME follows a more adaptive approach with a backlog in support of standard Kanban boards, enhancing the visibility and organisation of DOME issues. This approach empowers developers to seamlessly transition items through various states based on ongoing releases and work

⁷ <https://www.atlassian.com/software/jira>

⁸ <https://www.scrum.org>

⁹ [https://en.wikipedia.org/wiki/Kanban_\(development\)](https://en.wikipedia.org/wiki/Kanban_(development))



progress. Releases are identified and crafted during project management meetings and individual issue progress is closely monitored during weekly project calls. This regular assessment within the Kanban methodology fosters effective communication and enables timely adjustments, ensuring the project stays on track.

Release management is a crucial aspect of our software development and project management strategy. In Jira software, releases are referred to as 'versions,' signifying specific points in time within the project. This framework allows us to schedule and control the delivery of features and improvements systematically (see picture below):



Fig. 6: Releases overview in Jira

All development teams are tasked with populating the Jira backlog and allocating items to specific releases based on project requirements. As teams actively engage with specific tasks, they are required to update the status of items, adhering to the workflow outlined in the Kanban Boards, which encompasses three key stages: "to do," "in progress," and "done":

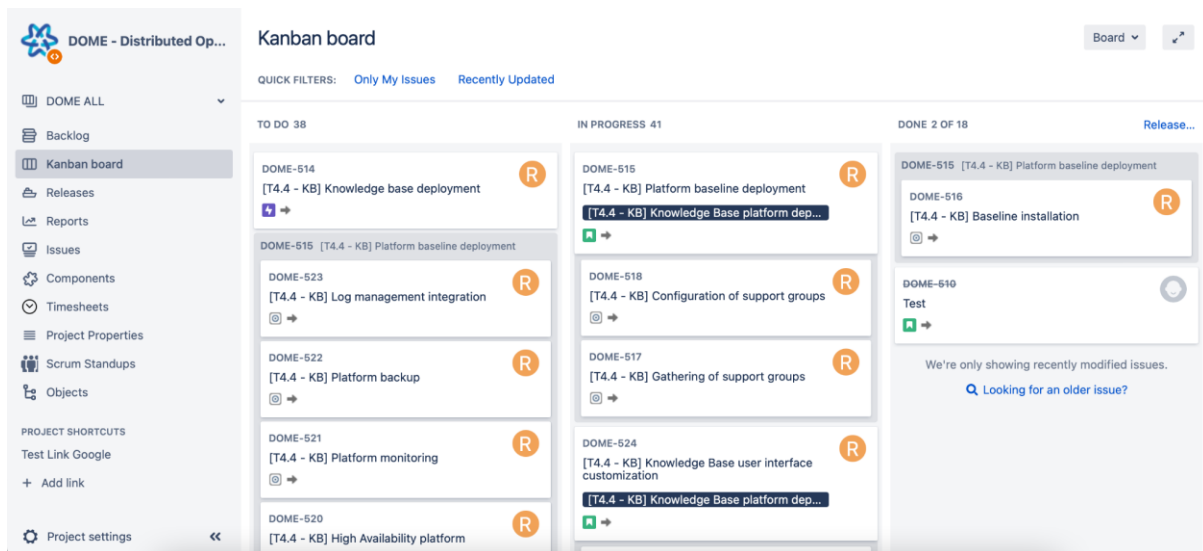


Fig. 7: Kanban boards overview

In this context, the adoption of agile hierarchical elements (such as Epics, Stories, and Sub-Tasks) brings several benefits, enhancing the overall organisation, planning, and execution of tasks within the project:

- **Epics** serve as high-level containers for large bodies of work, encapsulating major functionalities or features. In our scenario, Epics are instrumental in providing a macro-level view of project objectives and milestones. By breaking down the project into Epics, we achieve better organisation and strategic planning.
- **Integration with Components:** Each Epic is associated with specific Jira components, representing categorised modules within the project each associated with specific functionalities, parts of software, or project work packages (e.g., DOME Portal, Payment, TMForum API, and Persistence Layer). This integration ensures that Epics align with the logical breakdown of the project, allowing for focused development efforts within distinct functionalities
- **User Stories**, on the other hand, provide a granular and user-focused description of software features or functionalities. They serve as concise narratives outlining the user, their goal, and the benefit derived from the described feature. In our scenario, Stories enable a detailed and user-centric breakdown of work associated with each Epic.
- **Sub-Tasks** represent specific, granular units of work associated with a User Story. While Stories provide a user-focused description of a software feature, Tasks split those user stories into manageable components, allowing for detailed planning and tracking of development efforts.

In summary, Jira is a pivotal tool for DOME's success. The tailored configuration, integration of the Kanplan methodology (Kanban+Backlog), logical project components, and strategic use of releases contribute to efficient, adaptable, and transparent project goals. This approach reflects our commitment to delivering high-quality software solutions within the project's dynamic environment.



4 DOME Marketplace releases

As introduced in the previous section, the project technical team decided to implement an approach combining Agile with the subsequent delivery of a Minimum Viable PoC (Proof of Concept that, at a certain point, based on maturity of features, will turn into a Minimum Viable Product). The idea is to continually deliver updates of a MV PoC (first phase) and Product (second phase) based on a plan of subsequent milestones.

Within this overall approach, a few further considerations have been done:

- A MV-PoC/P is available on the DOME runtime environment at the end of every milestone, ready for showcasing;
- A number of features will be implemented and incorporated in the product from one milestone to the next;
- Distance between two consecutive milestones will not be the same along the project; the time needed to deliver a milestone depends of the number and complexity of features to be implemented and integrated for one milestone;
- Development for a given milestone may start earlier than finalisation of the previous one, since the goal is to adjust the Work In Progress (WIP) to the capacity of the various development teams.

4.1 Release Plan

Based on the above approach, an initial release plan has been drafted, with associated features and timing and associated content until M18 (i.e. June 2024) and a more rough planning for the rest of the project. The current plan of milestones is presented below:

Milestone “Amaro” (Oct. 2023, M10) delivered basic trading features and initial support for trust framework for users. This release features a single centralised marketplace where service providers can register products and offerings and receive product orders. Customers can navigate through the catalogue of products, select offerings and order products. Authentication of users (both customers and providers) is based on VCs.

Milestone “Corno Grande” (Oct. 2023, M10) allowed providers to describe products as a combination of independently managed services and resources. It also laid-out the foundation to support transparency and ability to audit lifecycle of product/service/resource specifications, offerings and orders, providing proof of associated transactions (relying on blockchain technology); on top of this, basic monitoring of audit records and creation of basic audit reports is feasible.

Milestone “Olympus” (Nov. 2023, M11) consists of a first step towards a federated approach: federated marketplaces (other than the DOME one) can be connected to the ecosystem, each through a DOME access node. By sharing their offerings in the federation, service providers will allow customers to discover and navigate their product offerings at different endpoints, thus enhancing the visibility of their business.

Milestone “Musala” (Dec. 2023, M12) will basically incorporate usability improvements, overall quality enhancement and will constitute the basis for a first live version in a restricted



environment for invited users. It will be the first major release of the DOME marketplace, as documented in this deliverable D.3.2

Milestone “Aneto” (Jan. 2024, M13) will deliver the ability to implement trust at DOME users level based on the defined Participation Agreement framework. In particular, this release will enable onboarding of DOME users (providers and customers) verifying in an automatic manner some credentials and the compliance to existing EU legal regulations or sector regulations EU wishes that providers in DOME have to comply with. Also, initial support for catalogue replication control and for transactional ordering scenarios will be available in this release.

Milestone “Mulhacen” (Mar. 2024, M15) will offer customers trust on the products they acquire rights to use (trust at product-level) by associating them with verifiable characteristics issued by recognised Trusted Issuers that may be verified by customers. The DOME central marketplace will support for provisioning based on federated requests and will exploit usage data generated by federated providers; a keyword-based search and a multi-parameter filtering will allow to fully browse the federated catalogue; also, providers will benefit from initial data visualisation and dashboards to get insights on their business activity.

Milestone “Antelao” (Apr. 2024, M16) will enable configuration of authorisation, based on defined DOME roles and environment variables, as part of a more solid operation. In particular, it will deliver an implementation of PDP functions based on policies expressed in ODRL, considering attributes of TMForum entities on which operations act upon as well as defined DOME roles

Initial production of DOME is planned to be rolled-out in June 2024 (M18), able to support customers and gather valuable feedback from them. Besides accumulated features from previous releases, this version will improve DOME users’ satisfaction and usability of DOME services by enclosing an in-depth revision of existing marketplace user experience. More advanced reporting on business activity and service usage will be available as well as initial support for billing. Search features will be enriched with support for categories and tags. Also, this release will include user support based on an AI-based (natural language generative) chatbot, a knowledge base and a ticketing system.

At the time of writing, the detailed content of releases beyond June 2024 has not been defined yet. As a baseline, three further major versions will be planned up to the end of the project (the one Dec. 2024, will be the second major release of the DOME marketplace and will be documented in D3.5; a further intermediate release will be issued in June 2025; a final one in Dec. 2025 will include all project results and will be documented in D3.8).

Nevertheless, in continuity with the current approach, a more fine-grained release plan will be established in the near future, with more frequent versions delivering value to stakeholders as soon as features are stable and robust enough.

4.2 First Major Release

This section reports the most recent versions of available components of the DOME platform available as of December 2023 (project month M12). This release corresponds to Milestone 3 (codenamed *Musala*) presented in the previous section.

With this Minimum Viable PoC, the DOME platform embeds an initial and consistent set of marketplace features enabling customers and providers to experience basic trading along



their respective journeys (see fig. 8 below). Within the DOME central marketplace (based on the FIWARE Business API Ecosystem, BAE), catalogue management functionalities have been made available, as well as initial order placement and corresponding service activation and delivery to the customer. Secure access to the marketplace is granted through user authentication based on Verifiable Credentials. Furthermore, this release lays out the foundation to support transparency and audit lifecycle of product/service/resource specifications, offerings and orders, providing proof of associated transactions; in fact, *Musala* includes a first implementation of the TMForum APIs on top of a combination of a blockchain and an off-chain local storage.

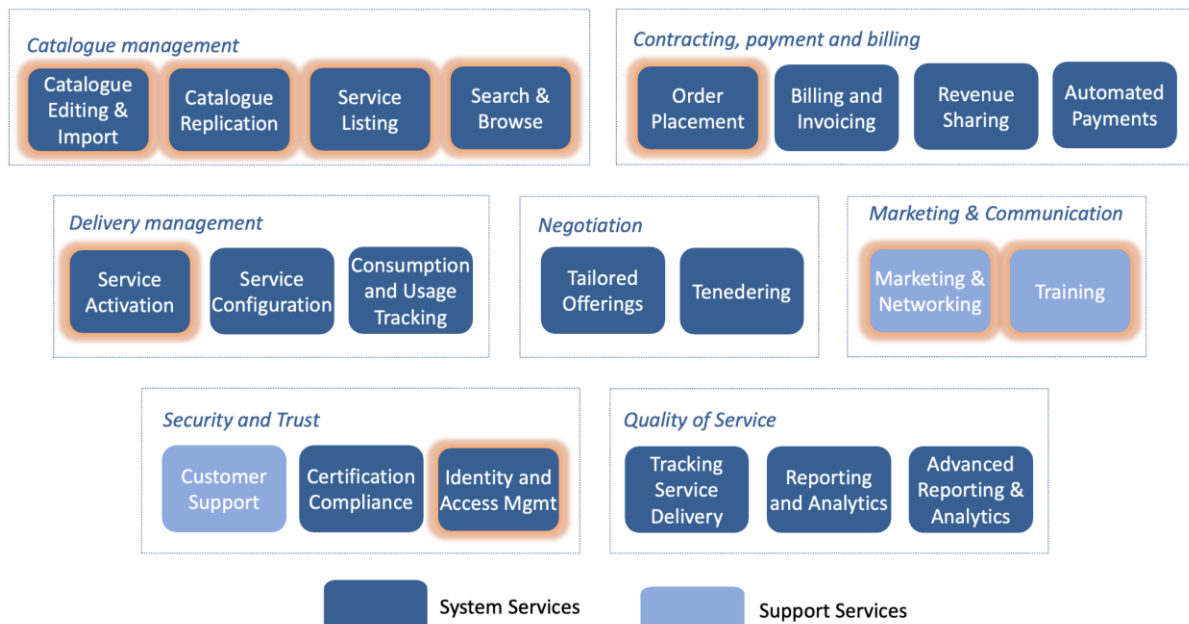


Fig. 8: Features included in the first DOME release *Musala*

Following the deployment strategy introduced in the previous section, a live instance of this release has been made available with a twofold purpose: it allows live showcasing to the interested community; it recently served as the basis for a live demo run with relevant people from EC Cloud and Software Unit (CNECT.E2) and for a following demo at the 3rd General Assembly of the European Alliance for Industrial Data, Edge and Cloud. As a second, internal purpose, this instance also supports service providers and marketplace owners in the consortium to get familiar with the technology in order to integrate their services within the federation. It can be accessed at <https://dome-marketplace.org>

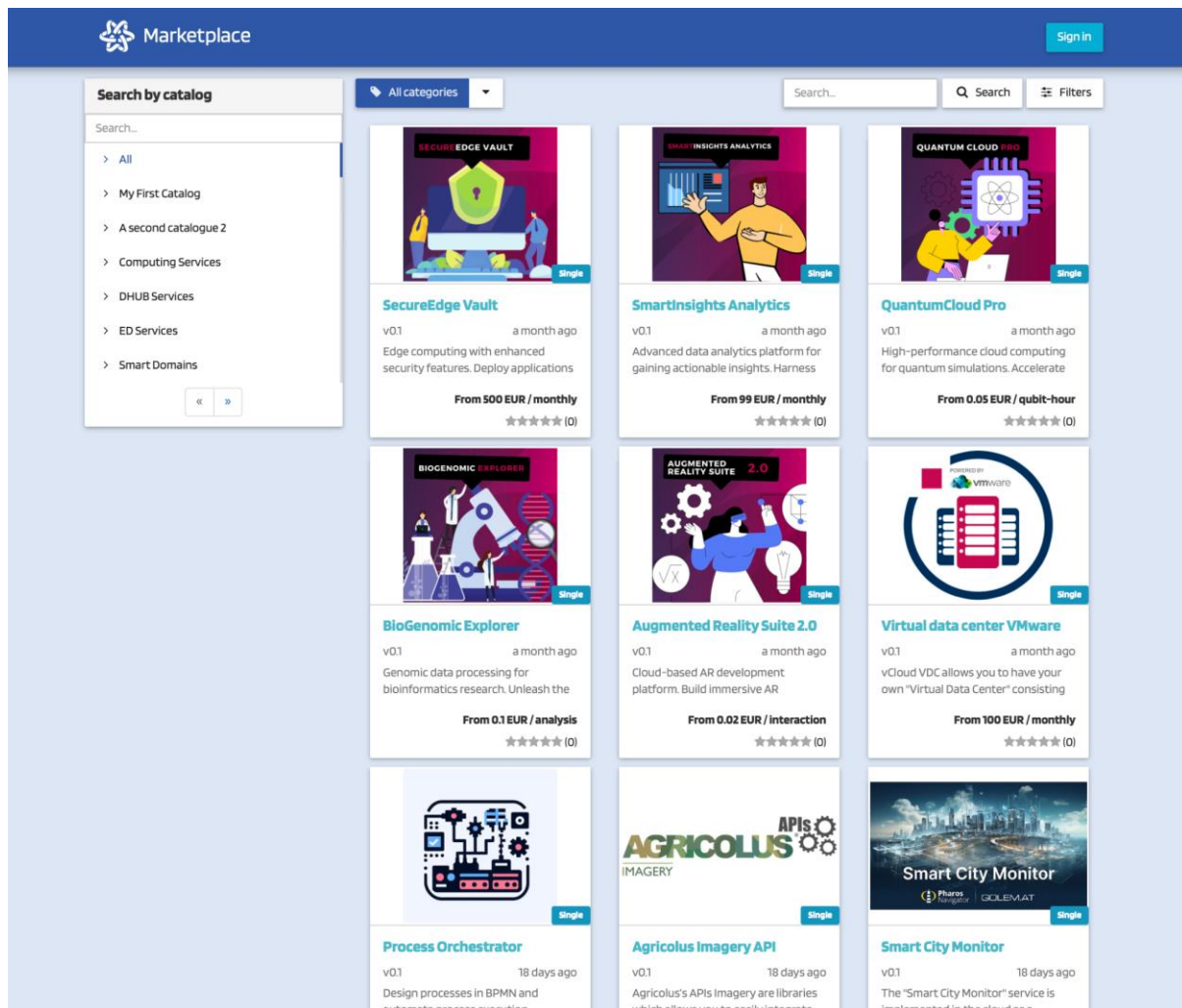


Fig. 9: Live instance of the DOME central marketplace

4.3 Release Content

This section describes the first major release of the DOME platform. In particular it provides useful information and references for the following released modules, grouped according to the three major areas introduced in Section 2.

Trust and IAM Framework

The Trust and IAM Framework is built around different components ensuring trust between the participants (onboarding services and a Trusted Participants Registry). It also provides the needed functionalities to enable authentication based on VCs (Keycloak VC Provider, VCVerifier, Credentials Config Service) and to control access to resources (Kong PEP Plugin, DSBA PDP).

- Credentials Config Service
- Keycloak VC Provider
- Kong PEP Plugin



- DSBA PDP
- Trusted Issuers List
- VCVerifier

Decentralised Persistence Layer

The Decentralised Persistence Layer operates through interconnected Access Node instances on a Blockchain enabling the interconnection of individual marketplaces. Access to local data (managed by Scorpio Context Broker and its associated adapter) is granted through uniform interfaces (TMForum API). Blockchain Connector and DLT Adapter are instead responsible for all interactions with the Blockchain in a transparent way.

- TMForum API
- Scorpio Context Broker
- Broker Adapter
- Blockchain Connector
- DLT Adapter

Marketplace Features

The Marketplace features are being implemented in DOME as part of the FIWARE BAE. This system is made up of two different software components that orchestrate and provide business logic to the TMForum API implementations

- **Business Ecosystem Logic Proxy:** this component is the entry point of the BAE, providing access to the TMForum APIs, validating and orchestrating the different requests, and providing a graphical user interface for accessing the marketplace via web browsers.
- **Business Ecosystem Charging Backend:** this component includes the business logic of the BAE, processing product orders, calculating billing and providing procurement and activation features.

Value-added services

No value-added service has been included in this first DOME release. However, development and integration are ongoing in all mentioned areas (i.e. certification, payment, advanced search, reporting and analytics, customer support). Early features will be available in March 2024 and gradually introduced within next year (i.e. will be reported in the next deliverable D3.5).

In the remainder of this chapter, a specific section for each released component provides summary information and references to external resources, namely:

- *Release Notes:* listing the main features included in the current release;
- *Source Code:* reference to the public repository where the source code of the component can be found;
- *Deployment:* documentation (or reference to it) about where to retrieve artefacts and how to deploy and configure them in the target environment;
- *Further documentation:* references to further documentation (e.g. administration, usage, migration, troubleshooting, etc.);
- *Licence* under which the component has been released;



- *Support and Contacts*: instructions (or reference to them) on how to report bugs, submit requests and post any other code-related issue.

4.3.1 Credentials Config Service

The Credentials Config Service manages and provides information about services and their required credentials to Verifier.

In the DOME Platform, it for example serves the Credentials information for the marketplace instances or the TMForum APIs.

Release Notes

The first release includes CRUD functionality for services and their credentials.

Source code

The source code of the component is available at:

<https://github.com/FIWARE/credentials-config-service>

Deployment

The component is distributed as Container Image available at:

<https://quay.io/repository/fiware/credentials-config-service>

The component can be installed through the Helm Chart at:

<https://github.com/FIWARE/helm-charts/tree/main/charts/credentials-config-service>

Further documentation

Detailed and further documentation (deployment, configuration, usage, troubleshooting, etc.) for component is available at:

<https://github.com/FIWARE/credentials-config-service/blob/main/README.md>

Licence

The component is released under the Apache 2.0 License.

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

<https://github.com/FIWARE/credentials-config-service/issues>

4.3.2 Keycloak VC Provider

The Keycloak VC Provider is an extension to the [Keycloak IAM](#) to support issuance of Verifiable Credentials through the OID4VCI Standard.

Release Notes

The first release includes support for issuance of credentials through OIDC4VCI in the formats:



- ldap_vc
- jwt_vc

Source code

The source code of the component is available at:

<https://github.com/FIWARE/keycloak-vc-issuer>

Deployment

The component is distributed as Container Image available at:

<https://quay.io/repository/fiware/keycloak-vc-issuer>

The component can be installed through Keycloak's standard mechanisms for provider:

<https://www.keycloak.org/server/configuration-provider>

Further documentation

Detailed and further documentation (deployment, configuration, usage, troubleshooting, etc.) for component is available at:

<https://github.com/FIWARE/keycloak-vc-issuer/blob/main/README.md>

Detailed documentation for Keycloak is available at:

<https://www.keycloak.org/>

Licence

The component is released under the Apache 2.0 License.

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

<https://github.com/FIWARE/credentials-config-service/issues>

4.3.3 Kong PEP Plugin

The Kong PEP Plugin is a plugin for the [Kong API Gateway](#).

It enables Kong to act as a Policy Enforcement Point in the DOME Platform, delegating Policy Decisions to the DSBA-PDP.

Release Notes

The first release includes support for the following PDPs:

- DSBA-PDP
- Keyrock

Additionally, a plugin with built-in PDP for NGSI-ISHare Policies is available.

Source code

The source code of the component is available at:

<https://github.com/FIWARE/kong-plugins-fiware/tree/main/kong-pep-plugin>



Deployment

The plugin is bundled in a self-built Kong image, which is based on the default public Kong image. The Kong image is distributed as Container Image available at:

<https://quay.io/repository/fiware/kong>

The component can be installed through Kong standard Helm-Chart:

<https://github.com/Kong/charts>

Further documentation

Detailed and further documentation (deployment, configuration, usage, troubleshooting, etc.) for component is available at:

<https://github.com/FIWARE/kong-plugins-fiware/blob/main/kong-pep-plugin/README.md>

Detailed documentation for Kong is available at:

<https://github.com/Kong/kong/blob/master/README.md>

Licence

The component is released under the MIT License.

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

<https://github.com/FIWARE/kong-plugins-fiware/issues>

4.3.4 DSBA PDP

The DSBA PDP is responsible for evaluating policies about incoming requests. It will provide the PEP with the decisions to enforce.

Release Notes

The first release supports evaluation of iSHARE policies in an i4Trust conformant way. Attribute based decisions can be made on NGSI-LD conformant requests, path-based decision making is supported for all other HTTP-Rest calls.

Source code

The source code of the component is available at:

<https://github.com/FIWARE/dsba-pdp>

Deployment

The component is distributed as Container Image available at:

<https://quay.io/repository/fiware/dsba-pdp>

The component can be installed through the Helm-Chart:

<https://github.com/FIWARE/helm-charts/tree/main/charts/dsba-pdp>

Further documentation



Detailed and further documentation (deployment, configuration, usage, troubleshooting, etc.) for component is available at:

<https://github.com/FIWARE/dsba-pdp/blob/main/README.md>

Licence

The component is released under the Apache 2.0 License.

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

<https://github.com/FIWARE/dsba-pdp/issues>

4.3.5 Trusted Issuers List

The Trusted Issuers List is an implementation of the [EBSI Trusted Issuers Registry API](#). It provides the Verifier information about issuers of Verifiable Credentials and their capabilities in an instance of the DOME Marketplace.

Release Notes

The first release supports version 3 and 4 of the EBSI Trusted Issuers Registry. Additionally, a proprietary management API is provided.

Source code

The source code of the component is available at:

<https://github.com/FIWARE/trusted-issuers-list>

Deployment

The component is distributed as Container Image available at:

<https://quay.io/repository/fiware/trusted-issuers-list>

The component can be installed through the Helm-Chart:

<https://github.com/FIWARE/helm-charts/tree/main/charts/trusted-issuers-list>

Further documentation

Detailed and further documentation (deployment, configuration, usage, troubleshooting, etc.) for component is available at:

<https://github.com/FIWARE/trusted-issuers-list/blob/main/README.md>

Licence

The component is released under the Apache 2.0 License.

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

<https://github.com/FIWARE/trusted-issuers-list/issues>



4.3.6 VCVerifier

The VCVerifier provides the endpoints required for [OID4VP](#) compliant authentication flows.

In the DOME Platform it is required for authentication of users, verifies their Verifiable Credentials and exchanges them with JWT Tokens for usage in down-stream components.

Release Notes

The first release supports the Same-Device and the Cross-Device Flow, as defined by the OID4VCI specification.

Source code

The source code of the component is available at:

<https://github.com/FIWARE/VCVerifier>

Deployment

The component is distributed as Container Image available at:

<https://quay.io/repository/fiware/trusted-issuers-list>

The component can be installed through the Helm-Chart:

<https://github.com/i4Trust/helm-charts/tree/main/charts/vcverifier>

Further documentation

Detailed and further documentation (deployment, configuration, usage, troubleshooting, etc.) for component is available at:

<https://github.com/FIWARE/VCVerifier/blob/main/README.md>

Licence

The component is released under the Apache 2.0 License.

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

<https://github.com/FIWARE/VCVerifier/issues>

4.3.7 TMForum API

Implementation of the TMForum APIs. Serves as the main API for the Marketplace implementations to interact with. Detailed information about the APIs and their usage in DOME can be found in D3.3 DOME APIs.

Release Notes

The first release includes implementations of the following APIs:

- Party Catalog
- Customer Bill Management
- Customer Management



- Product Catalog
- Product Inventory
- Product Ordering Management
- Resource Catalog
- Resource Function Activation
- Resource Inventory
- Service Catalog
- Account Management
- Agreement
- Usage Management

Source code

The source code of the component is available at: <https://github.com/FIWARE/tmforum-api>

Deployment

The component is distributed as Container Images available at:

- <https://quay.io/repository/fiware/tmforum-party-catalog>
- <https://quay.io/repository/fiware/tmforum-product-inventory>
- <https://quay.io/repository/fiware/tmforum-service-catalog>
- <https://quay.io/repository/fiware/tmforum-resource-catalog>
- <https://quay.io/repository/fiware/tmforum-resource-function-activation>
- <https://quay.io/repository/fiware/tmforum-resource-inventory>
- <https://quay.io/repository/fiware/tmforum-customer-management>
- <https://quay.io/repository/fiware/tmforum-customer-bill-management>
- <https://quay.io/repository/fiware/tmforum-product-catalog>
- <https://quay.io/repository/fiware/tmforum-agreement>
- <https://quay.io/repository/fiware/tmforum-account>
- <https://quay.io/repository/fiware/tmforum-usage-management>

The component can be installed through the Helm Chart at:

<https://github.com/FIWARE/helm-charts/tree/main/charts/tm-forum-api>

Further documentation

Detailed and further documentation (deployment, configuration, usage, troubleshooting, etc.) for component is available at:

<https://github.com/FIWARE/tmforum-api/blob/main/README.md>

Licence

The component is released under the Apache-2.0 License.

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

<https://github.com/FIWARE/tmforum-api/issues>



4.3.8 Scorpio Context Broker

Scorpio is an NGSI-LD compliant Context Broker. It implements the full [NGSI-LD](#) API as specified by the ETSI Industry Specification Group on cross cutting Context Information Management ([ETSI ISG CIM](#)).

As part of the DOME Platform, it serves as the storage backend to the TMForum APIs and provides the NGSI-LD API as a connection interface between the Access Node Instances.

Release Notes

The first release includes a full implementation of the NGSI-LD API.

Source code

The source code of the component is available at:

<https://github.com/ScorpioBroker/ScorpioBroker>

Deployment

The component is distributed as Container Image available at:

<https://quay.io/repository/fiware/scorpio>

The component can be installed through the Helm Chart at:

<https://github.com/FIWARE/helm-charts/tree/main/charts/scorpio-broker>

Further documentation

Detailed and further documentation (deployment, configuration, usage, troubleshooting, etc.) for component is available at:

<https://scorpio.readthedocs.io/en/latest/>

Licence

The component is released under the BSD 3-Clause License.

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

<https://github.com/ScorpioBroker/ScorpioBroker/issues>

4.3.9 Broker Adapter

The Broker Adapter is a component designed to act as an intermediary between Context Brokers and the Blockchain Connector. Its primary function is to facilitate the communication and data exchange between different systems and protocols. By bridging the gap between disparate data sources, the Broker Adapter ensures seamless integration and efficient data processing, making it an indispensable tool for achieving operational synergy within the platform.

Release Notes

The first release includes the following features:



- Set communication between Context Broker and Blockchain Connector.
- Implement CRUD operation to Context Broker entities.

Source code

The source code for the Broker Adapter component is available at:

<https://github.com/in2workspace/broker-adapter>

Deployment

The Broker Adapter component is distributed as Docker Images available at:

<https://hub.docker.com/r/in2kizuna/broker-adapter>

The Broker Adapter component can be installed through Helm following the instructions at:

<https://github.com/in2workspace/helm-charts/tree/main/charts/broker-adapter>

Further documentation

Detailed and further documentation (deployment, configuration, usage, troubleshooting, etc.) for component Broker Adapter is available at:

<https://github.com/in2workspace/broker-adapter/blob/main/README.md>

Licence

The Broker Adapter component is released under the Apache 2.0 Licence.

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

<https://github.com/in2workspace/broker-adapter/issues>

4.3.10 Blockchain Connector

The Blockchain Connector component plays a critical role in the DOME platform, acting as a bridge between blockchain technologies and the local storage. It facilitates secure and efficient interactions with blockchain, enabling seamless integration of data-based services. This component is essential for ensuring interoperability and operational efficiency in blockchain operations within the platform.

Release Notes

The first release includes the following features:

- Set communication between Broker Adapter and DLT Adapter.
- Set subscriptions to Entities and Events.
- Create the transactions events to the blockchain network.
- Retrieve data using blockchain events transactions.
- Audit all transactions processed, as a Producer and a Consumer.

Source code

The source code for the Blockchain Connector component is available at:

<https://github.com/in2workspace/blockchain-connector>



Deployment

The Blockchain Connector component is distributed as Docker Image available at:

<https://hub.docker.com/r/in2kizuna/blockchain-connector>

The Blockchain Connector component can be installed through Helm following the instructions at:

<https://github.com/in2workspace/helm-charts/tree/main/charts/blockchain-connector>

Further documentation

Detailed and further documentation (deployment, configuration, usage, troubleshooting, etc.) for component Blockchain Connector is available at:

<https://github.com/in2workspace/blockchain-connector/blob/main/README.md>

Licence

The Blockchain Connector component is released under the Apache 2.0 licence.

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

<https://github.com/in2workspace/blockchain-connector/issues>

4.3.11 DLT Adapter

The DLT Adapter, also referred to as the DLT Interface is a component used by the Blockchain Connector to interact with the Blockchain technology in a transparent way. It is provided as a REST API. The main objective of this component is to isolate the complexity of the interaction with the federated blockchain networks by providing an abstraction layer on top of them.

Release Notes

The main features of the first release of the DLT Adapter (v1.2) are:

- Configuration of the connection to the chosen blockchain node.
- Publication of events derived from the TM Forum APIs to the blockchain.
- Subscription to events of interest.

Source code

The source code of this component is available at:

https://github.com/alastria/DOME-blockchain_connector-dlt_interface/tree/release/v1.2.1

Deployment

The DLT Adapter component is distributed as a Docker image available at:

<https://quay.io/repository/digitelts/dlt-adapter>

The DLT Adapter component can be installed through the following Helm chart:

https://github.com/alastria/DOME-blockchain_connector-dlt_interface/blob/release/v1.2.1/helmCharts

Further documentation



Detailed and further documentation (deployment, configuration, usage, troubleshooting, etc.) for DLT Adapter component is available at: https://github.com/alastria/DOME-blockchain_connector-dlt_interface/blob/release/v1.2.1/README.md and https://github.com/alastria/DOME-blockchain_connector-dlt_interface/tree/release/v1.2.1/docs in the case of the OpenAPI specification.

License

The DLT Adapter component is released under the Apache 2.0 licence.

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

https://github.com/alastria/DOME-blockchain_connector-dlt_interface/issues

4.3.12 Business Ecosystem Logic Proxy

The Business Ecosystem Logic Proxy component of the BAE provides the entrypoint to the system. This component serves the web interface that can be used to interact with the BAE features. On the one hand, It receives the requests to the different TMForum APIs, validates them and orchestrates the different flows calling the appropriate APIs.

Release Notes

The first release includes the following features:

- Update of the integration with TMForum APIs to use the new implementation. This includes Party Management API, Product Catalog Management API, Product Ordering Management API, Product Inventory Management API and Usage Management API
- Integration of new TMForum APIs, including Account Management, Service Catalog Management, Resource Catalog Management, Service Inventory Management and Resource Inventory Management APIs
- Improve support for Verifiable Credentials
- Use of Service Specifications for the management of the assets handled by plugins instead of the ad-hoc Asset model.

Source code

The source code is available at:

<https://github.com/FIWARE-TMForum/business-ecosystem-logic-proxy>

Deployment

The component is distributed as Docker Images available at:

<https://quay.io/repository/fiware/biz-ecosystem-logic-proxy>

The component can be installed through the Helm chart:

<https://github.com/FIWARE/helm-charts/tree/main/charts/business-api-ecosystem>

Further documentation

Detailed and further documentation for the component is available at ReadTheDocs:

<https://business-api-ecosystem.readthedocs.io/en/latest/>

Licence

The component is released under the AGPL v3 licence

Support and Contacts

Bugs, requests and other code-related issues can be reported at:

<https://github.com/FIWARE-TMForum/business-ecosystem-logic-proxy/issues>

4.3.13 Business Ecosystem Charging Backend

The Business Ecosystem Charging Backend includes the business logic of the BAE, processing product orders, calculating billing and providing procurement and activation features.

Release Notes

The first release includes the following features:

- Update the integration with TMForum APIs to use new implementations, including Product Catalog Management API, Product Ordering Management API, Product Inventory Management API and Usage Management API
- Integrate new APIs for procurement, including Resource Inventory Management API and Service Inventory Management API
- Update plugin management to use Service Specifications instead of the ad-hoc Asset model
- New implementation of the Revenue Sharing system as part of the BAE charging backend, replacing the previous separate component

Source code

The source code is available at:

<https://github.com/FIWARE-TMForum/business-ecosystem-charging-backend>

Deployment

The component is distributed as Docker Image available at:

<https://quay.io/repository/fiware/biz-ecosystem-charging-backend>

The component can be installed through the Helm chart:

<https://github.com/FIWARE/helm-charts/tree/main/charts/business-api-ecosystem>

Further documentation

Detailed and further documentation for the component is available at ReadTheDocs:

<https://business-api-ecosystem.readthedocs.io/en/latest/>

Licence

The component is released under the AGPL v3 licence

Support and Contacts



Bugs, requests and other code-related issues can be reported at:

<https://github.com/FIWARE-TMForum/business-ecosystem-charging-backend/issues>



5 Conclusions

This document presented the first major release of the DOME platform, showcasing an initial set of marketplace features and realising a PoC of the envisioned federated approach. Its availability in a publicly-accessible environment promoted visibility of the initial project results; it has been successfully exploited for demo purposes and is being used for training, experimentation and communication.

DOME integrates, customises and develops a number of components built on top of cloud-native technologies like Docker, Kubernetes and Helm in order to achieve a high level of modularity, robustness and interoperability. The development, integration and release activities behind initial software releases this year will follow a more structured and consistent approach by following WP4 guidance starting from the second project year.

A release plan has been defined, covering a number of intermediate releases until M18, when a production-level deployment is expected to support early customers, providers and marketplaces. The major aim of this initial roll-out and operation, conducted within WP4, will be to validate the technical solution (but also the governance and the business model) and gather valuable feedback to fine-tune the planning of next intermediate and major releases.

Two further major versions of the DOME platform will be issued (at M24 and M36) including more features and components and will be documented in deliverables D3.5 and D.3.8.