OPEN CALL #1

TECHNICAL GUIDELINES FOR THE COP-PILOT PLATFORM

Revision: V1 (February 2026)

COVER PAGE

Document Revision History

Version	Date	Description of change	List of contributors
V0.1	05/10/2025	1st edit	*Rafael Oliveira Rodrigues (D4P)*
V0.2	11/11/2025	2nd edit	*Rafael Oliveira Rodrigue (D4P)*
V0.3	08/01/2026	Review of the document and addressing PO's comments	Georgios P. Katsikas (UBI),
V0.4	16/01/2026	Addressing Technical Coordinator's comments	All Clusters
V1.0	26/01/2026	Final Submission	WP6

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

TABLE OF CONTENTS

1 THE COP-PILOT TECHNICAL FRAMEWORK

COP-PILOT is developing a Collaborative Open Platform (COP) designed for robust, end-to-end service orchestration across complex, heterogeneous environments that span the IoT-to-edge-to-core compute continuum. The platform's primary goal is to enable the deployment of secure, intelligent, and automated applications that operate across multiple administrative domains and industrial sectors. This framework is being validated through large-scale piloting clusters in strategic areas such as mining, smart buildings/cities, agriculture, and energy (see section 2).

To accelerate market uptake, broaden the platform's capabilities, and foster a mature European supply chain, COP-PILOT is engaging third-party innovators, with primary focus on SMEs and startups, through Open Calls. Participants in these calls are expected to design, integrate, and validate their own innovative services and applications or integrate new domains and functionalities to existing piloting clusters. By leveraging the COP-PILOT framework, successful applicants will test their solutions against real-world industrial challenges, demonstrate new cross-sector applications, and contribute to the platform's growing ecosystem.

- **Vision**

The convergence of smart IoT devices, edge computing, and advanced 5G connectivity has created immense opportunities for market digitization. However, realizing the full potential of "edge intelligence" is blocked by several significant challenges:

- Complex Interoperability: Integrating diverse technologies, domains, and administrative sectors under a common orchestration framework remains difficult.
- Massive IoT Scale: The immense increase in connected IoT devices requires intelligent management and automation features far beyond current capabilities.
- Limited Visibility: Inadequate visibility into the data plane and coarse monitoring make it difficult to effectively manage modern systems and guarantee service quality.
- Intelligent Orchestration: Orchestrators must evolve from simple resource allocation to intelligently analyzing complex data to make scalable, actionable decisions.
- Security and Trust: Establishing trust for data sharing and secure applications across collaborating sectors is essential but increasingly complex.

To address these challenges, COP-PILOT is developing a multi-layer orchestration platform designed to manage the entire service lifecycle across different domains and infrastructure providers. The COP-PILOT platform supports this vision by providing a set of critical functionalities across the platform’s interaction layers, (from the end user business interface to the infrastructure connectivity layer). At the top layer a unified portal for all stakeholders simplifies interaction through a novel LLM-based interface with smart service-oriented intent management capabilities. Below this, an end-to-end Service Orchestrator provides the onboarding and lifecycle management of services across multiple domains. A key innovation inserted that eases multi-domain and collaborative operation is the Secure Integration Fabric which acts as a zero-trust "network-as-a-service" solution for securely connecting these domains without complex manual configuration. At the domain level, the Distributed Domain Orchestrator manages local resources, offering "resource-as-a-service" and handling domain-specific data management. This entire software stack interfaces with the underlying heterogeneous Infrastructure Layer, allowing it to manage diverse compute, network, and IoT resources.

Co-funded by
the European Union

It is noted that a core principle of the COP-PILOT platform design is to move beyond fragmented, proprietary systems by building an open and interoperable ecosystem. The platform architecture is founded on open, standardized APIs and data models, aligning with key industry bodies like TMForum (TMF) and ETSI (specifically OSL, ZSM, and CIM). This standards-based foundation is the key enabler for the platform's core vision: open collaboration. The COP-PILOT platform is being explicitly designed to be expanded, providing simple, secure, and automated pathways for third parties to join the ecosystem.

This is the essence of the Open Calls, which target **two** primary types of collaboration:

- **Onboarding New Services:**
  - The platform provides a unified, high-level interface for service providers (like SMEs and startups) to easily onboard, deploy, and manage new, innovative applications. A novel LLM-based portal further simplifies this process, translating high-level user goals (e.g., "deploy my service") into the complex orchestration blueprints required by the system.
- **Integrating New Domains**:
  - The platform features an "Auto-Pilot" capability that allows infrastructure owners to easily register their private domains (e.g., testbeds, factory floors, or compute clusters) with the COP-PILOT ecosystem. A central Secure Integration Fabric (SIF) handles all the complex security and networking, allowing new domains to connect "on-the-fly" and securely expose their resources (like data or compute) to trusted partners using 'zero trust principles'—without the traditional burden of manual VPN and network configuration.

This vision of a unified, secure, and easy-to-join ecosystem, where new services and new domains can seamlessly collaborate, is what COP-PILOT aims to achieve. The Open Calls are the primary mechanism to test, validate, and grow this collaborative environment with new, innovative partners.

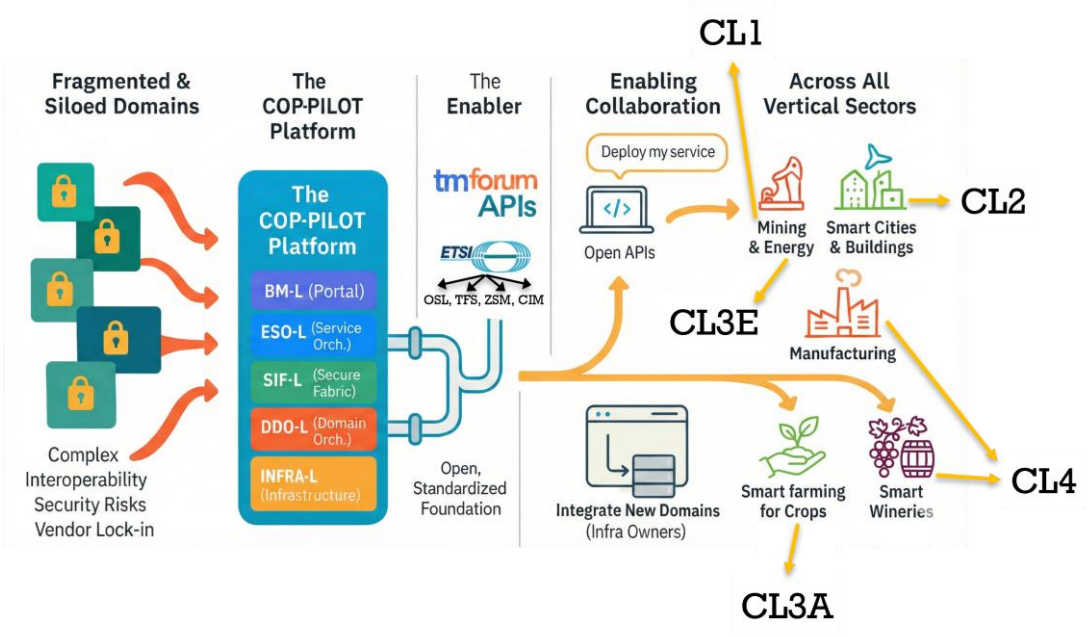

Figure 1 – The COP-PILOT vision towards the support of collaborative services and infrastructure domains.

1.1 THE COP-PILOT PLATFORM ARCHITECTURE: HIGH-LEVEL FUNCTIONAL OVERVIEW

This section provides a simplified functional overview of the COP-PILOT architecture that abstracts away low-level details without losing essential context. Figure 2 visualizes a simplified view of the initial COP-PILOT architecture, which is briefly explained in the rest of this section. This figure boils down the complex COP-PILOT architecture into 3 essential pieces:

- **(i)** The COP-PILOT domains where the platform offers domain-level orchestration and Data Management services to every individual domain stakeholder in the COP-PILOT ecosystem for managing compute, network, and data resources,
- **(ii)** A secure integration fabric for allowing secure on-the-fly data and service-level interactions between multiple domains, and
- **(iii)** An end-to-end service orchestration platform for managing services that span across multiple domains, thus highlighting the "collaborative nature" of the COP-PILOT system.

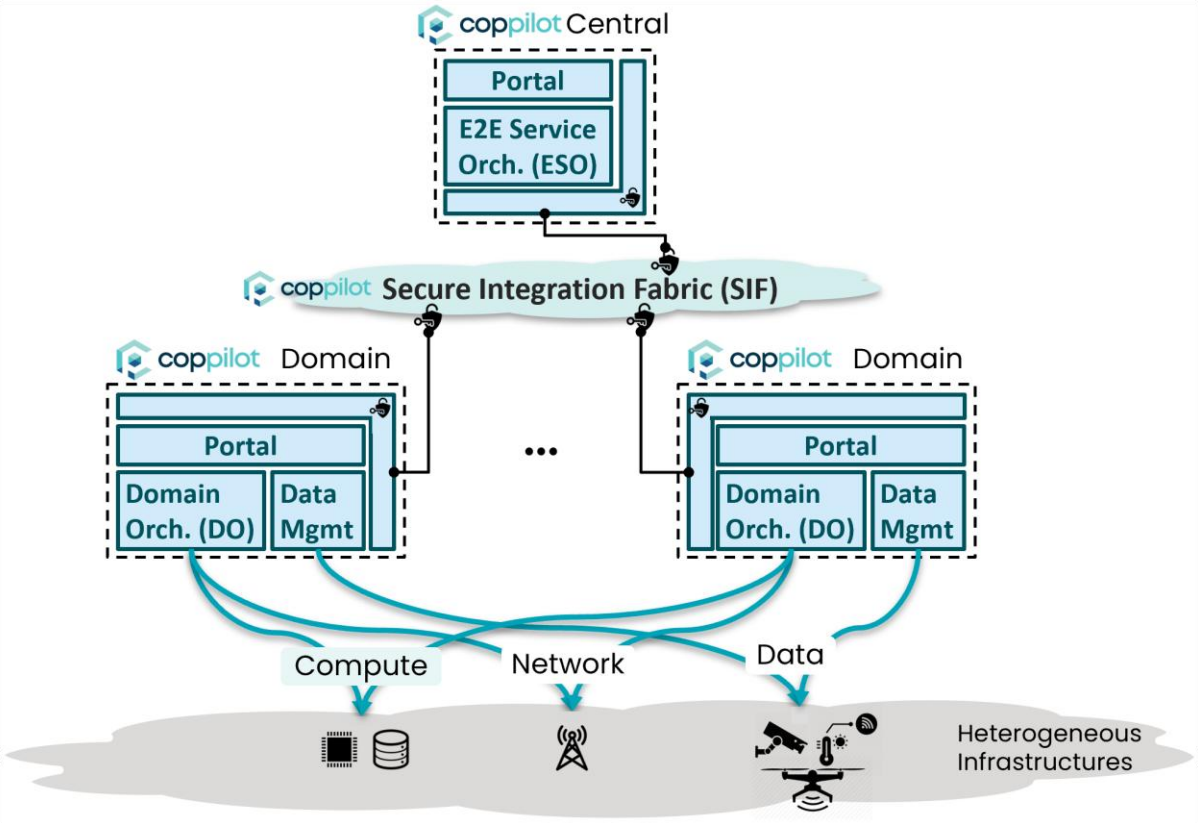

Figure 2 – A minimal view of the initial COP-PILOT architecture (delivered in October 2025).

Note that the minimal view of the initial COP-PILOT architecture shown in Figure 2 focuses solely on COP-PILOT elements. To make this view more complete, Figure 3 presents additional items to sketch the entire ecosystem around COP-PILOT: **(i)** the layers of the initial architecture shown at the left hand side in Figure 3, **(ii)** the span of the infrastructure at the bottom part in Figure 3, **(iii)** the various stakeholders associated with COP-PILOT (see user icons in Figure 3), **(iv)** the primary vertical sectors that COP-PILOT facilitates via the piloting activities in WP4 (top part in Figure 3), and **(v)** the way COP-PILOT and third-party services interact with the platform. Note also that the pink-highlighted boxes in Figure 3 are presented as distinct aspects of the platform in the following paragraphs.

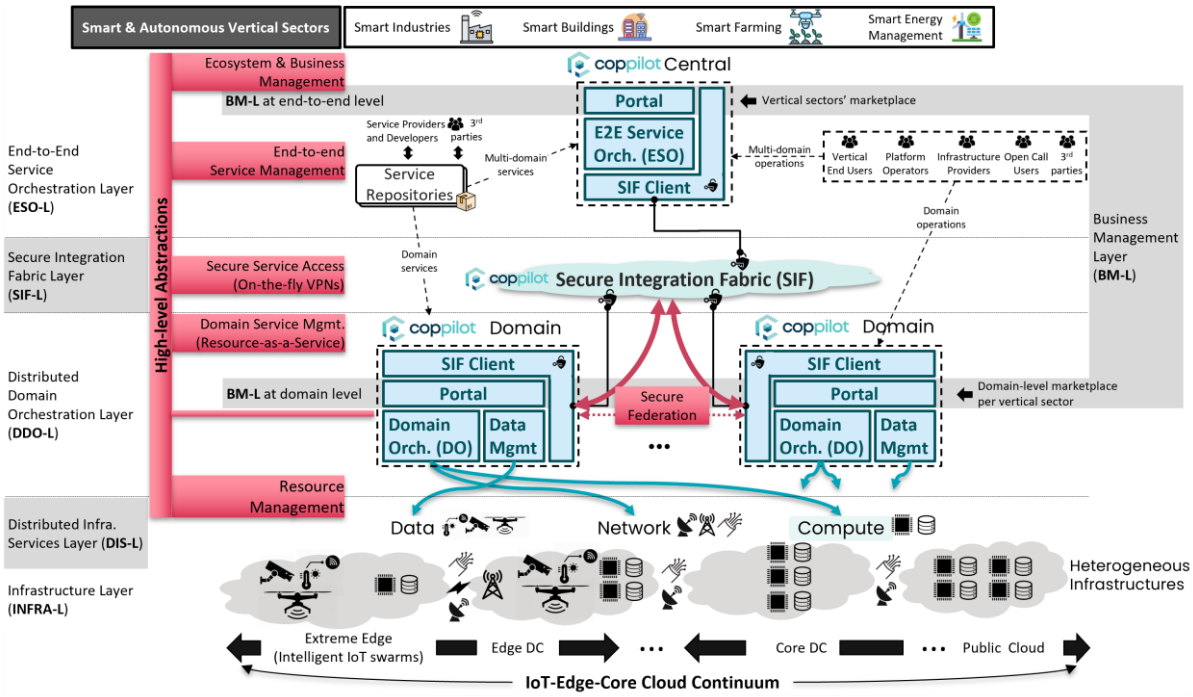

Figure 3 – A simplified equivalent of the initial COP-PILOT architecture (delivered in October 2025).

Resource management: Hide hardware details of complex, large scale, and multi-tenant testbeds.

At the bottom part in Figure 3, the infrastructure layer (INFRA-L) is depicted as a large pool of compute, network, and data resources spread across the entire IoT-edge-to-core continuum. Specifically, **(i)** extreme edge and edge domains offer various types of sensors with or without compute capabilities, while the latter domains may also host private networking infrastructure (e.g., private 5G) for channelling the data of these sensors towards a (core/edge) datacenter, **(ii)** core domains offer larger amounts of compute capacity for hosting either applications of core network functions (e.g., those corresponding to a private 5G core network), and **(iii)** hyperscalers are always available for further increasing the required compute capacity with public cloud VMs/clusters when needed. This layer includes not only the large COP-PILOT clusters' infrastructure, but also potential infrastructure that may appear during the two Open Call rounds or third parties that wish to integrate their infrastructure with the COP-PILOT platform in the future. On top of this layer, infrastructure providers typically expose a set of Distributed Infrastructure Services for managing the underlying hardware without directly exposing it to overlay systems. This layer - titled DIS-L – is the interface of the COP-PILOT platform with any testbed, as it is responsible for direct resource management over the physical resources. The objective of COP-PILOT is to demonstrate ability to integrate with any kind of infrastructure service, thus embrace heterogeneous testbeds that pertain to different vertical sectors (not necessarily the 4 vertical sectors addressed by the COP-PILOT clusters).

Domain-level service & data management: Foster any Resource-as-a-Service.

Today's infrastructures tend to be multi-domain, while different administrative entities appear as operators/owners in each domain. On top of every such domain, COP-PILOT employs a domain-level management platform that is comprised of: **(i)** a Domain Orchestrator instance, **(ii)** a Data Management instance, and the overlay Business Portal at the domain level. Both DO and DM establish bindings with the underlying domain infrastructure services (DIS-L) for exposing compute, network, and data-as-a-service using standardized service, resource, and data management APIs (see Figure 3). This is a key abstraction of COP-PILOT that 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.

Secure service access management: Security and trust built-in COP-PILOT.

Within every domain, COP-PILOT's Secure Integration Fabric (SIF) undertakes to create encrypted connections for exposing certain domain-level services towards the outside world (i.e., another domain or the upper layer). The COP-PILOT SIF undertakes to offer zero-trust on-the-fly VPNs towards any service within the entire COP-PILOT ecosystem, introducing a key abstraction that ensures security and trust. This abstraction takes away the huge complexity of managing traditional VPNs in large ecosystems of multiple administrative entities, while being far more secure by default. To establish traditional VPNs within COP-PILOT, multiple IT departments in every cluster should issue VPN accounts, credentials, and configuration files for several external parties, not to mention that this needs to happen across additional IT departments by the partners who will appear during two open call rounds.

COP-PILOT revisits the way private networking should be done nowadays by leveraging recent advancements in secure software-defined overlay networking using OpenZiti (provided by the COP-PILOT consortium member *TATA*). This allows COP-PILOT to set up a root-of-trust domain where the control plane of the secure integration fabric 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 to register a new domain under the COP-PILOT realm and consequently declare domain resources and services that the domain owner wishes to expose through the integration fabric.

Secure federation: Apart from exposing domains and domain services in a secure and trusted manner, the COP-PILOT SIF will be used as a secure and trusted data federation fabric to east-west interconnect multiple COP-PILOT Data Management platform instances. This will allow secure sharing of data between domains to enable collaborative scenarios throughout the COP-PILOT piloting activities.

End-to-end service management: Manage services across a complex private continuum.

A higher-tier orchestration layer is added above the SIF to provide the notion of end-to-end management of services and resources in the multi-domain and multi-stakeholder/tenant era of 6G. This layer (titled ESO-L) introduces COP-PILOT's End-to-end Service Orchestrator (ESO), which binds with multiple DO instances across multiple domains to federate domain-level marketplaces into a large multi-domain marketplace of services offered to the COP-PILOT stakeholders. This is done via another dedicated business portal on top of the ESO (complementary to the business portal atop the DO), which introduces a business management layer at the level of multiple domains and end-to-end services. This is the reason that the Business Management Layer (BM-L) of COP-PILOT appears vertically in Figure 3; conceptually speaking, this layer is the top-most layer of the COP-PILOT architecture as it appears in the detailed version of the architecture in Section 1.1. However, when the architecture gets instantiated into a real multi-domain platform, the business layer appears both within every domain (to enable domain level business operations), but also in an end-to-end fashion (atop ESO-L).

Ecosystem and Business Management: Finally, a modern user portal design complements the rest of the platform with views that allow all kinds of stakeholders (i.e., domain-level stakeholders in every domain, infrastructure owners, platform administrators, service providers, vertical end users, 3rd parties, etc.) to exploit the rich set of APIs offered by the DO and DM in every domain as well as the ESO in the central management domain in order to perform platform, service, resource, and data management operations in a straightforward and user friendly manner. Key workflows for COP-PILOT stakeholders get triggered from the portal, thus abstracting the low-level details that appear in the ESO, DO, and DM APIs.

1.2 THE COP-PILOT PLATFORM WORKFLOWS

This section describes how the potential third parties can interface with the clustering pilots through the COP-PILOT platform. In principle, the platform is designed to radically simplify how third parties interact with complex, multi-domain ecosystems. **Two primary workflows** are provided for Open Call participants, depending on their integration role in the project:

► ***For Service Providers*:** A workflow to easily onboard and deploy a new, innovative vertical service, able to coexist with other services and ideally promote cross-domain collaboration. This workflow supports the full-service lifecycle, managing not just the initial deployment but also runtime operations like updates and scaling. Providers can leverage the platform's advanced, built-in mechanisms for effective data collection (telemetry) to monitor their application's performance or deploy and attach new mechanisms as a service. Crucially, the platform simplifies this entire process through a novel LLM-based portal for easy onboarding and provides automated proactive SLA assurance to guarantee service quality without manual intervention.

► ***For Infrastructure/Domain Owners*:** A workflow to securely and automatically register a new private infrastructure domain (e.g., a testbed, lab, or compute cluster) with the COP-PILOT ecosystem. This "Auto-Pilot" workflow allows owners to easily expose their unique domain resources (like compute, networks, or IoT data) "as-a-service" to the federated platform. The platform's Secure Integration Fabric (SIF) is central to this process, providing identity-first, zero-trust, on-the-fly secure connectivity that eliminates the need for complex, manual VPNs. Owners retain full control through programmable, policy-driven mechanisms that define exactly which resources are shared and which trusted parties can consume them.

Workflow synergies: While these workflows can be addressed independently, applicants are highly encouraged to combine them. The integration of a new infrastructure domain (Workflow 2) should aim to demonstrate interoperability and collaboration with existing COP-PILOT piloting cluster domains. This integration can be further extended to:

(a) Include the deployment of a new, collaborative service (following Workflow 1) onto that newly added domain.

(b) Demonstrate the scalability of an existing COP-PILOT use case service by extending its operation to the new domain.

1.2.1 Workflow 1: Onboarding and Deploying New Services

This workflow is designed for service providers who want to deploy their applications onto the federated COP-PILOT infrastructure. The process is broken into four phases, automating the most complex integration tasks.

- Phase #1: Automated Service Onboarding
  - Instead of complex API interactions, the provider simply publishes their application to a git-based software repository. This includes their application container images and existing deployment manifests (e.g., Helm charts, Kubernetes manifests, or Docker Compose files).
  - A Continuous Integration (CI) pipeline automatically detects the new artifacts and translates them into standardized service specification blueprints, understood by the orchestrators
  - These blueprints are onboarded onto the orchestrator(s) catalog(ues).
  - This approach will also be used to onboard all the Open Call applications, given that the application artifacts can be uploaded (by the OC partners) to the central COP-PILOT service registry.
- Phase #2: Auto-peering among the COP-PILOT orchestrators
  - Another automation (CI) pipeline undertakes to synchronize the catalogues of the various Domain Orchestrators – spread across multiple domains within the COP-PILOT clusters – with the COP-PILOT ESO. This process is titled “Peering between orchestrators”.
  - This way, the ESO obtains an up-to-date view of the marketplace (i.e., available services) of each cluster with a blink of an eye.
  - At the end of the day, the main benefit of this peering process is that the ESO gathers multiple marketplaces from different vertical sectors and exposes them altogether via the same portal (and the use of a standardized TMF interface). Users may see the entire spectrum of COP-PILOT services (classified across multiple catalogues and categories) in one view.
- Phase #3: LLM-assisted and Intent-driven Service Ordering
  - This is the primary interface for a new service provider.
  - Before being able to order a service, this service must be onboarded (as per Phase #1) onto a certain orchestrator instance, and this orchestrator must have completed the peering with the ESO (as per Phase #2).
  - Only then, the service provider can interact with the COP-PILOT LLM chat provided by the COP-PILOT Business Portal. The service provider may ask the LLM to locate the service and order it with suitable configuration.
  - The LLM parses the user’s informal intent and browses through the COP-PILOT Marketplace to pinpoint the service.
  - Once found, the LLM creates a formal intent that requests the specific service to be ordered atop a certain infrastructure (i.e., a Cluster testbed or set of testbeds). Depending on the nature of the intent (multi-domain vs. domain-level), this is dispatched either to the ESO or the right DO.
    - If the ESO receives the intent, it coordinates with the right set of DOs to perform a multi-domain service deployment.

- If a DO receives the intent, it carries out the service deployment itself (as this is a single-domain service).

- In either case, the platform uses the Secure Integration Fabric (SIF) to automatically and securely gain access to the remote private clusters, thus conducting the deployment of the service.

- Phase #4: Automated SLA Assurance (Built-in)

  - Crucially, Open Call participants do not need to build their own complex SLA management. The platform does this automatically.

  - During the interaction of the end user with the LLM (Phase #3), the formulated intent captures the (additional) needs of the service by means of security, intelligence, automation, etc.

  - When the intent is translated into a service order, the ESO automatically couples the user's service with a corresponding "platform service chain," which is visualized as an "Ultra service graph".

  - This hidden service chain ensures the vertical's SLA through a "Trinity" of automated capabilities.

    - Intelligence: Proactive, AI-based forecasting to predict potential SLA violations.

    - Automation: Policy-based, zero-touch invocation of platform APIs (e.g., scaling, migration) to reconfigure the service before the SLA is breached.

    - Security: Guarantees that these automated actions are applied in a secure and trusted manner.

  - This phase is not explicitly triggered by the user but rather incorporated into Phase #3 to properly accommodate the service runtime needs of the user.

➔ **Applicant Responsibilities and Opportunities**: While the platform provides this automated framework, its effectiveness depends on the applicant's application and contributions.

- For Vertical Application Compliance: The vertical application must be observable. It should be packaged (e.g., containerized) and expose relevant performance metrics (e.g., latency, throughput, custom KPIs). This is essential for the platform's "Intelligence" service to monitor its health and predict SLA violations.

- For Providing New Platform Services: Applicants are also highly encouraged to propose their own innovative services that can be integrated into this open "Trinity" framework. For example, your proposal could include:

  - A new "Intelligence" Service: A custom-trained AI/ML model (e.g., for specialized anomaly detection or predictive maintenance) that is more advanced than the platform's default.

  - A new "Automation" Service: A "zero-touch" function that defines a custom response to a predicted violation (e.g., "offload traffic to a specific partner domain" or "trigger a custom data-caching mechanism"), closing the loop with the orchestrators.

- For Integrating new application and services: These new platform services must be onboarded just like any other application (see Phase #1) and expose standardized APIs. Any custom "Intelligence" service must output a standard event that the platform can understand, and any "Automation" service must be able to receive this event and correctly invoke the Orchestrators' standardized APIs (e.g., a TMF service update API) to trigger the desired action.

1.2.2 Workflow 2: Integrating New *Private* Infrastructure Domains

This workflow is designed for infrastructure owners (e.g., universities, labs, or private companies) who want to connect their *private* domain to the COP-PILOT ecosystem, make its resources available, and collaborate with service providers.

- Registration and Authentication:
  - The infrastructure owner uses the COP-PILOT Business Portal to register as a new party in the ecosystem. They authenticate to the Portal using state-of-the-art OAUTH2 mechanisms.
- Domain Expansion:
  - Using a dedicated "domain expansion" view in the portal, the owner registers their new private domain. This "Auto-Pilot" process, powered by the Secure Integration Fabric (SIF), uses a secure authentication token and automatically establishes the necessary zero-trust, encrypted connections, eliminating the need for manual VPN setup.
- Automated synchronization of distributed marketplaces:
  - Once the domain is federated, the local Domain Orchestrator (DO) exposes an API that the COP-PILOT ESO exploits to "pull" the domain's marketplace (peering).
  - This peering process makes the domain's services discoverable through the COP-PILOT multi-domain marketplace (ESO level).

➔ **Applicant Responsibilities and Opportunities in Workflow 2**:

- While the platform automates the secure connection and registration, the Infrastructure Owner/Provider/Manager is responsible for preparing their local domain for this integration.
  - For Domain Compliance: To be federated, your domain must run the COP-PILOT domain-level management components, i.e., the Domain Orchestrator (DO) and, if data is shared, the Data Management (DM) platform. You are responsible for:
    - Integration: Ensuring the DO/DM components can interface with your local infrastructure. This means using compatible infrastructure controllers (e.g., Kubernetes) that the platform can manage.
    - Resource Cataloguing: Correctly describing your available compute, network, and data resources as standardized "as-a-service" specifications in your local DO/DM catalogues. This is what allows the central ESO to discover and order them.
    - Data Connectivity: If you are exposing data (e.g., from IoT sensors), you are responsible for providing the Data Connectors (at the DIS-L) that feed your local Data Management platform.

- For New Opportunities: Once your domain is successfully integrated, you unlock new collaborative and commercial opportunities. You can:
  - Monetize Your Resources: Offer your newly exposed compute or data services to the entire COP-PILOT ecosystem via the multi-domain marketplace, creating a new value stream.
  - Host Collaborative Services: Use your own domain to host new, innovative cross-domain services by following Workflow 1, either for your own projects or in partnership with other service providers.
  - Scale Existing Use Cases: Collaborate with existing COP-PILOT partners to test the scalability of their services by extending them into your newly available domain.

These workflows demonstrate how the platform is built for open collaboration. They allow participants to focus on their core innovation, either a novel service (Workflow 1) or a unique infrastructure (Workflow 2), while the platform handles the complex orchestration, security, and integration.

1.3 THE COP-PILOT PLATFORM TECHNOLOGIES AND PROTOCOLS

This section links the COP-PILOT platform assets (i.e., elements of the COP-PILOT architecture described in Section 1.1) with relevant technologies, protocols, etc as shown in Table 1.

Table 1: Map of COP-PILOT Platforms Assets related to relevant technologies and protocols.

COP-PILOT Component Name	Architectural layer of COP-PILOT Component	Relevant Technologies, Protocols, etc.
Business Portal	BM-L	Next.js (The React Framework for the Web) React Keycloak (Identity and Access Management) LangChain Ollama or vLLM Hugging Face Transformers Vector Databases (e.g. FAISS, Qdrant)
End-to-End Service Orchestrator (ESO)	ESO-L	Next.js (The React Framework for the Web) React Apache Quarkus (Java framework) Keycloak (Identity and Access Management) Redhat Kogito (Business Process Model and Notation) Prometheus (Metrics and monitoring) Grafana (Loki, Tempo, Dashboard) Hashicorp Vault (Secret management) Apache Kafka (Distributed event streaming) Technologies that the ESO orchestrates: - RedHat Open Cluster Management (OCM) multi-domain orchestration - Kubernetes (Container orchestration) - RedHat OpenShift
Domain Orchestrator (DO)	DDO-L	Angular (Web application framework) Spring Boot (Java framework) Keycloak (Identity and Access Management) Flowable (Business Process Model and Notation) Apache ActiveMQ (Distributed event streaming) MCP (Standard for connecting AI applications to DO) Prometheus (Metrics and monitoring) Grafana (Loki, Dashboard) Elasticsearch (Central log aggregation) Kroki (Diagram creation tool) Technologies that the DO orchestrates: - Kubernetes (Container orchestration) - RedHat OpenShift - ColonyOS - GitOps exchange model - LF Sylva Framework - NFV Orchestrators (implementing SOL005) - External Orchestrators (following TMF ODA)

Data Management (DM)	DDO-L	- Orion Context Broker - Scorpio Context Broker - Stellio Context Broker - Eclipse Arrowhead
Secure Integration Fabric (SIF)	SIF-L	- OpenZiti - Go (Golang) — primary implementation language for controllers, routers, and SDKs - X.509 / PKI — cryptographic identities for services, workloads, and endpoints - mTLS (Mutual TLS) — mandatory mutual authentication and encrypted transport - Libsodium — modern cryptographic primitives (key exchange, encryption, signatures) - SDKs / Client Libraries — application embedding of secure connectivity (e.g. Go, C, Java, JS)
CI/CD Platform	SIF-L	Github (Git based cloud platform for hosting and managing software projects) Jenkins (CI automation) Harbor (Container & charts registry) Portainer (Container management) Kubernetes (Container Orchestration) Keycloak (Single sign-on & identity management)
Compute Controller	DIS-L	Kubernetes (vanilla), microk8s, k3s, k9s, and more flavours. RedHat's OpenShift, Open Cluster Management (OCM), ColonyOS
Network Controller	DIS-L	Any controller for managing private 5G resources (RAN, Transport, Core) can be integrated with OpenSlice, especially if it is described as a Kubernetes Custom Resource (CR). Example 5G controller integrations already in place: 1) NOKIA's Network as Code (NaC) controller, 2) OTE's Slice Manager based on Ericsson's hardware, and 3) ETSI OSM 5GC provisioning based on Open5GS helm charts (other 5GC soft. Stack is supported if packaged as a cloud native software)
IoT Controller	DIS-L	Any IoT controller that complies to the standardized ETSI NGSI-LD or NGSI-v2 data models (mentioned above) or Eclipse Arrowhead southbound protocols

1.4 THE COP-PILOT PLATFORM STANDARDS AND COMMUNITIES

This section links the COP-PILOT platform assets (elements of the COP-PILOT architecture described in Section 1.1) with relevant standards and open-source (software) communities as shown in Table 2.

Table 2: Map of COP-PILOT Platforms Assets related to relevant standards and open-source communities.

COP-PILOT Component Name	Architectural layer of COP-PILOT Component	Relevant Standards	Relevant Open-source Communities
Business Portal	BM-L	TMF 633 Service Catalog Mgmt. TMF 641 Service Ordering Mgmt. TMF 638 Service Inventory Mgmt. TMF 634 Resource Catalog Mgmt. TMF 652 Resource Ordering Mgmt. TMF 639 Resource Inventory Mgmt.	Main component: Open-source COP-PILOT Business Portal (GitHub) → Will be open soon Based on: AG-UI: the Agent-User Interaction Protocol (GitHub)
End-to-End Service Orchestrator (ESO)	ESO-L	TMF 633 Service Catalog Mgmt. TMF 641 Service Ordering Mgmt. TMF 638 Service Inventory Mgmt. TMF 640 Service Activation Mgmt. TMF 634 Resource Catalog Mgmt. TMF 652 Resource Ordering Mgmt. TMF 639 Resource Inventory Mgmt. TMF 642 Alarm Mgmt. TMF 632 Party Mgmt. TMF 669 Party Role Mgmt. TMF 673 Geographic Address Mgmt. TMF 674 Geographic Site Mgmt. ETSI GS ZSM 003 v1.1.1 (2021-06): Zero-touch network and Service Management (ZSM); End-to-end management and orchestration of network slicing ETSI GS ZSM 008 v1.1.1 (2022-07): Zero-touch network and Service Management (ZSM); Cross-domain E2E service lifecycle management	Main component: ETSI OpenSlice HypO Module Development Group (MDG) created through COP-PILOT on January 14, 2026 Part of ETSI OpenSlice Software Development Group (SDG) The COP-PILOT ESO recently demonstrated compliance with ETSI ZSM via PoC 16
Domain Orchestrator (DO)	DDO-L	TMF 620 Product Catalog Mgmt. TMF 622 Product Ordering Mgmt. TMF 637 Product Inventory Mgmt. TMF 633 Service Catalog Mgmt. TMF 641 Service Ordering Mgmt. TMF 638 Service Inventory Mgmt. TMF 640 Service Activation Mgmt. TMF 634 Resource Catalog Mgmt.	Main component: ETSI OpenSlice Software Development Group (SDG)

		TMF 652 Resource Ordering Mgmt. TMF 639 Resource Inventory Mgmt. TMF 702 Resource Activation Mgmt. TMF 628 Performance Mgmt. TMF 632 Party Mgmt. TMF 669 Party Role Mgmt. TMF 673 Geographic Address Mgmt. TMF 674 Geographic Site Mgmt. ETSI GS ZSM 003 v1.1.1 (2021-06): Zero-touch network and Service Management (ZSM); End-to-end management and orchestration of network slicing ETSI GS ZSM 008 v1.1.1 (2022-07): Zero-touch network and Service Management (ZSM); Cross-domain E2E service lifecycle management ETSI GS NFV-SOL 005 V4.3.1 (2022-08) ETSI GS NFV-SOL 018 V5.1.1 (2024-07) IETF: A YANG Data Model for the RFC 9543 Network Slice Service	The COP-PILOT DO demonstrated compliance with ETSI ZSM via PoC 16 and PoC 2 Components that integrate with OSL (thus be potentially exploited): 1. ETSI Open-Source MANO (OSM) Software Development Group (SDG) 2. ETSI TeraFlowSDN Software Development Group (SDG)
Data Management (DM)	DDO-L	ETSI NGSI-LD by ETSI GS CIM 009 V1.9.1 2025-07	Main components: • FIWARE Orion-LD • Scorpio broker • Stellio Context Broker • Lepus for combined NGSI-v2 and NGSI-LD data spaces • Eclipse Arrowhead
Secure Integration Fabric (SIF)	SIF-L	CNCF SPIFFE OpenID Connect (OIDC) authentication protocol built on top of the OAuth 2.0 framework RFC 7519: JSON Web Token (JWT) mTLS extension of OAuth 2.0 RFC 5280: X.509 Public Key Infrastructure Certificate	Main component: OpenZiti OpenZiti GitHub
CI/CD Platform	SIF-L	OAuth 2.0 Authorization protocol (RFC 6749) OpenID Connect (OIDC) Identity layer on OAuth 2.0 SAML 2.0 Authentication/SSO standard JWT (JSON Web Token)	Jenkins Community Forum Harbor Community Page Portainer Community Forum Keycloak Community Kubernetes Discuss Forum

		OCI Image Specification OCI Distribution Specification	
Compute Controller	DIS-L	Any standards applicable to the entire Kubernetes ecosystem	All communities related to the entire Kubernetes ecosystem
Network Controller	DIS-L	Any standards applicable to private (beyond) 5G systems	All communities related to private (beyond) 5G systems
IoT Controller	DIS-L	(Commercial) IoT platforms that comply with standardized ETSI NGSI data models	All IoT communities that comply with standardized ETSI NGSI data models

The relation of the above platform components with open-source software and relevant standards is shown in the following Figure 4.

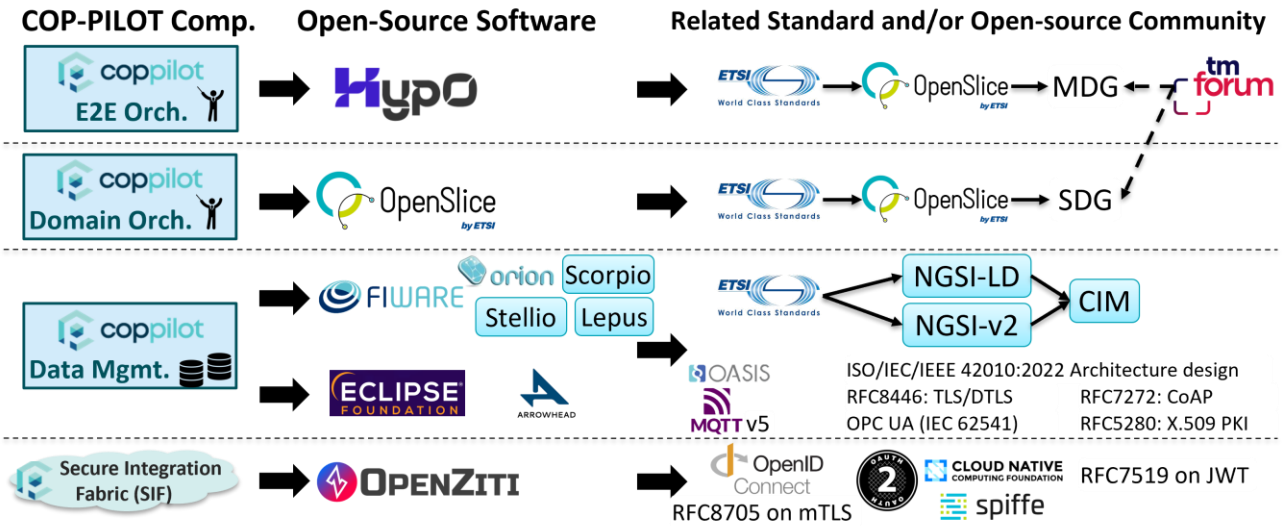

Figure 4: Mapping of COP-PILOT platform components to open-source software and standards.

1.5 EXPECTED BENEFITS FROM OPEN CALLS AND LINK TO THE PLATFORM

Proposals submitted to the COP-PILOT Open Calls will be evaluated on their ability to deliver tangible benefits to the ecosystem. All funded projects must successfully demonstrate the following outcomes, which are directly linked to the validation and expansion of the COP-PILOT platform:

- **Platform Validation and Feedback**

  Applicants must deploy their solutions using the core components of the COP-PILOT platform. The primary goal is to validate the platform's key innovations, including the LLM-assisted service onboarding, the end-to-end service orchestration (COP-PILOT ESO and DO components), the automated SLA assurance, and the "Auto-Pilot" secure domain integration (SIF). A mandatory outcome for all projects is the delivery of concrete validation data (e.g., performance KPIs, service deployment times, scalability metrics) and qualitative feedback on the platform's usability, robustness, and features.

- **Ecosystem Expansion and Innovation**

  Applicants must introduce tangible, innovative assets that expands the COP-PILOT ecosystem. Such assets are expected to be new vertical application or platform service (following Workflow 1) relevant to the project's core piloting clusters (Mining, Smart City/Building, Agriculture, Energy, and Manufacturing), as well as a new infrastructure domain (following Workflow 2) that offers unique resources (e.g., a new testbed, a private compute cluster, a unique data-as-a-service offering) to the federation.

- **Demonstrate Cross-Domain Collaboration**

  Proposed solutions must not operate in a silo. A key expected outcome is the demonstration of tangible cross-domain collaboration. This must be achieved by interacting with, consuming data from, or deploying services to at least one of the existing COP-PILOT piloting domains. Alternatively, it may offer the scaling of an existing COP-PILOT use case or service into a newly federated third-party domain.

- **Adherence to Open Standards and Interoperability**

  All proposed solutions must strictly comply with the platform's open, standards-based interfaces. This includes using the specified TMForum (TMF) based APIs for service and resource orchestration and management as well as the NGSI-LD and smart data models for standardized data management. This adherence is critical to ensure interoperability and to validate the core COP-PILOT vision of a unified, open, and standardized European ecosystem.

1.6 APPLICATION OF COP-PILOT ACROSS 5 VERTICAL SECTORS

The generic COP-PILOT platform is applied across 5 heterogeneous vertical sectors in Europe, the details of which are provided in the following links:

- **Cluster 1** – Business Integration in Mining
- **Cluster 2** – Smart Sustainable IoT Solutions in Valencia
- **Cluster 3A** – AgriTech Transformation and Sustainability Initiative (ATSI)
- **Cluster 3E** – Edge Intelligence for Enhancing Grid Reliability in RES-Rich Distribution Grids
- **Cluster 4** – Smart Vineyards and Sustainable Winery Ecosystems