Open Source MANO by ETSI

OCI-Registries for OSM xNFs

Guillermo Gomez (Eviden, an ATOS business)
Guillermo.gomezchavez@eviden.com

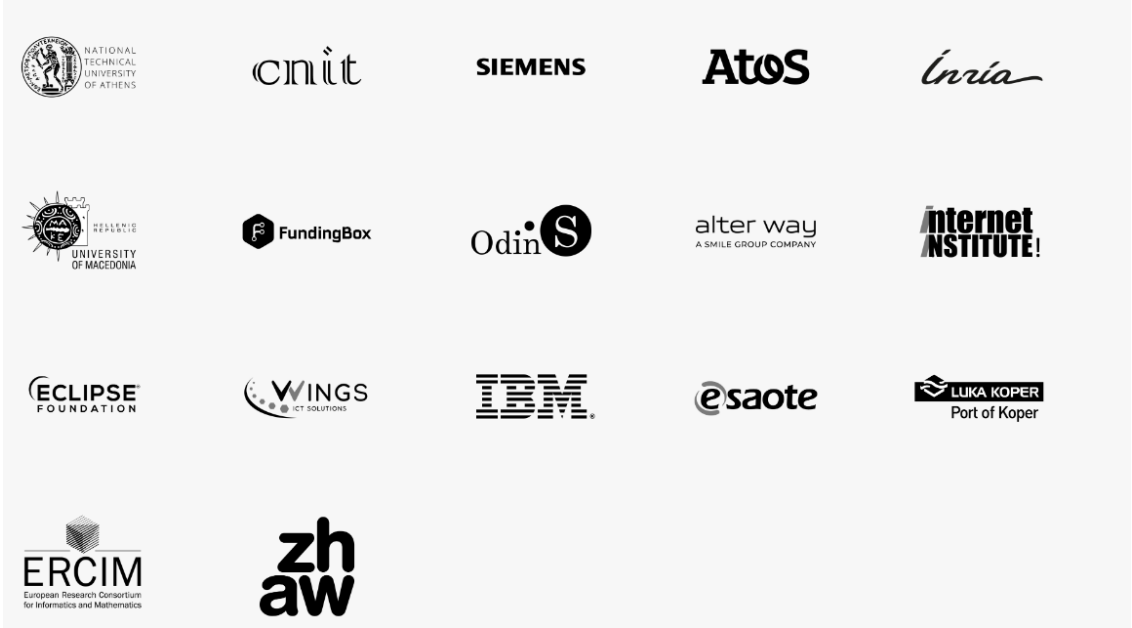
29/11/2023

Agenda

- Nephele Project
- Motivation & Objectives
- Technologies
- Demo implementation
- Demo
- Conclusions and next steps

Nephele Project

- **Research and Innovation Action for "Future European platforms for the Edge: Meta Operating Systems".**
- Enable the efficient, reliable and secure end-to-end orchestration of **hyper-distributed applications (HDAs)** over programmable infrastructure that is spanning across the **compute continuum from Cloud-to-Edge-to-IoT**, **removing existing openness and interoperability barriers** in the convergence of IoT technologies against cloud and edge computing orchestration platforms, and introducing automation and decentralized intelligence **mechanisms powered by 5G** and distributed AI technologies.

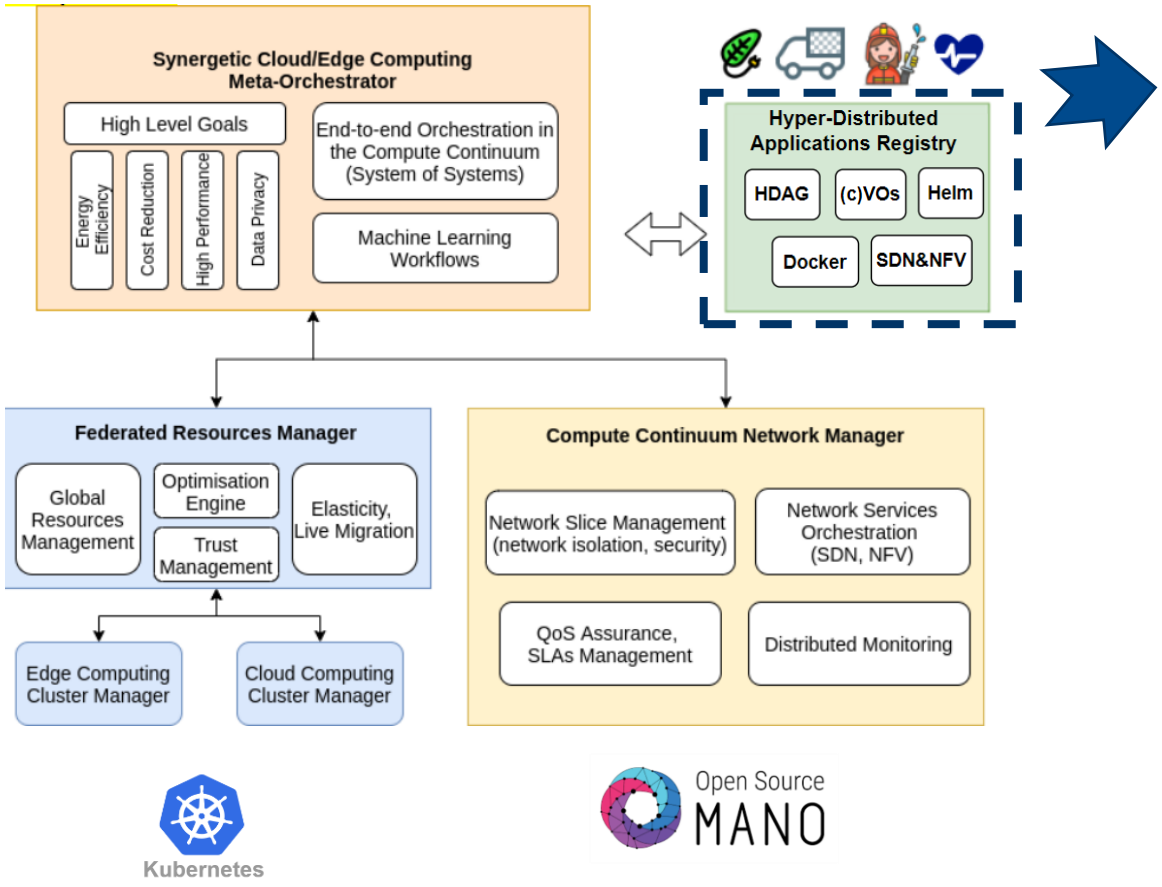


nephele
Part of EUCloudEdgeIoT.eu

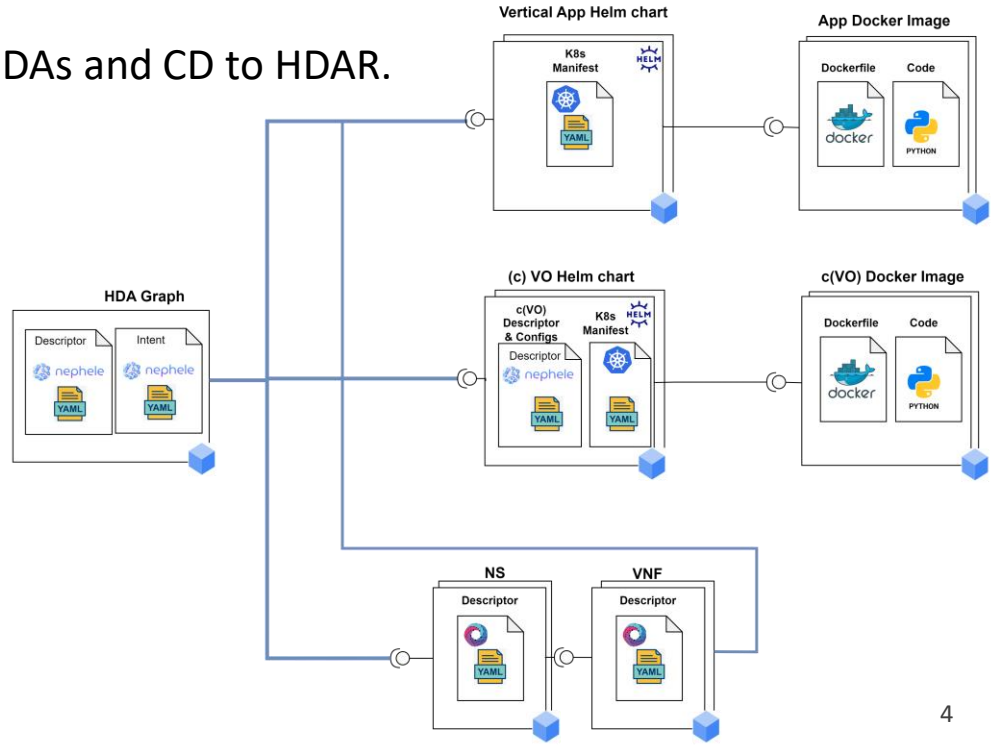
Grant Agreement No. 101070487

Nephele Project

- **Synergetic Meta-Orchestration Framework for the Compute Continuum.**



- **Hyper-Distributed Applications Registry**.
  - HDA: App distributed across the continuum (extremeEdge to Cloud)
  - R: Storage & Distribution System for all artifacts involved in the deployment of a HDA.
- **Development Environment for HDAs**
  - CI of HDAs and CD to HDAR.

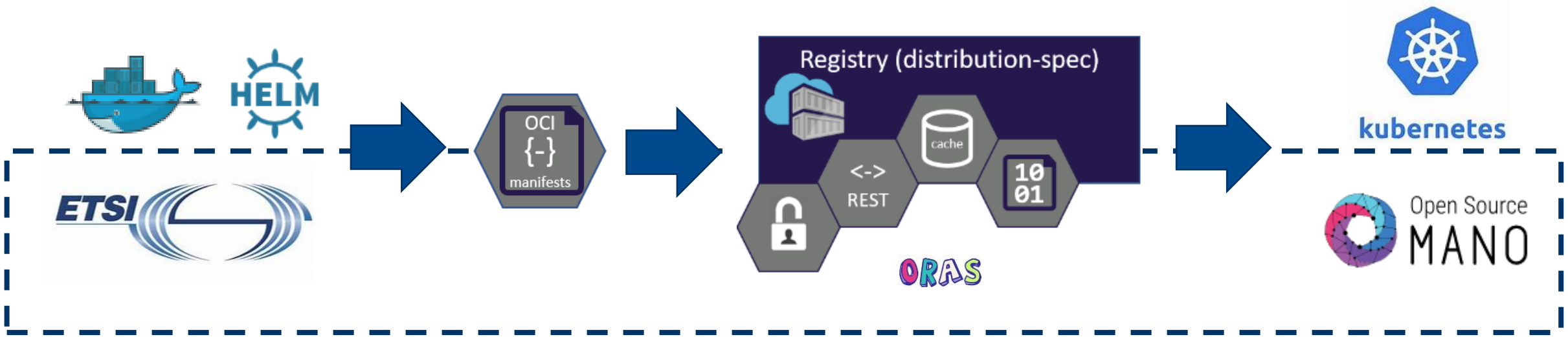


Motivation & Objective

Motivation: Simplify and improve the development and management (storage and distribution) of xNFs artifacts (NS, VNFs, Slices) in simple and complex scenarios by **leveraging commercial-grade technologies from the CN ecosystem.**

- Simplify and unify artifact management in OpenSource codebase.
- Contribute to the **ETSI & CNCF alignment**: ETSI & CNCF agreement renewal, OCI registry for Helm OCI registries for helm (#10997) · Issues · osm / Features · GitLab (etsi.org)

Objective: To demonstrate how we can **harmonize the storage and distribution mechanisms used for artifacts** in the Telco and CN industries **significantly improving both, the developer and the stakeholder experience and interoperability**.

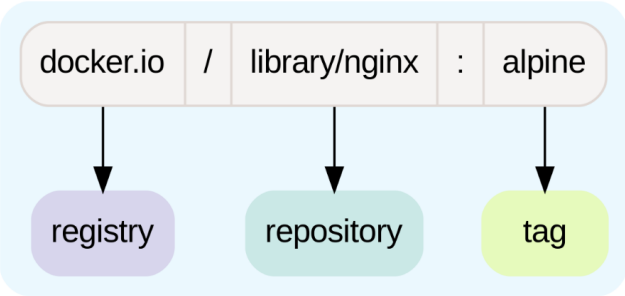


Demo

Technologies

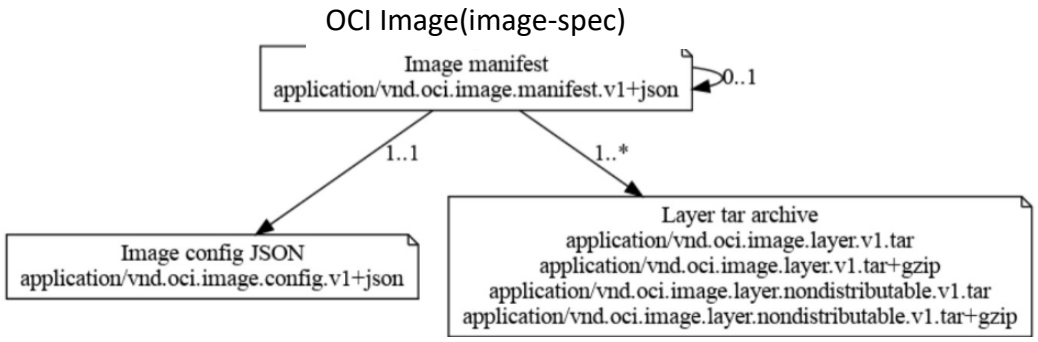
Background:

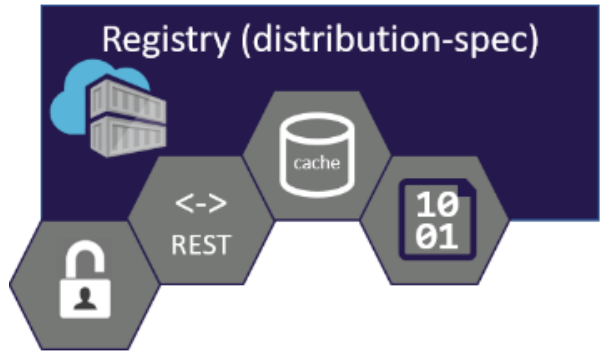
- **Artifact**: Distributable outcome of a software development in the form of a filesystem bundle, essentially, anything that a system or user might need to successfully run a software.
- **Registry**: A registry is a storage and distribution system for those artifacts
- **Repository**: A registry is organized into repositories, where a repository holds all the versions of a specific artifact



OCI Specifications:

- The Runtime Specification (**runtime-spec**) details how to run a "filesystem bundle" that is unpacked on disk.
- The Image Specification (**image-spec**) outlines how to build and process the "filesystem bundle" which results in the creation of an OCI Image.
  - The **image manifest** structure and mediaType is fixed to be compatible with the distribution-spec implementers
  - The **image config manifest** and mediaType is what enables the runtime-spec implementers to know if the image is meant for their system and how to process the filesystem bundle. → This is what needs to be standardize for NFV artifacts.
- The Distribution Specification (**distribution-spec**) defines an API protocol to facilitate and standardize the distribution of OCI images.





Demo implementation

nephele

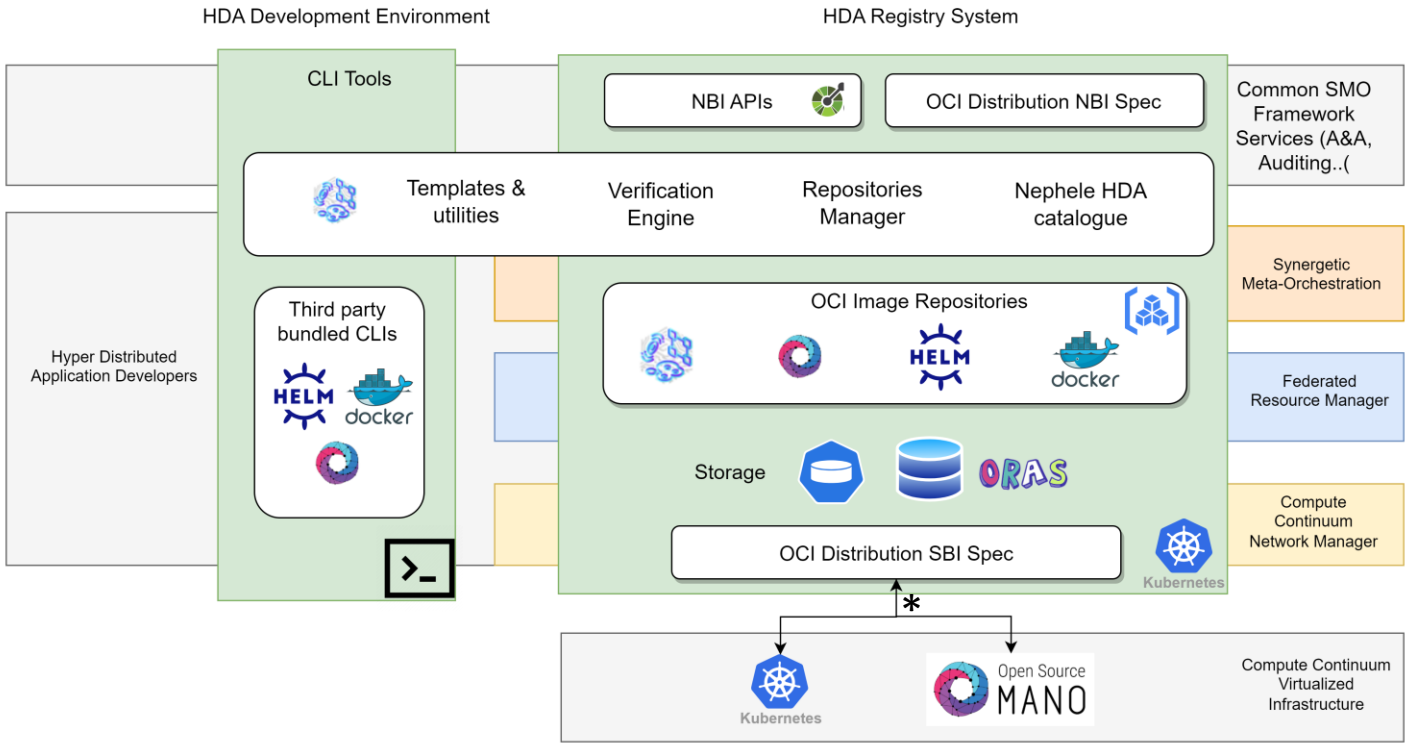
Grant Agreement No. 101070487

The HDAR CLI (inspired in Helm CLI) implements the custom image-spec for NEPHELE and OSM artifacts to Pull&Push them to the OCI-Registry. Supports

- **HDAG, (c)VOs, NS, VNF**
- Other CLIs support: **Helm, SBOM, Digital signatures, Provenance data, Attestations and Vulnerability reports [OCI artifacts on Docker Hub | Docker Docs]**

The HDAR System consist of (regarding OSM):

- An **OCI-Registry** (distribution-spec) which stores custom OCI images**, including OSM artifacts (NS and VNFs).**
- **NFV Config JSON proposal with supporting info for deployment and verification.**
- Integrated custom CLI and OSM CLI to onboard artifacts to OSM.



* Helm pull expected by OSM Rel15.

* OSM pull under development and to be proposed as feature.

Demo

OSM VNF image-spec image manifest JSON


```
{
    "schemaVersion": 2,
    "mediaType": "application/vnd.oci.image.manifest.v1+json",
    "config": {
        "mediaType": "application/vnd.etsi.nfv.config.v1+json",
        "digest": "sha256:c8afbc0b197e8d4f56374412b68bb823aa8179d909ceb6f492cba605f9ed89cd",
        "size": 421
    },
    "layers": [
        {
            "mediaType": "application/vnd.etsi.nfv.config.v1+json",
            "digest": "sha256:c0a70ca449b9c570280f750d9b9b4f87a934746b182d86be5021abfd186291b5",
            "size": 321,
            "annotations": {
                "org.opencontainers.image.title": "openldap_knf-1.0.tar.gz"
            }
        }
    ],
    "annotations": {
        "org.opencontainers.image.authors": "Telefonica",
        "org.opencontainers.image.created": "2023-11-21T11:34:50Z",
        "org.opencontainers.image.description": "KNF with single KDU using a helm-chart for openldap version 1.2.3",
        "org.opencontainers.image.title": "openldap_knf",
        "org.opencontainers.image.version": "1.0"
    }
}
```


OSM NFV image-spec image config JSON


```
{
    "schemaVersion": "1",
    "mediaType": "application/vnd.etsi.nfv.config.v1+json",
    "packageSpec": "SOL004",
    "dataModelSpec": "SOL006",
    "descriptorFile": "openldap_knf/openldap_vnfd.yaml",
    "descriptorType": "VNF",
    "descriptorId": "openldap_knf",
    "descriptorVersion": "1.0",
    "dependencies": [
        {
            "descriptorType": "HELM",
            "descriptorId": "stable/openldap",
            "location": "external",
            "mode": "soft"
        }
    ]
}
```

Conclusions and next steps

While most of the existing implementations for any of the three OCI specifications focus on containerized workloads in the context of Kubernetes. **The image-spec and the distribution-spec are by no means limited to K8s and has already been adopted by multiple projects** OCI artifacts on Docker Hub | Docker Docs.

Using an OCI-Registry to store OSM artifacts **simplifies the developers work and greatly contributes to the interoperability of xNF providers and MNOs**.

What could come next?

- Simplification of the OSM CLI codebase: **Unification of the K8s and OSM Repositories codebase.**
- Creation of a xNF Martketplace and **upgrade of the OSM Packages GitLab repository**.
- **Public and Private Registries**, externally and internally.
- Quick support for **xNF artifacts versioning.**
- OCI-Registry replication and proxies for multi-stakeholder environments.
- **OSM storage profile reduction**:
  - Offloading to specialized commercial-grade services.
  - Layer blob sharing, automated garbage collection.
  - Pull-based orchestration.

Open Source MANO by ETSI

Thank You!