# D2.2

# COP-PILOT ARCHITECTURE AND FUNCTIONALITIES

Report on the final version of the COP-PILOT architecture and functionalities

Revision: v1.0

# D2.2: COP-PILOT architecture and functionalities

COP-PILOT's final design specifications

| | |
|---|---|
| **Work package** | WP2 |
| **Task** | T2.1- T2.3 |
| **Due date** | 30/06/2026 |
| **Submission date** | 30/06/2026 |
| **Deliverable lead** | Georgios P. Katsikas (UBITECH) |
| **Version** | 1.0 |
| **Authors** | Georgios P. Katsikas, Dimitrios Manolopoulos, Dimitrios Klonidis (UBITECH), Kostis Trantzas (UOP), Christos Tranoris (PNET), *Ioanna Drigkopoulou* , Konstantinos Fragkos, Panos Mantzakos (NETC), Philip Griffiths (TATA), Benjamin Ertl (AGE), Jesus Iglesias, Emilio Garrido (TID), Luis Ferreira, Luis Rosa (ONE), *Martin Engelmark (LTU Business), Nuria Molner (UPV), Gregory Mygdakos (AGA), Lefteris Mylonas (UOP), Anastasis Tzoumpas (NOVA)* |
| **Reviewers** | Jesus Iglesias Maqueda (TID), Emilio Garrido (TID) |
| **Abstract** | This deliverable presents the final architecture and core operational workflows of the COP-PILOT platform, constituting the definitive technical reference for the platform's implementation and integration activities. Building on the ecosystem requirements established in D2.1, D2.2 introduces a six-layer architecture spanning the full IoT-to-edge-to-core computing continuum, from heterogeneous physical infrastructure through distributed domain orchestration and secure zero-trust networking, to end-to-end service management and a business portal with native LLM-assisted interaction. The document specifies the fundamental platform workflows covering domain onboarding, service federation, ordering, and reconfiguration, and provides a detailed service modelling engineering view for each of the five piloting clusters, mapping use case requirements to concrete infrastructure configurations and deployment specifications. The outputs of D2.2 directly feed the platform development activities of WP3 and the cluster integration activities of WP4. |
| **Keywords** | Edge computing, service orchestration, cloud continuum, distributed infrastructure, multi-domain management, secure integration fabric, SLA preservation, platform architecture, open platform, interoperability. |

## Document Revision History

| Version | Date | Description of change | List of contributors |
|---|---|---|---|
| V0.1 | 03/03/2026 | Table of Contents | *Georgios P. Katsikas (UBITECH)* |
| V0.2 | 28/04/2026 | Updated architecture<br><br>Initial draft of workflows | *Georgios P. Katsikas (UBITECH), Kostis Trantzas (UOP), Philip Griffiths (TATA), Konstantinos Fragkos, Panos Mantzakos (NETC), Benjamin Ertl (AGE), all WP2 partners*<br><br>*Georgios P. Katsikas (UBITECH)* |
| V0.3 | 21/05/2026 | Abstract, Executive Summary, Introduction,<br><br>Workflows | *Georgios P. Katsikas, Dimitris Manolopoulos (UBITECH)*<br><br>*Benjamin Ertl (AGE), Kostis Trantzas (UOP), Christos Tranoris (PNET), Georgios P. Katsikas (UBITECH)* |
| V0.4 | 05/06/2026 | Service models<br>- Framework<br>- Per cluster adaptation<br>- Service Marketplace | *Donal Morris, Ksama Patil (RZ),*<br>*Georgios P. Katsikas (UBI),*<br>*Cluster 1: Martin Engelmark (LTU Business)*<br>*Cluster 2: Nuria Molner (UPV),*<br>*Cluster 3A: Gregory Mygdakos (AGA)*<br>*Cluster 3E: Lefteris Mylonas (UOP)*<br>*Cluster 4: Luis Rosa (ONE)* |
| V0.5 | 21/05/2026 | Abstract<br>Executive Summary<br>Section 1 (Introduction)<br>Section 6 (Conclusion) | Dimitrios Manolopoulos (UBITECH) |
| V0.6 | 07/06/2026 | Final editing | *Georgios P. Katsikas (UBITECH)* |
| V0.7 | 12/06/2026 | Internal review | Jesus Iglesias Maqueda, Emilio Garrido (TID) |
| V0.8 | 22/06/2025 | Addressed review comments | *Georgios P. Katsikas (UBITECH)* |
| V1.0 | 25/06/2026 | Submission | *Ioanna Drigkopoulou (NETC)* |

**Grant Agreement No:** 101189819 | **Topic:** HORIZON-CL4-2024-DATA-01-03
**Call:** HORIZON-CL4-2024-DATA-01 | **Type of action:** HORIZON-IA

# DISCLAIMER

Co-funded by
the European Union

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs,
Education and Research EAER
**State Secretariat for Education,**
**Research and Innovation SERI**

Co-funded by the European Union (COP-PILOT, 101189819). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

# COPYRIGHT NOTICE



| | | |
|---|---|---|
| Project Co-funded by the European Commission in the Horizon Europe Programme | | |
| Nature of the deliverable: | R | |
| Dissemination Level | | |
| PU | Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page) | ✔ |
| Classified R-UE/ EU-R | EU RESTRICTED under the Commission Decision No2015/ 444 | |
| Classified C-UE/ EU-C | EU CONFIDENTIAL under the Commission Decision No2015/ 444 | |
| Classified S-UE/ EU-S | EU SECRET under the Commission Decision No2015/ 444 | |

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

DATA: Data sets, microdata, etc.

DMP: Data management plan

ETHICS: Deliverables related to ethics issues.

SECURITY: Deliverables related to security issues

OTHER: Software, technical diagram, algorithms, models, etc.

# TABLE OF CONTENTS



Co-funded by the European Union

# EXECUTIVE SUMMARY

The COP-PILOT project (Collaborative Open Platform for Piloting services Across Emerging Smart IoT and Edge computing environments) aims to develop and validate an open, standards-based orchestration platform capable of managing collaborative applications across heterogeneous IoT-to-edge-to-core computing environments. Deliverable D2.2 presents the final COP-PILOT architecture and its associated workflows, building upon the ecosystem requirements and initial design established in D2.1 to provide the complete technical blueprint that guides the platform's implementation across WP3 and WP4.

The final COP-PILOT architecture is organised into six horizontal layers. The Infrastructure Layer (INFRA-L) encompasses the full spectrum of physical and virtual resources, from intelligent IoT swarms at the extreme edge to hyperscaler public cloud environments. Above it, the Distributed Infrastructure Services Layer (DIS-L) provides industrial-grade software services that interface with this heterogeneous hardware. The Distributed Domain Orchestration Layer (DDO-L) sits atop DIS-L and introduces a modular Domain Orchestrator (DO) and Data Management (DM) platform for each administrative domain, exposing compute, network, and data resources as standardised, marketable services using TMF APIs. The Secure Integration Fabric Layer (SIF-L) provides encrypted, zero-trust overlay networking across all COP-PILOT domains, eliminating the administrative burden of traditional VPN management and enabling secure cross-domain collaboration. The End-to-End Service Orchestration Layer (ESO-L) coordinates services spanning multiple domains by federating the individual DO marketplaces into a unified, multi-domain service catalogue. Finally, the Business Management Layer (BM-L) exposes a Business Management Portal (BMP) acting as the primary business-facing interface of the COP-PILOT platform. Beyond providing a unified product marketplace, the BMP implements core Business Support System (BSS) capabilities, including product catalogue management, product offering lifecycle management, and product order handling. Additionally, the BMP integrates a native LLM-based assistant that facilitates user interaction by translating high-level intents into validated product configurations and corresponding service orders, significantly reducing the complexity of multi-domain service consumption.

The document further introduces nine key platform workflows covering the full operational lifecycle: binding the ESO with the secure network fabric, expanding the platform to new private domains in a multi-step automated process, designing and federating domain-level services, ordering and reconfiguring services through the orchestrators, and designing and ordering products via the Business Management Portal, including LLM-assisted ordering. These workflows demonstrate how COP-PILOT achieves its ambition of reducing complex orchestration tasks, such as domain onboarding and service provisioning, from hours of manual effort to minutes of guided interaction.

The deliverable concludes with a summary of the project’s service modelling approach that translates each piloting cluster's use cases into concrete infrastructure specifications and deployment configurations. Across all five clusters (mining operations in Sweden, smart city applications in Valencia, Agritech in Central Macedonia - Greece, energy grid management in Preveza - Greece, and sustainable resource management across Portugal and Spain) a consistent engineering methodology is applied, identifying infrastructure blocks, configuration action sequences, and gaps against the evolving COP-PILOT service library. This analysis directly informs the integration and validation activities planned through the remainder of the project, supporting progressive platform releases up to the final milestone at Month 36.

# LIST OF FIGURES

# LIST OF TABLES

# ABBREVIATIONS

| Term | Description |
|---|---|
| 3GPP | 3rd Generation Partnership Project |
| BM-L | Business Management Layer |
| BMP | Business Management Portal |
| BSS | Business Support System |
| CD | Continuous Delivery |
| CI | Continuous Integration |
| DC | Datacenter |
| DDO-L | Distributed Domain Orchestration Layer |
| DIS-L | Distributed Infrastructure Services Layer |
| DM | Data Management |
| DO | Domain Orchestrator |
| ESO | End-to-end Service Orchestrator |
| ESO-L | End-to-end Service Orchestration Layer |
| ETSI | European Telecommunications Standards Institute |
| INFRA-L | Infrastructure Layer |
| IoT | Internet of Things |
| OSS | Operations Support System |
| SIF | Secure Integration Fabric |
| SIF-L | Secure Integration Fabric Layer |
| TMF | TM Forum |
| WP | Work Package |

# 1 INTRODUCTION

The COP-PILOT project pursues the development of a Collaborative Open Platform for Piloting services Across Emerging Smart IoT and Edge computing environments. At its core, COP-PILOT seeks to establish an open, standards-based orchestration framework capable of managing collaborative applications across massively heterogeneous infrastructure landscapes (from intelligent IoT swarms at the extreme edge to hyperscaler public cloud environments) while delivering unprecedented levels of automation, security, and interoperability. Realising this ambition requires a carefully designed architecture that not only accommodates today's infrastructure diversity but is also sufficiently modular and extensible to evolve alongside the rapidly changing edge computing ecosystem.

This deliverable, D2.2, marks the culmination of Work Package 2 (WP2) design activities and presents the final, definitive version of the COP-PILOT platform architecture together with its core operational workflows and service modelling specifications. It represents the primary technical reference document that bridges the design phase of the project with the implementation and integration phases carried out in WP3 and WP4 respectively.

## 1.1 OBJECTIVE

The principal objective of this deliverable is to consolidate and finalise the COP-PILOT architecture, translating the ecosystem requirements, stakeholder needs, and use case specifications established throughout WP2 into a coherent, actionable technical blueprint. Whereas D2.1 provided the foundations (ecosystem characterisation, functional and non-functional requirements, initial architectural decomposition, and preliminary platform design) D2.2 advances this work to its final form, resolving open design questions, incorporating feedback from early integration activities, and introducing the detailed workflows and service configurations that will govern the platform's operation in practice.

Specifically, D2.2 pursues four interrelated objectives. First, it presents the final COP-PILOT architecture in both a high-level form, accessible to a broad audience of stakeholders, and a detailed form that exposes the internal structure of each architectural layer and the interfaces between them. Second, it specifies fundamental operational workflows of the platform, covering the full lifecycle of platform expansion, service federation, service ordering, and product management through the business portal. Third, it introduces the service modelling engineering view for each of the five piloting clusters, mapping use case requirements to concrete infrastructure configurations and deployment specifications. Fourth, it identifies the current state of the COP-PILOT service library and outlines the catalogue of services expected by the project's conclusion at Month 36.

Collectively, these contributions ensure that WP3 and WP4 partners have a shared, unambiguous understanding of what the platform must do and how it must be structured, reducing integration risk and providing a stable reference against which platform releases can be evaluated throughout the project's remaining development cycles.

Co-funded by
the European Union

## 1.2 RELATION WITH OTHER DELIVERABLES

D2.2 complements the extensive D2.1 [1] with the last bits and pieces to finalize the COP-PILOT architecture specification and important workflows to define key operations. D2.2 then directly feeds D3.1 [2] and D4.1 [3] with abstract architecture blocks already mapped to WP3 and WP4 activities that should materialize these blocks into concrete platform instances. The architecture is instantiated in the context of WP5 and all its Use Case and Open Call validation and evaluation activities.

## 1.3 DOCUMENT STRUCTURE

This document is organized in four (4) sections as follows:

- **Section 2** presents the **final COP-PILOT architecture** in two complementary representations: a minimal high-level view designed to convey the platform's essential structure to a broad audience, and a detailed view that articulates the internal composition of each architectural layer. The section concludes with an explicit mapping of the architecture to the technical work packages responsible for realising each component.

- **Section 3** introduces the fundamental **operational workflows** of the COP-PILOT platform, covering new private domain onboarding, service federation into a common Marketplace, service ordering and reconfiguration, and product management through the Business Management Portal, including LLM-assisted ordering.

- **Section 4** presents a summary of the **service modelling** engineering view for each of the five clusters, translating the value definitions established in D2.1 into concrete infrastructure blocks, deployment configurations, and gap analyses against the COP-PILOT service library. A view of the current (M18) and expected (by M36) service marketplace is provided.

- **Section 5** draws **conclusions** about COP-PILOT’s technical specifications and wraps up the work in WP2.

# 2 FINAL COP-PILOT ARCHITECTURE

This section introduces the final COP-PILOT architecture in two complementary versions: (i) a minimal version to convey the high-level aspects of the platform (Section 2.1) and (ii) a detailed version outlining internal details of the basic components (Section 2.2). An overview of the architecture's components is provided in Section 2.3, where links to technical Work Packages and deliverables are also provided.

## 2.1 FINAL COP-PILOT ARCHITECTURE: HIGH-LEVEL VERSION

To better convey COP-PILOT to the EU citizens and ecosystem of potential users, we deem important to provide a simplified view of the COP-PILOT architecture (see Figure 2-1) that abstracts away low-level details without losing essential context. This figure boils down the complex COP-PILOT architecture into five (5) essential pieces:

- Distributed COP-PILOT domains across the COP-PILOT Clusters in Europe. Individual domain stakeholders exploit domain-level orchestration (DO) and Data Management (DM) services for jointly managing compute, network, and data resources.
- A Secure Integration Fabric (SIF) for allowing secure on-the-fly data and service-level interactions between multiple domains,
- An End-to-end Service Orchestrator (ESO) for managing services that span across multiple domains, thus highlighting the "collaborativeness" of the COP-PILOT system.
- A Business Management Portal (BMP) offers a central marketplace of products and an LLM-driven assistant to turn stakeholders' intents into actionable platform operations.
- A vertical CI/CD platform ensuring continuous integration and delivery of new features and bug fixes across every piece of software in Europe.

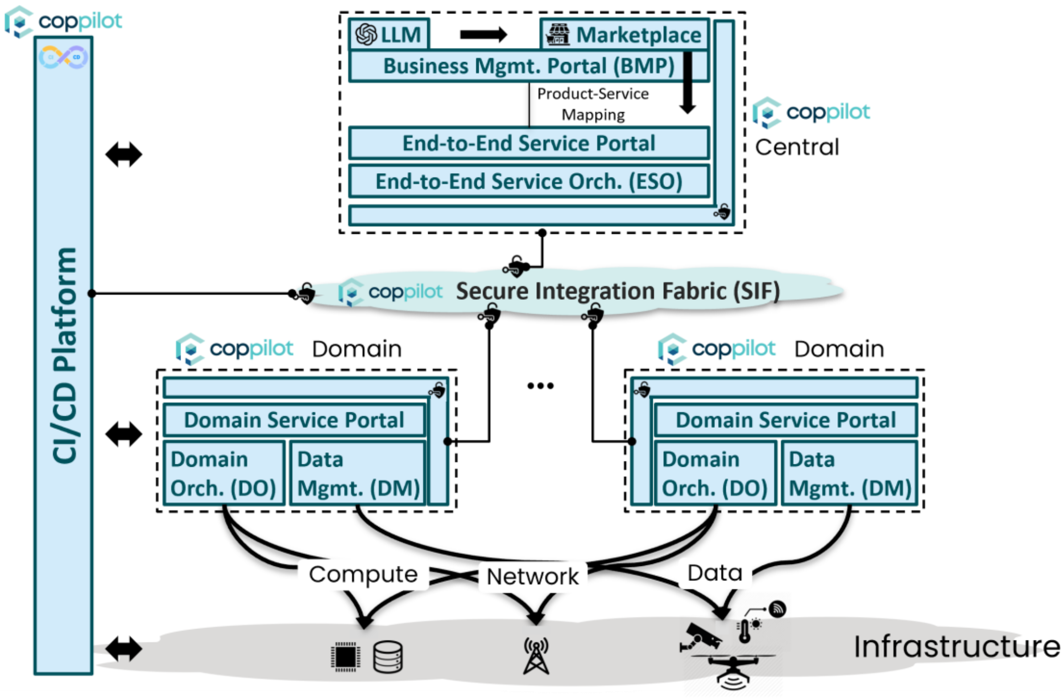


*Figure 2-1: A minimal view of the final COP-PILOT architecture.*

Note that the minimal view of the initial COP-PILOT architecture shown in Figure 2-1 focuses solely on COP-PILOT elements. To make this view more complete, Figure 2-2 adds additional items to sketch the entire ecosystem around COP-PILOT: (i) the layers of the architecture shown at the right hand side in Figure 2-2, (ii) the span of the infrastructure at the bottom part in Figure 2-2, (iii) the various stakeholders associated with COP-PILOT (see user icons in Figure 2-2), (iv) the primary vertical sectors that COP-PILOT facilitates via piloting activities (top part in Figure 2-2), and (v) the way COP-PILOT and 3rd party services interact with the platform. Note also that the pink-highlighted boxes in Figure 2-2 are presented as distinct abstractions of the platform in the rest of this section.

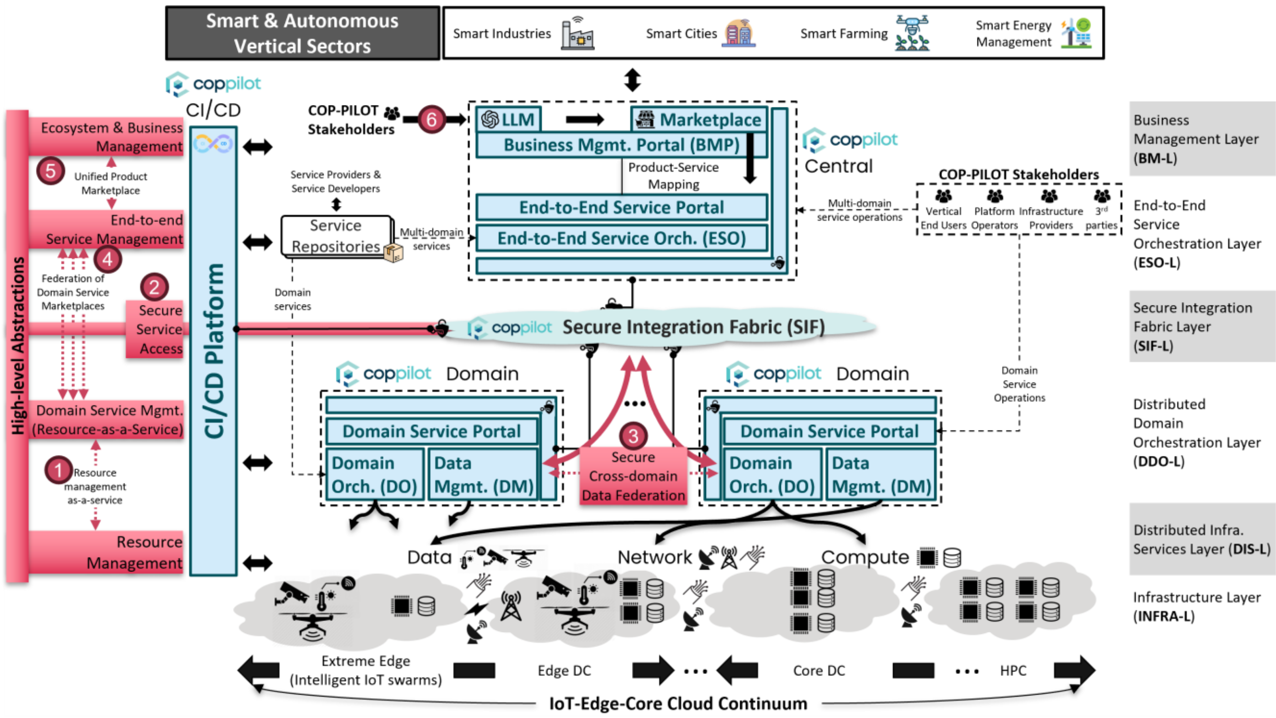


*Figure 2-2: A simplified equivalent of the final COP-PILOT architecture.*

**Key COP-PILOT Abstractions:**

**(1) Resource management in an as-a-service manner:** Despite dealing with a massively heterogenous continuum of physical and virtual compute, network, and data resources in every domain, COP-PILOT hides the details of these resources under a resource-as-a-service model. This allows infrastructure owners to easily expose the abilities of their hardware without disclosing resource-level details to the consumers. Having resources-as-a-service also allows the infrastructure owners to monetize these services in the COP-PILOT marketplace, linking them with products that can be purchased by interested stakeholders.

**(2) Secure Service Access:** COP-PILOT offers identity-first, zero-trust connectivity services within the entire COP-PILOT ecosystem, eliminating the need for managing traditional "connect first, verify later" VPNs in large ecosystems of multiple administrative entities.

**(3) Secure Cross-domain Data Federation:** The synergy between COP-PILOT's SIF and DM components allows secure sharing of data between domains to enable collaborative scenarios throughout the COP-PILOT piloting activities.

**(4) Federation of Domain Service Marketplaces:** COP-PILOT's end-to-end service orchestrator binds with multiple domain orchestrator instances across multiple domains to federate domain-level marketplaces into a large multi-domain marketplace of services offered to the COP-PILOT stakeholders.

**(5) A unified Product Marketplace:** Maps the end-to-end services offered by the COP-PILOT ESO to standardized products and product offerings of a unified Pan-European marketplace offered to all COP-PILOT stakeholders.

**(6) LLM-driven interaction with the Product Marketplace:** COP-PILOT offers a native LLM assistant as part of a business management portal to greatly facilitate the way stakeholders consume the COP-PILOT products and product offerings.

## 2.2 FINAL COP-PILOT ARCHITECTURE: DETAILED VERSION

Figure 2-3 introduces the final version of the COP-PILOT architecture. This figure takes the simplified version of the architecture introduced in Figure 2-1 to the next level, introducing additional details in every layer of the architecture as described in the rest of this section.

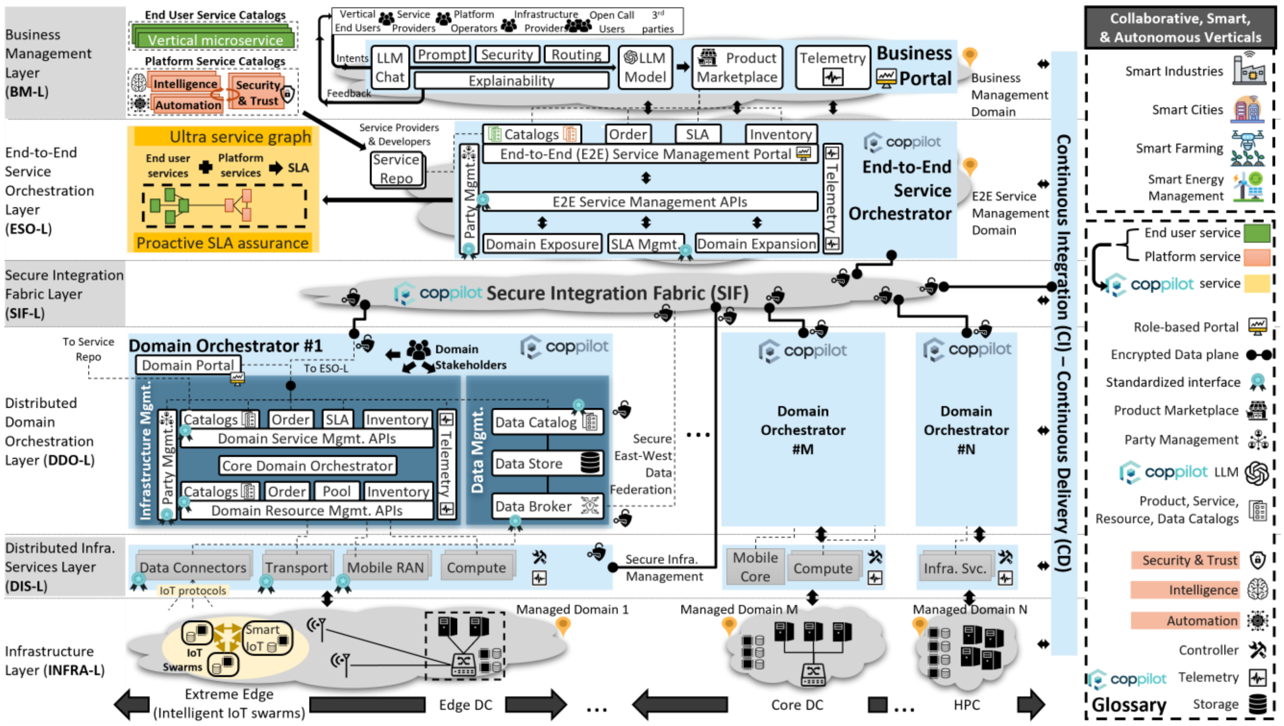


*Figure 2-3: The final COP-PILOT architecture in detail.*

### 2.2.1 Infrastructure Layer (INFRA-L)

Today's infrastructures exhibit a great amount of heterogeneity due to the rapid expansion of the network towards the end users, where user equipment (UE) is surrounded by a plethora of individual or collaborative processing-capable IoT devices, programmable network elements, off-the-shelf, and specialized edge devices. This largely heterogeneous device manifests itself across multiple domains of different scales ranging from extreme edge domains of intelligent IoT swarms to private edge domains, core network domains and datacenters, as well as hyperscaler-level public cloud domains as shown at the bottom part in Figure 2-3. This is the physical or "Infrastructure layer" – also abbreviated as INFRA-L - on top of which the COP-PILOT architecture must develop and provide solutions. Note that this layer covers any kind of infrastructure, including potential infrastructure that may appear during the Open Calls, third parties, or future customers that wish to integrate their infrastructure with the COP-PILOT platform.

### 2.2.2 Distributed Infrastructure Services Layer (DIS-L)

The distributed infrastructure landscape outlined in INFRA-L creates the need for developing industrial-grade infrastructure services which directly interact with the underlying hardware through a multitude of protocols, while providing unified northbound management knobs for taming the southbound complexity. The global IT community has been striving for decades to build open-source platforms that undertake this challenging role of dealing with so much hardware heterogeneity, producing outstanding projects that provide industrial-grade solutions for compute, network, and data management. The technologies around these services will be studied and analysed in the context of WP3, however, from an architectural perspective, COP-PILOT looks at these services as a distinct layer atop INFRA-L, titled "Distributed Infrastructure Layer" and abbreviated as DIS-L (see Figure 2-3). This layer is highly distributed, as does the underlying INFRA-L, while the owners of the infrastructure may mix and match legacy vs. advanced and open vs. proprietary technologies of their choice to manage the underlying infrastructure. Nevertheless, COP-PILOT shall provide a solid overlay platform for dealing with the plethora of northbound interfaces of the DIS-L, as explained next.

### 2.2.3 Distributed Domain Orchestration Layer (DDO-L)

**Compute and Network Management:** To facilitate business within every large or small administrative domain, while ensuring maximum security and independence, COP-PILOT fosters a low-tier orchestration layer for managing every single resource and service within a domain. Because the COP-PILOT ecosystem comprises of multiple domains, this layer is also distributed (like INFRA-L and DIS-L), titled "Distributed Domain Orchestration Layer" and abbreviated as DDO-L (see Figure 2-3). The role of this layer is to interact with the northbound APIs for the underlying infrastructure services and expose the resources that these services manage as-a-service. To do so, the industry has been attempting to standardize resource and service orchestration for decades, designing and evolving APIs for (i) describing a resource and/or service using resource/service specifications, (ii) classifying resource and/or service specifications across relevant categories, and (iii) organizing these categories in resource and/or service catalogues, effectively creating the notion of a resource/service marketplace. Additional APIs are also designed for ordering resources and services from the marketplace, towards a specific infrastructure and managing the instantiated resources/services (the so-called resource/service inventory) once being successfully provisioned as a result of an order. COP-PILOT leverages these well-structured standardization efforts to introduce a Domain Orchestration (DO) platform that implements all these APIs for ultimately offering compute and network resource-as-a-service. Alongside resources and services, the COP-PILOT Domain Orchestration platform offers APIs for encoding the role of domain stakeholders (either individuals or organizations) as parties in the domain ecosystem, while also capturing the state of the domain's resources via compute and network telemetry. The COP-PILOT Domain Orchestration platform is visualized in Figure 2-3 as a core part of the DDO-L.

**Data Management:** Modern infrastructures pose extra requirements for managing data in tandem with compute and network resources and services, especially since the emergence of Internet of Things (IoT). For this reason, COP-PILOT caters for a dedicated Data Management platform withing DDO-L (aside to the DO) for serving the complementary purpose of offering Data-as-a-Service. This platform interacts with Data Connectors from the underlying DIS-L, brings their data in, and provides essential services, such as (i) data manipulation and/or pre-processing, (ii) data persistence, (iii) east-west data sharing, and (iv) data cataloguing/exposure towards overlay service as shown in Figure 2-3. The COP-PILOT Data Management platform is also abbreviated as “Data Mgmt.” or DM.

**Domain-level Portal:** A dedicated portal (see Figure 2-3) facilitates domain-level operations that need to be taken by e.g., the domain administrator/owner, domain service providers, and domain infrastructure owners. Such operations could be resource and service management (i.e., ordering, lifecycle management, and termination) within a domain, management of parties within a domain, management of domain-level SLAs, and domain-level visibility tools (i.e., resource and service telemetry). This portal addresses the role of a domain-specific Operations Support System (OSS).

**Modularity:** Note that the DDO-L is designed to be modular. This means that for domains that require only a DO (due to absence of data) or only a DM (due to absence of compute and network resources), the architecture allows us to deploy only the necessary part of the Domain Orchestration stack, thus covering only the true needs of every domain.

### 2.2.4 Secure Integration Fabric Layer (SIF-L)

COP-PILOT is a highly distributed system that spans across multiple self-managed domains across Europe; as such, most of the COP-PILOT components, mainly from the DDO-L (domain services, resources, and data) towards the INFRA-L (i.e., hardware), may be deployed within private infrastructures, the owners of which may wish not to expose publicly. To integrate these components with the rest of the system and allow collaborative cross-domain service deployments, there is an imminent requirement to provide secure connectivity among COP-PILOT components. Doing so via a classical VPN is cumbersome as a huge administrative effort is required per IT department to issue VPN accounts, credentials, and configuration files for several external parties, not to mention that this needs to happen across tens of different IT departments by the partners who provide infrastructure and/or services. COP-PILOT revisits the way secure and private networking should be done nowadays by leveraging recent advancements in secure software-defined overlay networking, effectively offering programmable VPN-as-a-Service. This service resides in COP-PILOT’s Secure Integration Fabric Layer, also abbreviated as SIF-L, as shown in Figure 2-3.

Using SIF, COP-PILOT sets up a root-of-trust domain where the SIF control plane will be publicly exposed as a cloud-managed service to the entire ecosystem. COP-PILOT parties who wish to enter the COP-PILOT ecosystem will exploit the COP-PILOT Business Management Portal (provided by the BM-L below) to register to the platform, authenticate using state-of-the-art OAUTH2 mechanisms, and use the portal’s dedicated domain expansion view to register a new domain under the COP-PILOT realm through an authentication token, and consequently declare domain resources and services that the domain owner wishes to expose through the integration fabric. This information will flow from the COP-PILOT Business Management Portal to the secure integration fabric backend, triggering the creation of dynamic policies that will associate the domain’s resources and services with certain parties and party services who are eligible to consume these endpoints. Both the SIF control and data planes are encrypted; thus, no unprotected packet traverses the public internet, creating a secure programmable overlay network dedicated to the COP-PILOT platform. This is aligned with the Integration Fabric component of the ETSI ZSM reference architecture and goes one step further by adding key security and trust features.

### 2.2.5 End-to-End Service Orchestration Layer (ESO-L)

**End-to-end Service Management:** Modern services may span across multiple domains of different scales ranging from extreme edge to edge, core, and public cloud domains, thus falling outside of a DO’s scope. For this reason, COP-PILOT introduces a high-tier orchestration platform titled “End-to-

End Service Orchestrator" (ESO) to manage services across multiple geo-distributed COP-PILOT domains in an end-to-end manner (see the top part in Figure 2-3). The COP-PILOT ESO consumes the standardized service and resource management interfaces exposed by each COP-PILOT DO instance to acquire the state of each domain by means of available/consumed compute and network resource services. At the northbound, the ESO exposes similar standardized TMF APIs to (i) design and onboard multi-domain end user services on a desired catalogue, (ii) order a multi-domain service or service bundle (composite service chain) with certain SLA requirements, and (iii) monitor a multi-domain service instance's runtime through a service inventory API and a service telemetry engine. When such an end-to-end service is ordered, the ESO breaks this service down to one or more service order requests (towards one or more DO instances), thus partitioning the service graphs across domains. Each DO instance manages its own partition of the end-to-end service, while ESO ensures secure connectivity between service components either within a domain or across domains. Dynamic service runtime updates are supported by the COP-PILOT ESO, either through update operations on the service order and inventory APIs or via a policy API which allows us to construct custom rules and apply them in various checkpoints of a service's lifecycle. Finally, the ESO possesses APIs to manage COP-PILOT's SIF in a dynamic way. These APIs allow to (i) attach/detach new SIF instances under the ESO, thus expand the COP-PILOT platform network accordingly, (ii) manage the identities (i.e., stakeholders) within every SIF instance, and (iii) create/delete encrypted tunnels on-the-fly. This allows ESO to drive the expansion of the platform towards new private domains and automate the process of deploying new COP-PILOT DO instances in these domains, hence transforming new domains into orchestrated platform environments in a glimpse of time.

**End-to-end Service Portal:** A dedicated portal is provided by the ESO (see Figure 2-3) to visualize key end-to-end service operations, such as service design, service onboarding on certain categories/catalogues, as well as service inventory management. Additional views (i) provide the state of the entire platform, visualizing the hierarchy of orchestrators (ESO and multiple DO instances) across domains and the underlying resources/services in each domain and (ii) allow the platform administrator to easily expand the platform to new domains, embracing more infrastructure owners under the COP-PILOT realm. This portal addresses the role of an OSS tailored for multi-domain service operations.

### 2.2.6 Business Management Layer (BM-L)

Today's service and network management platforms not only require powerful abstractions and APIs for managing heterogeneous resources and services but also means to render these capabilities friendly to the overly ecosystem of users. These users may range from platform administrators with domain-level expertise on orchestration to end users with zero understanding on how the platform works. Due to this heterogeneity of users and their backgrounds, a business layer – atop COP-PILOT's domain-level (DO-L) and end-to-end service management (ESO-L) layers – is introduced to complement the rest of the system with high-level tools that render important platform operations simple and intuitive. This layer is titled Business Management Layer – also abbreviated as BM-L – as shown in Figure 2-3. The core of the BM-L is a Business Management Portal (BMP) that resides on top of the COP-PILOT orchestrators. This layer plays the complementary role of a Business Support System (BSS), offering standardized Product-level APIs that map existing ESO services with actual products in the COP-PILOT marketplace. This mapping creates a powerful trinity of Resources-to-Services-to-Products that commonly appear in large-scale commercial systems today. COP-PILOT leverages the advent of Large Language Models (LLMs) and the emerging capabilities of AI for the benefit of the COP-PILOT users, offering an LLM – native to the Business Management Portal – to facilitate the interaction of end users with the platform by means of (i) guiding the user to the right product catalogues and categories and (ii) ordering composite products with certain SLA requirements on behalf of the user.

### 2.2.7 Cross-cutting Integration through the COP-PILOT CI/CD platform

Such a complex ecosystem of Pan-European scale could not be managed without automation tools that facilitate the provisioning of platform components across testbeds of multiple administrative stakeholders. The COP-PILOT CI/CD platform shown in Figure 2-3 (i) leverages the SIF to establish secure connections towards every testbed and (ii) invokes pipelines that continuously integrate features and bug fixes of the main platform components (i.e., BMP, ESO, all DO and DM instances), and (iii) schedules additional pipelines that roll-out new versions of COP-PILOT Vertical Applications (i.e., for UCs) through the orchestration platform.

## 2.3 MAPPING OF THE FINAL COP-PILOT ARCHITECTURE WITH TECHNICAL WORK PACKAGES

To break down the responsibilities for realizing the COP-PILOT architecture, a mapping of this architecture with key technical work packages (i.e., WP3 and WP4) and tasks is necessary. This mapping is visualized in Figure 2-4, which is an annotated version of Figure 2-3. In short, WP3 is responsible for the core COP-PILOT platform (BM-L, ESO-L, DDO-L, SIF-L highlighted in purple colour), while WP4 offers the ecosystem around the platform by means of (i) the underlying infrastructure (i.e., INFRA-L highlighted in green colour at the bottom part in Figure 2-4) and infrastructure services (i.e., DIS-L highlighted in grey colour in Figure 2-4) to be managed by the platform and (ii) the overlay vertical sector services, either stemming from the COP-PILOT Clusters or external (e.g., Open Calls or third-parties) to be managed by the platform (see green-highlighted top part in Figure 2-4). Note that the vertical pillar providing CI/CD tools across all layers is a joint responsibility between WP3 and WP4, as it touches upon platform and Cluster-specific aspects. More details about these mappings are available in Section 2 of Deliverables D3.1 [2] and D4.1 [3].

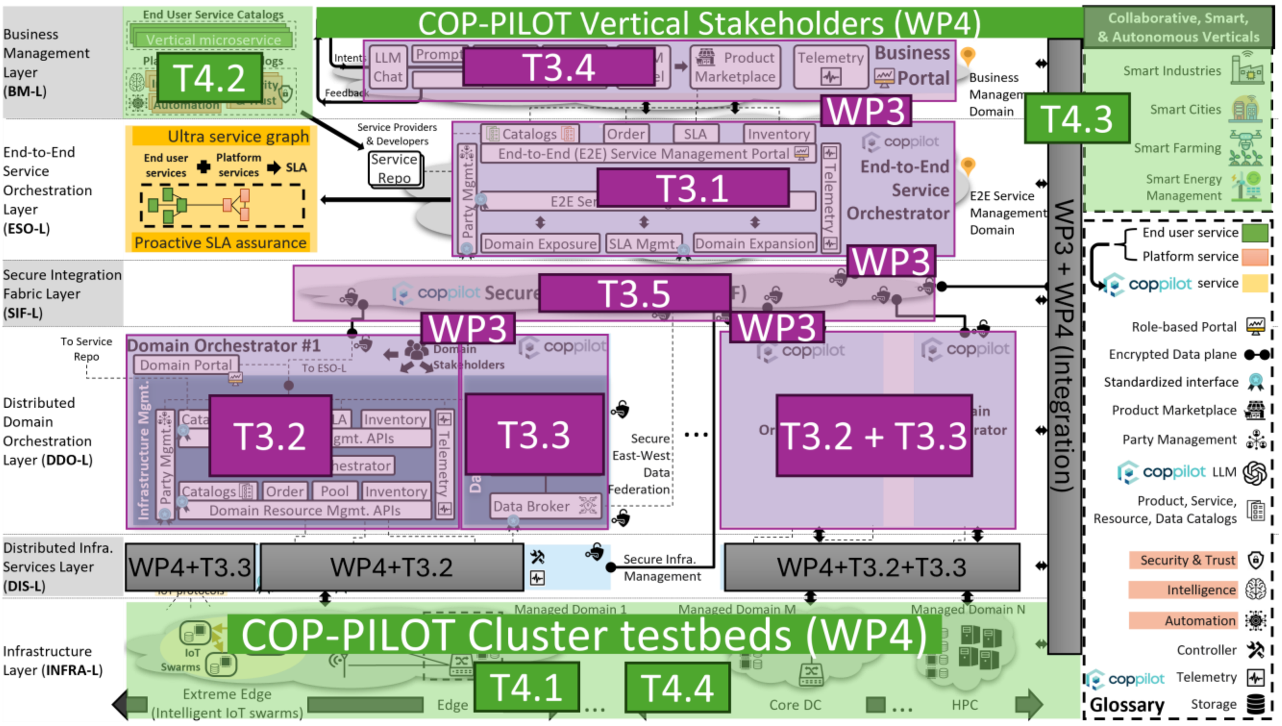


Figure 2-4: Final COP-PILOT architecture mapped to technical WPs.

# 3 BASIC COP-PILOT ARCHITECTURE WORKFLOWS

This section introduces some key workflows offered by the COP-PILOT platform at different layers of the system. First, the workflow in Section 3.1 visualizes how the COP-PILOT ESO peers with a secure network fabric (i.e., the COP-PILOT SIF) render the platform expandable to new domains in a secure and trusted manner. Next, Section 3.2 discusses how the COP-PILOT platform expands to new domains through a synergy of portal-driven operations and automated ESO-SIF-DO operations. Section 3.3 introduces a workflow that describes how a service designer may design new services using standardized APIs at the level of the orchestrators – using the DO as an example. In Section 3.4, a workflow visualizes how the COP-PILOT ESO peers with multiple COP-PILOT DOs to import all domain-level services spread across multiple DOs onto a centralized service marketplace offered by the ESO. Once services are designed and onboarded onto an orchestrator's catalogue, Section 3.5 introduces a workflow that describes the steps to order these services. Section 3.6 visualizes how a user may reconfigure a running service. Moreover, Section 3.7 introduces how stakeholders may leverage the COP-PILOT Business Management Portal to design Products mapped to ESO services, while Section 3.8 shows how such products may be ordered through the portal. Finally, Section 3.9 introduces a workflow that visualizes the way COP-PILOT stakeholders may leverage the Business Management Portal's LLM to order products via chatting with an AI-based assistant.

## 3.1 PLATFORM BINDING WITH A SECURE NETWORK FABRIC

This section describes how a platform administrator can bind the COP-PILOT ESO with a secure and programmable network fabric (i.e., the COP-PILOT SIF) to render the platform expandable to new domains. This workflow is visualized in Figure 3-1 assumes that the SIF is pre-deployed by the platform administrator through an automated COP-PILOT CI/CD Platform pipeline. The flow begins with the platform administrator issuing a new ESO account on the SIF platform. Once the account is created the platform administrator authenticates on the ESO's portal (also omitted for brevity) and browses through the platform management view, where the portal offers an option to "Add new SIF network". The form exposed by the portal requests (i) a name for the new SIF network, (ii) the SIF controller's endpoint, (iii) the SIF owner's party name (i.e., a TMF Party entity behind this system), (iv) the credentials of the ESO on the SIF platform (issued by the previous step), and (v) an indicative location of the SIF instance. When the administrator inputs this information and clicks the "Add" button, the ESO portal issues a request to the ESO Backend, which in turn tries to authenticate against the SIF controller's endpoint with the provided credentials. Upon success, the network is added in ESO's database, and an entry of the SIF network is visualized on the portal.

To verify that the SIF network is functional, the administrator may proceed with a test operation, such as creating a 'test' SIF identity. This can be done via the ESO Portal's dedicated SIF management view which allows to manage identities and connections. To do so, the platform administrator is prompted to insert (i) the SIF network through which the identity is created (i.e., the SIF network added in the previous step), (ii) a related organization towards which the identity will be issued (e.g., an existing DO instance in a remote COP-PILOT domain), and (iii) an indicative name for this identity. When the user fills in this information and presses the "Add" button, the ESO Portal dispatches the request to the ESO Backend, which in turn forwards the request to the SIF controller. When the identity is created, a positive response is returned to the ESO Portal through the ESO Backend and the identity is visualized.

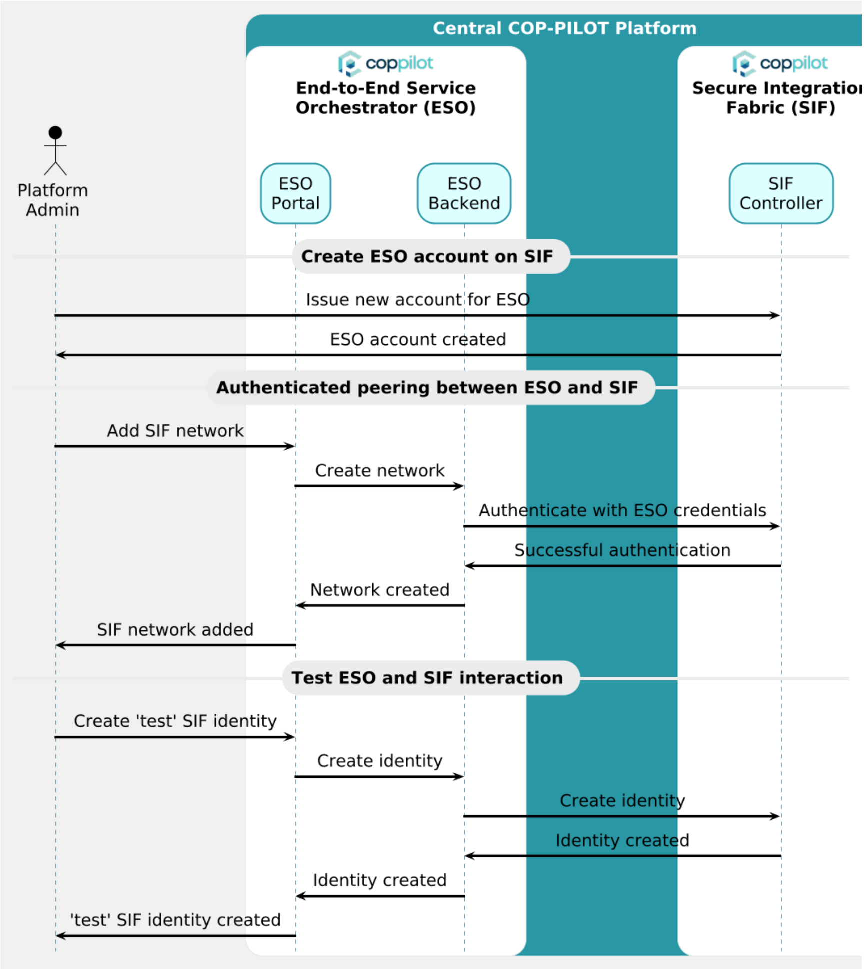


*Figure 3-1: COP-PILOT workflow for binding the ESO with a secure network fabric.*

## 3.2 PLATFORM EXPANSION TO NEW PRIVATE DOMAINS

This section outlines how infrastructure owners may exploit COP-PILOT's portals to onboard resources of their domain onto the platform. Platform expansion is a composite process that entails multiple steps; therefore, the workflow is split in parts as follows.

### 3.2.1 Registration of New Domain Owner

The process begins with the owner of the new domain requesting access to the COP-PILOT ESO as shown in Figure 3-2. To do so, he/she contacts the ESO platform administrator to request an account. The platform administrator asks relevant information from the user (i.e., contact information, affiliation with organization, registration purpose) and when the domain owner provides this information the platform administrator proceeds with a request to the ESO Portal to create a new user account. In case that the domain owner is affiliated with an organization, the ESO Portal requests additional information to associate the user with a TMF organization party before issuing the request to the ESO Backend. When the backend satisfies the user creation request, it returns a positive response to the ESO Portal, and the platform administrator shares the credentials with the domain owner.

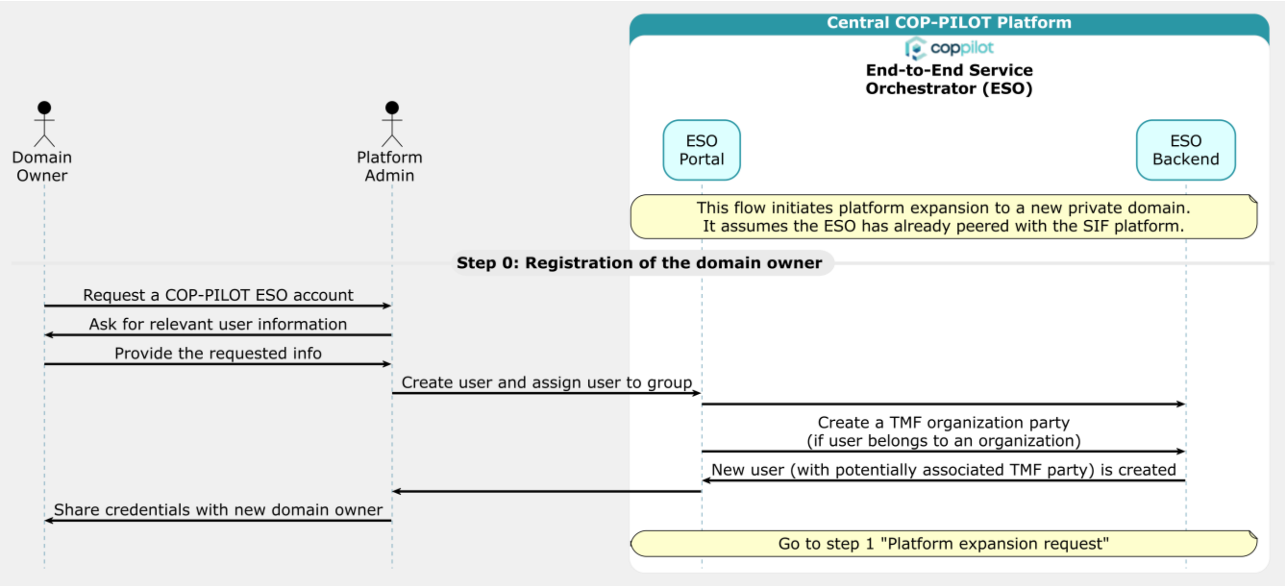


*Figure 3-2: COP-PILOT workflow for expanding the platform to new private domains – Step 0.*

## 3.2.2 Platform Expansion Request

Next, the domain owner authenticates against the ESO Portal to issue a platform expansion request as shown in Figure 3-3. First, the domain owner browses to the platform view where the ESO Portal offers the ability to create a new SIF identity for the new domain. To do so, the domain owner is prompted to insert (i) the SIF network through which the identity is created (i.e., the COP-PILOT SIF), (ii) a related organization towards which the identity will be issued (i.e., his/her own organization), and (iii) an indicative name for this identity. When the domain owner fills in this information and presses the "Add" button, the ESO Portal dispatches the request to the ESO Backend, which in turn forwards the request to the SIF controller. When the identity is created, a positive response is returned to the ESO Portal through the ESO Backend, and the identity is visualized on the ESO Portal. A checkout button allows the domain owner to download the identity (i.e., a security token) locally to proceed to the next step.

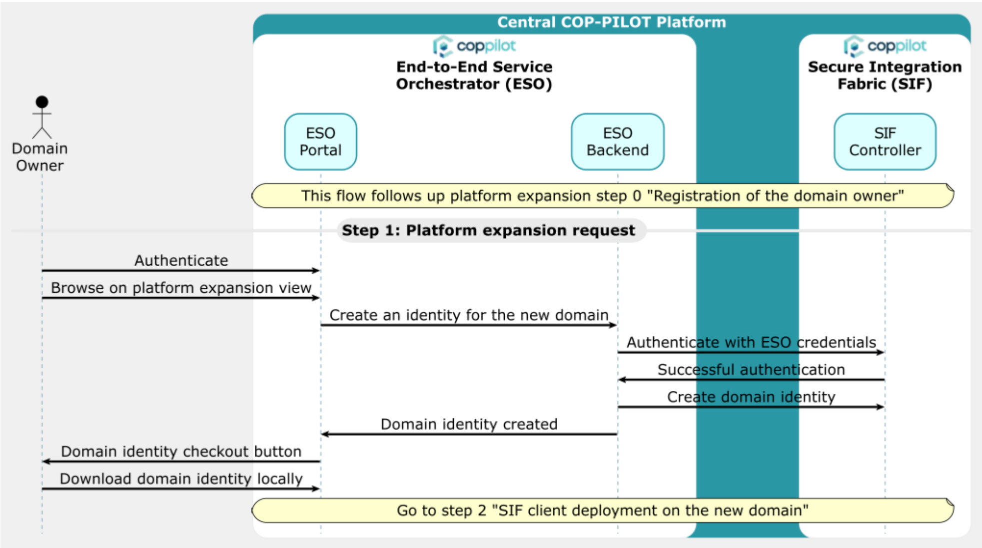


*Figure 3-3: COP-PILOT workflow for expanding the platform to new private domains – Step 1.*

### 3.2.3 SIF Client Deployment on the New Domain

At this point the domain owner has a SIF identity in his/her local system, therefore he/she can proceed with the deployment of a SIF client in the new domain as shown in Figure 3-4. The domain owner connects to a compute node in the new domain, stores the SIF identity in the node's local filesystem and issues a simple command to deploy a SIF client. When the client is provisioned, it reads the local SIF identity and uses this identity to authenticate against the COP-PILOT SIF. The COP-PILOT SIF controller gets the request from a valid identity (as this identity was issued by this controller in the previous step), thus marks this identity as "valid" and "active". At this point, the ESO Backend uses a periodic polling mechanism to retrieve active identities from the SIF, thus eventually discovers the active identity in the new domain, thus marking the domain as "connected".

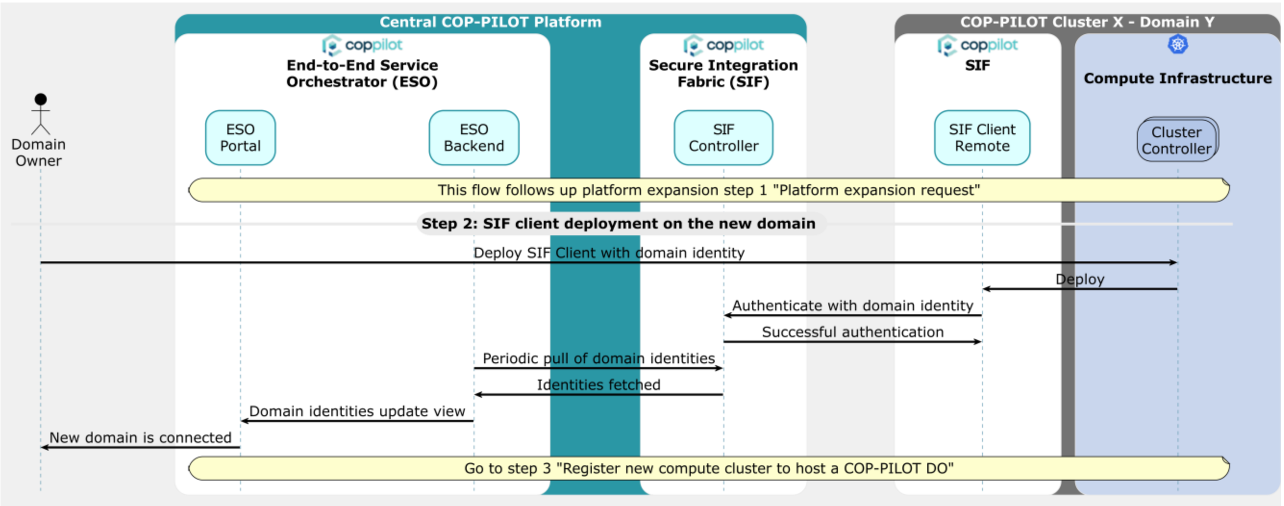


*Figure 3-4: COP-PILOT workflow for expanding the platform to new private domains – Step 2.*

### 3.2.4 Register New Compute Cluster to Host a COP-PILOT DO

Next, the domain owner should proceed with the registration of a compute cluster in the new domain under the ESO as shown in Figure 3-5. This is important for the ESO to be able to order a new dedicated COP-PILOT DO instance for the new domain. The process begins with the domain owner browsing on a dedicated view showing the new domain's registered clusters. At this point, the view shows no clusters (as the domain is new), however it offers an "Add" button to register a new cluster. The ESO Portal prompts the domain owner to fill in some relevant information (i.e., compute cluster's endpoint encoded into a cluster configuration file) to help the ESO locate this compute cluster. Once the user fills in this information, the ESO Portal dispatches the request to the ESO Backend, which in turn (i) requests the COP-PILOT SIF to create a new secure connection towards the domain's compute cluster and once this is granted (ii) authenticates and connects to this cluster. Upon successful completion, the cluster is visualized on the ESO Portal's list of compute clusters in this domain.

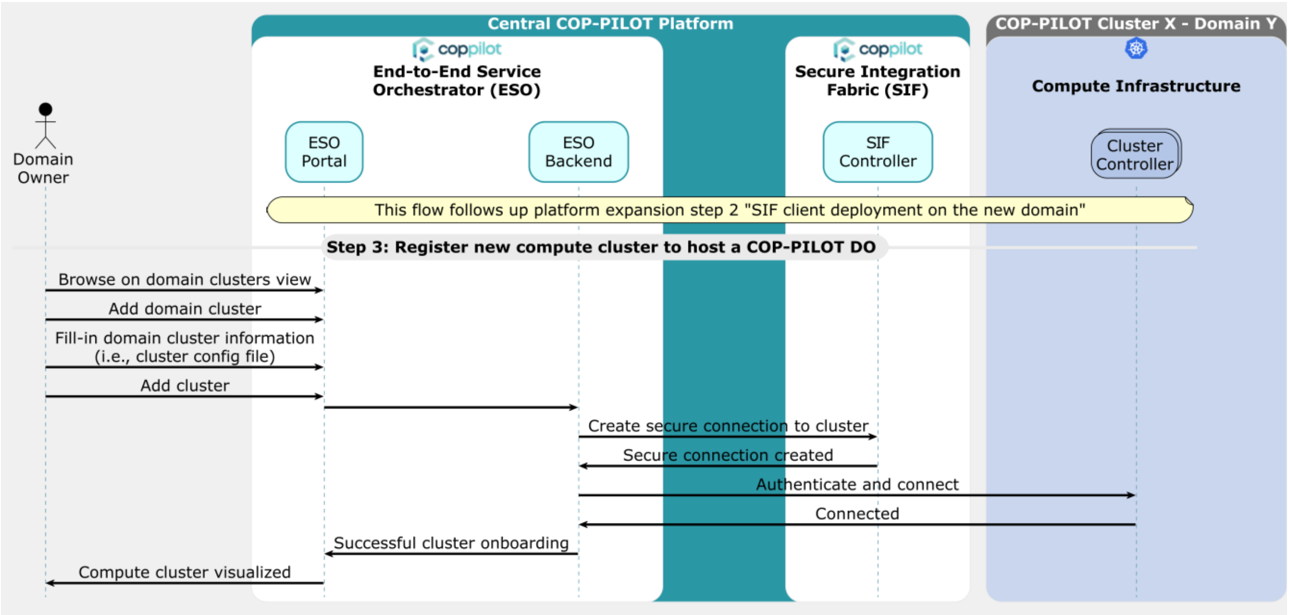


*Figure 3-5: COP-PILOT workflow for expanding the platform to new private domains – Step 3.*

### 3.2.5 Order a New COP-PILOT DO

At this point, a compute cluster is registered to the ESO for the new domain, thus the ESO can proceed with the order of a new dedicated COP-PILOT DO instance to be provisioned in the new domain as shown in Figure 3-6. To do so, the domain owner browses on the ESO's marketplace to find a relevant service specification (e.g., named as "COP-PILOT Domain Orchestrator service"). When found, the user may add this service specification into the shopping cart and configure the service order details, such as (i) an endpoint to a service registry where the DO's service package is stored, (ii) the name of this service artifact in the registry, (iii) the exact version of the DO service artifact to be deployed, (iv) the duration of the service order (i.e., until when this DO instance should be available in the domain), and (v) the compute cluster where the DO instance shall be deployed (i.e., the cluster registered to the ESO in the previous step). When this information is complete, the domain owner may click on the "Place order" button on the ESO Portal, which results in the ESO Portal submitting a service order to the ESO Backend. The ESO Backend parses the order information and understands which cluster will host the new order, thus connects to this cluster and issues a service deployment request. The cluster controller fetches the DO's service images and deploys all of them (i.e., the DO Portal and Backend services). Upon successful completion, the ESO Backend marks the service order as Completed.

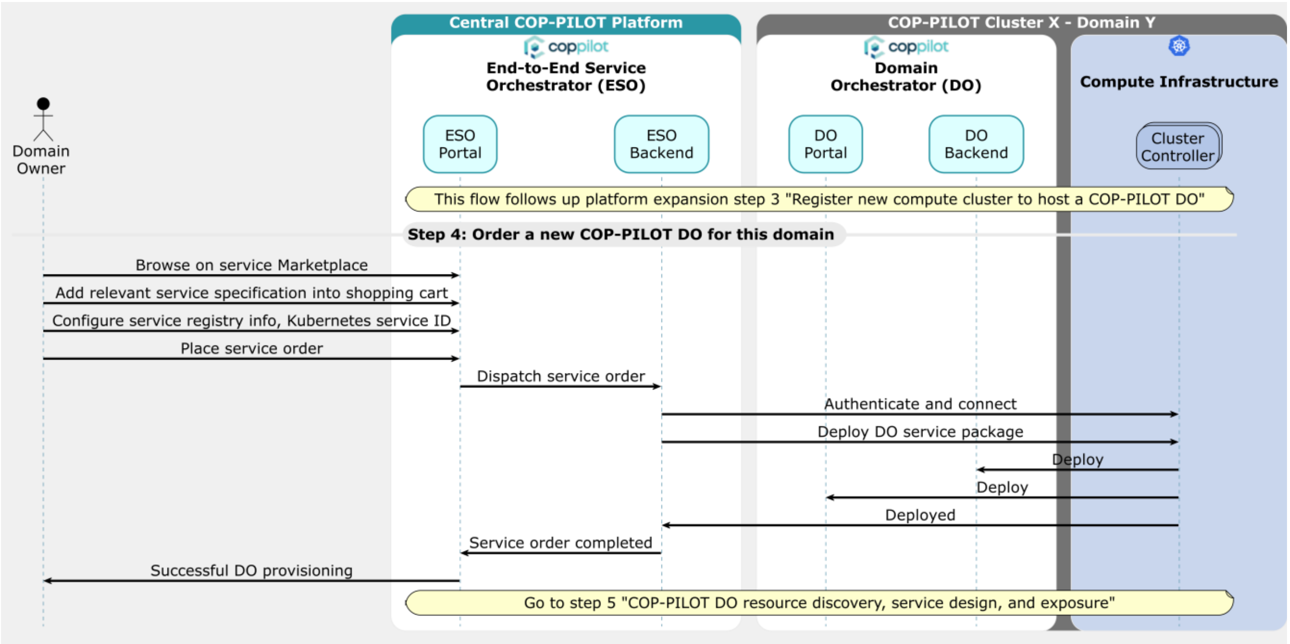


*Figure 3-6: COP-PILOT workflow for expanding the platform to new private domains – Step 4.*

### 3.2.6 COP-PILOT DO Resource Discovery, Service Design, and Exposure

The newly provisioned COP-PILOT DO within the target domain is now prepared to discover and expose available resources, as illustrated in Figure 3-7. This process follows a self-discovery mechanism, enabling the DO to identify and acknowledge infrastructure resource controllers operating within a cluster that has been previously registered as part of the service order details. More specifically, compute and network infrastructure controllers are registered as TMF Resource Specifications within the DO backend. These resources are subsequently exposed through the portal, making them available for the design and composition of services or service bundles that leverage the underlying infrastructure capabilities.

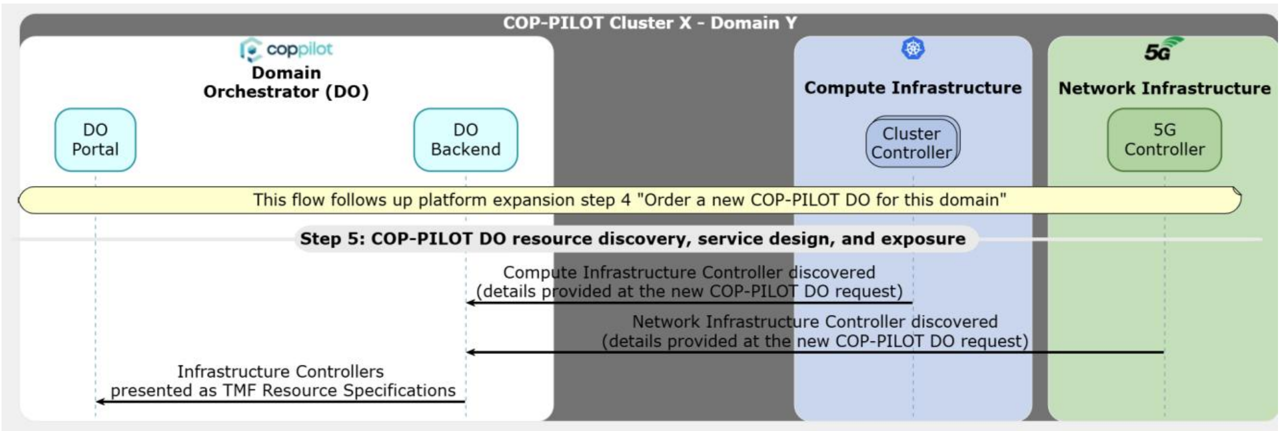


*Figure 3-7: COP-PILOT workflow for expanding the platform to new private domains – Step 5.*

## 3.3 SERVICE DESIGN

This section outlines how a COP-PILOT stakeholder may use the orchestrators to design a new service. The workflow designed in Figure 3-8 suggests a service design process using the COP-PILOT DO, however a similar process is followed at the ESO level, as both orchestrators rely on the same standardized TMF Service Management APIs. In the presented flow, domain infrastructure controllers are assumed to be onboarded as TMF Resource Specifications, which the Domain Owner first browses through the portal to identify the required capabilities. Based on these resources, the Domain Owner designs one or more resource-facing service specifications and then derives customer-facing service specifications that reference the previously defined resource-facing services. The Domain Owner further refines these customer-facing services by defining their characteristics, lifecycle rules, and catalogue organisation before saving the resulting specifications, which updates the service marketplace exposed by the DO.

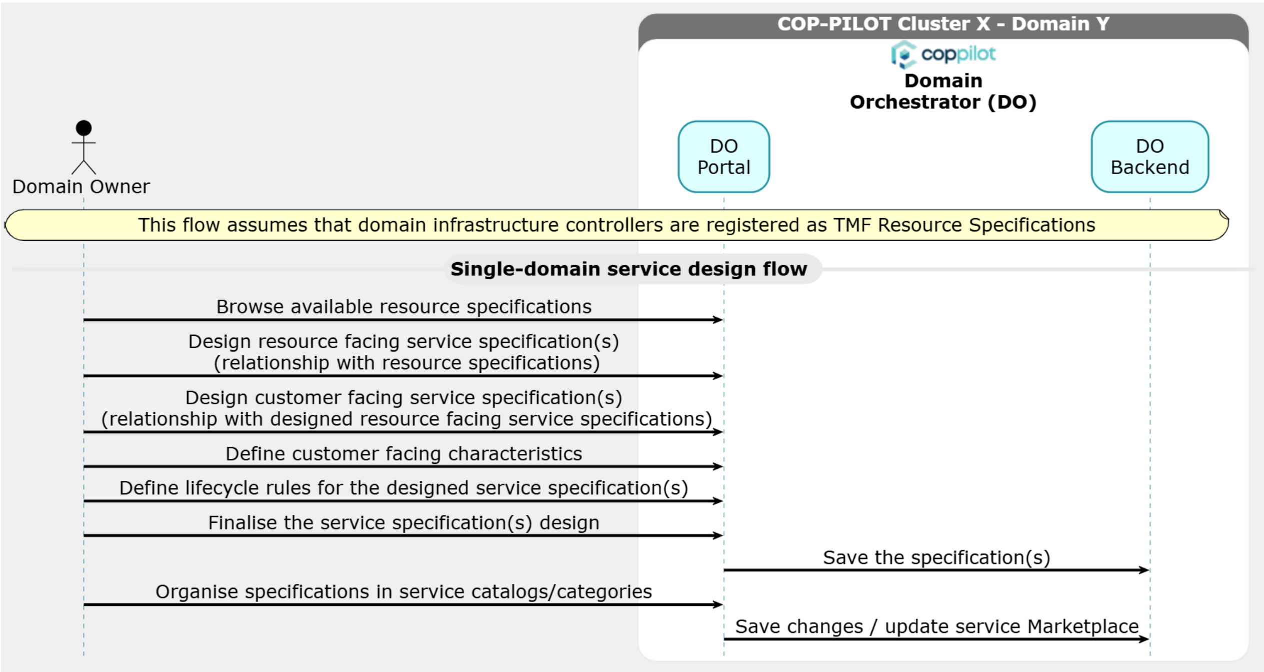


*Figure 3-8: COP-PILOT workflow for designing services via the COP-PILOT DO.*

It is important to clarify that service design activities are logically performed within the Service Catalogue, and not directly within the orchestration function. While the COP-PILOT DO Portal provides user interfaces to support service design, this functionality corresponds to the service catalogue management capabilities rather than to the runtime orchestration layer. In this context, the DO consumes service specifications stored in the catalogue as input for service fulfilment. Therefore, the design process should be understood as a catalogue-driven activity, aligned with TM Forum Service Catalogue Management principles, where service specifications are defined, versioned, and organized prior to their operational use by the orchestrator.

## 3.4 FEDERATION OF DISTRIBUTED DOMAIN SERVICE MARKETPLACES

This section outlines how the COP-PILOT ESO peers with one or more COP-PILOT DOs to synchronize their catalogues, categories, and services into an end-to-end marketplace of services. Figure 3-9 illustrates this interaction. First, the platform administrator browses on the platform view of the ESO Portal and clicks the “Add OSS” button.

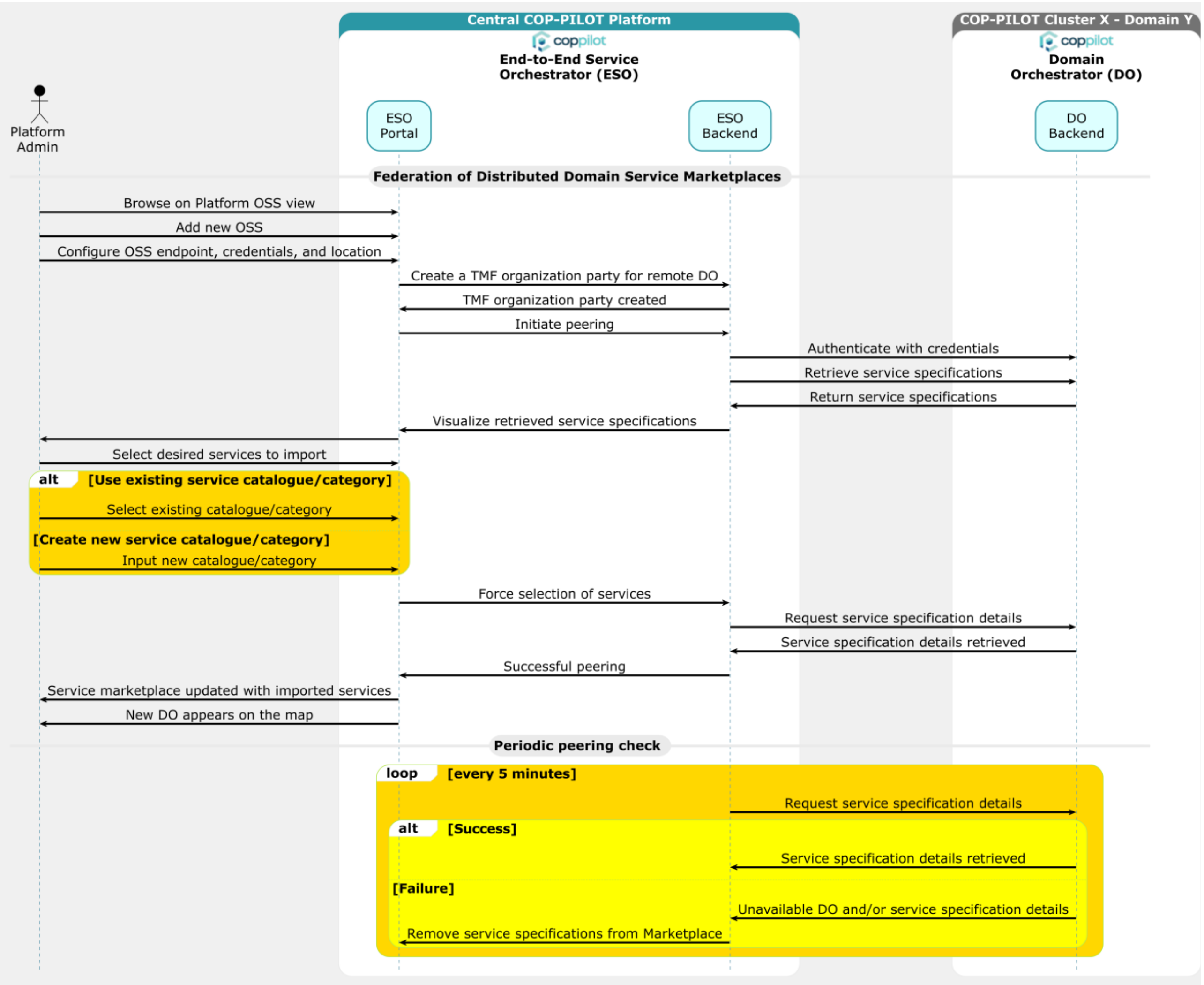


*Figure 3-9: COP-PILOT workflow for federating multiple domain services onto the ESO’s Marketplace.*

The portal loads a form which requests the platform administrator information on the remote DO instance, i.e., an endpoint (domain name or IP, Port pair) where the DO is deployed, credentials for authenticating the ESO against the remote DO, and the location of the remote DO instance (used later to visualize the DO on a map). When the platform administrator inputs this information and clicks “Add OSS”, the ESO Portal sends a request to the ESO Backend to create a new TMF organization party associated with the remote DO and when this is completed the peering is initiated by the ESO Backend towards the remote DO. The ESO Backend uses the credentials to authenticate against the remote DO and then retrieves the list of available service specifications of this DO. This list is returned to the ESO Backend and forwarded to the ESO Portal, where the latter allows the platform administrator to filter the desired service specifications from the entire list and either import these specifications in an existing service catalogue/category or create a new catalogue/category to host these service specifications (in the latter case a request is made to the ESO Backend to create catalogue/category). The platform administrator forces the filtering towards the remote DO through the ESO Backend and a filtered list of detailed service specifications is returned and stored into the ESO's database. At this point, the peering is successful and the Marketplace of the ESO visualizes the imported service specifications in the corresponding catalogue/category. At the same time, the ESO's platform view is updated with a new DO instance appearing on the map, at the location specified by the user above. Finally, the ESO schedules a background job to check for the availability of the remote DO instance by periodically retrieving the selected service specifications from the remote DO catalogue. If these specifications still exist, the job does nothing. Otherwise, if the remote DO becomes unavailable or the desired service specifications disappear from its catalogue, the background job removes these service specifications from the ESO Marketplace, to avoid failures while ordering these services in the future.

The retrieval and synchronization of service specifications between the ESO and the Domain Orchestrators should be implemented using standardized APIs. This ensures interoperability across domains and guarantees that service specifications, categories, and catalogue structures are consistently managed and exposed in a standardized way across the federated ecosystem.

## 3.5 END USER SERVICE ORDERING VIA THE ORCHESTRATORS

This section outlines how the synergy between the COP-PILOT ESO and one or more COP-PILOT DOs allows stakeholders to order a level 2[1] user service at a given vertical sector. Figure 3-10 visualizes this process for a single domain service. The flow assumes that a service was originally designed at the domain-level orchestrator as per the workflow described in Section 3.3. It also assumes that after the service design, peering between the ESO and the specific DO instance was performed as per the workflow in Section 3.4, to import this service into the ESO's marketplace. Finally, the workflow assumes that a compute cluster is already ordered in advance to this service, thus the compute cluster is already exposed via the COP-PILOT SIF.

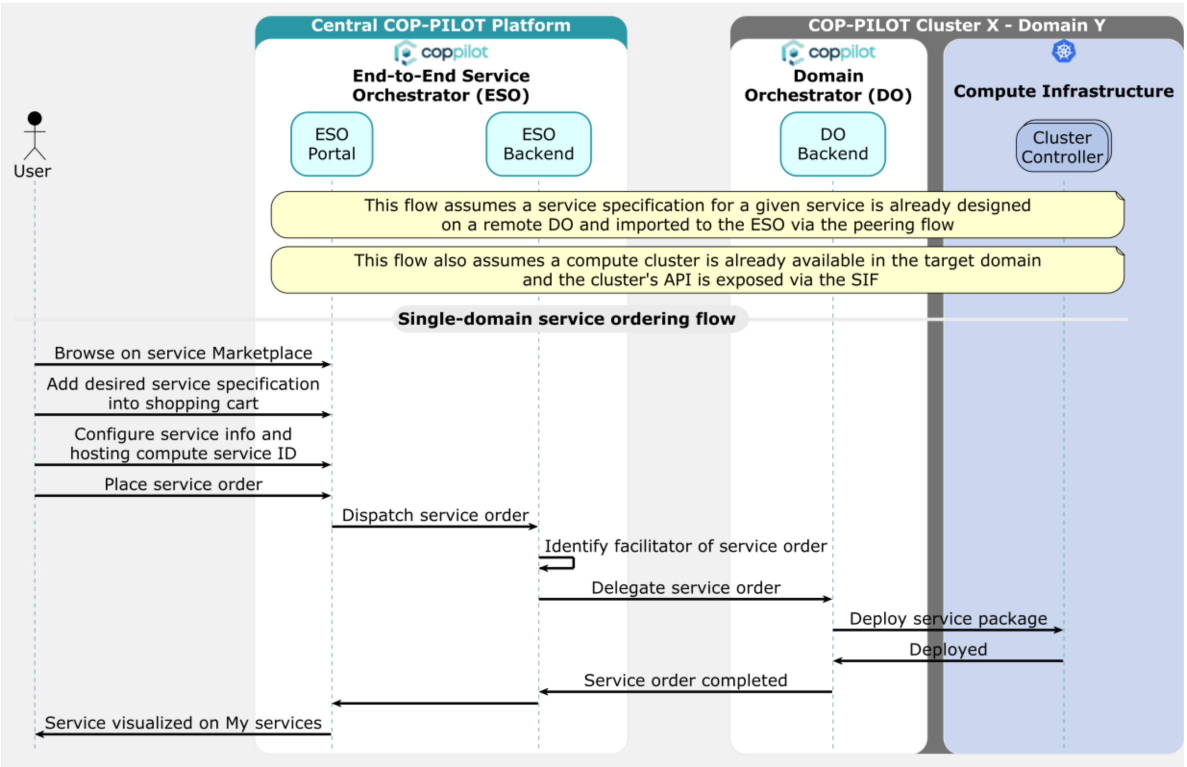


*Figure 3-10: COP-PILOT workflow for ordering end user service through the COP-PILOT Orchestrators.*

Given the above assumptions, the flow begins with the user browsing on the ESO's marketplace to find the desired service specification. When found, the user may add this service specification into the shopping cart and configure the service order details, such as (i) an endpoint to a service registry where the service package is stored, (ii) the name of this service artifact in the registry, (iii) the exact version of the service artifact to be deployed, (iv) the duration of the service order (i.e., until when this service instance should be available in the domain), and (v) the compute cluster where the service instance shall be deployed. When this information is complete, the user may click on the "Place order" button on the ESO Portal, which results in the ESO Portal submitting a service order to the ESO Backend. The ESO Backend parses the order information and understands which domain shall host the new order, thus selecting the right DO instance to facilitate this request. That said, a corresponding service order is dispatched at the specific DO, which communicates with the target cluster to deploy the desired services. Upon successful completion, the DO Backend marks

[1] Level 2 user corresponds to an operations or administrator user (e.g., platform operators or domain administrators) rather than end users. These users interact directly with service-level abstractions exposed by the orchestrators.

the service order as Completed, hence the corresponding ESO service order transitions to the same state and the service becomes available on "My services" view of the ESO Portal.

## 3.6 SERVICE RECONFIGURATION VIA THE ORCHESTRATORS

This section outlines how the synergy between the COP-PILOT ESO and one or more COP-PILOT DOs allows level 2 users to change the runtime characteristics of an already deployed service. Figure 3-11 depicts the service reconfiguration workflow. This flow assumes that a single domain end user service was previously ordered as per the service ordering workflow described in Section 3.5 and the service is still active. Given this assumption, the user browses on the ESO Portal "My Services" view to find the desired service and modify the customer-facing service characteristics through the dedicated service cart. When this is done, the level 2 user may press the update button, which results in the ESO Portal submitting a service update request to the ESO Backend. The backend identifies the facilitator DO of this service and delegates the service update request towards that DO. In turn, the DO interacts with the cluster that hosts the service and modifies the requested characteristics. The response is propagated back to the ESO and the ESO Portal visualizes the updated service to the user.

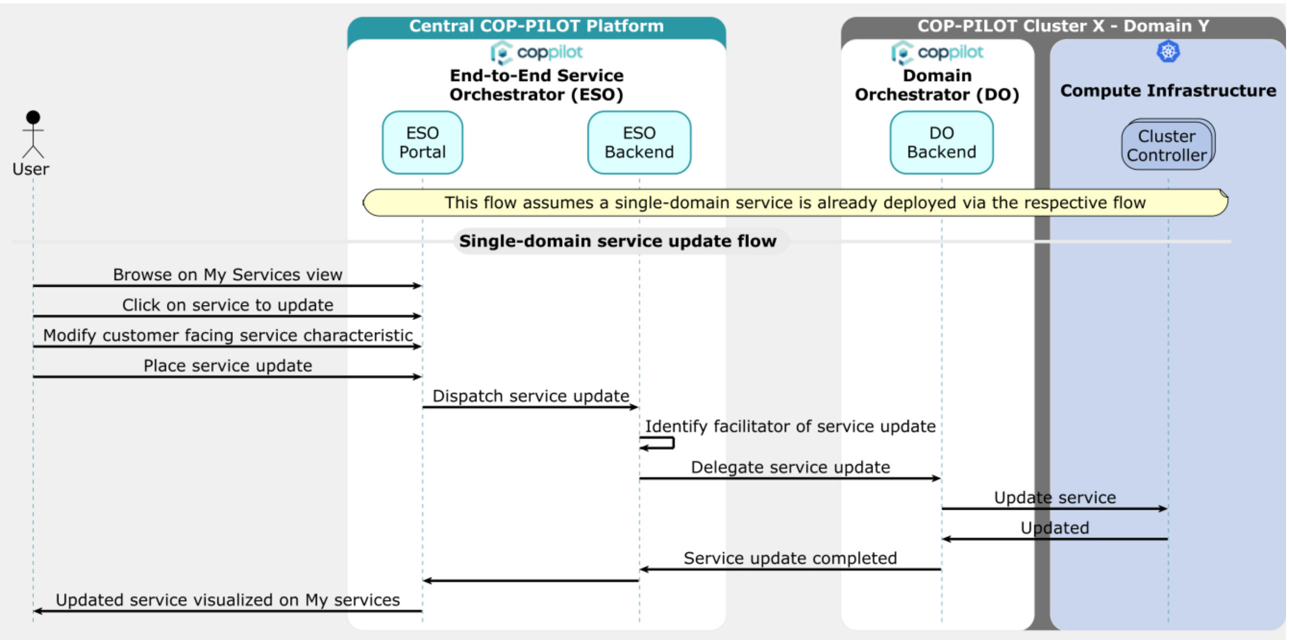


*Figure 3-11: COP-PILOT workflow for reconfiguring running services through the COP-PILOT Orchestrators.*

## 3.7 PRODUCT DESIGN AND EXPOSURE VIA THE BUSINESS PORTAL

This section guides the COP-PILOT stakeholders through the steps required to design a new product on the COP-PILOT Business Management Portal. Figure 3-12 visualizes this process.

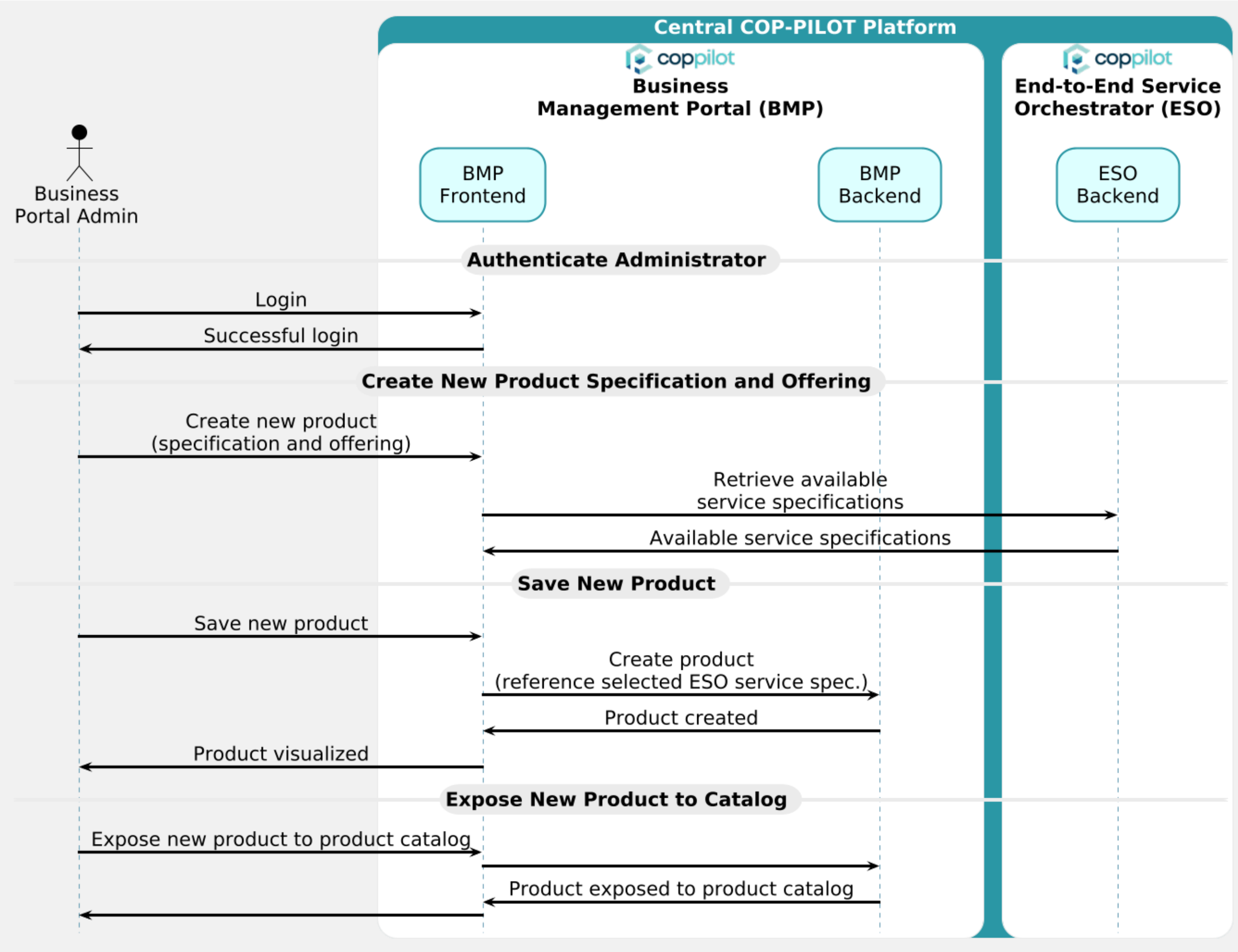


*Figure 3-12: COP-PILOT workflow for designing products via the Business Management Portal.*

This workflow describes how a Business Portal Administrator creates a new product in the COP-PILOT Business Management Portal (BMP) and subsequently publishes it in the product catalogue, making it available for discovery and ordering by users and applications.

The process begins when the administrator accesses the Business Management Portal. Before any management functions can be used, the administrator must authenticate through the portal's authentication mechanism. The submitted credentials are validated by the authentication entity associated with the BMP. Once the authentication process is successfully completed, the administrator is granted access to the product management capabilities of the portal.

After logging in, the administrator initiates the creation of a new product. To support this activity, the BMP retrieves the available Service Specifications from the End-to-End Service Orchestrator (ESO). These service specifications represent the technical services that can be delivered by the platform and serve as the foundation for business-facing products. The retrieved specifications are presented to the administrator through the BMP user interface, allowing the selection of the appropriate technical service on which the new product will be based. In this way, business-level products are directly associated with the underlying service capabilities managed by the ESO.

Once a suitable service specification has been selected, the administrator defines the business characteristics of the product, such as its name, description, and commercial attributes. When the product definition is complete, the administrator saves the new product. The BMP Front-End forwards the request to the BMP Backend, including a reference to the selected ESO Service Specification. The Backend then creates and persists the product definition within the portal. After the operation is successfully completed, a confirmation is returned to the administrator indicating that the product has been created and stored. At this stage, the product exists within the Business Management Portal but remains unavailable to end users because it has not yet been published in the product catalogue.

To make the product available for consumption, the administrator proceeds with the publication step. Through the BMP user interface, the administrator requests that the product be exposed in the catalogue. The BMP Front-End forwards this request to the Backend, which updates the product's status and marks it as publicly available. Following successful publication, the Backend returns a confirmation to the Front-End, which notifies the administrator that the product has been exposed successfully.

At the conclusion of this workflow, a new business-facing product has been created, linked to an existing ESO Service Specification, and stored within the Business Management Portal. The product is now published in the product catalogue and can be discovered, viewed, and ordered by authorized users. This workflow establishes the connection between business-oriented product management in the BMP and the technical service definitions maintained by the COP-PILOT End-to-End Service Orchestrator.

## 3.8 PRODUCT ORDERING VIA THE BUSINESS PORTAL

This section guides the COP-PILOT stakeholders through the steps required to order a product through the COP-PILOT Business Management Portal and the underlying service orchestrators. Figure 3-13 illustrates the steps for realizing product orders.

This workflow describes how a user discovers and orders a product through the COP-PILOT Business Management Portal (BMP), and how the resulting service request is orchestrated and monitored through the End-to-End Service Orchestrator (ESO) until fulfilment is complete.

The process begins when the user accesses the Business Management Portal. Before any marketplace or ordering functionality becomes available, the user must authenticate with the platform. The authentication request is processed by the BMP's authentication entity, which validates the user's credentials and grants access upon successful verification. Once authenticated, the user can access the product marketplace and all associated ordering functions.

After logging in, the user navigates to the product marketplace to explore the services and offerings available within the platform. To populate the marketplace, the BMP Front-End requests the list of published products from the BMP Backend. The Backend retrieves all products that have been created and published by platform administrators and returns them to the Front-End, which presents them to the user through the marketplace interface. This allows users to browse, compare, and select products that are available for consumption.

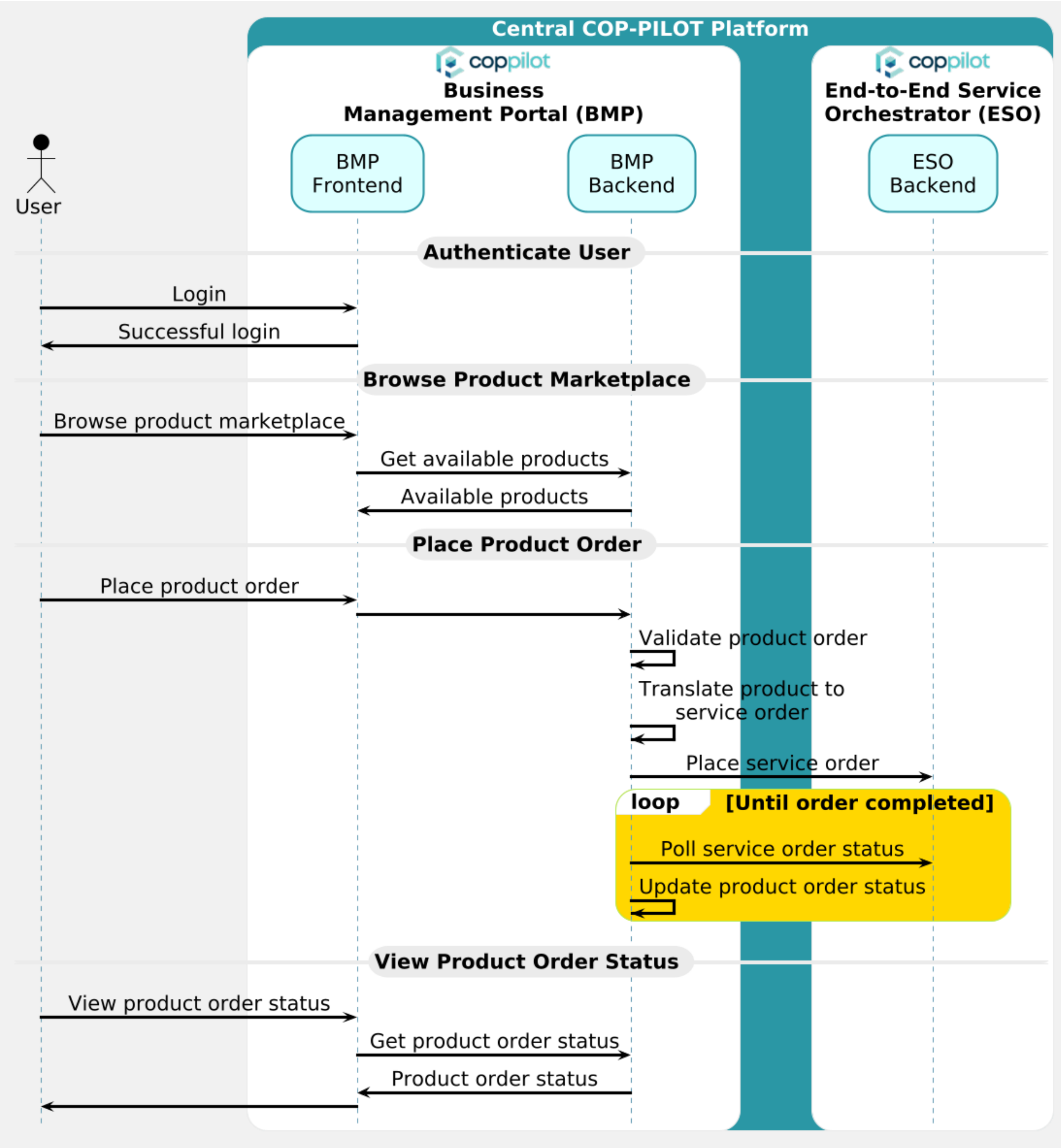


*Figure 3-13: COP-PILOT workflow for ordering products via the Business Management Portal.*

When the user identifies a suitable offering, the ordering process begins. The selected product is submitted through the BMP Front-End, which forwards the product order request to the BMP Backend. Before initiating the fulfilment process, the product order undergoes a validation phase within the BMP. This includes eligibility checks, feasibility validation against current resource availability, and enforcement of business and technical policies. Once the product order is validated and accepted, the BMP translates the product offering into corresponding service orders that can be processed by the End-to-End Service Orchestrator (ESO).

The ESO initiates the fulfilment process and coordinates the deployment and configuration activities required to deliver the requested service. Depending on the nature of the service, the ESO may interact with one or more Domain Orchestrators (DOs) to provision resources, deploy applications, or configure infrastructure across the distributed COP-PILOT environment. This translation mechanism separates business-facing products from the underlying technical services and infrastructure, allowing users to interact with the platform at a business level without being exposed to implementation details.

Once the service order has been submitted, the Business Management Portal continuously monitors its progress. The BMP Backend periodically queries the ESO for updates regarding the status of the service order. Each status update received from the ESO is reflected in the corresponding product order maintained by the BMP. This synchronization process continues throughout the entire fulfilment lifecycle until the service order reaches a final state, such as Completed, Failed, or Cancelled. As a result, the product order status displayed to users always reflects the current execution state of the underlying service order.

At any point during the fulfilment process, users may request information about the status of their orders. When a status request is submitted through the BMP Front-End, the request is forwarded to the BMP Backend, which retrieves the latest information available for the corresponding product order. The status is then returned to the Front-End and presented to the user. Typical order states include Submitted, Accepted, In Progress, Provisioning, Completed, Failed, and Cancelled, providing users with a clear view of how their request is progressing through the platform.

At the conclusion of the workflow, the user has successfully ordered a business-level product through the Business Management Portal. The BMP has translated this request into a service order and delegated its execution to the End-to-End Service Orchestrator. Throughout the fulfilment process, the BMP continuously synchronizes product order information with the underlying service order status, enabling users to monitor progress through a unified business-oriented interface. This workflow establishes a clear separation between business-level product management and ordering within the Business Management Portal and the technical orchestration and provisioning functions performed by the COP-PILOT End-to-End Service Orchestrator.

## 3.9 LLM-ASSISTED PRODUCT ORDERING

This section illustrates how COP-PILOT facilitates product ordering via an LLM assistant that stands between the stakeholders and the platform to simplify product ordering (see Figure 3-14). The LLM-based assistant operates as an intelligent interaction layer on top of the Business Management Portal. It does not replace existing catalogue or ordering mechanisms but rather orchestrates them by interpreting user intents and translating them into structured product queries and configurations. Internally, the assistant interacts with the product catalogue, ordering, and validation APIs to ensure that all actions remain consistent with platform rules and data models.

This workflow describes how users interact with the COP-PILOT Business Management Portal (BMP) through an LLM-based Ordering Assistant to discover, configure, validate, order, and monitor products using natural language. By combining conversational AI, automated validation, and service orchestration, the workflow simplifies access to complex services while hiding the technical details of the underlying platform.

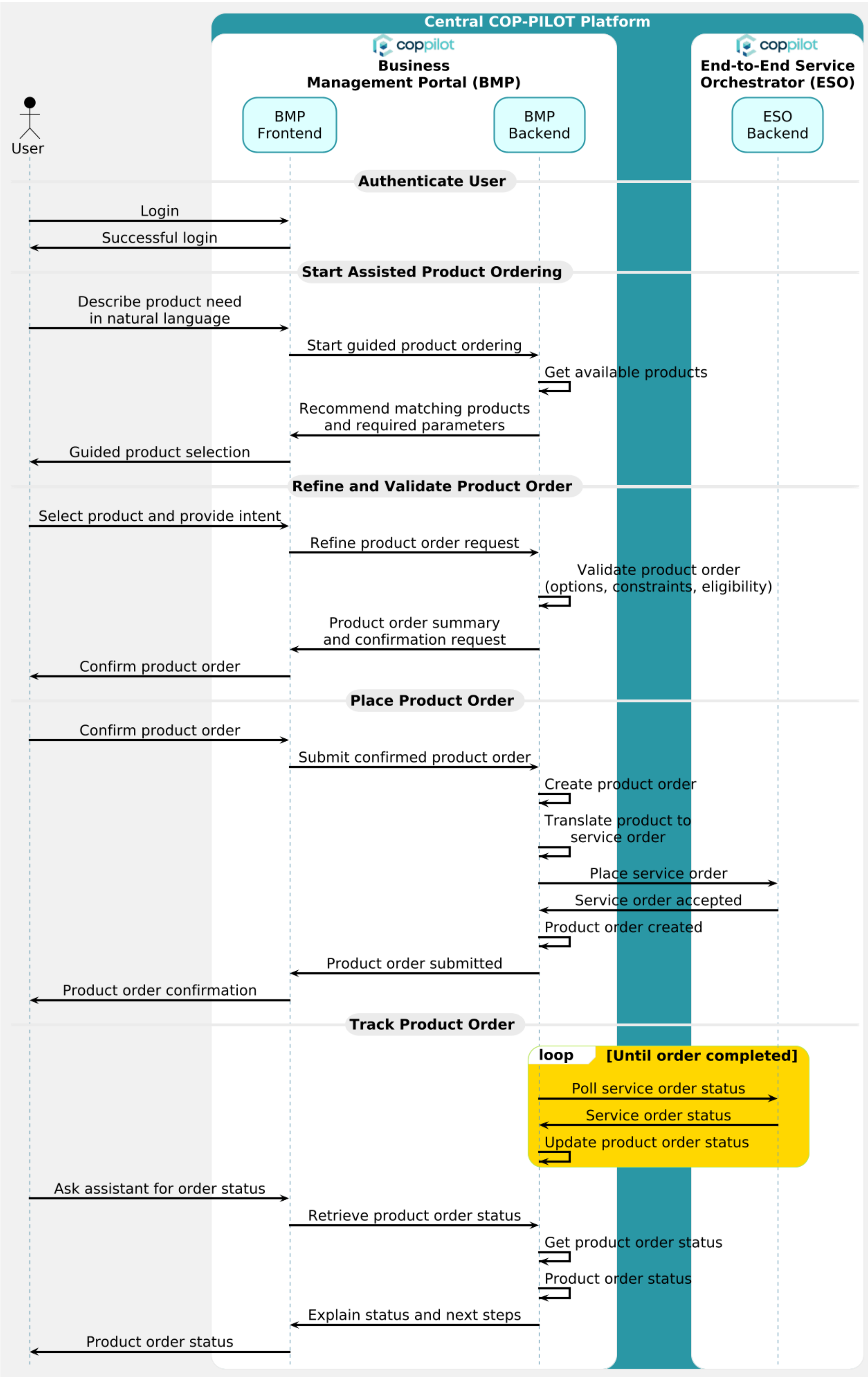


*Figure 3-14: COP-PILOT workflow for ordering products via Business Management Portal's LLM assistant.*

The process begins when the user accesses the Business Management Portal and authenticates with the platform. The authentication request is processed by the BMP authentication entity, which validates the user's credentials and grants access to both the portal's standard functionality and the AI-assisted ordering capabilities. Once authenticated, the user can interact with the platform through a conversational interface rather than navigating traditional catalogues and configuration screens.

Instead of manually browsing available offerings, the user begins by describing the desired product or service in natural language through the BMP Front-End. The Front-End initiates a guided ordering session with the LLM-based Ordering Assistant, which retrieves the available product offerings from the BMP Backend. Using information from the product catalogue together with the user's requirements, the assistant analyses the request, identifies suitable products, and determines any additional information or configuration parameters that may be required. Based on this analysis, the assistant presents one or more recommended products and guides the user through the selection process.

As the conversation progresses, the user selects a proposed product and provides any additional intent, requirements, or configuration preferences requested by the assistant. The Ordering Assistant continuously refines the product request by incorporating the information provided during the interaction. This conversational approach significantly reduces the complexity associated with navigating technical product catalogues and configuring sophisticated services, allowing users to focus on their business objectives rather than platform-specific implementation details.

Before the order can be submitted, the assistant validates the proposed configuration. To do so, it forwards the refined product request to the BMP Backend, which evaluates it against the available product options, product constraints, eligibility rules, mandatory parameters, and applicable business policies. Once the validation process is complete, the results are returned to the assistant. The assistant then generates a human-readable summary of the proposed order, clearly presenting the selected product, configured parameters, intended product characteristics, and any assumptions or constraints that may influence the final deployment. This summary is presented to the user as part of a confirmation request, ensuring that the proposed configuration is both technically valid and aligned with the user's expectations.

After reviewing the generated summary, the user explicitly confirms the order. At this point, the system possesses both a validated configuration and the user's approval to proceed. The BMP Front-End forwards the confirmed request to the LLM-based Ordering Assistant, which submits the finalized order to the BMP Backend. The Backend creates a corresponding product order and then translates it into a technical service order suitable for execution by the platform orchestration layer. This service order is submitted to the End-to-End Service Orchestrator (ESO), which validates and accepts the request before initiating the necessary provisioning and orchestration activities. Once the order has been successfully registered, a product order reference is generated and returned through the BMP Backend to the Ordering Assistant. The assistant informs the user that the order has been successfully submitted and provides the corresponding order reference for future tracking. This stage establishes the relationship between the business-facing product order managed by the BMP and the underlying service order executed by the ESO.

Following submission, the Business Management Portal continuously monitors the progress of the fulfilment process. The BMP Backend periodically queries the ESO to obtain the latest service order status. Each update received from the ESO is reflected in the corresponding product order maintained within the BMP, ensuring that business-level order information remains synchronized with the underlying technical execution state. This monitoring process continues until the service order reaches a final state, such as Completed, Failed, or Cancelled.

At any point during the fulfilment lifecycle, the user can request updates through the conversational assistant. When a status inquiry is made, the BMP Front-End forwards the request to the Ordering Assistant, which retrieves the latest product order information from the BMP Backend. Rather than simply displaying status codes, the assistant interprets the available information and provides a contextual explanation of the current order state. This explanation may include the progress achieved so far, activities currently in progress, pending steps, and expected next actions. By presenting fulfilment information in natural language, the assistant helps users understand the operational status of their requests without requiring knowledge of the underlying orchestration processes.

At the conclusion of the workflow, users can discover and order products through natural language interactions rather than manually navigating complex product catalogues. The LLM-based Ordering Assistant recommends suitable offerings, guides users through the configuration process, and automatically validates requests against both business and technical constraints. Approved product orders are translated into service orders and executed by the End-to-End Service Orchestrator, while the Business Management Portal continuously synchronizes product and service order states throughout the fulfilment lifecycle. Through conversational status tracking and contextual explanations, users gain clear visibility into fulfilment progress and next steps. The result is a unified user experience that combines AI-assisted product discovery and configuration, business-level service ordering, and automated service orchestration, significantly simplifying access to complex multi-domain services within the COP-PILOT platform.

# 4 SERVICE MODEL & INFRASTRUCTURE CONFIGURATION PER UC

Deliverable D2.1 details COP-PILOT's ecosystem definition and requirements, addressing clusters operating across diverse sectors. Annex 1 of D2.1 initially (M10) outlined the service value requirements that use cases within these clusters can deliver to communities. The same annex has now been extended (M18) – in the context of D2.2 – with additional context detailing the engineering view of the service models, which articulate the values to be delivered to the community alongside the technical foundations required to implement each use case.

## 4.1 COP-PILOT SERVICE MODELLING BUSINESS VIEW SUMMARY

The service model's business view detailed in Annex 1 of D2.1 establishes the conceptual and commercial foundation for understanding how COP-PILOT creates value across European industries. Central to this is the COP-PILOT Service Value Model, which frames every deployment around two interlocking dimensions: Value and Engineering. Value is created at the intersection of Suppliers, the technology and service providers who bring capabilities to the platform, and Customers, the industries and organisations that consume and benefit from those capabilities. Engineering, in turn, is realised through the deliberate configuration of technical infrastructure that enables the service to function reliably and at scale.

This model reflects a clear commercial approach: sustainable impact requires both a compelling value proposition for the customer and a technically sound delivery mechanism on the engineering side. A well-configured system with no clear customer benefit fails to drive adoption, while a strong value proposition built on weak infrastructure cannot deliver at the scale and reliability that industrial environments demand.

This model is further explored through use cases in this annex, such as a 5G-connected factory managing 1,000 IoT endpoints, or an AI-driven anomaly detection service for robotic systems, showing how the same model structure applies consistently regardless of the sector or complexity of the use case. In each instance, the Supplier, Customer, COP-PILOT Service, Configuration, and Infrastructure elements are clearly defined, making the value creation logic transparent and replicable.

This repeatable structure is then applied across COP-PILOT's five industry clusters, spanning mining, smart cities, agritech, energy grids, and viticulture. Across all of them, the service model functions as a common language for articulating where value lives, who captures it, and what technical conditions must be met to sustain it. This foundation sets the stage for the detailed service requirements that follow.

## 4.2 COP-PILOT SERVICE MODELLING ENGINEERING VIEW SUMMARY

Having explored the Value side of the COP-PILOT Service Model which defines what each service does, for whom, and why it matters, the focus now shifts to the Engineering side: how each service is built, deployed, and configured in practice.

For every use case, the project moves beyond service definitions and articulates the concrete infrastructure underpinning each deployment and the specific configuration actions required to bring it to life. This implies identifying a minimum of five infrastructure blocks per use case and no more than ten configuration steps, keeping the exercise practical and actionable rather than exhaustive.

To structure this engineering view, clusters first identify their most mature service, then study how it is currently built, covering components, packaging technology, endpoints, data sources, compute and networking requirements, and current deployment status. This information would be translated into explicit infrastructure diagrams and ordered configuration action lists, before being mapped against the COP-PILOT Service Library to identify which existing marketplace elements each use case can draw on and where gaps remain.

## 4.3 SERVICE LIBRARY

The extended D2.1 Annex 1 presents the engineering layer of the COP-PILOT service model across all clusters, translating the value definitions and service requirements established in D2.1 into concrete infrastructure and configuration specifications. While the initial D2.1 Annex 1 defined what each service does and for whom, its updated version defines how each service is built, deployed, and configured, the precise layer where engineering decisions become running systems.

The D2.1 Annex 1 is structured around four cluster-level engineering views, each following a consistent methodology: identifying the most mature use case as the starting point, mapping its infrastructure components (hardware, platform, and compute layers), specifying deployment strategies, and defining a bounded set of configuration actions. A gap analysis against the COP-PILOT service marketplace is performed for each use case, distinguishing between service library elements that are already ready and those that remain to be designed.

### 4.3.1 COP-PILOT Service Marketplace in P1 (M1-M18)

Figure 4-1 illustrates the current view of the COP-PILOT Service Marketplace in the first period of the project (i.e., up to M18).

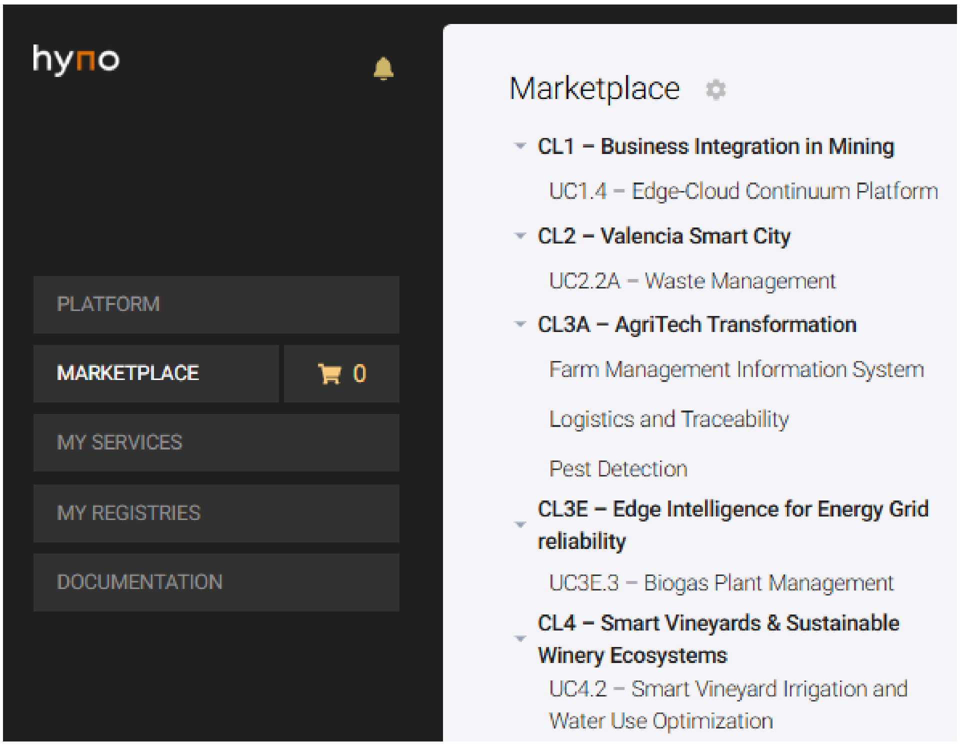


*Figure 4-1: The COP-PILOT Service Marketplace catalogues and categories in P1 (M18).*

This marketplace is offered by the COP-PILOT ESO and is organized in service catalogues based on the TMF 633 Service Catalogue Management API v4.0.0 [4]. Each service catalogue depicted in Figure 4-1 corresponds to a COP-PILOT Cluster, therefore this is the reason that 5 service catalogues are depicted in total. Each service catalogue contains one or more service categories, where the COP-PILOT Clusters place their service specifications. Both service categories and services specifications are defined by the same TMF 633 API [4].

Figure 4-2 and Figure 4-3 illustrate the COP-PILOT service specifications that are designed during the first period of the project to accommodate use cases across the 5 COP-PILOT Clusters and their piloting activities. Fourteen (14) service specifications are designed in total, with the share per Cluster being analysed below:

- 1 service specification (ColonyOS Blueprint Reconciler) corresponds to Cluster 1,
- 2 service specifications (Waste Collection Alerting and Smart City Event Management) correspond to Cluster 2,
- 3 service specifications (AgroApps 360, AI Pest Detection, PowerFleet) correspond to Cluster 3A,
- 3 service specifications (K8saaS OCM, Biogas Sensing, Biogas Visualization) correspond to Cluster 3E,
- 2 service specifications (Aquaview and theTorre) correspond to Cluster 4.
- The remaining 3 service specifications are generic service specifications provided by the COP-PILOT ESO.

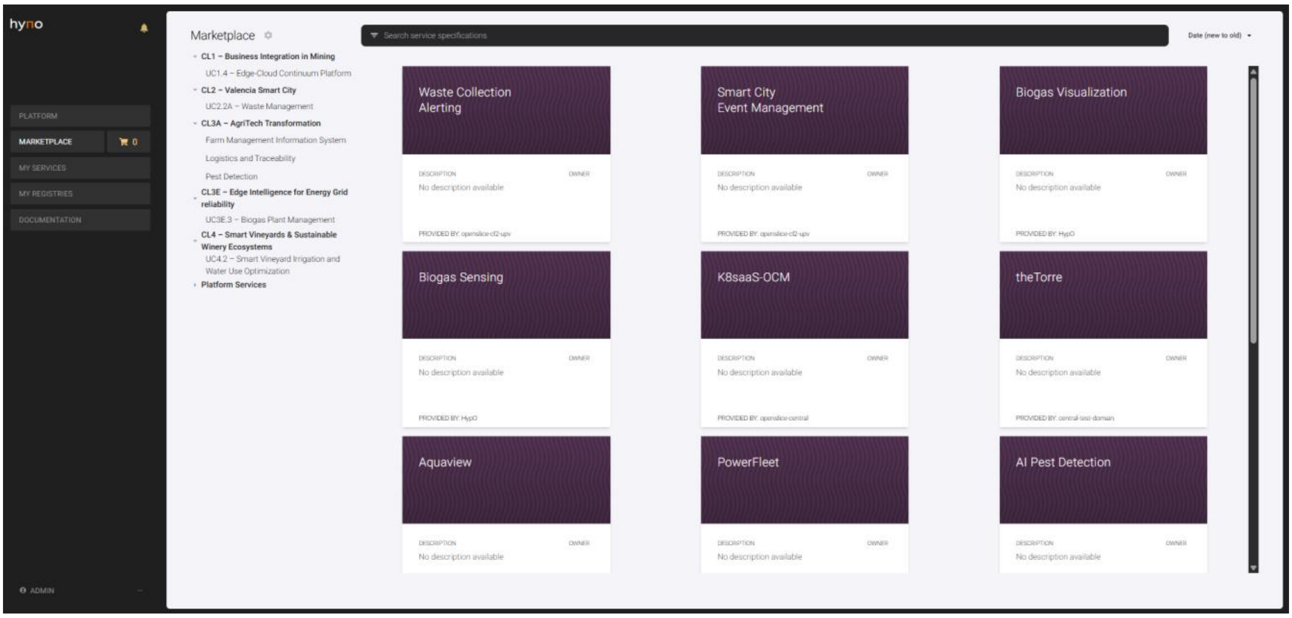


*Figure 4-2: The COP-PILOT Service Marketplace service specifications in P1 (M18) – Part #1.*

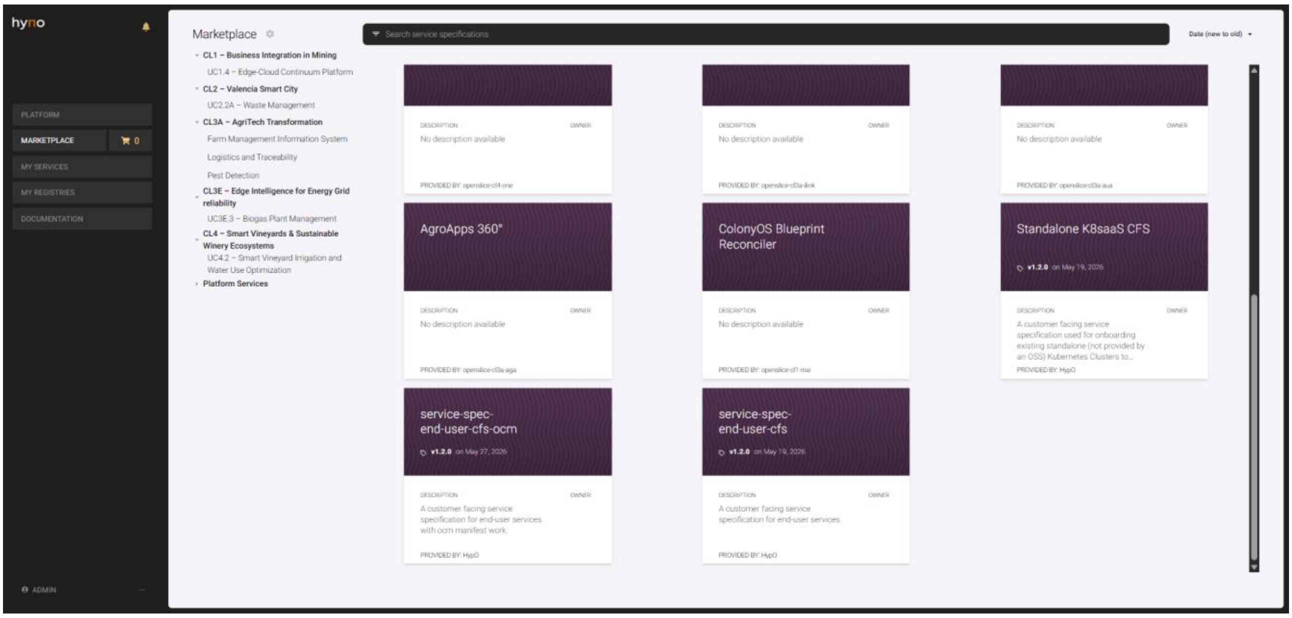


*Figure 4-3: The COP-PILOT Service Marketplace service specifications in P1 (M18) – Part #2.*

The COP-PILOT Service Marketplace reached this state by M13 (January 2026). A proof-of-concept demonstration video that describes how COP-PILOT created this marketplace is available on the COP-PILOT YouTube channel (video is here), while the same video demonstrates the COP-PILOT Service Marketplace at EuCNC 2026 in Malaga, Spain, where COP-PILOT partners participated at the ETSI Booth promoting COP-PILOT's standardized open-source platform.

### 4.3.2 COP-PILOT Service Marketplace Expected by M36

Table 4-1 illustrates the expected view of the COP-PILOT Service Marketplace by the end of the project, classifying services across 5 service catalogues (one per COP-PILOT Cluster) and multiple service categories.

*Table 4-1: Expected services in the COP-PILOT Service Marketplace by M36.*

| COP-PILOT Cluster Service Catalogue | Service Categories in COP-PILOT Cluster Service Catalogue | List of COP-PILOT Service Specifications per Category | Expected date to appear in the Marketplace |
|---|---|---|---|
| CL1 – Business Integration in Mining | UC1.1 – Seismic processing | - BEMIS Multi-sensing Seismic Processing | Dec. 2026 |
| | | - BEMIS Auto-scaling Seismic Processing | Dec. 2026 |
| | | - BEMIS Distributed Seismic Processing | Dec. 2026 |
| | UC1.2 - Asset tracking | - EvoMining Asset tracking | Dec. 2026 |
| | UC1.3 - Condition monitoring | - Splaice Onboarding Automation | Dec. 2026 |
| | | - Conveyor Onboarding Automation | Dec. 2026 |
| | | - Conveyor Decision Support UI Update Automation | Dec. 2026 |

| | | | |
|---|---|---|---|
| | UC1.4 – Edge-Cloud Continuum Platform | - ColonyOS Blueprint Reconciler | Feb. 2026 (Done) |
| CL2 – Valencia Smart City | Cross-UC Smart City Stack | - Smart City Stack<br>+ Core<br>+ Event Manager<br>+ Dashboards<br>+ Open Data<br>+ AI | June 2026 |
| | UC2.1A – 5G-Connected Radars for Traffic Classification and Vehicle Counting | - Smart City Stack<br>- Alert Notification<br>- YOLO-based Verification | June. 2026<br>July 2026<br>Sep. 2026 |
| | UC2.1B – Flood Warning and Mitigation via Radar Sensing | - Smart City Stack<br>- Weather Forecast Monitoring<br>- Frequency Modifier | June. 2026<br>Oct. 2026<br><br>Oct. 2026 |
| | UC2.1 + UC2.2 + UC2.4 –IoT-Driven Smart Building and Environmental Management | - Smart City Stack<br>- Alert Notification | June 2026<br>July 2026 |
| | UC2.3 – Maritime Traffic Monitoring and Berthing Assistance | - Smart City Stack<br>- Maritime Traffic<br>- 5G Network Management | June. 2026<br>Sep. 2026<br>Sep. 2026 |
| CL3A – AgriTech Transformation | Farm Management Information System | - AgroApps 360º | Feb. 2026 (Done) |
| | Pest Detection | - AI Pest Detection | Feb. 2026 (Done) |
| | Logistics and Traceability | - PowerFleet<br>- Hyperledger Fabric | Feb. 2026 (Done)<br>Sept. 2026 |
| | Private 5G Network Management | - Private 5G-aaS at the Farm | Sept. 2026 |
| CL3E – Edge Intelligence for Energy Grid Reliability | UC3E.1 - Flexibility Management | - Power System Simulation<br>- Flexibility Estimation | June 2026<br>Oct. 2026 |
| | UC3E.2 - EV Chargers Management | - EV Chargers Sensing<br>- EV Chargers Visualisation<br>- Predictive Maintenance | June 2026<br>June 2026<br>Oct. 2026 |
| | UC3E.3 – Biogas Plant Management | - Biogas Sensing<br>- Biogas Visualisation<br>- Anaerobic Digestion Forecasting | Feb. 2026 (Done)<br>Feb. 2026 (Done)<br>Oct. 2026 |
| CL4 – Smart Vineyards & Sustainable Winery Ecosystems | UC4.1 – Recycling, Maintenance and Logistics of IoT Sensors | - IoT Sensor/Recycling Platform<br>- euID Wallet Service | Oct. 2026<br>Oct. 2026 |
| | UC4.2 – Smart Vineyard Irrigation and Water Use Optimisation | Aquaview | Feb. 2026 (Done) |
| | UC4.3 – Sustainable Winery Production Line Optimisation | TheTorre | April 2026 (Done) |
| | UC4.4 – Sustainable Vinyards | - 5G IMSI provisioning & Weather & Soil virtual stations<br>- Energy measurements<br>- Energy AI optimizations | June 2026 (Done)<br><br>Oct. 2026<br>Oct. 2027 |

### 4.3.3 COP-PILOT Product Marketplace Expected by M36

In addition to service modelling, COP-PILOT introduces a product modelling approach aligned with TM Forum best practices. Products are defined as customer-facing abstractions that encapsulate one or more underlying service specifications managed by the ESO.

The product model includes:

- Product Specifications defining the functional and commercial attributes of the offering
- Product Offerings representing the commercial instantiation exposed in the marketplace
- Product Orders representing the lifecycle of user requests.

This model enables the decoupling between business-level interactions and technical service orchestration, allowing the platform to evolve towards full commercial capabilities, including pricing, customer management, and SLA-driven offerings.

While the initial implementation focuses on the Service Marketplace exposed by the ESO, a complementary Product Marketplace exposed by the BMP must be defined as the primary entry point for end users. The Product Marketplace aggregates product offerings derived from underlying service specifications and provides a business-oriented view of the platform capabilities, enabling discovery, comparison, and ordering without exposing technical complexity.

The COP-PILOT Product Marketplace will be designed in the second half of the project (i.e., M19-M36), therefore reported in D3.2 next year.

Co-funded by
the European Union

# 5 CONCLUSIONS

This deliverable has presented the final architecture and operational design of the COP-PILOT platform, hence completed the design phase of the project and established the technical foundation upon which all subsequent implementation and integration activities will be built. Through the progression from high-level architectural abstraction to detailed layer specifications, workflow definitions, and cluster-level engineering views, D2.2 provides a comprehensive and actionable reference that equips WP3 and WP4 partners with a shared and unambiguous understanding of the platform they are collectively building.

The final COP-PILOT architecture, structured across six horizontal layers and unified by a vertical CI/CD platform, reflects a deliberate set of design principles: heterogeneity tolerance at the infrastructure level, modularity and reusability at the orchestration level, zero-trust security as a first-class architectural concern, and simplicity of interaction at the business management level. These principles are not abstract aspirations but are concretely expressed in the architectural components described in Section 2 and operationalised in the workflows presented in Section 3. Taken together, the Domain Orchestrator, End-to-End Service Orchestrator, Secure Integration Fabric, and Business Management Portal form a coherent and scalable system capable of supporting the diverse deployment scenarios presented by the five COP-PILOT piloting clusters.

The service modelling engineering analysis presented in Section 4 and D2.2 Annex 1 reveal both the readiness of the platform's foundational service library elements and the areas where further design work is required ahead of full deployment. Across all clusters, the shared infrastructure patterns (Kubernetes deployments, OpenZiti-managed secure connectivity, and TMF-compliant orchestration interfaces) confirm that the COP-PILOT architecture is sufficiently general to accommodate highly diverse application domains while remaining tractable from an integration standpoint. The gap analyses conducted for each cluster provide a clear and prioritised set of service library items to be addressed during the platform's second development cycle, ensuring that the catalogue available by Month 36 fully supports the validation and demonstration objectives of WP5.

Looking ahead, D2.2 directly triggers two parallel streams of activity. Within WP3, the architectural specifications and workflow definitions presented here will be progressively materialised into the first and subsequent releases of the COP-PILOT platform, with D3.1 reporting on the initial implementation outcomes at Month 18. Within WP4, the infrastructure configurations and deployment specifications established in the cluster engineering views will guide the preparation and integration of each piloting cluster, with D4.1 capturing progress at Month 20. Together, these activities will advance the COP-PILOT platform from its current design maturity towards the high-TRL, large-scale deployments targeted by the project's conclusion.

Co-funded by the European Union

# REFERENCES


[1] COP-PILOT D2.1 “Ecosystem Definition and Requirements”, Available: https://cop-pilot.eu/download/d2-1-deliverable/?wpdmdl=8359&masterkey=4g8GOLW8SBrRwk2E00AgGZWzulumTnledmEdiDVx60a6ZRzDRiuUeVS1Zfvui7ReVh73PD1C_hnX6esZAw4rKSlgkYe9HViSHlJwKIGHt3U

[2] COP-PILOT D3.1 “First Version of the COP-PILOT Platform”, Available: https://cop-pilot.eu/download/d3-1-first-version-of-the-cop-pilot-platform/?wpdmdl=8303&masterkey=BacPvuTVP2ZNmrybg4_x-2WS8ck8putGlqatM7WsOxuLwoaG8Z5kg0CyK3MAa9oGCs_Pb7L1FgwidJXqHFuzF4L0cQzCx3KotQ_H0f_UXc8

[3] COP-PILOT D4.1 “Cluster and UC Preparation & Integration - Intermediate”, Available: https://cop-pilot.eu/download/d4-1-clusters-and-uc-preparation-integration-intermediate/?wpdmdl=8304&masterkey=vkbwFw_c7f9nu6fbWVzhHOvkBHIjkz7-Q6DVjvdoQRqd9Cah-oqjEJOnYlj8-8XxjLjsS668G2iGhRIs21_KDql-Le2PC4inM0xxHtSjsac

[4] TMForum, “TMF633 Service Catalogue Management API v4.0.0”, 2021, Available: https://www.tmforum.org/open-digital-architecture/open-apis/service-catalogue-management-api-TMF633/v4.0.