Where is ECOFED active?

WS 3

WS 2

D.2.1. Federation and Multicloud Meta-Orchestration

MCK - Multi-Cluster Kubernetes

D.2.2. XaaS Service Layer

TEC2 - Telco & Edge Cloud

D.2.3. Integration, Management, Monitoring and Optimisation

virt8ra - Multi-Provider Multi-Cloud Virtualization Management and Orchestration

D.2.4. Operating System and Virtualization

D.2.6. Crosscutting Security
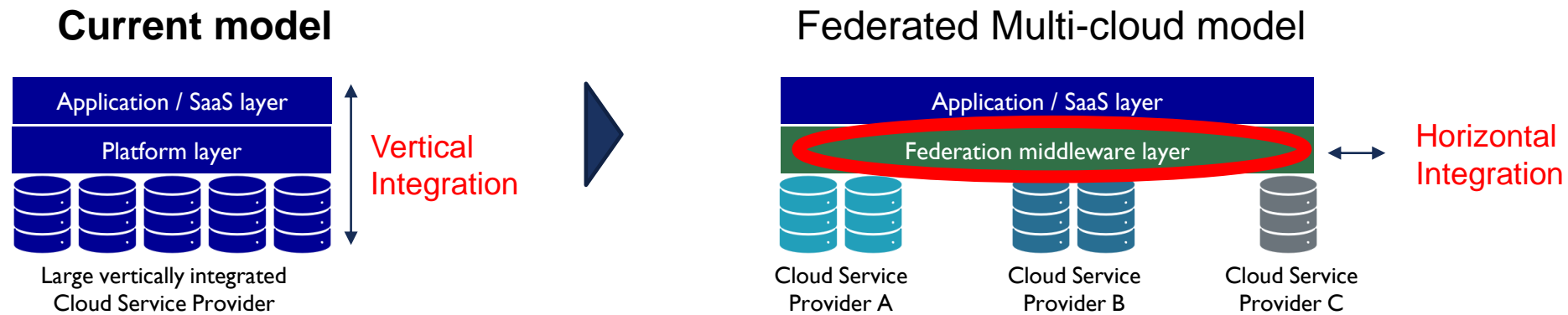
WS 1
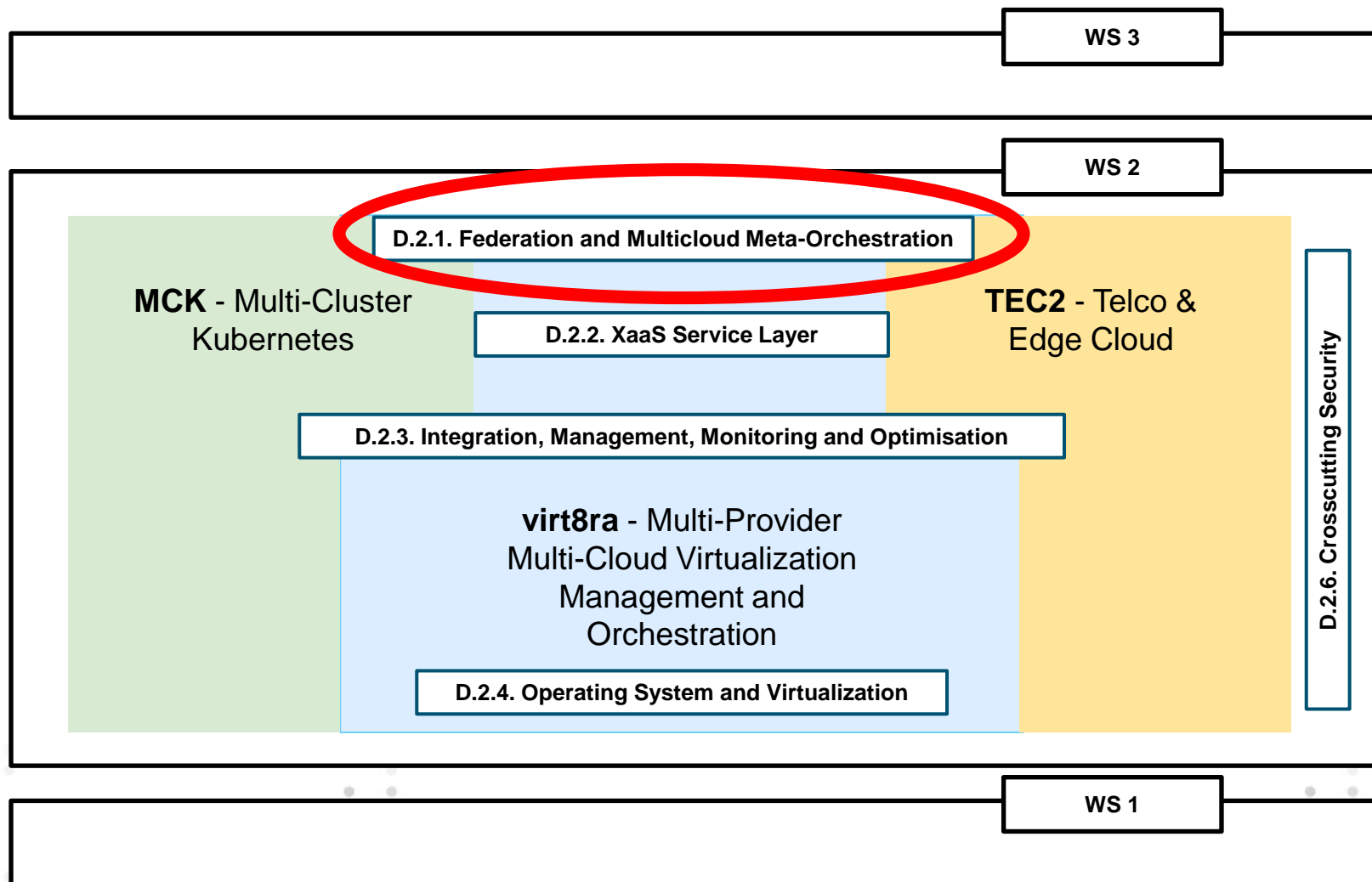
What will the ECOFED project do?

Quick recap from earlier WS2 sessions

- The objective of the ECOFED project is to develop a technical basis for a more open cloud usage model, facilitating interoperability, federation and switching between providers.

Current model

Application / SaaS layer

Platform layer

Vertical Integration

Large vertically integrated Cloud Service Provider

Federated Multi-cloud model

Application / SaaS layer

Federation middleware layer

Horizontal Integration

Cloud Service Provider A

Cloud Service Provider B

Cloud Service Provider C

- The open source software deliverables will enable cloud service providers to build more flexible, scalable and federated cloud infrastructure services.

- Benefits; lower risk of lock-in effects due to easier cloud switching and more flexible capacity scaling through cloud federation without the intervention of one central provider.

ECOFED: **Our key technical objectives**
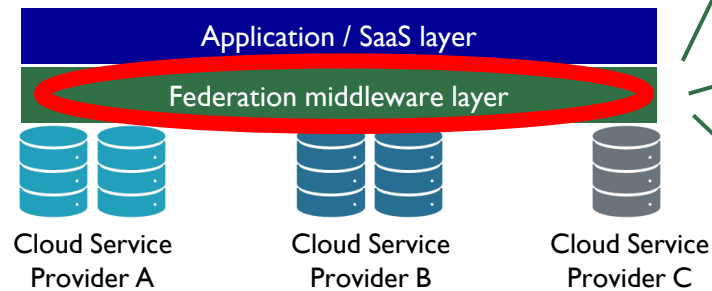
- An **open and technology-agnostic interface** suitable for the delivery of all kinds of heterogeneous services across the cloud stack.

  Such common protocol enables 1) cloud operators to implement a common service layer and 2) cloud customers to interact with all providers in the same manner.

- A **reference implementation of an orchestrator** that works within the protocol of our unified interface.

  The ability to develop orchestrators (and other tooling) that is compatible with the unified interface helps in delivering the same convenience as the hyperscale-platforms deliver.

- **Integrations with service catalogues** and/or marketplaces.

  Enable developers to discover services and to reason about the compatibility between service offerings of different providers.
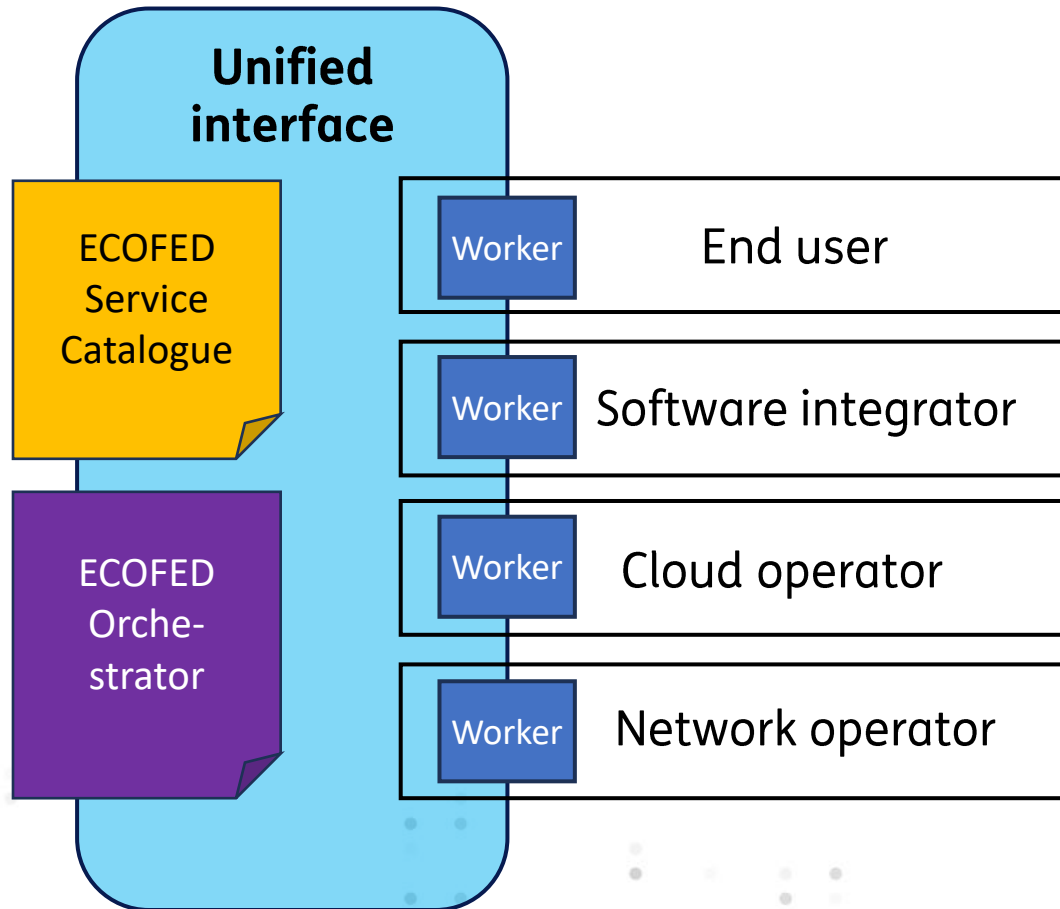
Federated Multi-cloud model

Application / SaaS layer
Federation middleware layer

Cloud Service Provider A   Cloud Service Provider B   Cloud Service Provider C

DOME MARKETPLACE

Think of Crossplane, Terraform/OpenTofu without hacking around provider specific APIs as ECOFED providers already speak the same language.

In AWS/GCP/Azure all services within the platform integrate perfectly together and can be displayed in one fancy portal. In our open model not every provider is identical, therefor open discovery and automated compatibility resolution are needed.

Solution requirements

Unified interface

ECOFED Service Catalogue

ECOFED Orche-strator

Worker	End user
Worker	Software integrator
Worker	Cloud operator
Worker	Network operator

The **unified interface** aims to

- bridge the gap between a lot of services with a lots of configurations across the entire cloud stack with one common language.

- make it easy for a provider to participate and become interoperable with the protocol.

- fit the business needs for all types of different cloud players in the open ecosystem.

- be convenient for tool developers to deliver ease of use to the ecosystem, as this is the hyperscaler's biggest selling point in their platform ecosystems.

Design principles

- Principles **Unified interface**
  1. There is one unified interface
  2. All parties/roles involved connect via a unified interface
  3. Parties can have a subset of the unified interface, with only functions necessary for the role
  4. The unified interface is extensible
  5. We prefer a declarative description over an imperative description to allow parties as much freedom as possible in the implementation

- Principles **Service catalogue**
  1. There can be one or more service catalogues
  2. All parties can publish their service offerings to a service catalogue
  3. All parties involved can use a third-party service catalogue to offer services
  4. All parties involved can use a private service catalogue to offer services
  5. Providers can use service catalogues to offer their services

- Principles **Orchestrator**
  1. There can be one or more orchestrators
  2. The orchestrator offers and uses the unified interface
  3. The orchestrator is decentral in nature
  4. The orchestrator can be run by any party
 5. The orchestrator organizes service deployments and changes

TL;DR:

*Our core value to be **open** and **technology agnostic** have had their impact on the system design.*

Introducing Multi Provider Linked Infrastructure (MPLI)

- We're building a specification that will allow CSP's to implement a common service using a common protocol to allow services to be described and delivered automatically.

- This unified interface will allow one user to interact with all CSP's in the same manner.

- This unified interface will exist in a space that CSP's have not moved into, the gap between the customer and the provider.

- It acts as middleware and allows total freedom for providers and consumers to implement their services as they wish and consume their services as they wish.

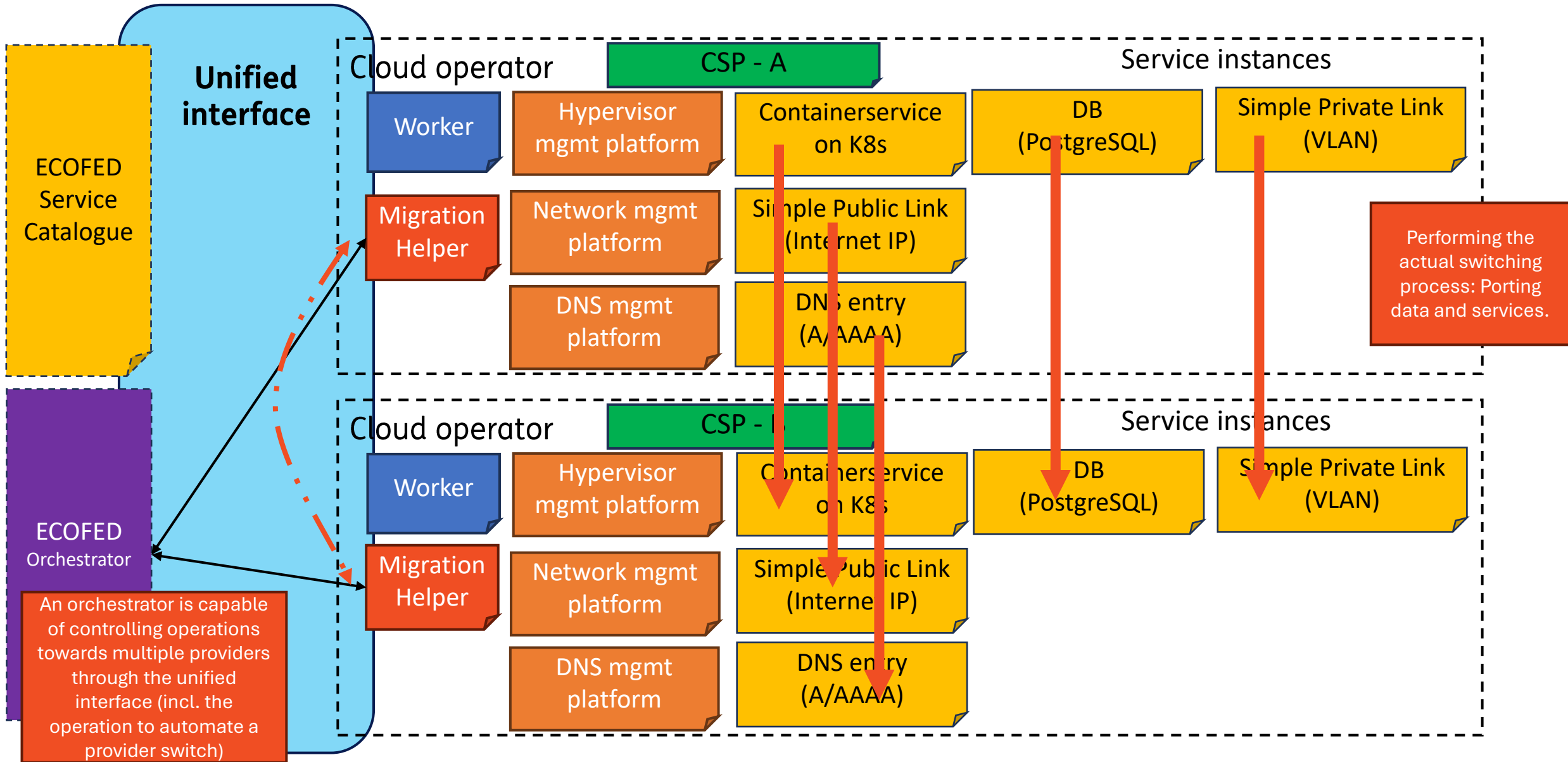- Providers do not need to rearchitect their infrastructure. It is 100% stack agnostic.

Example: Switching from provider A to B
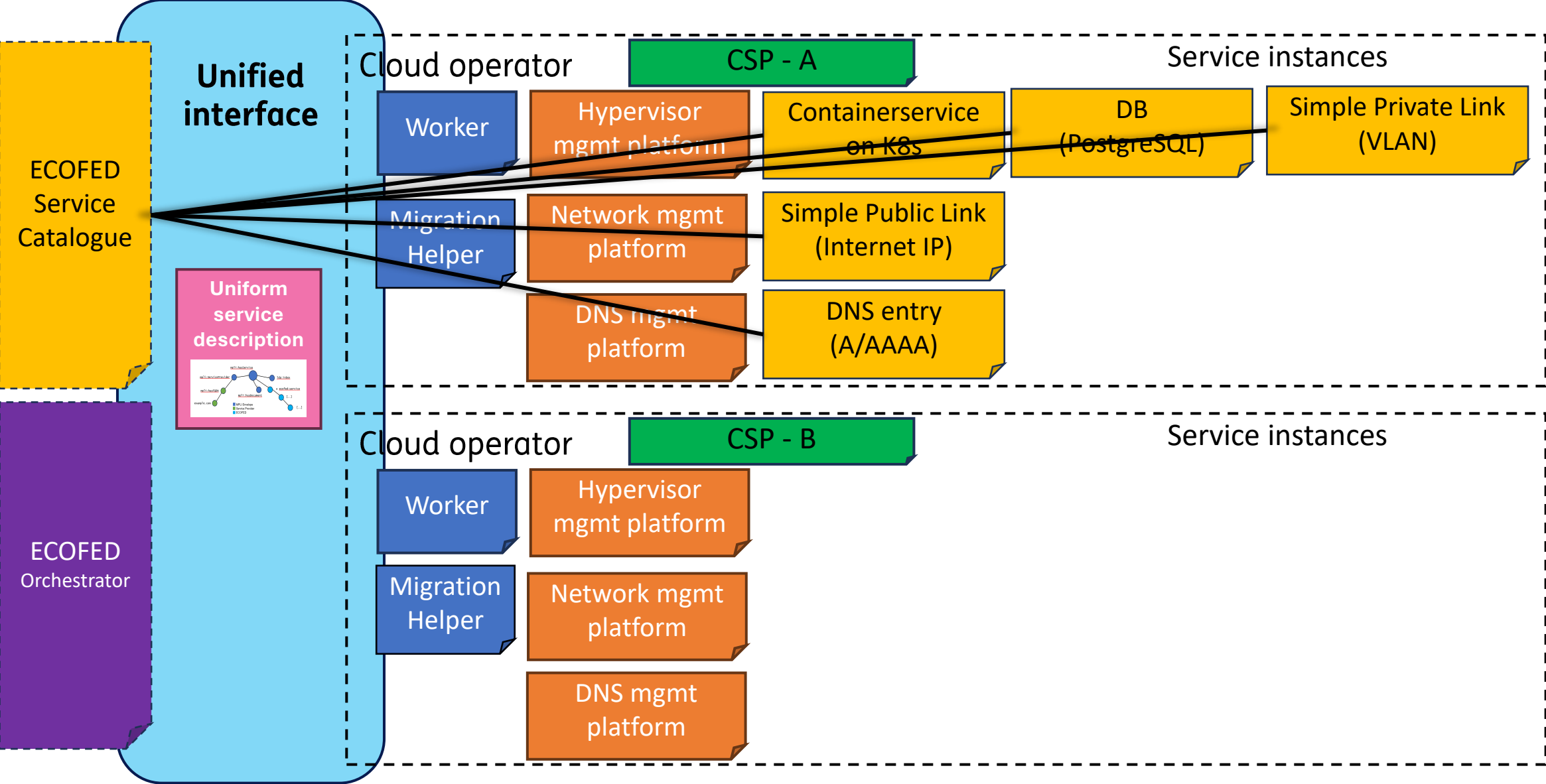
Example: Switching from provider A to B
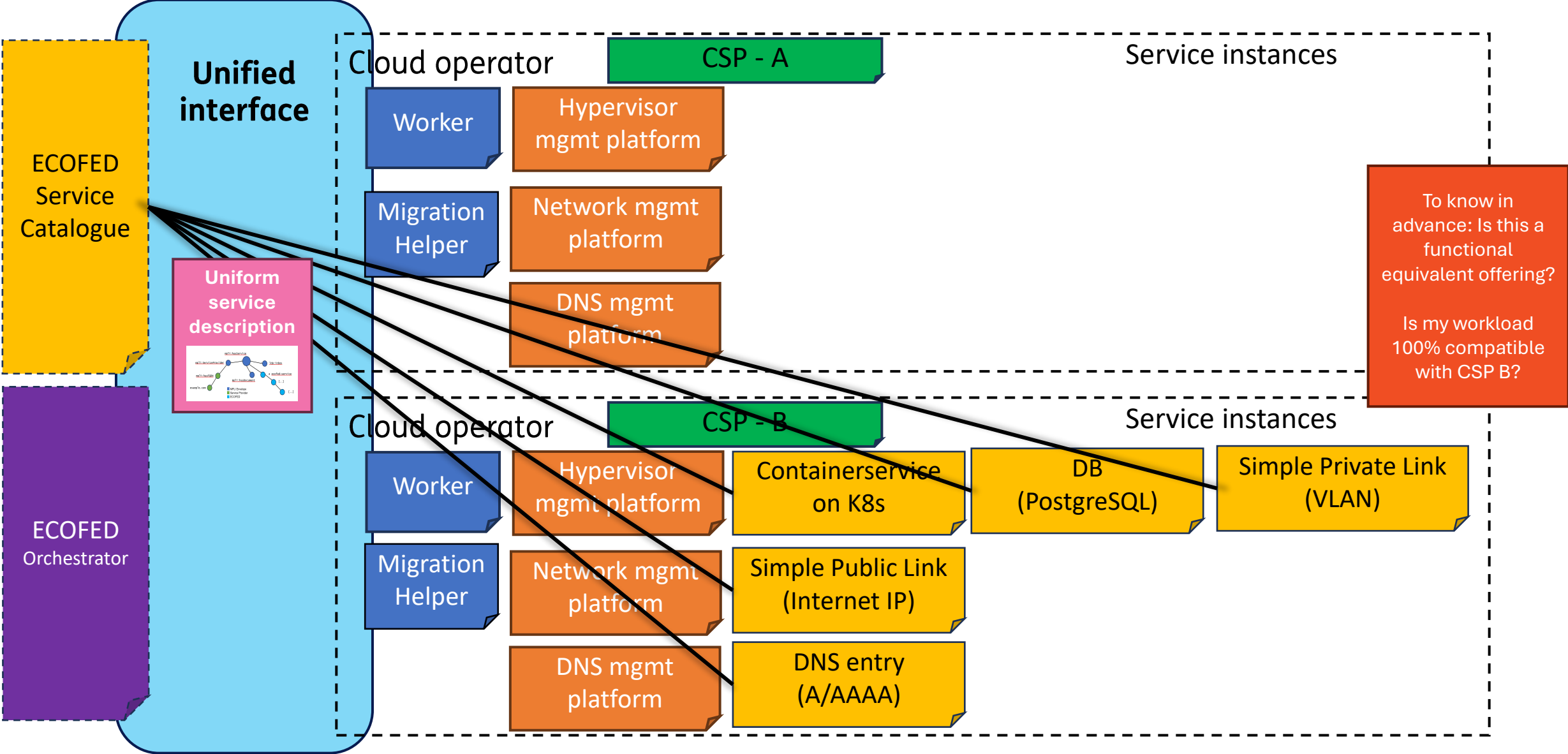
Example: Switching from provider A to B

Our next steps

- Building on top of our successful MVP 1.0 (December 2024)
  - "Backporting our actual experiences into the ECOFED architecture"

- Heading towards MVP 2.0: Stepping from a single VM towards a more complex workload (k8s + managed db)

- Gathering input from the community!

Contact and questions

- More info when available will be shared using the IPCEI-CIS communication possibilities.

- This info will also be published on https://ecofed.eu.

- Drop us an email on info@ecofed.eu.

- We love to hear what you think of our ideas/proposal. Would it be useful for you?

Backup slides

Introducing Multi Provider Linked Infrastructure

- Specially chosen W3C standards are used to work in tandem with each other.

- We use RDF and other graphing technologies to represent all services and service requirement information as a single coherent graph.

- We use Verifiable Credentials to asertain legitimacy with (legal) documents and credentials, to cut down red tape in a federated manner.

- We provide a single coherent REST endpoint for all of this.

What MPLI covers

- MPLI splits the problem domain into five sections:
    1. Discoverability
    2. Compatibility Resolution
    3. Trust
    4. Service Delivery
    5. Portability
- This becomes a fine gradient from bureaucracy to technology.

1. Discoverability

- People need to be able to find things, people need to be able to advertise things.
- The MPLI specification allows each CSP to have available a machine readable manifest that provides a detailed description of every single service sold by them.
- Multiple marketplaces can request this manifest, it is public knowledge, and even internal infrastructure tools can use it.
- Marketplaces can discover CSP's.
- ECOFED has an interest in projects like DOME, as its first target for a compatible Marketplace.

2. Compatibility Resolution

- The same manifest that provides the service description also entails prerequisites for doing business with them, such as an eIDAS VC, it can describe their certifications (ISO) or geographical location.

- A consumer creates, or have a marketplace create it for them, a similar style of manifest that entails the service needs (technical, legal) of this customer.

- MPLI compatible software will take care of comparing the consumer manifest with the manifests of all CSP's.

A sample graph (manifest)
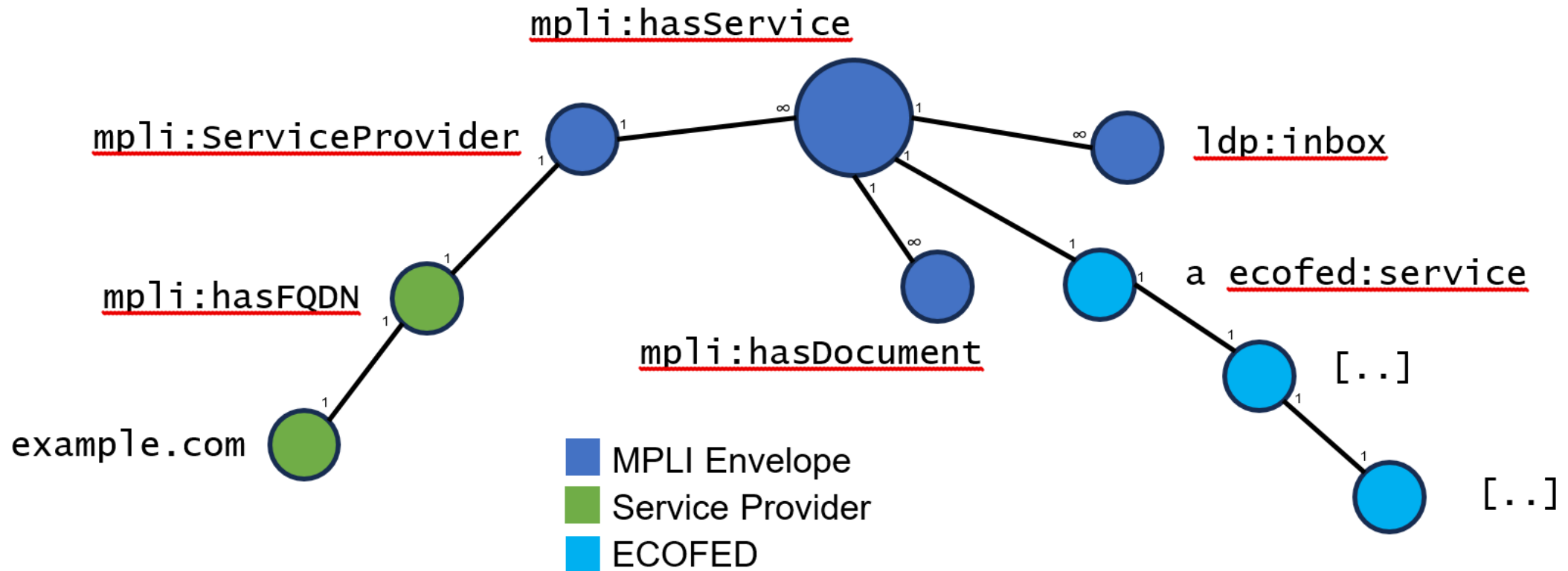
3. Trust

- Trust must be federated too. You can choose who to trust.

- We base our trust system on Verifiable Credentials.

- We initially look towards the Gaia-X Digital Clearing House as a basis for verifying identity and certification documents

- Verification can be automatic, it can include Service Level Agreements or any other required document.

- We intend to leverage this for:
  - Billing
  - SLAs
  - Certification
  - Individual Identity

4. Service Delivery

- The actual provisioning of services will not be done by MPLI.
- Any existing or new infrastructure deployment strategies can be used like Terraform, OpenTofu and Ansible.
- MPLI will tap into these existing provisioning tools and have them take care of the delivery/provisioning.
- A minimum set of new technologies and tools are required to join a federated cloud enabled by MPLI.
- A layered delivery should be possible which allows for white-labeling/automatic use of other MPLI powered services.

5. Portability

- Portability is obliged in the Data Act and MPLI will solve this problem.

- You will be able to move seamlessly/frictionless between services from different CSP's.

- Allows for portability within the current contract with already agreed terms, using the same trust framework, etc.

Current state

- The overall design of MPLI is done.

- MPLI has been validated against different use cases.

- As a proof of concept a virtual machine has been migrated using MPLI from an OpenStack hypervisor to an OpenNebula hypervisor to a custom libvirt-based hypervisor. All three hypervisors were running in different networks not directly connected to each other.