Open Source MANO

by ETSI

Scalable Cloud-Native Operation of Multi-Cloud Deployments

Francisco-Javier Ramón

(Telefónica, ETSI OSM Chair)

13/11/2024

Open Source MANO

Manage infrastructure, platforms and applications across cloud platforms in the easiest way.

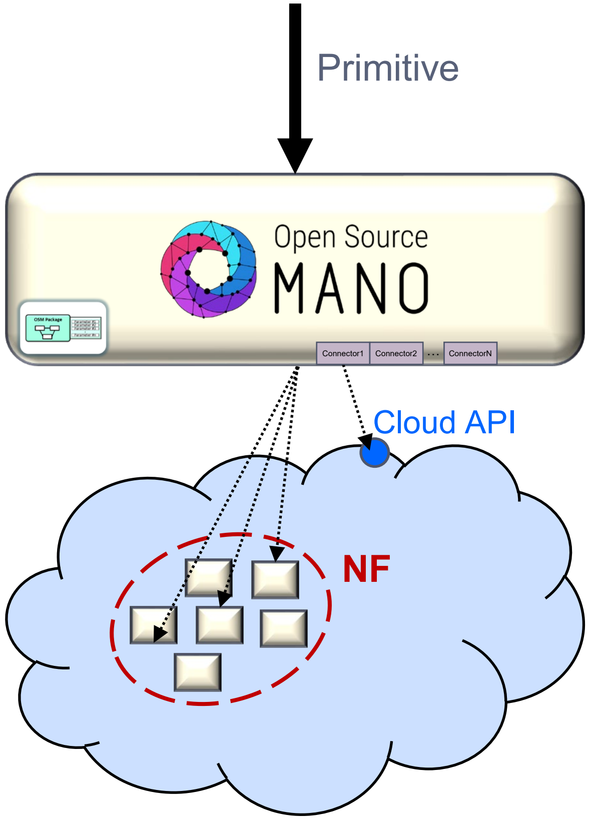
OSM provides a platform to create Network Services dynamically and to manage them conveniently later

OSM manages the low-level setup for **Network Functions** and **Cloud Infrastructure**, so that they are ready for use.

- It covers in 100% the role of a kind of **specialized PaaS for Network Functions**, with 2 key features:
  1. **Complex connectivity** setup, including EPA and underlay network scenarios.
  2. Solve **inter-NF relations**.
  3. Create any cloud infra required for the Network Service.

- Returns: **Service ready for its use and properly connected**:
  - Exposes **the "service" and its lifecycle, not its components**.
  - Presented as a whole (i.e., abstracts from low-level details).
  - Easy (standardized) access to NS's lifecycle.

- This follows well-known paradigms in **IT** and **public clouds**.



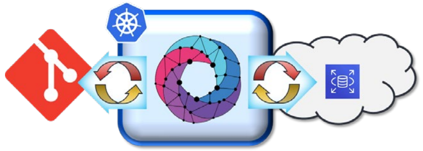
Release SIXTEEN brings a revolution in OSM's internal architecture, unlocking a huge set of features

Open Source MANO

Release SIXTEEN

Available at:
osm.etsi.org

Cloud-native operations in OSM



- Management cluster for cloud-native management of infra and applications.
- Workflow for cloud-native operations in OSM following GitOps model.
- VIM account registration for cloud-native operations.
- Setup of Git repo during OSM installation for continuous deployment operation.

Management of Kubernetes clusters

- Full life-cycle cluster management from OSM.
- Management of PaaS clusters from public clouds
- Infrastructure profiles for K8s clusters.
- Update profiles in a cluster.
- App profiles for K8s clusters.
- Kubernetes SW units deployable over K8s clusters.

Security enhancements

- Modification of audit logs in NBI for password change and NS operations.
- Password recovery in OSM.

Enhanced operational capabilities

- Add option for CNF upgrade to reset values for upgrade operation on helm charts.
- NS config templates as first-class citizens in OSM.
- NBI support for deletion of multiple NS Instances from OSM UI.
- Add labels to Kubernetes objects created by OSM.
- Enhancement of vertical scale feature and merge in update API.
- Service KPI Metric Based Scaling of VNF using exporter endpoint in NGSA.

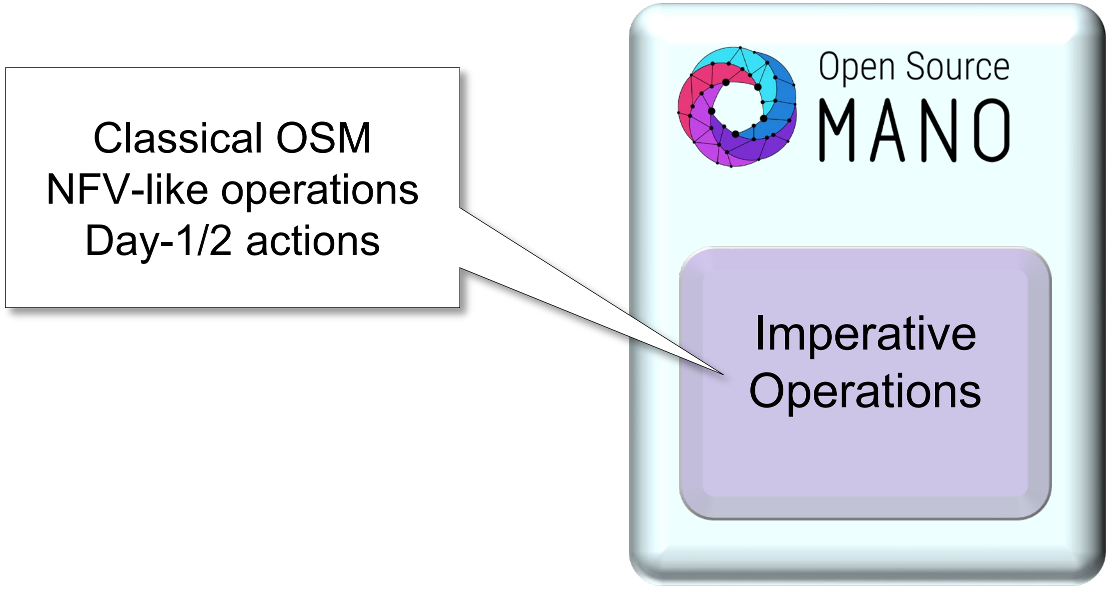
OSM installation

- Installation of ingress controller in OSM community installer.
- Enable K3S as Kubernetes distro for OSM installation.
- Publication of OSM helm chart externally in Gitlab.
- Use of upstream helm charts for Prometheus and Grafana in OSM installation.
- Removal of Zookeeper from OSM installation.
- Integration of OSM Helm Chart with different databases.

With Release SIXTEEN, OSM has extended substantially its scope and architecture

Open Source MANO

Release FIFTEEN

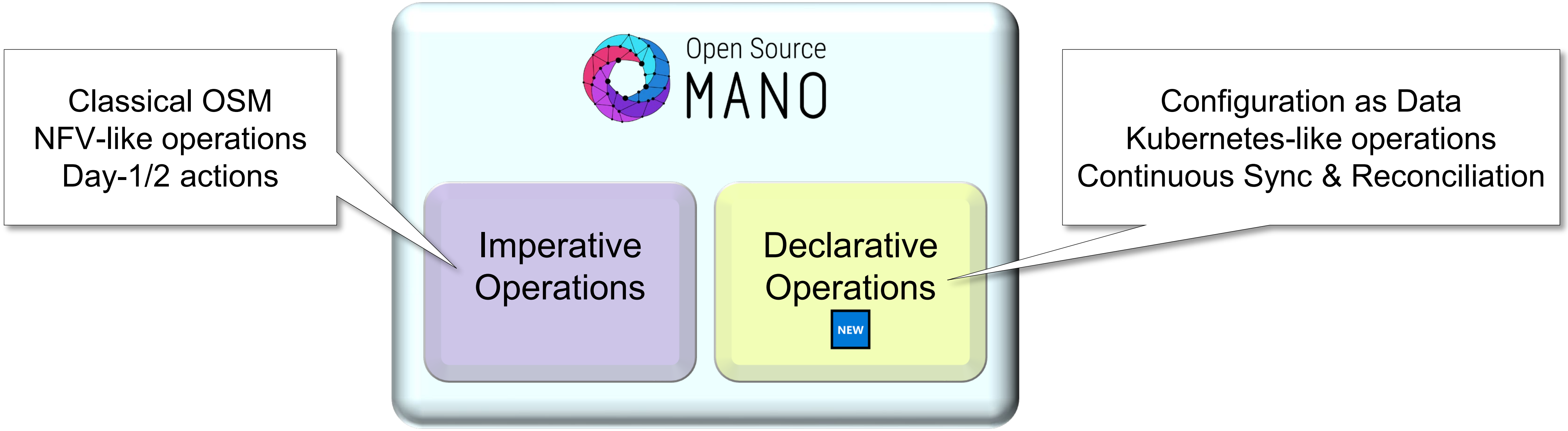


With Release SIXTEEN, OSM has extended substantially its scope and architecture

Open Source MANO

Open Source MANO

Release SIXTEEN

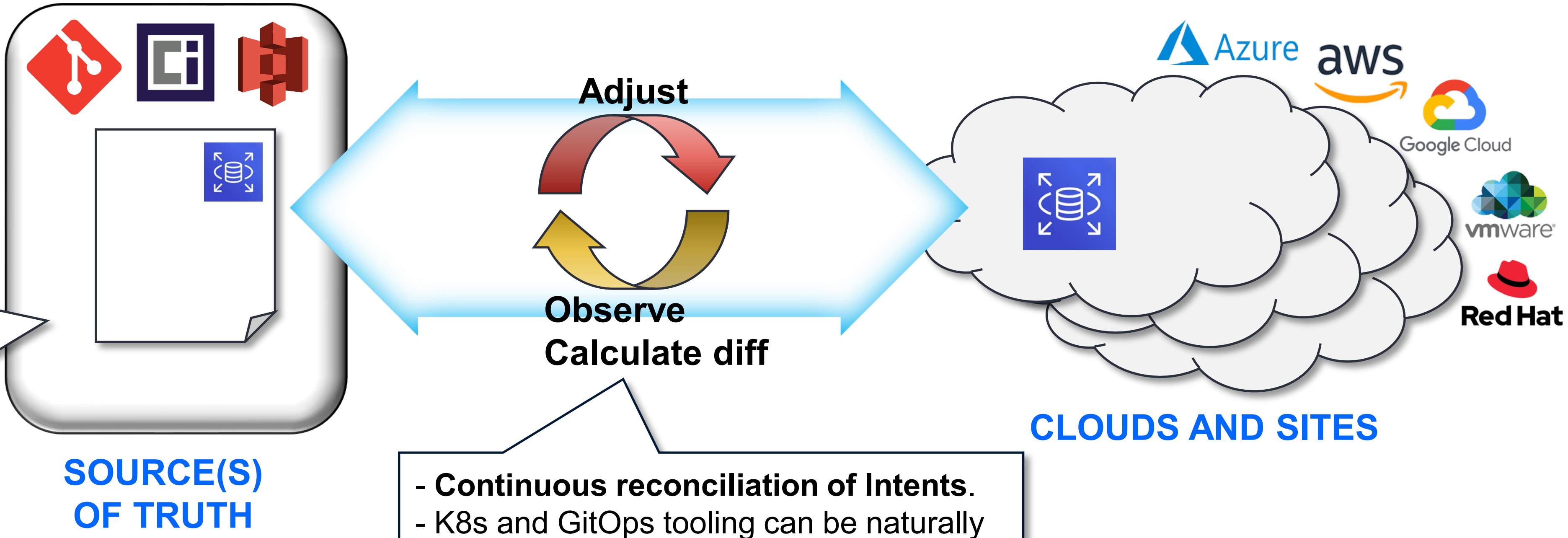


Classical OSM
NFV-like operations
Day-1/2 actions

Configuration as Data
Kubernetes-like operations
Continuous Sync & Reconciliation

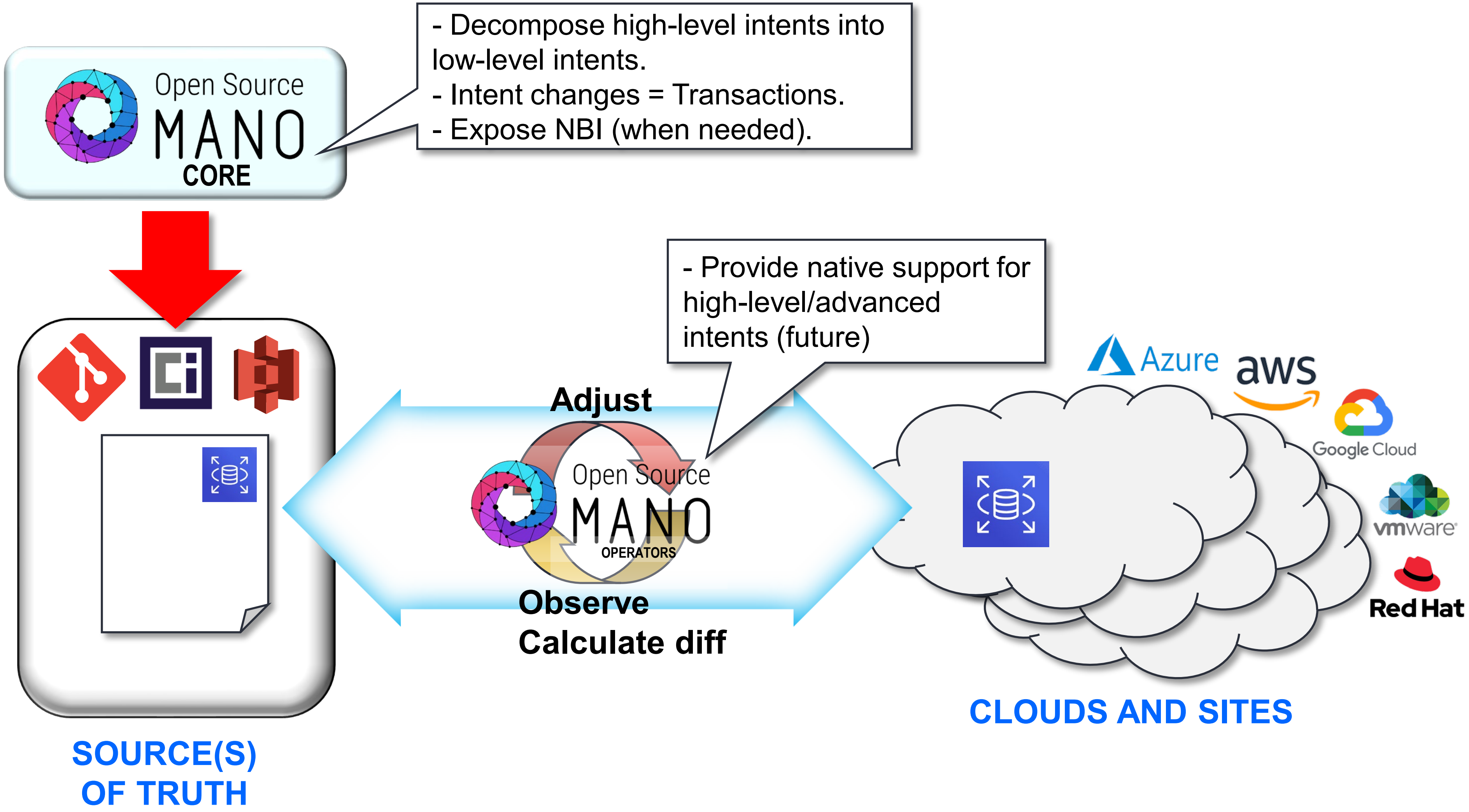
What are declarative operations about?

- **Safe storage** for all Intents.
- History **versioned**, may work as “time machine” for rollbacks.
- Tight **authoring & access** control.



- **Continuous reconciliation of Intents**.
- K8s and GitOps tooling can be naturally applied here.

What are declarative operations about?



- Decompose high-level intents into low-level intents.
- Intent changes = Transactions.
- Expose NBI (when needed).

- Provide native support for high-level/advanced intents (future)

When applicable, OSM’s *declarative operation* solves elegantly some practical challenges

OSM’s IMPERATIVE OPERATION

- “Desired state” based on a sequence of actions
  - **New deployment = Re-doing steps**
  - Case-specific **rollbacks** (how to “undo” an action?)
  - **Debugging** requires figuring out state first.

- Onboarding often requires
  - Deep app-specific knowledge
  - NFV-like adaptations

- OSM code can leverage on Python libraries

OSM’s DECLARATIVE OPERATION

- “Desired state” unambiguously captured
  - **New deployment = Copy state**
  - Trivial **rollbacks** (use *history of states*)
  - **Debugging** starts by checking state.

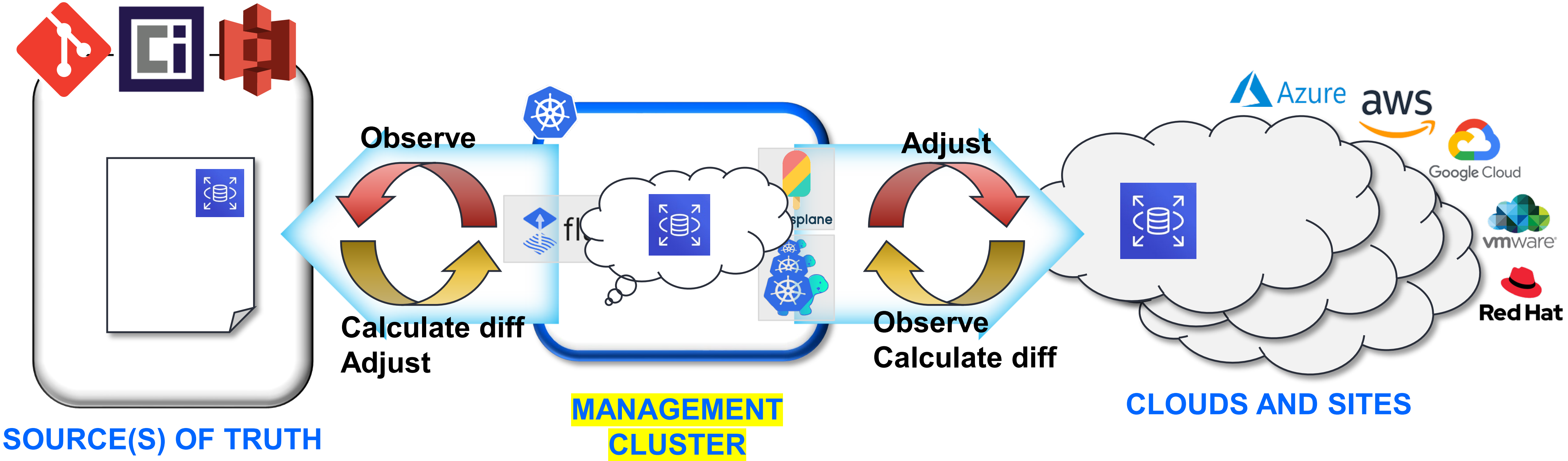
- Simple onboarding of cloud-native components
  - Reusability of many pre-existing artifacts.
  - Shallower learning curve.

- OSM code can leverage on both Python libraries and K8s operators
  - High-level support for a huge variety of cloud resources

The **New OSM** behind the hood

OSM’s declarative operations behind the hood

- Syncs are delegated to **Management Cluster** and **Workload Clusters**.
  - **Flux:** Sync with Sources of Truth (Git, OCI repos, object storage, etc.).
  - **Crossplane**, **CAPI**: Sync with target clouds (Management Cluster only).



Why naïve GitOps is not enough?

Need of granting synchronization order

- E.g., A new K8s cluster needs to be ready before attempting to deploy an App.

Massive duplication of intents for large deployments

- Maintenance of duplicated intents may be a recipe for disaster.

Grant protection of sensitive data, so that it is never saved in clear (e.g., secrets)

Creation of resources in the clouds

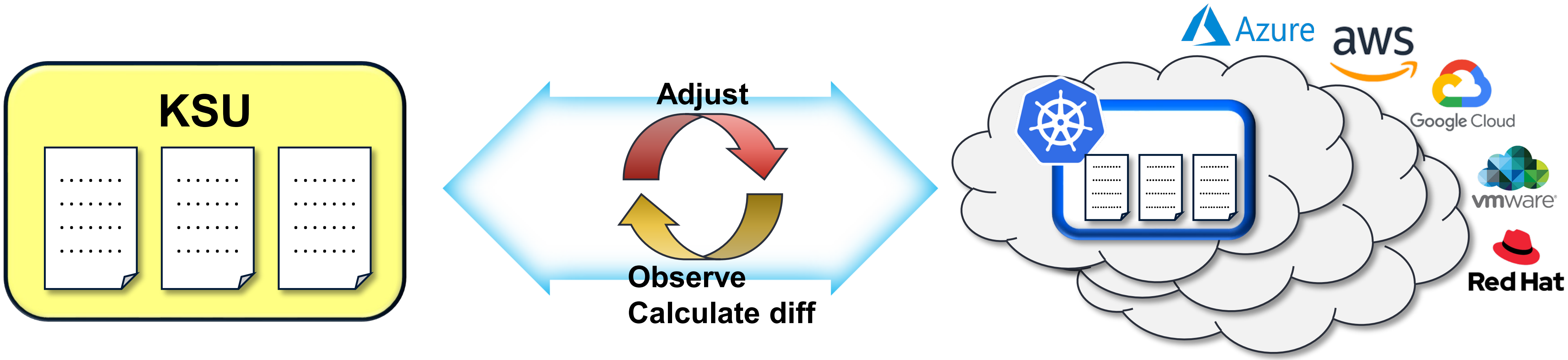
Bootstrap of remote clusters, which, in turn, become "resource containers"

- One cluster (*Management*) may create others (*Workload*), which, in turn, may host Apps/CNFs.

Under these premises, complexity grows wildly with size!

New OSM's concepts:
KSU as minimal unit of state to sync

- KSU = Kubernetes Software Unit
  - Set of manifests.
  - May be based or include references to an OSM blueprint (OKA).

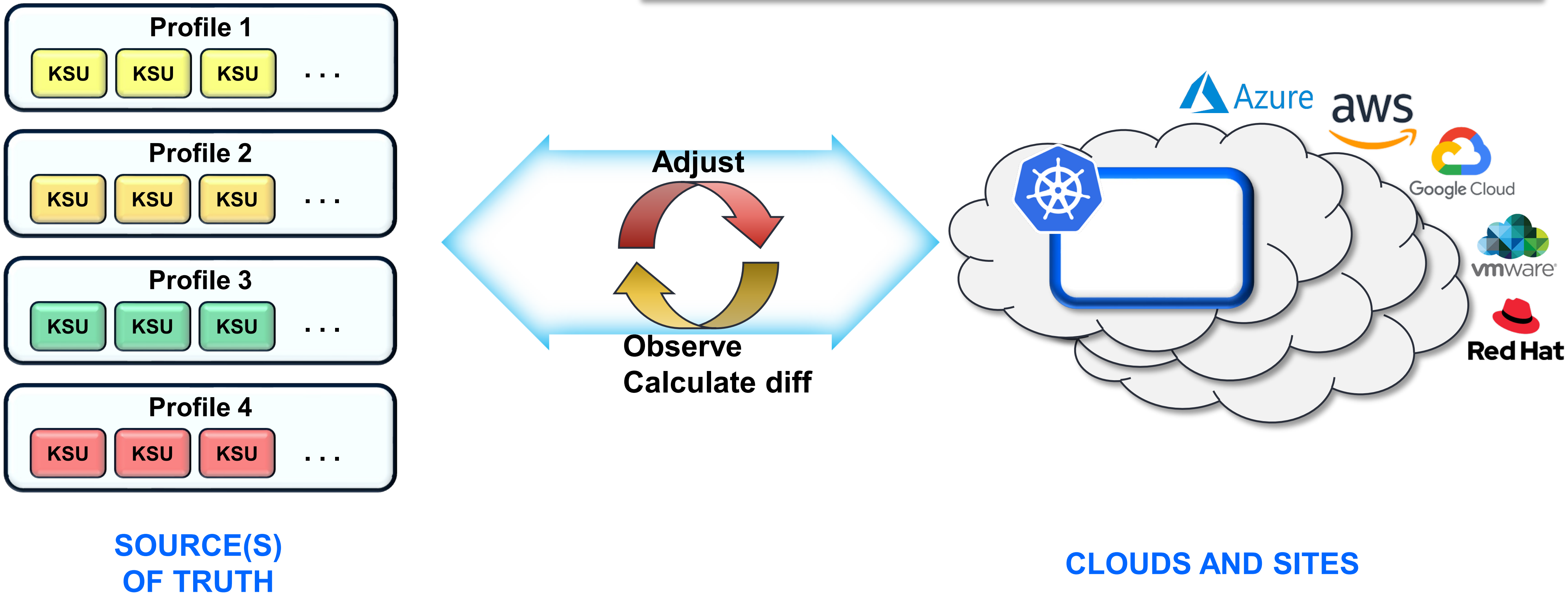


SOURCE(S) OF TRUTH

CLOUDS AND SITES

New OSM's concepts: **Profile** collects KSUs to be sync'd together

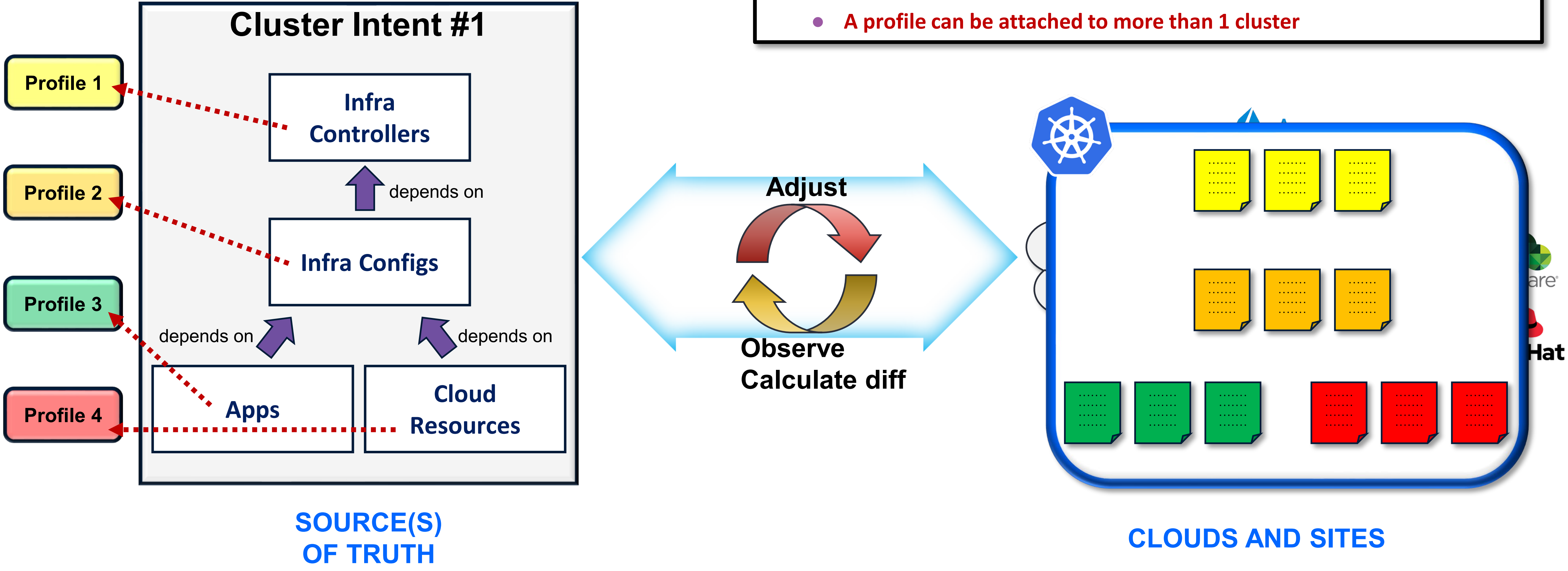
- Profile = Collection of KSU to be sync'd together
- Ready to attach to a layer of synchronization of clusters



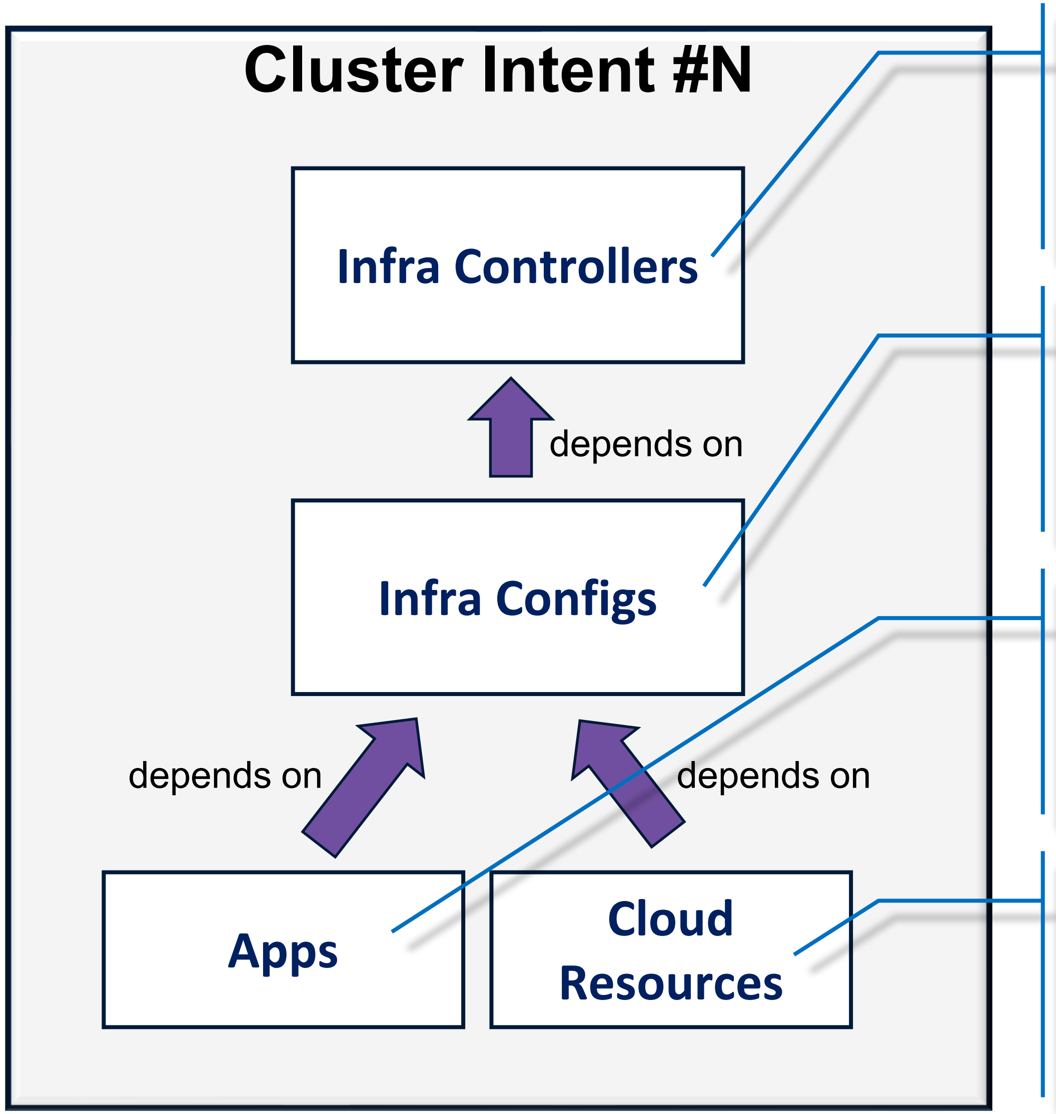
New OSM's concepts:
Cluster Intent refers to Profiles to apply to a given cluster

Open Source MANO

- Each cluster has 4 layers of synchronization
- Profiles are ***attached*** to layers of synchronization
  - **A profile can be attached to more than 1 cluster**



New OSM's concepts:
Cluster Intent refers to Profiles to apply to a given cluster



Infrastructure add-ons that need to be present in the cluster so that it has the right functionality.

- **Examples:** Multus Controller, Cert-manager, Service Mesh, K-Native, Ingress Controller, Flux Controller, Crossplane Controller, etc.

Add-on configurations, **service account setups**, or any other tunning required for the cluster services to be ready.

- **Examples:** Multus configuration, setup of cert authority, Service Mesh config, Crossplane ad-hoc Compositions, etc.

Regular **K8s Apps** and/or **CNFs**.

Main case for regular **Workload Clusters**.

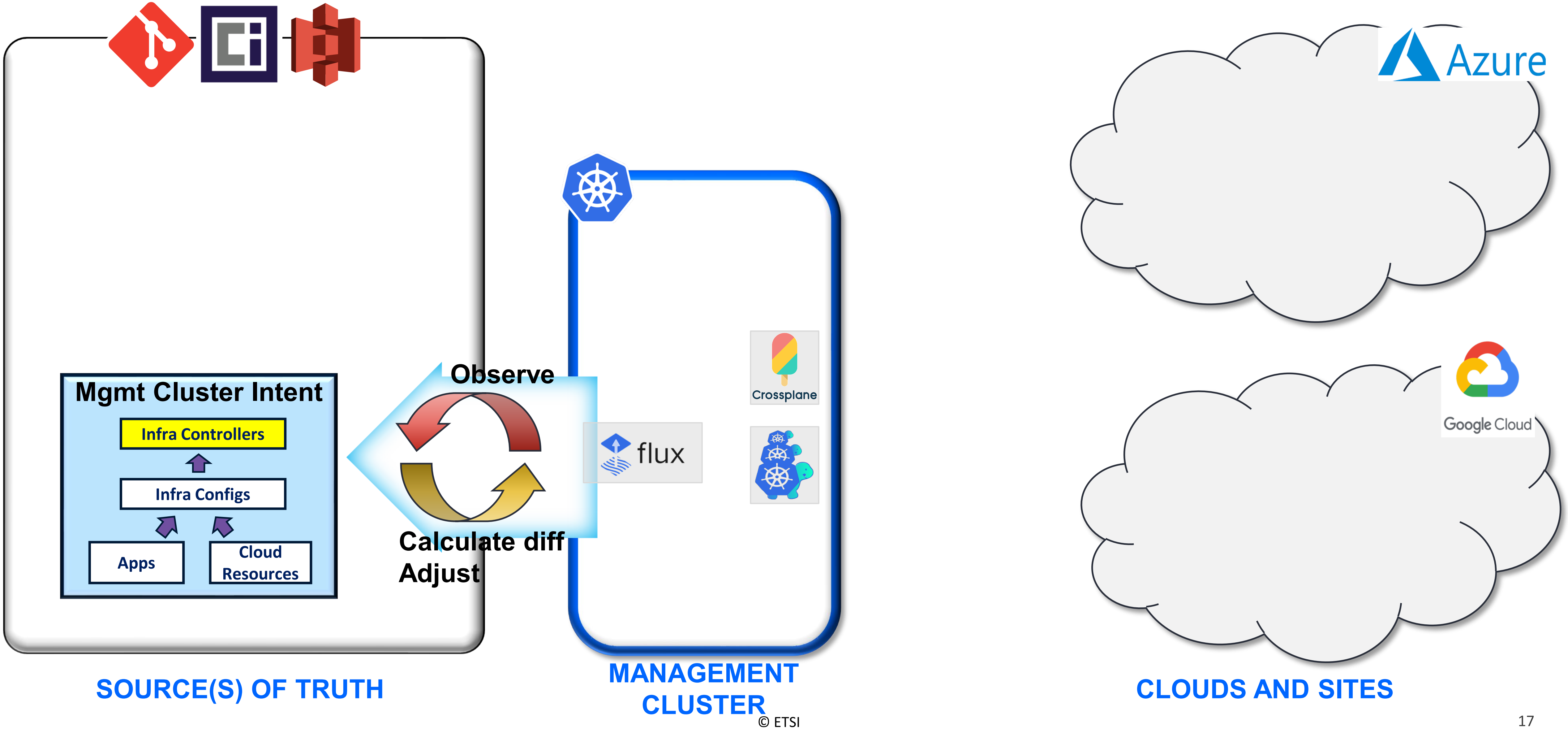
- **Examples:** 5GCore, IMS, MySQL, GitLab, etc.

K8s objects representing cloud resources (quite often, PaaS/SaaS).

Main case for the **Management Cluster(s)**.

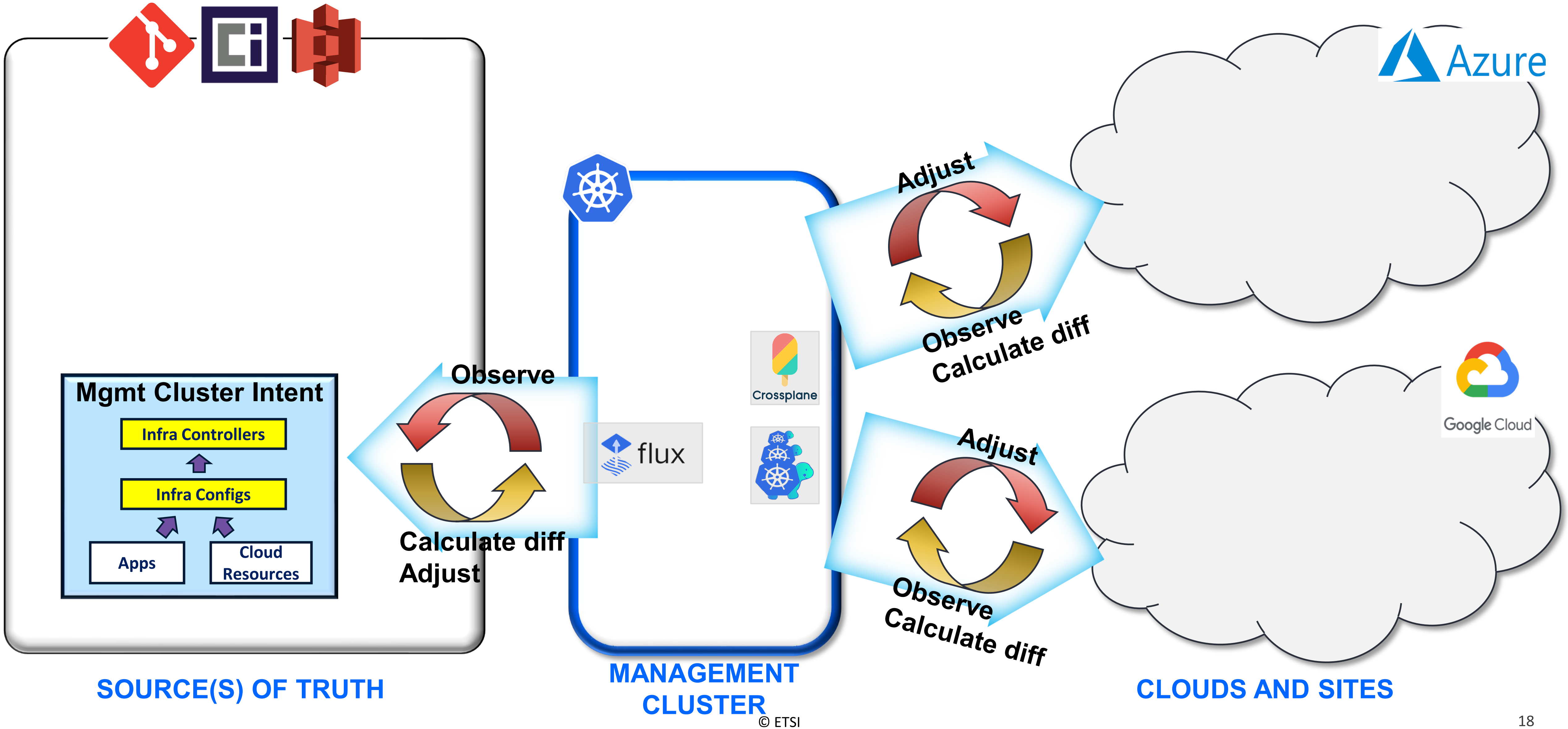
- **Examples:** K8s clusters, OCI registry, cloud databases, S3 storage, etc.

Management Cluster vs. Workload Clusters

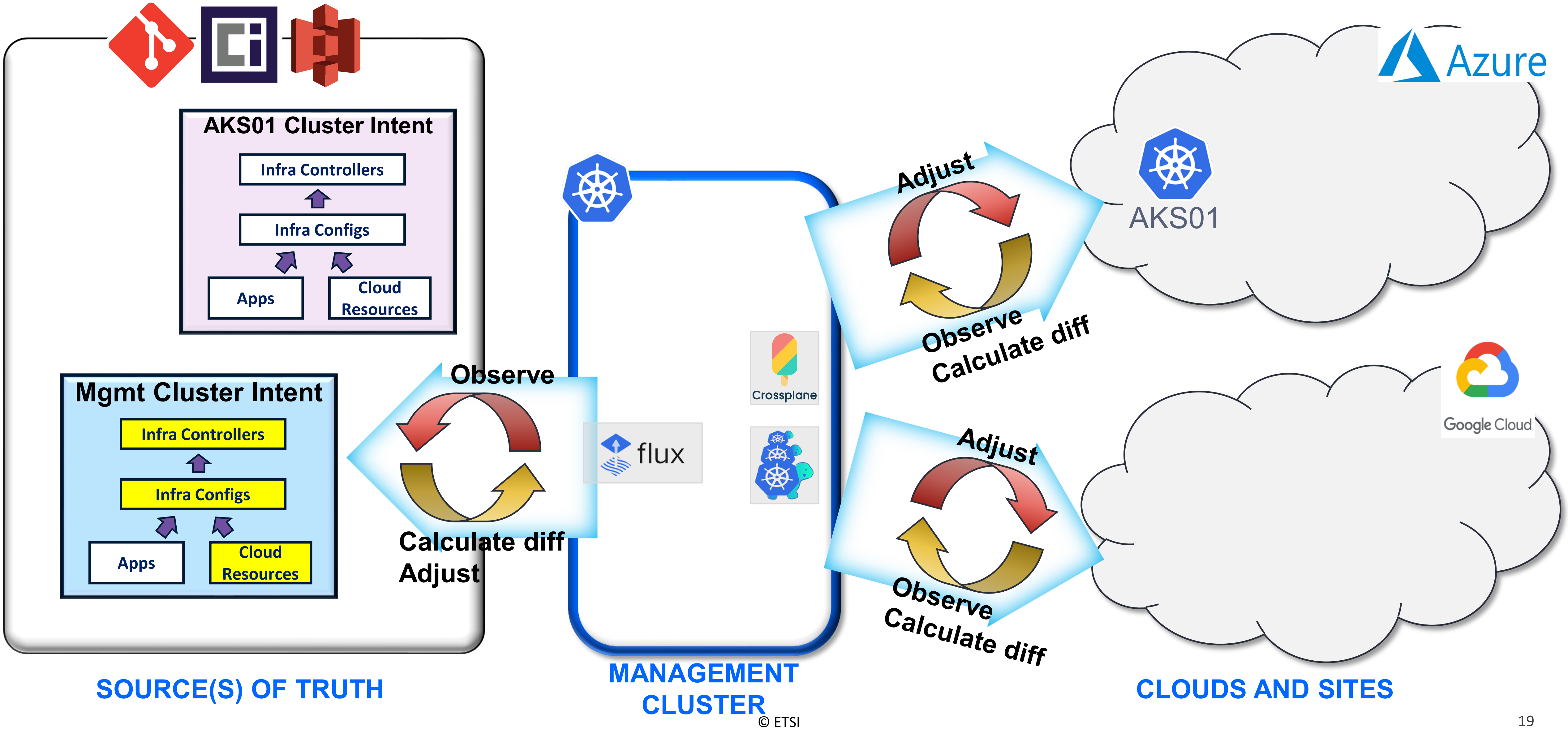


© ETSI

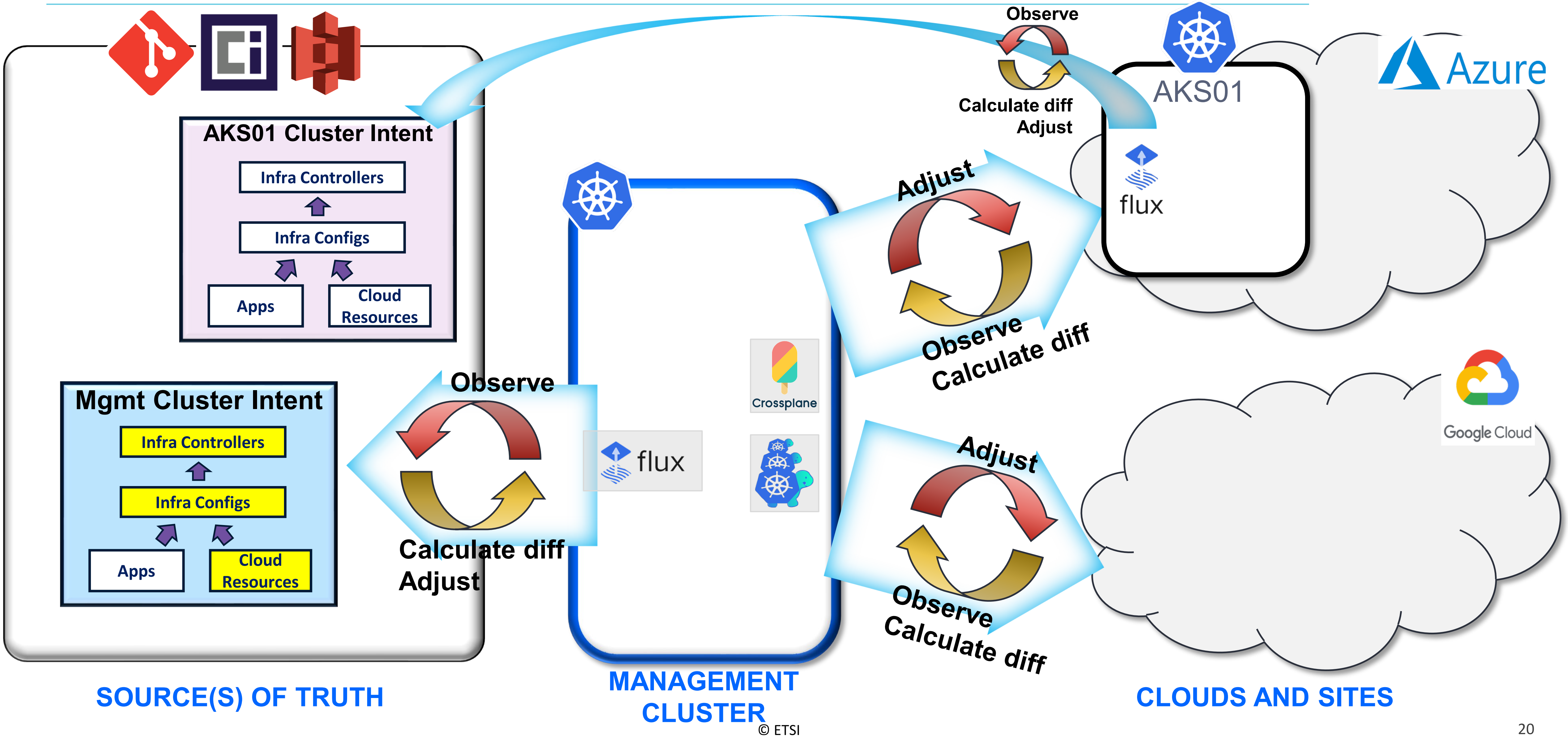
Management Cluster vs. Workload Clusters



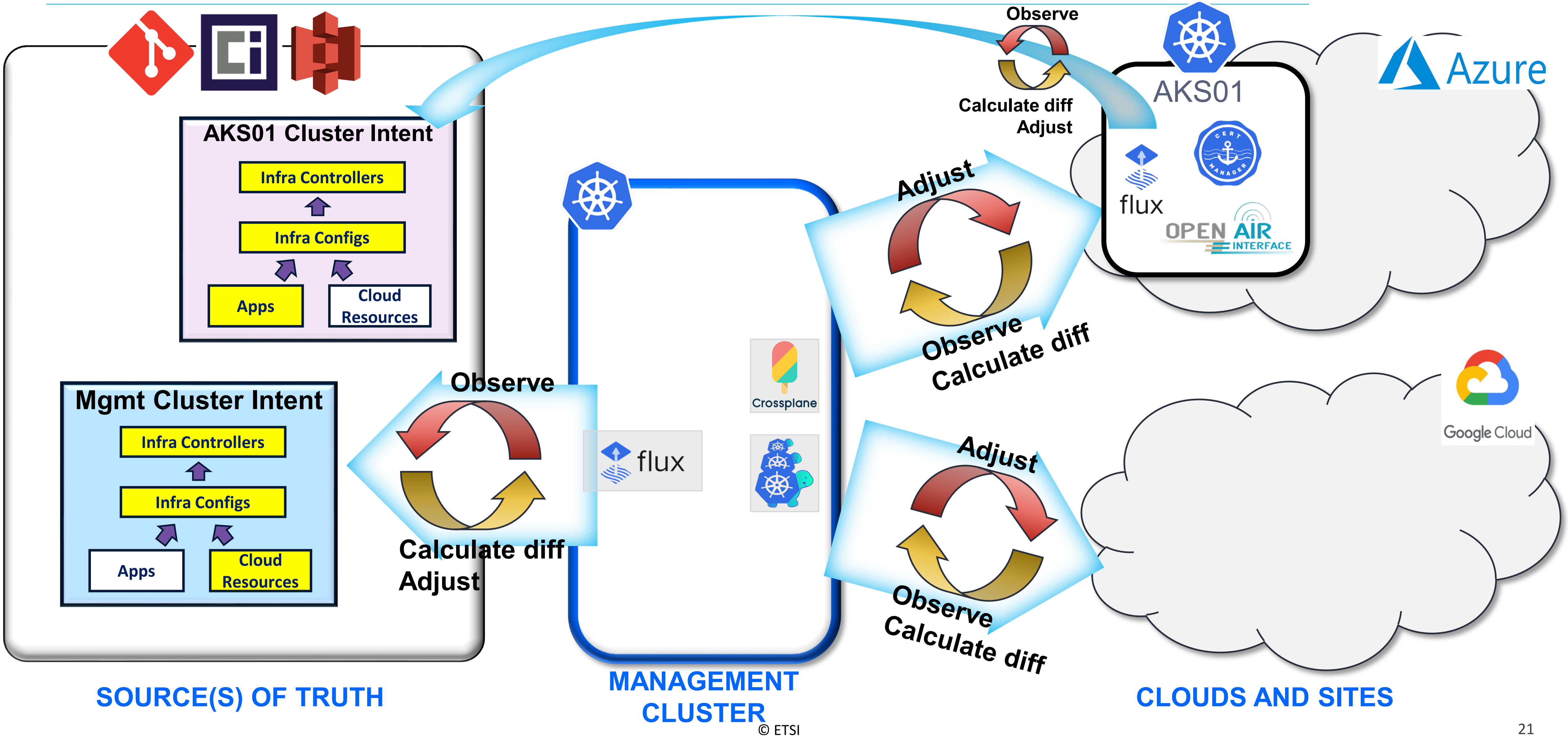
Management Cluster vs. Workload Clusters



Management Cluster vs. Workload Clusters

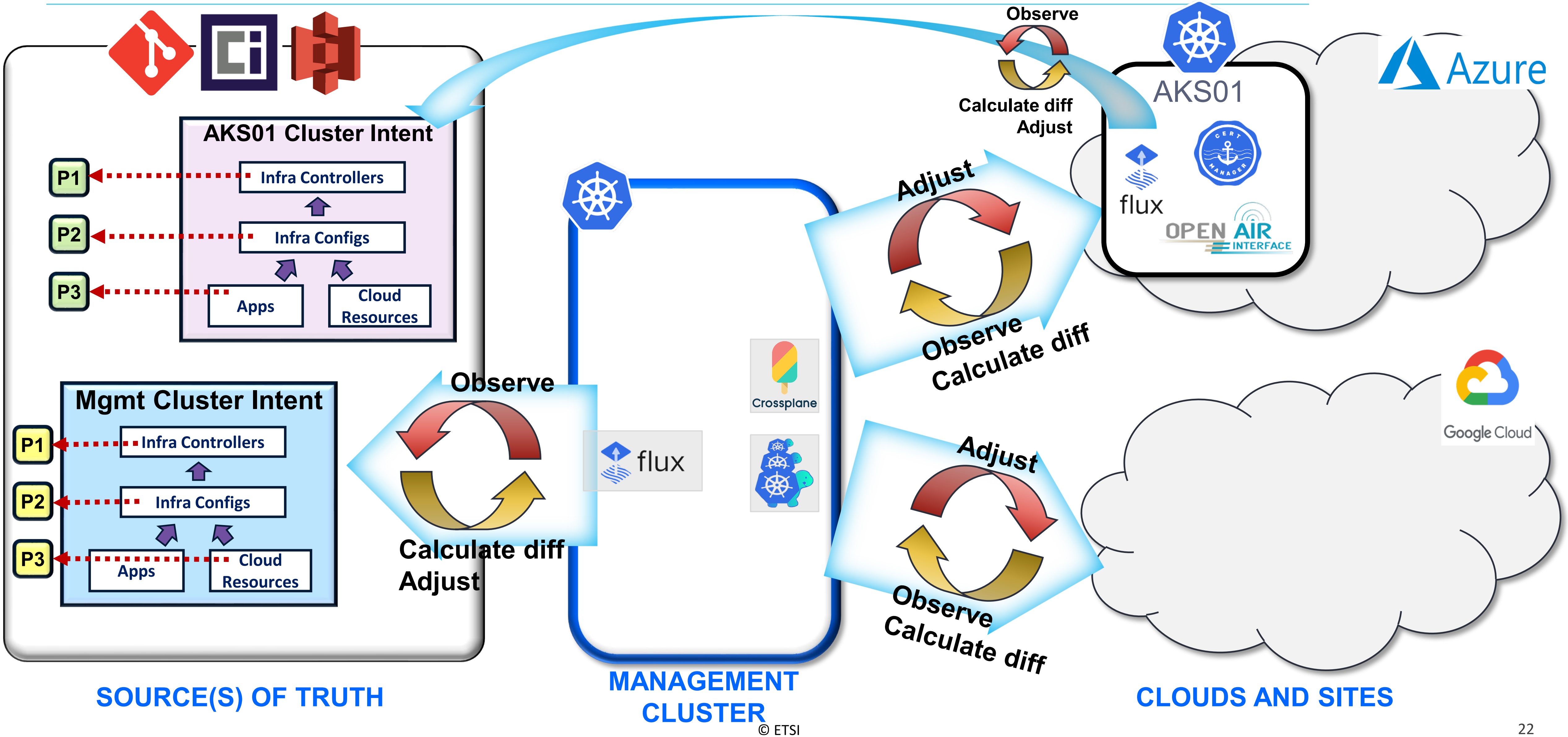


Management Cluster vs. Workload Clusters



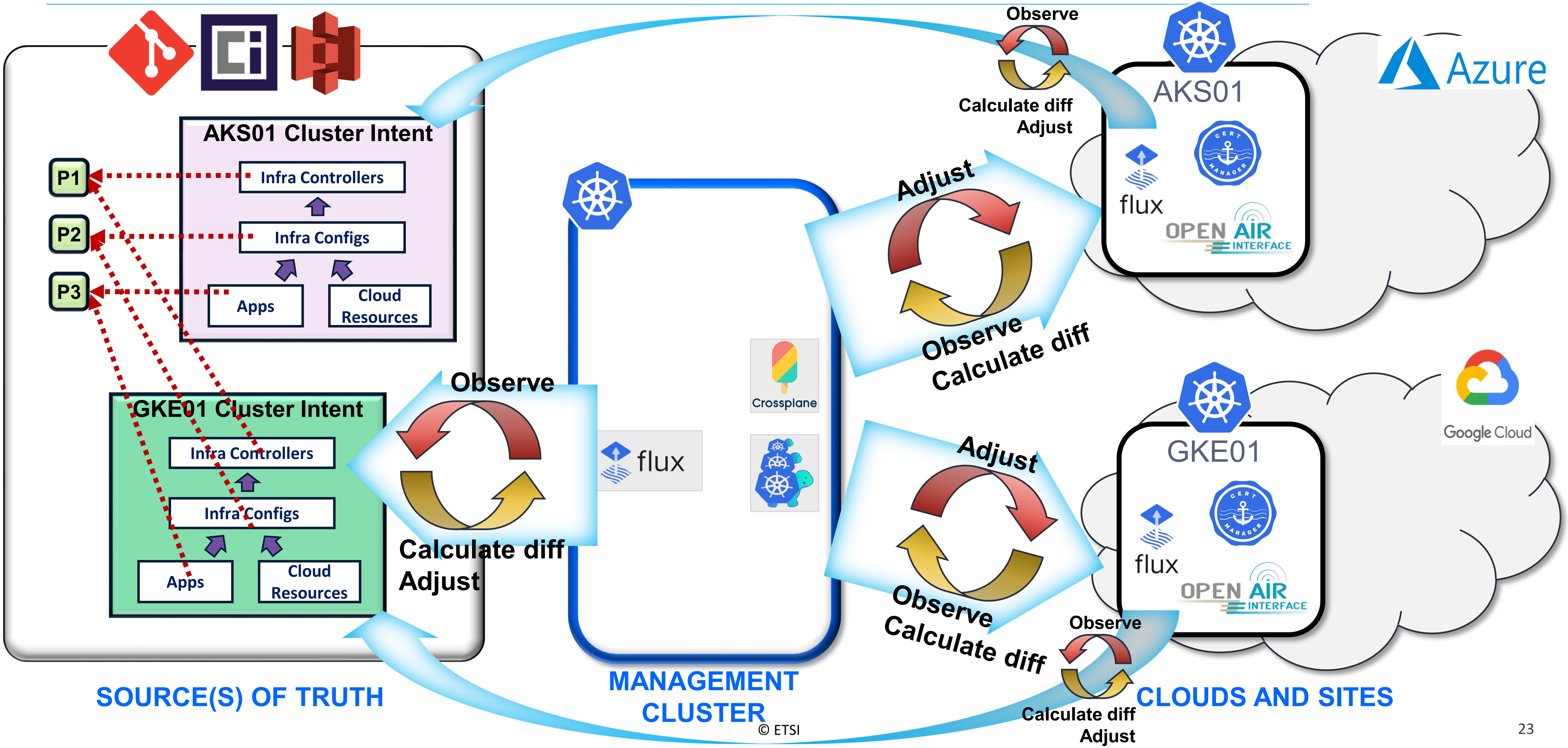
© ETSI

Managing a fleet of clusters as if they were one

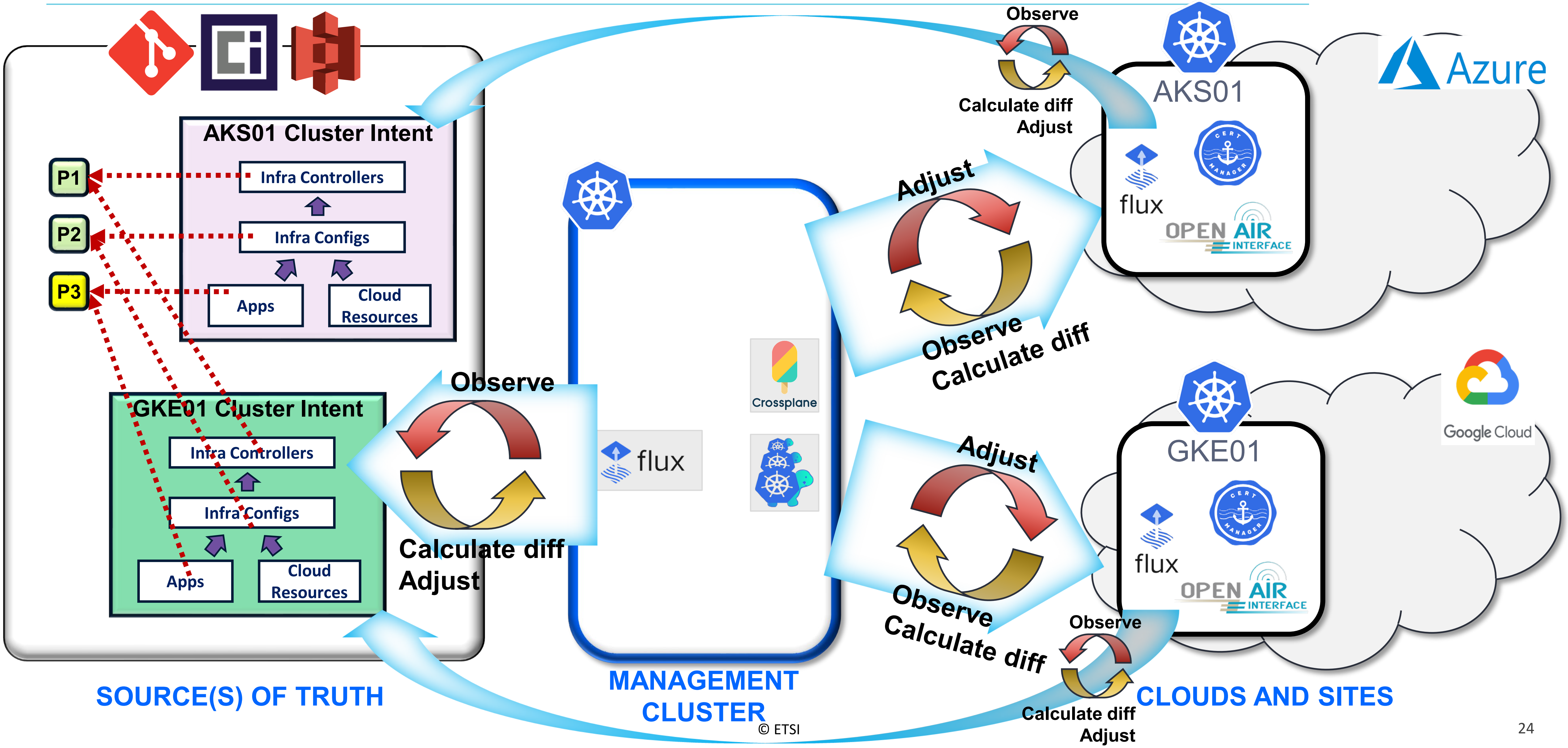


© ETSI

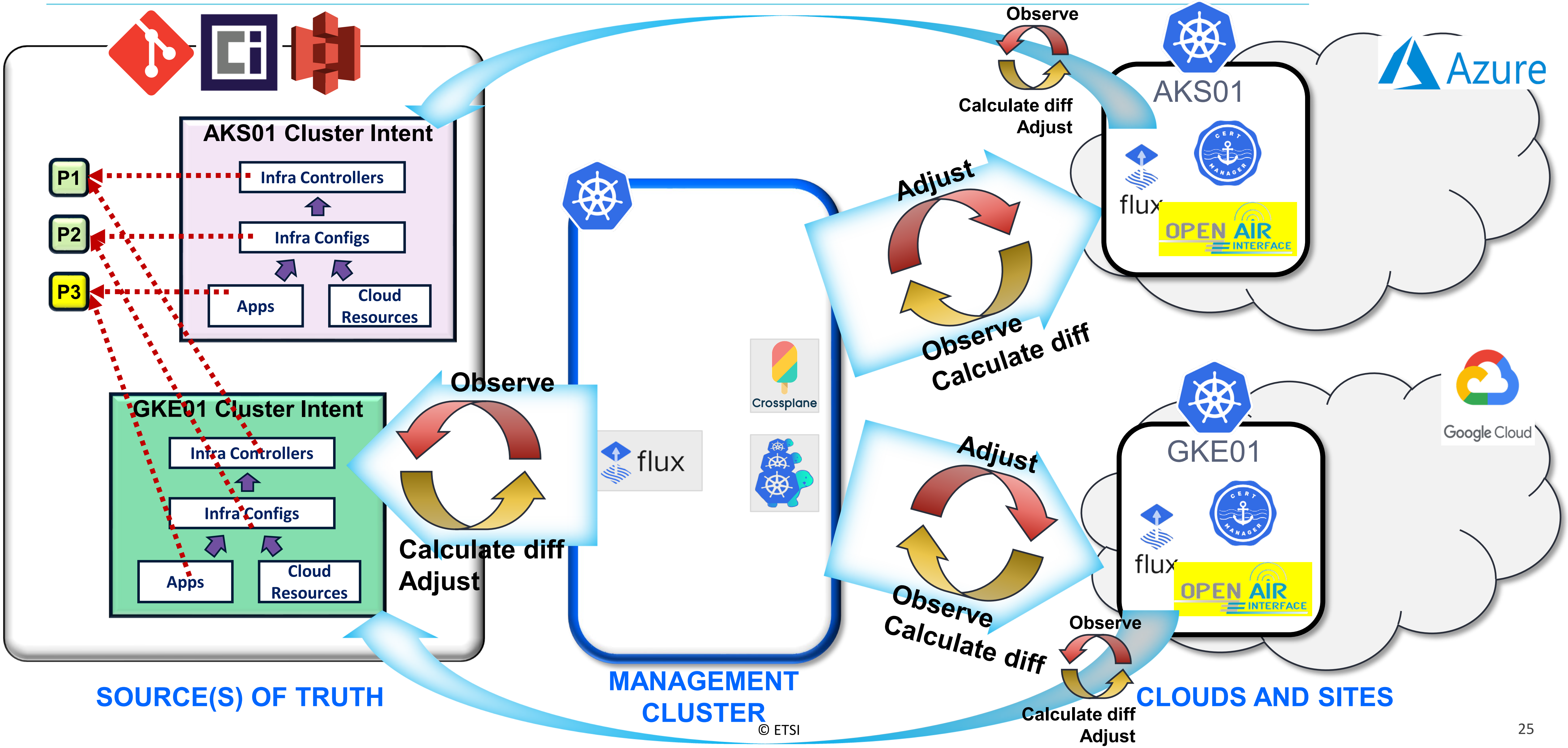
Managing a fleet of clusters as if they were one



Managing a fleet of clusters as if they were one



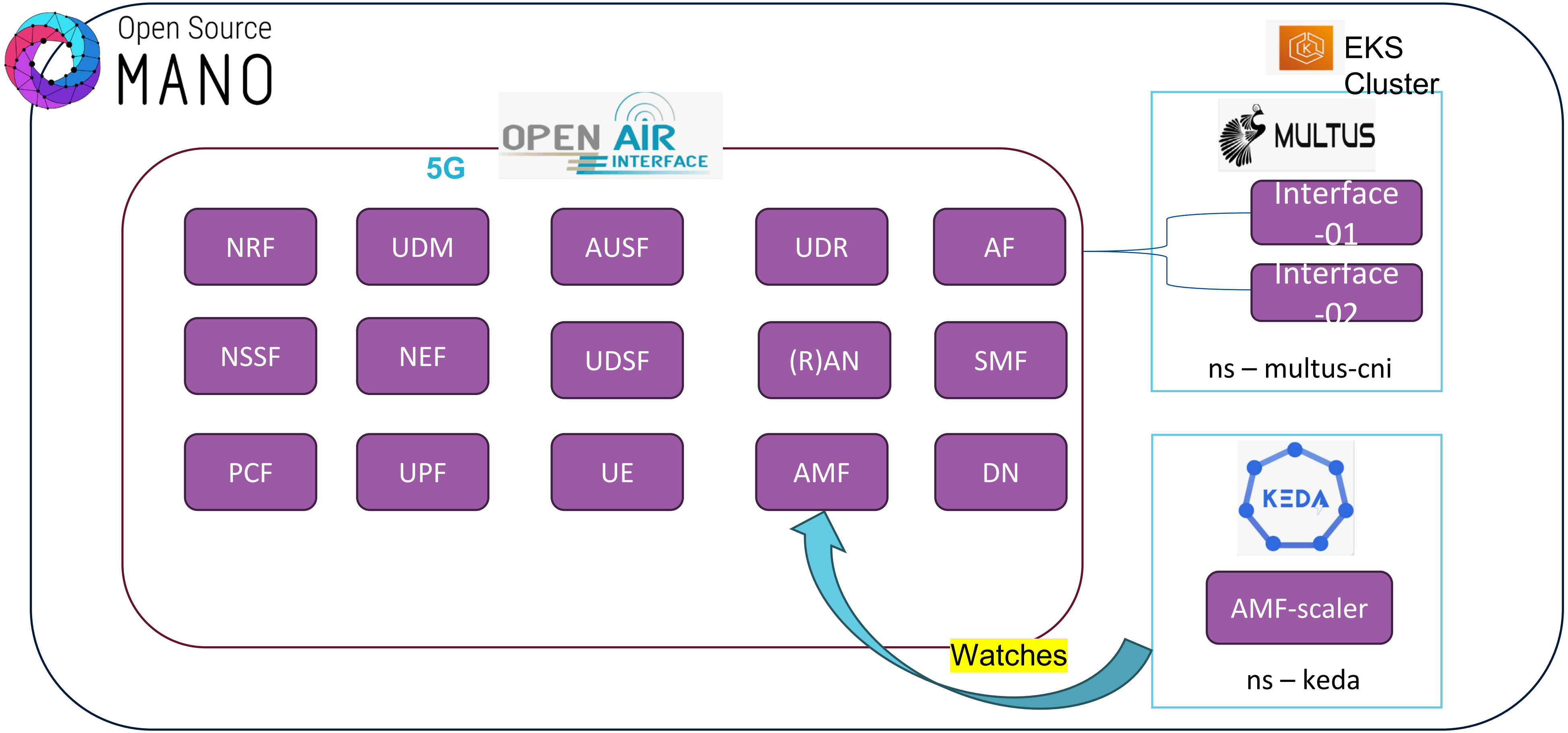
Managing a fleet of clusters as if they were one



Demo: Full 5G Use case

Play online demo:

https://www.etsi.org/events/webinars?commid=624338



Takeaways

OSM Rel SIXTEEN marks a pivotal milestone in OSM's journey, introducing significant changes in its scope, architecture, and functionality

- The new scope involves the management of **infrastructure**, **platforms** and **applications** across cloud platforms.

Naïve GitOps-like operations can become highly complex for large multi-cloud deployments

- Synchronization order, reuse of intents or protection of sensitive data (among others) are not trivial to operate at scale.
- Orchestration can become of great help here to prevent a exploding complexity.

The new OSM engine for **declarative operations** (which coexists with the **NFV-like** one), is key to manage large cloud-native infrastructures successfully

OSM defines new concepts, such as **KSUs**, **OKAs**, **Profiles** and **Cluster Intents** to orchestrate large declarative deployments in a more convenient fashion

- OSM leverages on well-stablished SW components whenever feasible to facilitate CNF/Apps onboarding and avoid reinventing the wheel, focusing on its core purpose.

Open Source MANO

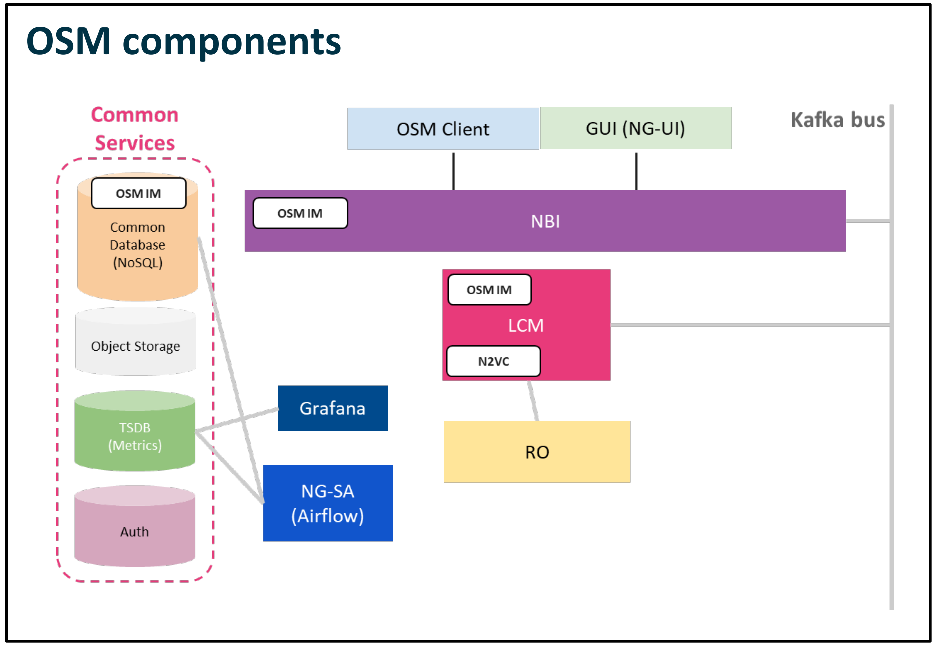
by ETSI

Thank You!

osm.etsi.org
osm.etsi.org/docs/user-guide
osm.etsi.org/wikipub

Components in an OSM installation in Release SIXTEEN

Open Source MANO



OSM components

Common Services

OSM IM

Common Database (NoSQL)

Object Storage

TSDB (Metrics)

Auth

OSM Client

GUI (NG-UI)

Kafka bus

OSM IM

NBI

OSM IM

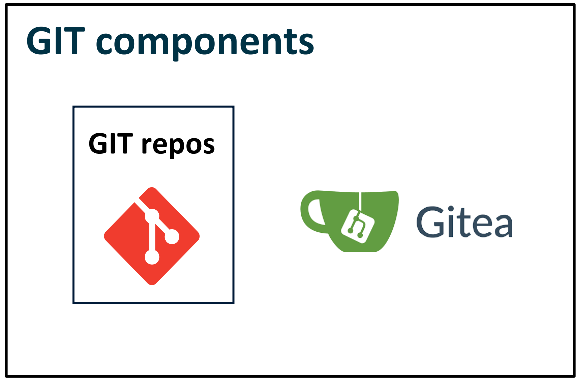
LCM

N2VC

RO

Grafana

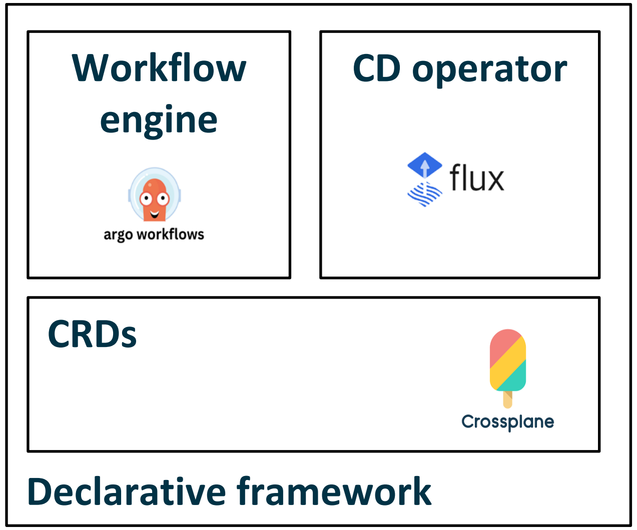
NG-SA (Airflow)



GIT components

GIT repos

Gitea



Workflow engine

argo workflows

CD operator

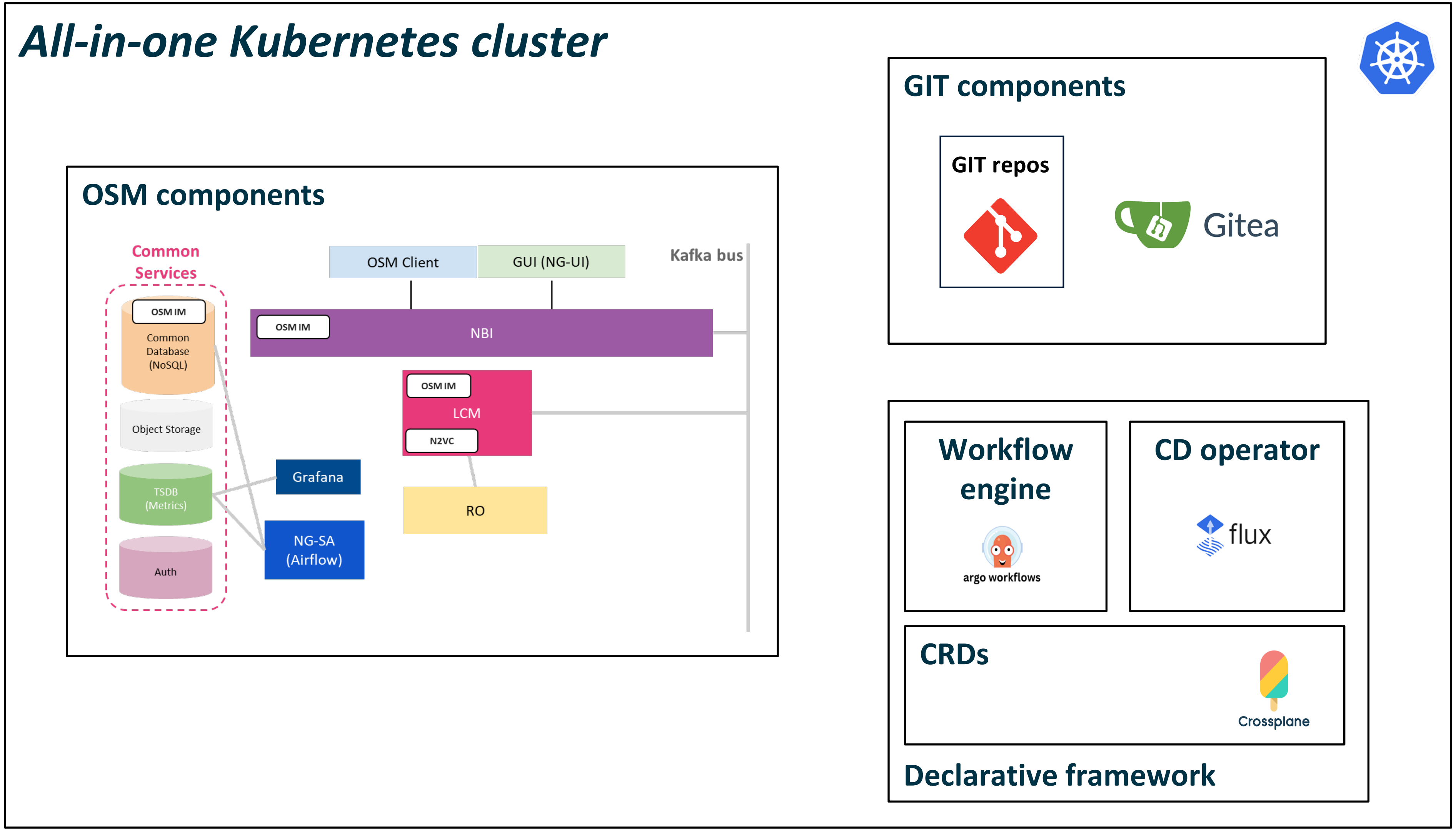
flux

CRDs

Crossplane

Declarative framework

Different deployment options



All-in-one Kubernetes cluster

OSM components

Common Services

OSM IM

Common Database (NoSQL)

Object Storage

TSDB (Metrics)

Auth

OSM Client

GUI (NG-UI)

Kafka bus

OSM IM

NBI

OSM IM

LCM

N2VC

Grafana

NG-SA (Airflow)

RO

GIT components

GIT repos

Gitea

Workflow engine

argo workflows

CD operator

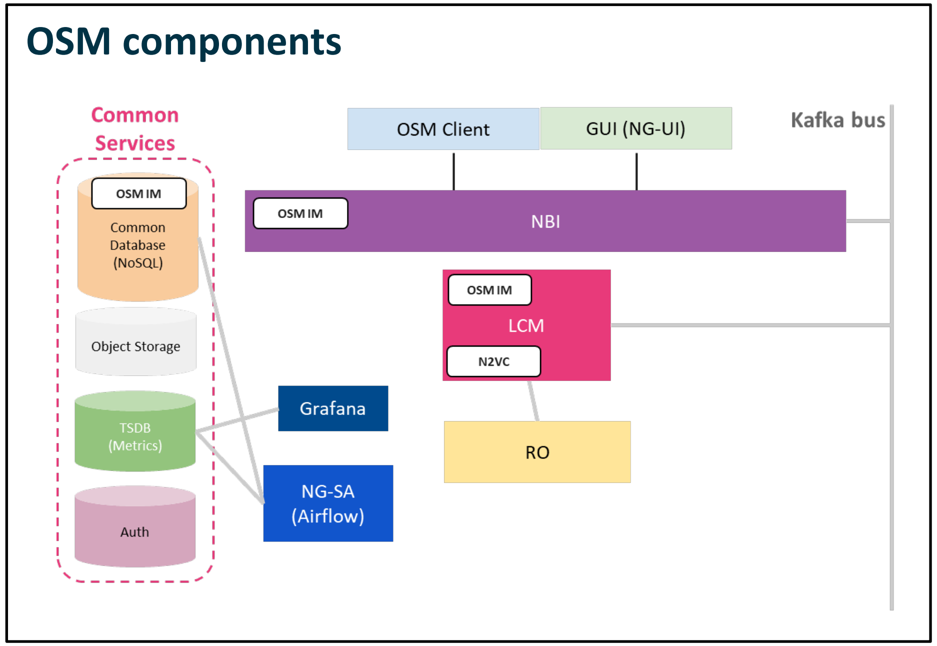
flux

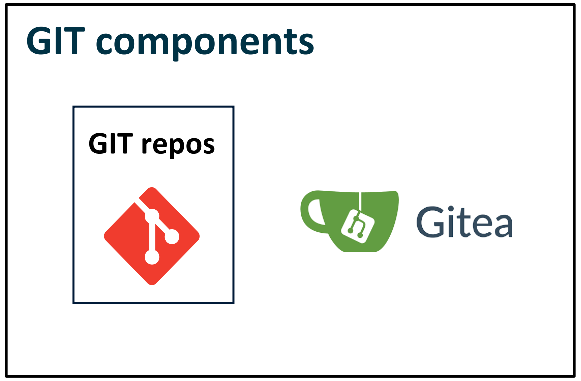
CRDs

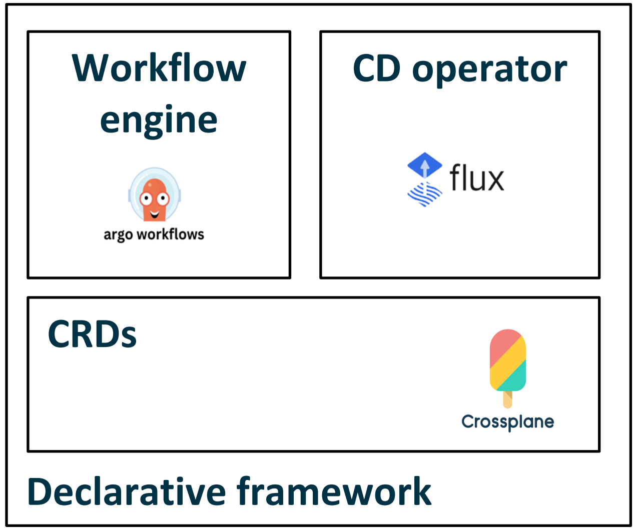
Crossplane

Declarative framework

Different deployment options

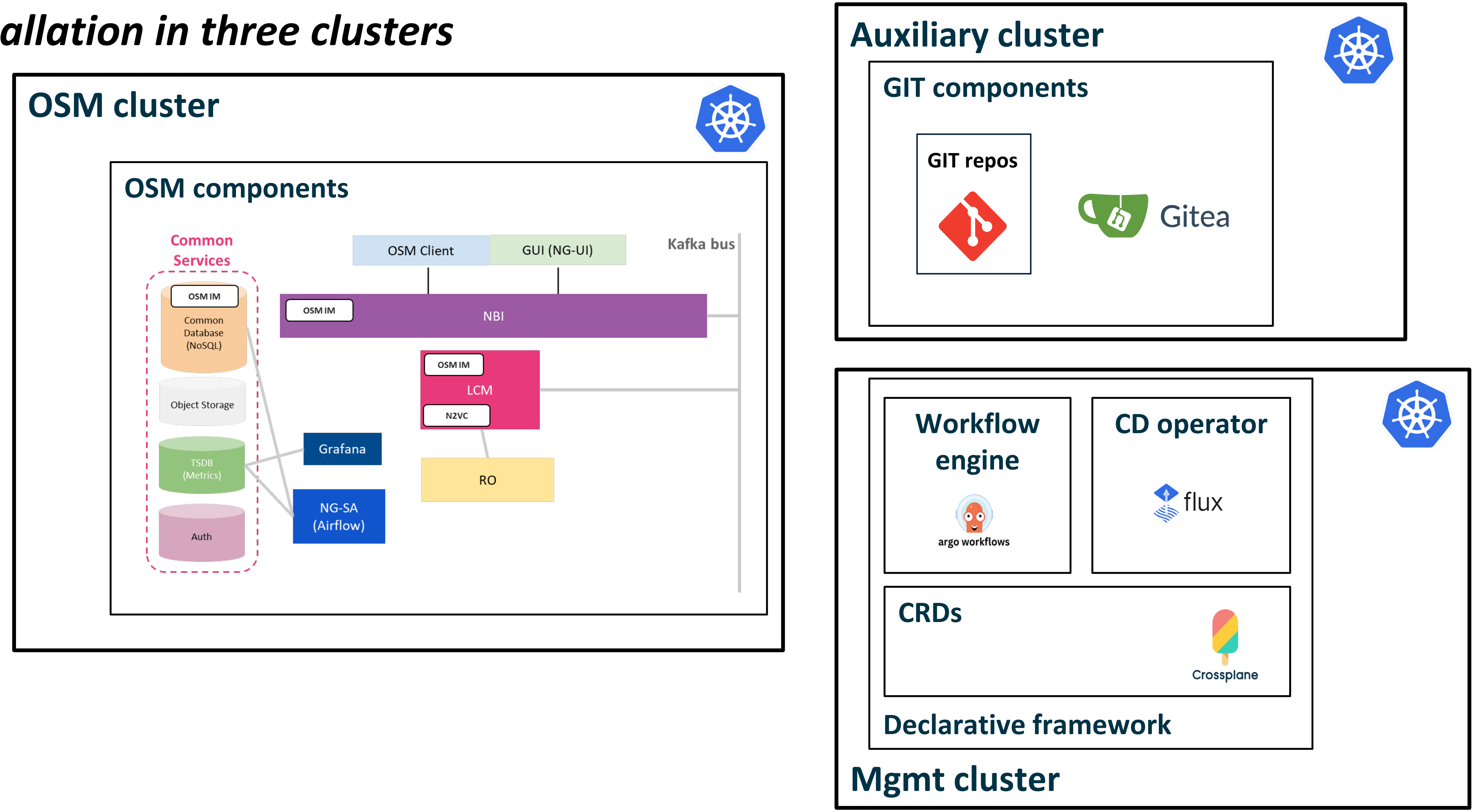






Different deployment options

Installation in three clusters



OSM cluster

OSM components

Common Services

OSM IM

Common Database (NoSQL)

Object Storage

TSDB (Metrics)

Auth

OSM Client

GUI (NG-UI)

Kafka bus

OSM IM

NBI

OSM IM

LCM

N2VC

Grafana

RO

NG-SA (Airflow)

Auxiliary cluster

GIT components

GIT repos

Gitea

Workflow engine

argo workflows

CD operator

flux

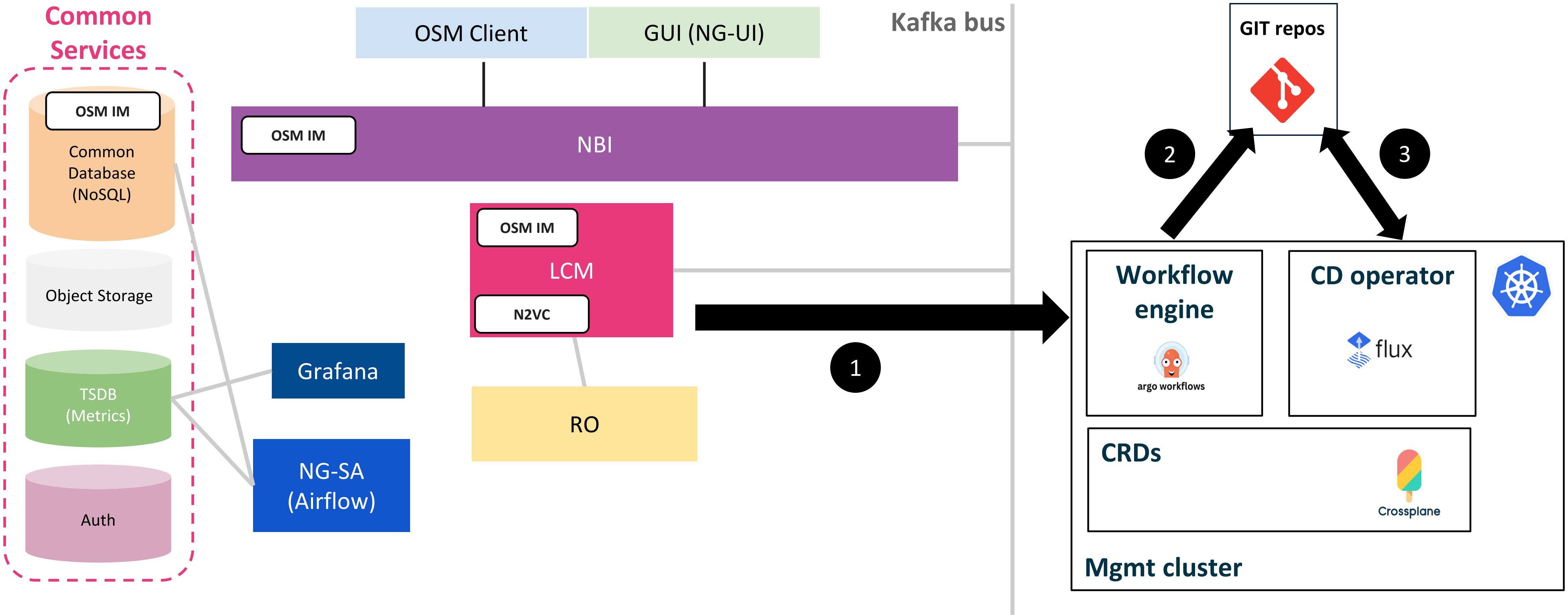
CRDs

Crossplane

Declarative framework

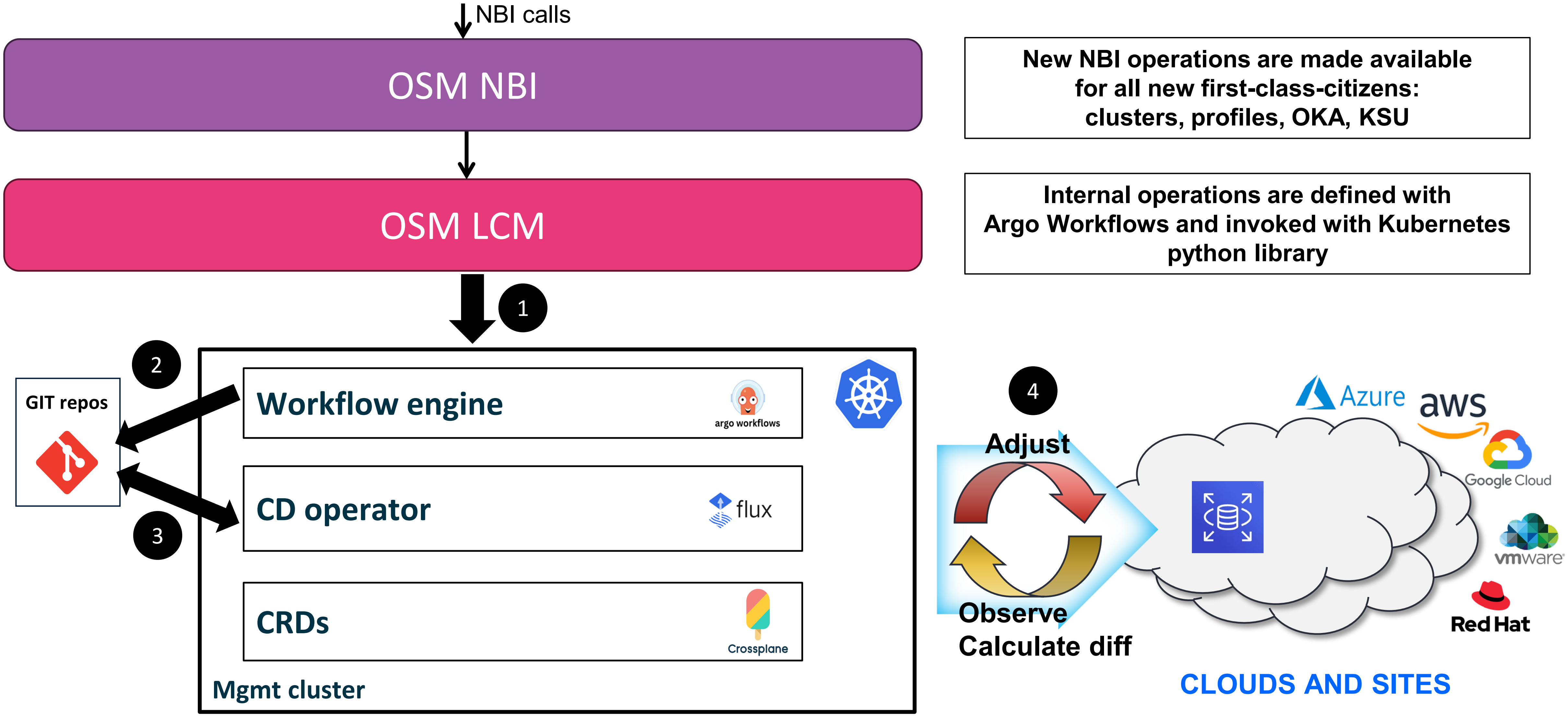
Mgmt cluster

Workflows for cloud-native operations in OSM follow GitOps model



Architecture in Release SIXTEEN

Workflow for cloud-native operations in OSM following GitOps model

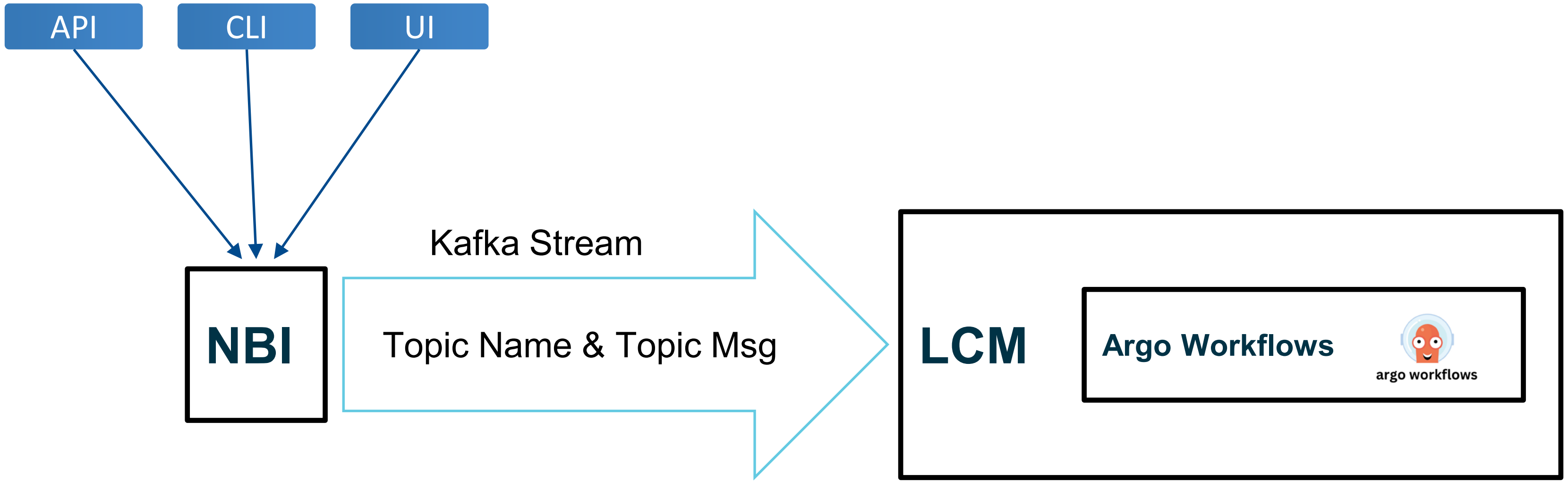


New NBI operations are made available for all new first-class-citizens: clusters, profiles, OKA, KSU

Internal operations are defined with Argo Workflows and invoked with Kubernetes python library

Architecture in Release SIXTEEN

Workflow for cloud-native operations in OSM following GitOps model



API

CLI

UI

NBI

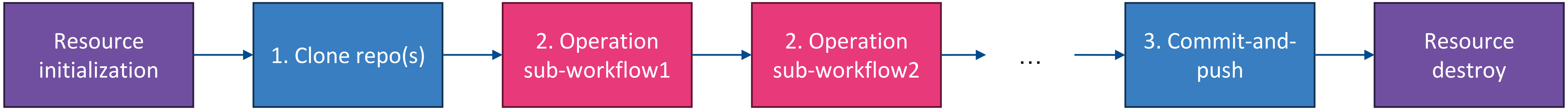
Kafka Stream

Topic Name & Topic Msg

LCM

Argo Workflows

Generic workflow for all operations



Resource initialization

1. Clone repo(s)

2. Operation sub-workflow1

2. Operation sub-workflow2

…

3. Commit-and-push

Resource destroy