

# ELIOT R5 - IoT Gateway Architecture Document

## Blueprint overview/Introduction

ELIOT is a project under Akraino approved blueprint family and part of Akraino Edge Stack, which intends to provide an platform for a fully integrated edge network infrastructure and running edge computing applications, on lightweight Edge Nodes. ELIOT targets on making the edge node a lightweight software stack which can be deployed on edge nodes which have limited hardware capacity by leveraging lightweight OS, a container running environment and container orchestration applications.

Below are the high-level features which this blueprint will go on to implement in phase-wise.

- Lightweight IOT-Gateway Solution
- Autonomous Edge Sites.
- Unified Portal for platform management and for App developers.
- Sandbox with SDKs and tools chains for IOT app developers
- Heterogeneous deployment on Multi-Arch.
- Software Rolling update to edge sites and IOT devices
- Interoperability between the devices and applications at the IOT Edge
- IOT protocol support like OPC-UA
- Industrial IOT enabling by TSN etc.
- Middle ware for IOT-Framework like EdgeX

## Use Case

Many diverse business applications require a converged IoT gateway. In this use case, the plan is to support the following scenarios:

- IOT sensor application
- OPC-UA over TSN application
- AI Camera Application

Industry Sector: Enterprise, diverse IoT applications in various vertical market segments, Carrier

## Overall Architecture

The ELIOT platform architecture consists of a One Click Deployment , ELIOT Master (Controller) Cluster (Cloud) and single to multiple ELIOT IOTGateway / Edge Nodes .

The ELIOT Master node can be a single node or a multiple node K8S Cluster.

Overall architecture diagram of the blueprint.

## ELIOT Arch

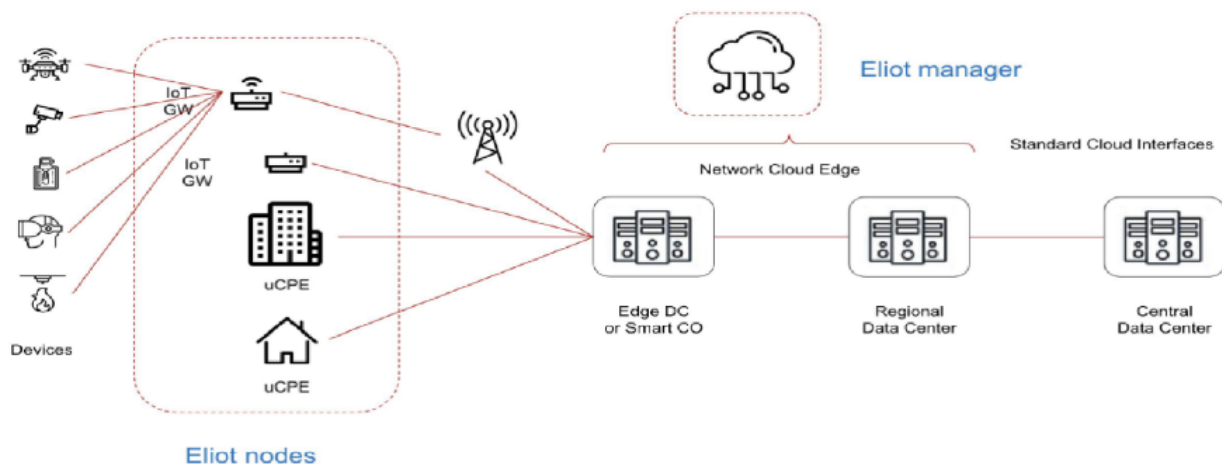


Figure 1 – ELIOT IOT-Gateway R5 architecture

## ELIOT Manager:

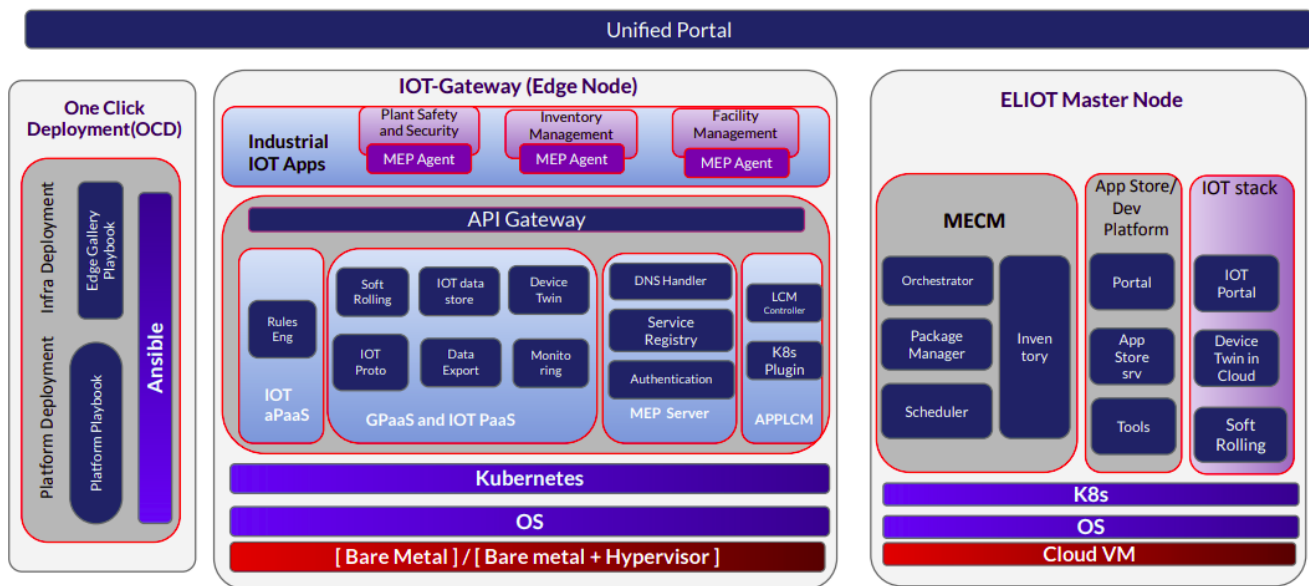
ELIOT Manager is a central node which provision and manage several IOT-Gateways deployed on Edges.

## IoT-Gateway Node:

IOT-Gateway is a lightweight platform which connect and manages various IOT devices (including, provision, discovery, device twin of devices), support industrial IOT protocols, TSN for factory automation use cases etc. user IOT applications leverage cloud native environment on resource constraint devices and manages from central Node(ELIOT Manager).

# Platform Architecture

The below diagram gives a overall architecture of the ELIOT IoT Gateway blueprint as given in Release 5.



**Note:** ELIOT IoTGateway Blueprint Deployment has been tested on Cloud VM and is not tested on Bare-Metal Environment. Though, theoretically deployment should work in bare metal, provided hardware and software prerequisites are met. Kindly refer [ELIOT R5 - IoT Gateway Test Document](#) to get details on the tested deployment.

## ELIOT Master Components

### IOT-Stack

- IOT - Portal - ELIOT Portal uses Angular 9.0. UI to provide Network management, monitoring, Application deployment, Dashboard. It internally can be able to view and modify the ELIOT cluster i.e ELIOT Master and IOTGateway nodes.
- Software rolling update: To provide software rolling updates to devices, in ELIOT Master we are integrating Eclipse Hawkbit platform. Eclipse [HawkBit](#) For rolling out software updates to constrained edge devices. We are using Hawkbit container environment with latest version.

### MECM:

- Portal: The dashboard provides user interface for the MECM services where edge nodes, APPLCM can be registered and apps can be distributed.
- Orchestrator: Provides application orchestration based on static/ dynamic host selection.
- Inventory: Maintains system level view of all Edges, including edge capabilities, deployed applications and their status etc.
- Scheduler: Responsible for selection of host for deployment.
- Package Manager: Maintains and manages application packages and is responsible of pushing images to edges.
- Common DB: Persistent Database.

### AppStore:

- Portal: Portal provides the user interface to publish and launch App applications. and has option for user to comment on the application in comment section and give ratings.
- Appstore Services (App Distribution, App Experience): App Store services where uploading an App package and verifies the upload. Also provides features like comment/like for user experience sharing.
- Repository : Open source PostgreSQL database is being used to store the information, Recommended version is 12.3.

### Developer Platform:

- Tools: Developer tools are used in Plugin Management, Plugin Life Cycle and grading systems making it more convenient for developers to develop applications and integrate them into the edgegallery platform.
- Workspace: Developers manage their own Application development process by creating projects or migrating projects. Developers can manage project, app package and test task management.

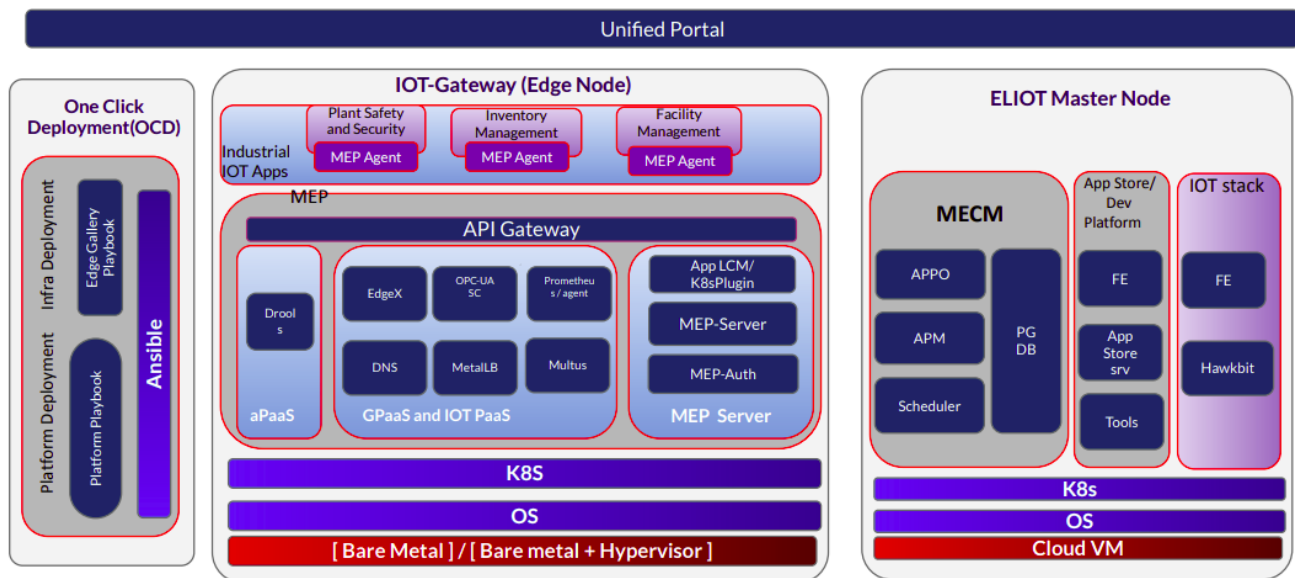
#### IoTGateway/ Edge Node Components:

MEP:

- aPaaS:
  - Rules Engine: Drools by EdgeX foundry
- GPaaS & IoT PaaS:
  - Software Rolling update: ELIOT includes Hawkbit from Eclipse community to provide software rolling update to Edge Sites / IoTGateway nodes
  - IOT Data store: storing data coming from IOT devices and which further IOT application can consume.
  - Basic Device Twin: device profile, configuration, metadata(cmds for devcies) etc.
  - IOT Protocol: OPC-UA industrial IOT protocol
  - Data Export: export data to Cloud applications and other on Prem applications
  - Monitoring: Graphana , prometheus agents to collect and display edge healths
  - CNI: k8s based networking for multi interface support in a POD for SFC scenarios, network policies etc.
- MEP Server:
  - DNS: DNS server and DNS record management service
  - Service Registry: Service registry and discovery, availability notifications
  - AppLCM: applications life cycle management
  - Network Isolation: Provides separation of control, management and data plan
- API Gateway: Provides secure End points for MEP services as well as producer Applications.

## Software Platform Architecture

The below image shows the software platform distribution and architecture for Release 5.0.



#### ELIOT Software Components

##### ELIOT Master -

Hawkbit:

- Eclipse [hawkBit](#) is an domain independent back end solution for rolling out software updates to constrained edge devices as well as more powerful controllers and gateways connected to IP based networking infrastructure.
- We are using Hawkbit container environment with latest version.

MECM Services:

- Front End: The MECM-FE management plane mainly provides the registration of edge nodes, APPLCM, and application distribution and deployment functions from the user interface perspective, which is divided into three functional areas. Map area provides switch between nodes in different regions. The KPI, MEP capabilities and application capabilities are displayed in detail according to the currently selected node.

- AppO: Application orchestra tor is the core module responsible for orchestrating life cycle maintenance operation of application. AppO orchestrates application deployment by executing specified process and automated sequence of tasks, rules and policies. AppO is responsible for retrieving the application instance information, edge host performance statistics and edge host platform capabilities.
- Inventory : Responsible for registrations like MEC Host, AppLCM, AppStore, application. Retrieves all the APPLCM records, MEC host records and applications.
- APM: Application package manager responsible for application package management including On boarding, distributing package to edge , retrieves the application package information and download the application package CSAR.
- Postgres DB: Open source PostgreSQL database is being used to store modules specific information.

#### AppStore:

- Front End: AppStore is a market for developers to publish and launch App applications. Appstore FE provides the registration interface to the new user and once login the store home is visible, where user can upload the application package across various industry domains and displays the list of application along with the details. User has option to comment on the application in comment section and give ratings. Deleting option is also available in "My App/My APP" section.
- Appstore Backend: AppStore is a market for developers to publish and launch App applications. After uploading an App package, it must pass the test first. Only the apps that pass the test can be officially launched.
- Postgres DB: The AppStore database uses an open source PostgreSQL database, and the recommended version is 12.2. Which Stores app information, application package information and review details of the application.

#### Developer Platform:

- Tools: Developer tools are used in Plugin Management, Plugin Life Cycle and grading systems making it more convenient for developers to develop applications and integrate them into the EdgeGallery platform.
- Workspace: Developers manage their own Application development process by creating projects or migrating projects. Developers can manage project, app package and test task management.

#### IOT-Gateway Node:

- aPaaS:
  - **Drools** - Rules engine. It is a part of EdgeX foundry services. In time sensitive use cases or environment where a lot of data is being generated by the connected things, rules engine will be connected to the data coming directly out of core data which will be generated by the gateway node.
- GPaaS & IOTPaaS:
  - **Hawkbitt**: Eclipse [hawkBit](#) is an domain independent back end solution for rolling out software updates to constrained edge devices as well as more powerful controllers and gateways connected to IP based networking infrastructure.
  - **EdgeX**: EdgeX is opensource , vendor neutral Edge IOT Middleware platform under the LF Edge umbrella. It collects data from sensors at the edge and acts as a dual transormation engine sending and receiving data to and from enterprise, cloud and on-premise applications. EdgeX enables autonomous operations and intelligence(AI) at the edge. EdgeX translates and transforms the infromation coming from ELIOT edge sites i.e from sensors and devices and delivers it to applications over network based protocols in formats and structures that meet the needs of customers. It also takes data from applications and delivers it to the iotgateway nodes to control and actuation.
  - **OPC-UA**: OPC Unified Architecture is a machine to machine communication protocol for industrial automation developed by OPC Foundation which primarily focus on communicating with industrial equipment and systems for data collection and control. OPC-UA installed on IOTGateway node.
  - **Prometheus Agent**: is an open-source systems monitoring and alerting toolkit. Main features include building a multidimensional data model with time series data identified by metric name and key/value pairs, PromQL to leverage monitoring, time series data collection , nodes are discovered via service discovery or static configuration. Prometheus is installed on IOT-Gateway node to collect primarily node metrics and to monitor the cluster.
- MEP Services:
  - **DNS**: The MEP platform provides the domain name resolution services to the applications deployed in the EdgeGallery MEC, which can be utilized by the device applications in UE. MEC platform receives the DNS configurations from the MEC management, which includes the FQDN (Fully Qualified Domain Name), IP address and related entries. MEC applications can later query or modify the state of these configurations.
  - **MEP server**: MEP platform services like service registry, discovery, DNS, traffic rules etc. MEP provides service registration, update, delete, and query related API interfaces. Applications can register, update, delete, and query services through MEP. The service registry provides visibility of the services available on the MEC server. It uses the concept of loose coupling of services, providing flexibility in application deployment. In addition, the service registry presents service availability (status of the service) together with the related interfaces and versions. It is used by applications to discover and locate the end-points for the services they require, and to publish their own service end-point for other applications to use. The access to the service registry is controlled (authenticated and authorised).
- AppLCM: LCM controller and K8s plugin. Handles the application life cylce, where lcmController is mainly communicates with K8s plugin to upload config, remove config, instantiate, terminate,and query an application. It also provides APIs to query mep capabilities and kpi information from prometheus.

Below are details of Upstream and Opensource software's used in various nodes of ELIOT Environment.

#### OCD Node

S. No.	Software Name	Category	Version Number	Remarks
1.	Ubuntu	OS	18.04	
2.	GIT	Version Control	2.17.1	
3.	Ansible	Scripting Tool	2.10.7	

### ELIOT Master Node

S. No.	Software Name	Category	Version Number	Remarks
1.	Ubuntu	OS	18.04	
2.	Kubernetes	Orchestration	1.18.7	
3.	Docker	CRI	18.09	
4.	Grafana	Analytics Dashboard	8.7	
5.	Edge Gallery	MultiAccess Edge Computing Platform	1.1.1	Opensource MEC Platform

### IoTGateway Node

S. No.	Software Name	Category	Version Number	Remarks
1.	Ubuntu	OS	18.04	Supports both 16.04 and 18.04
2.	Docker	CRI	18.09	
3.	K8s	Orchestration	1.18.7	
4.	Edge Gallery	MultiAccess Edge Computing Platform	1.1.1	Opensource MEC Platform
5.	CAdvisor	Container Metrics	0.36	
6.	RabbitMQ	Message Queue	3.7	
7.	EdgeX	Edge IoT middleware platform	Edinburgh	This old version is used due to K8s deployment
8.	OPC-UA	IoT protocol framework	Geneva	

## APIs

Please refer the API Documentation link : [ELIOT R5 IOTGateway API documentation](#)

## Hardware and Software Management

### Hardware Management

Currently for this blueprint Huawei Cloud Virtual Machines are being used for development , testing and CD hence there is no specific hardware management to be done.

### Software Management

Gerrit Repo : [ELIOT IoTGateway Gerrit Repo](#)

## Alignment with EdgeX community

Below table is to capture information and alignment with EdgeX community for better collaboration.

Due to K8s deployment, currently using older version.

This table can be used for track alignment with EdgeX

Sl. No.	Attributes	ELIOT	EdegX	Remarks
Sl. No.	Attributes	ELIOT	EdegX	Remarks
	EdgeX version	Edinburgh	Hanoi	Till ELIOT Rel5 plan, EdgeX was not providing K8s based deployment so used older version.  In next ELIOT release analyse and update latest EdgeX version if it support K8s based deployment

2.	API version	V1	V2	Once EdgeX version updated, API page will be updated for latest V2 APIs
3.	New components and services	supported OPC-UA		Analyse EdgeX and support new features in next release
4.	gaps and requirement	ELIOT use K8s based orchestration, but earlier EdgeX was providing Docker compose based services.		Analyse latest EdgeX release and see whether supported K8s based deployment.
5.	New features			Keep track with EdgeX community

## Licensing

- GNU/common license