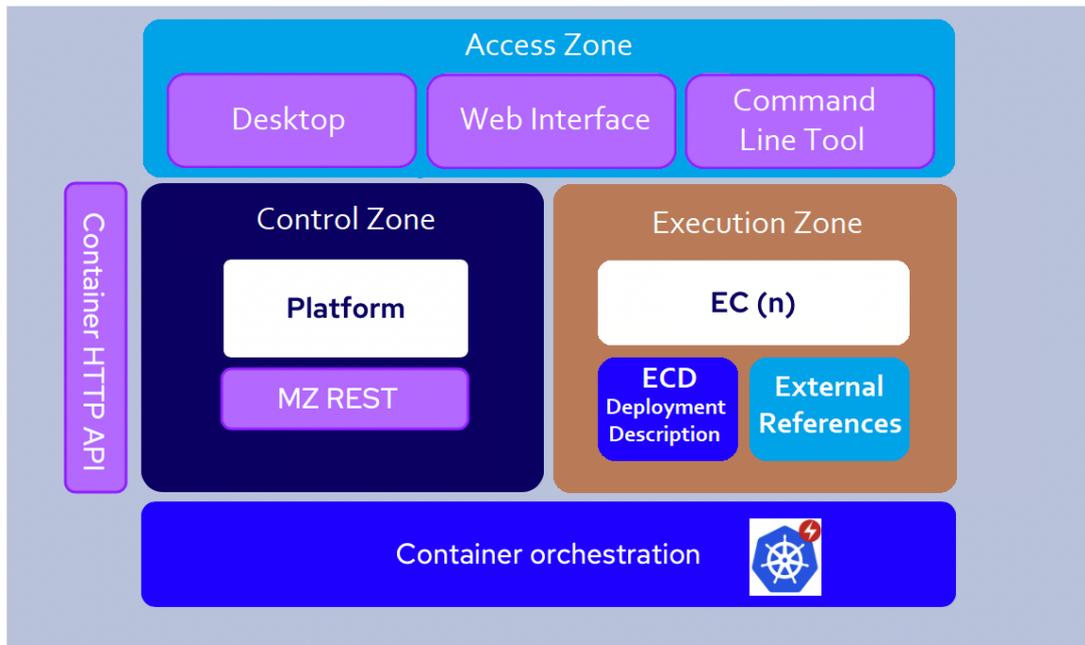


System Overview[hide]2.0[/hide]

Private Edition

The Usage Engine system architecture is designed to be completely distributed and scalable, capable of executing in both private and public clouds and using Kubernetes for clustering.

Development and Operation APIs



User Interfaces

There are three different user interfaces:

- **Desktop**, available both as a local client or in a browser, which is designed to be used by the solution designer; See [Desktop Overview](#)
- **Web Interface**, is designed to be used by for execution, scaling, monitoring,, logging, and troubleshooting, see [Web Interface](#)
- **Command Line Tool**, see [Command Line Interface](#), [mzcli](#) and [mzsh](#)

Zones

Architecturally, Usage Engine is comprised of three Zones.

- Control Zone, also called Platform
- Execution Zone
- Access Zone

The Zones are used for different purposes. The Control Zone is used to control Usage Engine and is intended for Administrators. The Execution Zone is where the various configured workflows are executed according to what has been configured by the end users (via the Access zone). The Control Zone contains the Platform and the Database.

Container Orchestration

The container orchestration is done through Kubernetes using the REST interface in Usage Engine.

The Kubernetes operator enables you to setup a dynamic cluster that can scale up or down automatically by using:

- Configurable monitoring and triggers
- Configurable conditions
- Configurable commands

In Usage Engine this is done using [Execution Context Deployments \(ECDs\)](#).

Container HTTP API

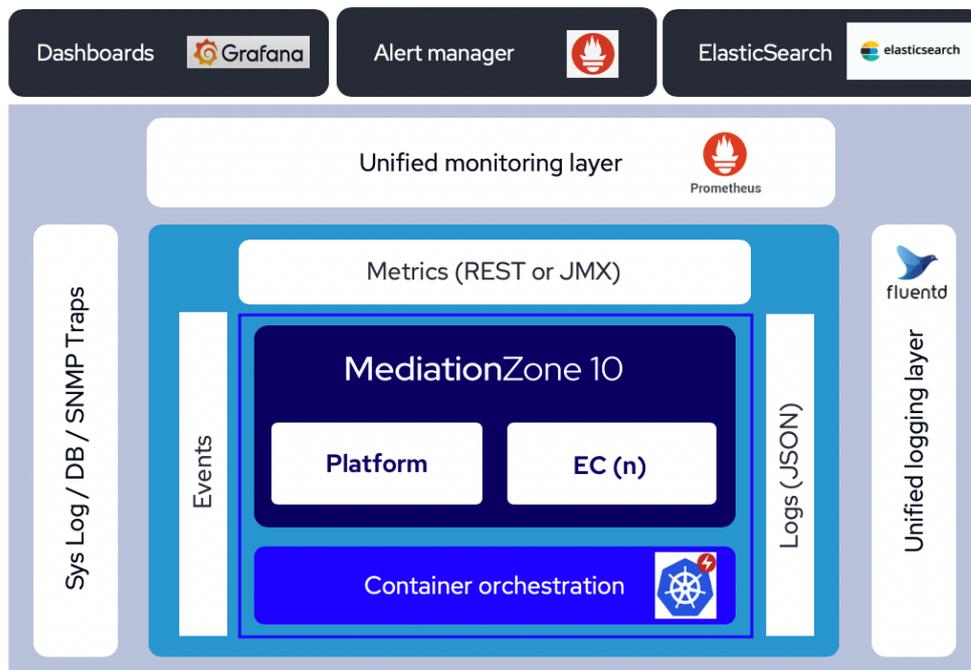
A HTTP API is provided for transferring files between containers.

External References

Profile that allows you to use configuration values that originate from property files or exported environment variables.

Monitoring APIs

The Usage Engine architecture uses best-of-breed industry standards for non-core functions like logging, monitoring, and alarming.



The APIs are standardized; JSON for logs and events, and REST or JMX for metrics.

The Events are filtered in the Event Manager, and apart from sending them to a database, or as a mail or SNMP Trap, you can also send them to System Log.

Lifecycle Management

In order to provide for Life Cycle Management that ensures repeatability and minimizes human errors and downtime for operational procedures like; deploying and updating use cases, scaling and upgrading, Usage Engine includes:

- Deployment using Helm and Docker repositories, see [Installation - Private Cloud](#) and [Installation - AWS](#)
- EC Deployment interface for use case deployment, see [Execution Context Deployments \(ECDs\)](#)
- Automatic scaling, see [Execution Context Deployments \(ECDs\)](#)
- Custom healing, see [Workflow Types](#)

This chapter includes the following sections: