GE VERNOVA

EDGE SOFTWARE & SERVICES

EDGE OVERVIEW

Documentation

GE VERNOVA

Proprietary Notice
The information contained in this publication is believed to be accurate and reliable. However, GE Vernova assumes no responsibilities for any errors, omissions or inaccuracies. Information contained in the publication is subject to change without notice.

Trademark Notices

We want to hear from you. If you have any comments, questions, or suggestions about our documentation, send them to the following email address:
doc@ge.com

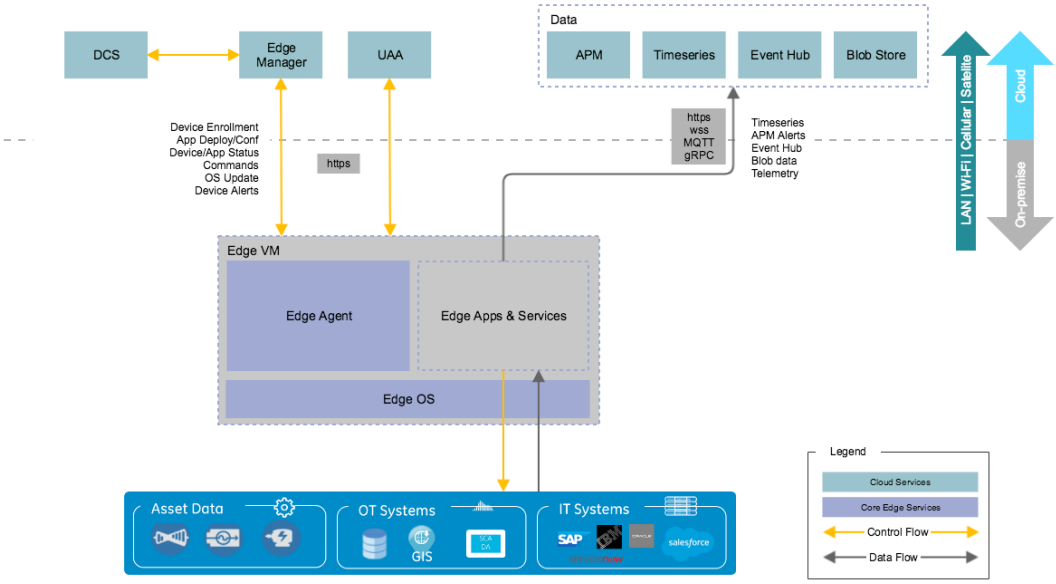
Contents

Predix Edge Architecture Overview

Edge Architecture Overview

Edge technology includes Edge OS - a secured Operating System; Edge Manager - a cloud application to remotely manage fleets of edge devices; and Cloud Connectivity Services - a set of optional services to securely connect Edge devices to Digital-hosted Cloud Applications and Services.

Figure 1. Edge Architecture Overview



Together with edge devices, edge applications, services and connectivity, they enable compute capabilities closer to machines, particularly to add operational value by executing workloads near the machines that generate the data.

By enabling data processing, storage, and analytics closer to machines, Edge technology creates a tight union between machines, control systems, and modern applications while easily and securely connecting edge devices to Edge services.

Edge Technology Benefits for Industrial Operations

Edge technology offers the following benefits:

- Rapid deployment: Turn industrial machines into smart, connected machines.
- Better, faster asset insights: Ability to use machine data faster and more efficiently, enabling quick decisions about performance, availability, and production optimization.
- Reliability and manageability: Enables IT teams to track, manage, and communicate with all edge devices and connected assets anytime, anywhere.

Edge 2.0 Key Capabilities

The 2.0 version of Edge has these capabilities:

- Connect:
  - Bi-directionally connect industrial assets over OPC-UA, Modbus, or MQTT to read and write data control tags.
- Process:
  - Run C, C++, Python, Node.js, or Java applications and analytics on the edge device in a docker-based edge execution environment.
- Transmit:
  - Securely transfer data to Digital-hosted Cloud Applications and Services.
  - Store and forward data during intermittent cloud connectivity.
- Manage:
  - Authenticate edge devices using Predix cloud identity and certificate-based device enrollment.
  - Enable centralized device and application lifecycle management of edge device fleet using Edge Manager.
- Extensibility:
  - Extend Edge 2.0 functionality through custom commands, custom software packages, and additional protocol adapters as multi-container Docker applications.

Edge 2.0 simplifies and accelerates the development and deployment of edge applications by providing a hardened, stable, and extensible platform.

The major components of the Edge 2.0 ecosystem are:

- **DCS:** Device Certificate Service.
- **Edge Manager:** Predix fleet management.
- **UAA:** User Account and Authorization is a multi-tenant identity management server. Its primary role is to issue tokens for client applications when they interact with the cloud services.
- **Edge VM:** In the GA release, the Edge OS is deployed as a Virtual Machine.

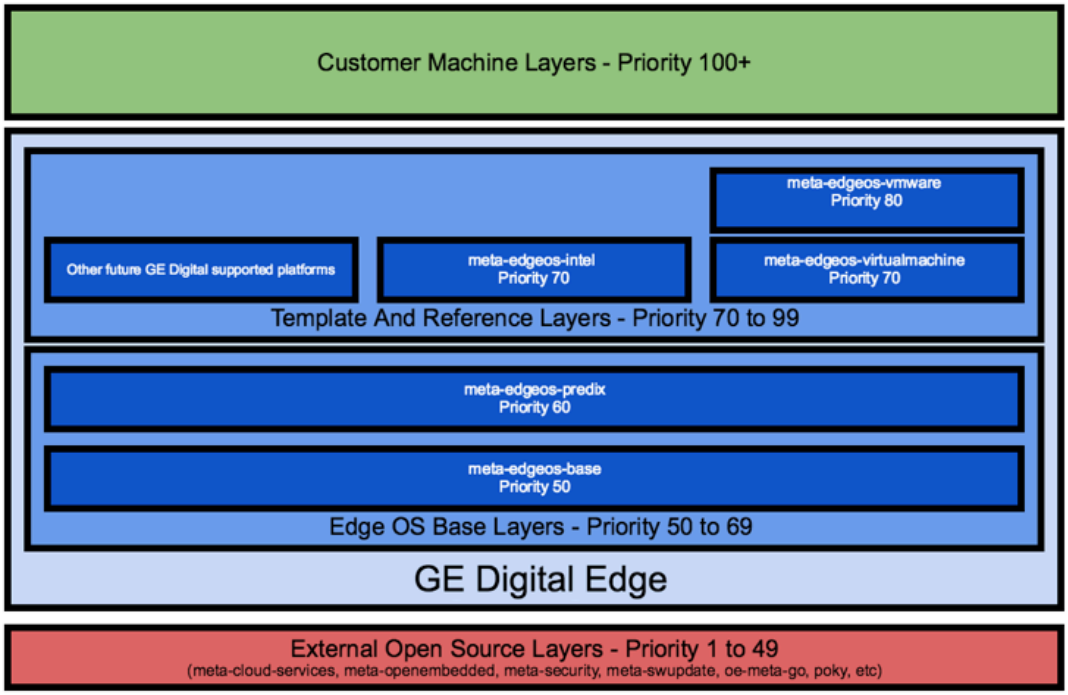
- **Edge OS:** Provides a secure base to run applications and common services to manage containers, applications, and the operating system.
- **Edge Agent:** Manages connectivity to Edge Manager.
- **Container Support:** Container support to allow independent applications to co-exist. (May not be available in lower capability devices.)
- **Cloud Data Services:** Cloud processing of the telemetry data sent from the edge devices. Examples of cloud data services are APM, Time Series, and Event Hub.

Predix Edge OS Architecture

Edge OS Architecture

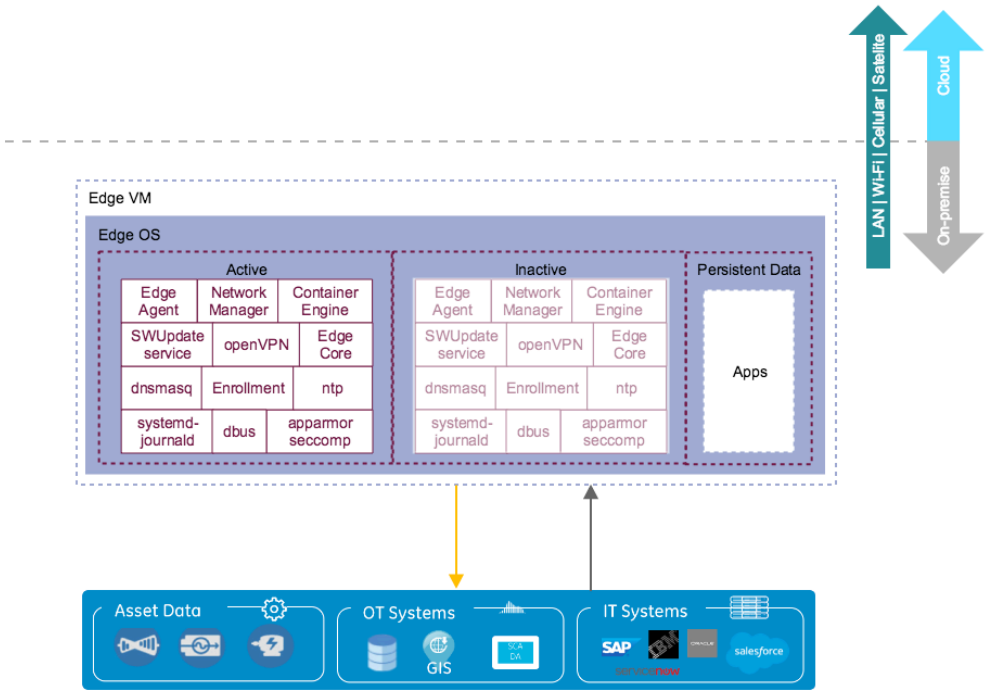
The Edge Operating System is a baseline Yocto Linux distribution that can be extended and ported to many bare metal platforms or virtual environments. It is architected around the design goals of small footprint, security, multi-container-based application hosting, and edge platform management.

Figure 2. Edge OS Overview



Partitions

Figure 3. Edge OS Architecture Overview



As seen in the above diagram, the Edge OS image is divided into three partitions (the scale of the partitions in the diagram do not reflect actual partition sizes):

1. **Active Partition:** This partition is the currently running version of the software. (Read Only)
2. **Inactive Partition:** This partition is used as part of the update process.
3. **Persistent Data:** This partition is used to store all mutable data and must endure device restarts and OS upgrades.

When the operating system is being updated, the following processes will occur:

- The active partition will be “snapshotted” to the inactive partition.
- Updates will be applied to the inactive partition.
- The inactive partition will become active.
- The system will attempt to reboot.

Predix Edge Agent Architecture

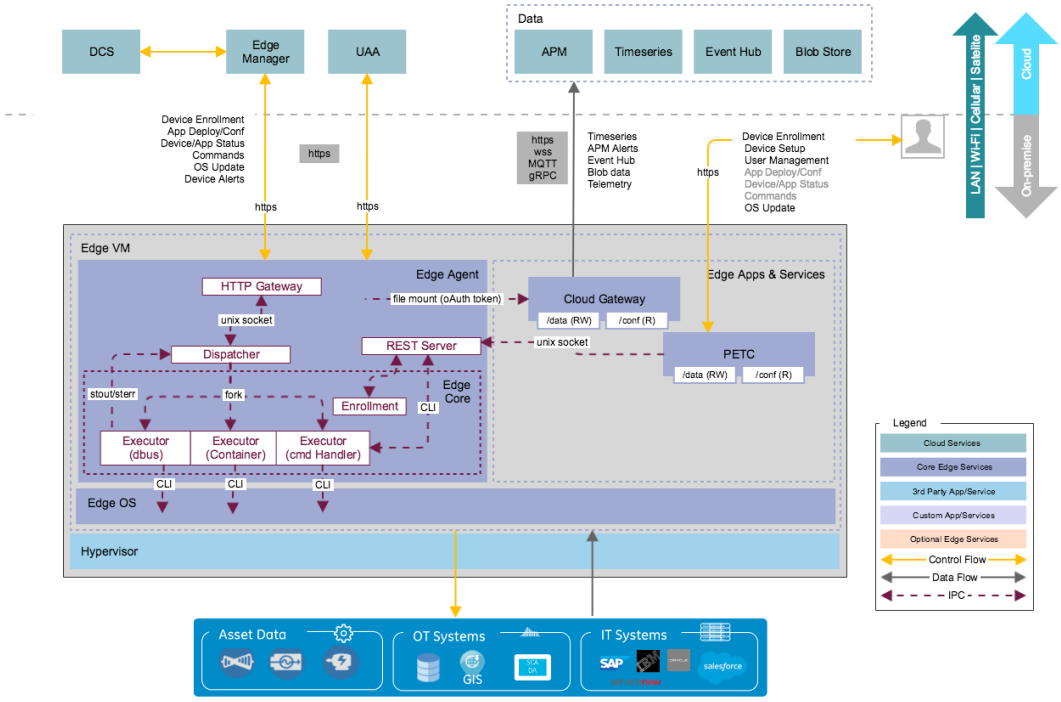
Edge Agent Architecture Overview

The Edge Agent is responsible for managing the state of the device and providing a common interface to update the state.

The Edge Agent is responsible for managing the state of the device and providing a common interface to update the state. Edge Agent capabilities include:

- Enrolling device to Edge Manager.
- Synchronizing device statuses to Edge Manager.
- Installing and updating applications and application configurations to devices.
- Maintaining security standards on devices.
- Ensuring Edge applications are always running.

Figure 4. Edge Agent Architecture



Edge Agent Gateway and Dispatcher

The HTTP Gateway provides connectivity and authentication management between the edge device and Edge Manager in the cloud. The HTTP Gateway performs the following tasks:

- Initiates conversation with Edge Manager on a scheduled basis.
- Receives information about the current state of the device as part of the initial connection.
- Ensures the connection is authorized and the “authentication token” is valid prior to any communication.
- Dispatches commands to appropriate command handlers.
- Responds with the status of the command execution.

Edge Core

The Edge Core is a common command execution framework that allows the abstraction of the command execution from Predix Edge Manager and the Edge Core API to execute the same commands.

The Edge Core provides a REST API that the Predix Edge Technician Console can use to dispatch commands to be issued locally. Initially, using Edge Manager on the cloud and using the technician console are mutually exclusive. A future enhancement will resolve any discrepancies between the edge device and Edge Manager.

Command Executors

Command Executors are functional handlers that can manage the state of the edge device, including Applications and Edge OS.

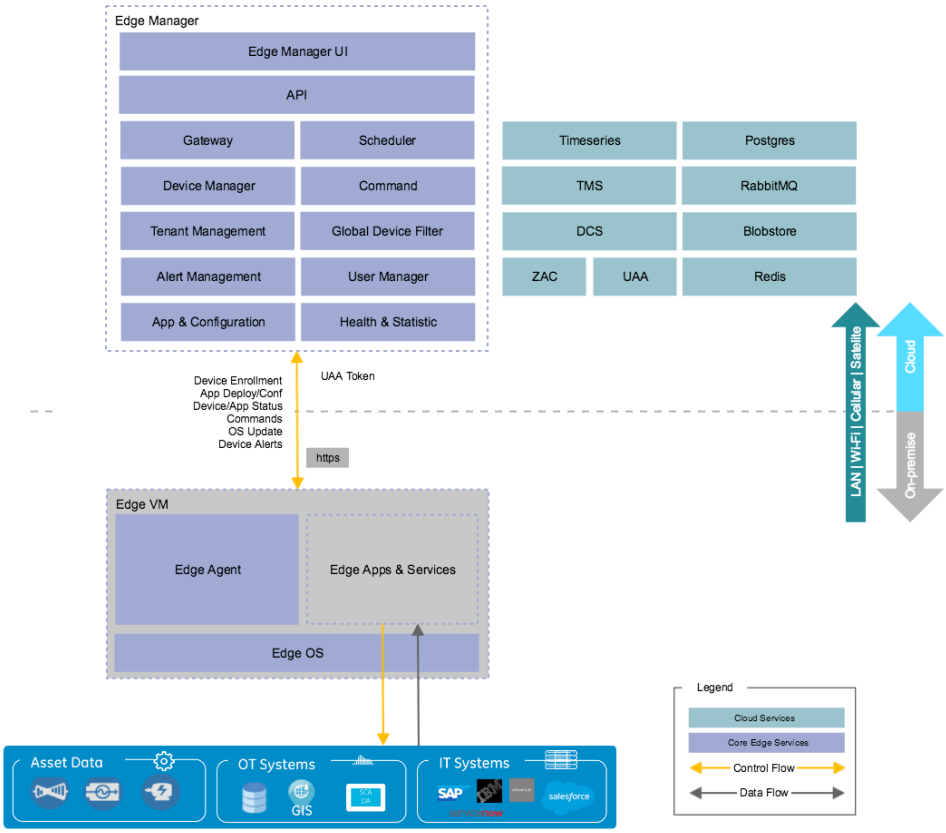
x

Predix Edge Manager Architecture

Edge Manager Architecture

Edge Manager is a cloud application consisting of a set of microservices to provide fleet management of edge devices. Its modular design allows for extensibility.

Figure 5. Edge Manager Architecture



Edge Manager Microservices

Edge Manager includes the following microservices:

- **Gateway** – enables the communication between cloud and edge devices
- **Tenant Management** – provisions tenants in Edge Manager and relevant Cloud services
- **Global Device Filter** – supports workflow to manage large, ad-hoc groups of devices
- **Edge Manager UI** – enables users to manage fleets of devices including enrollment, device management, and application lifecycle management

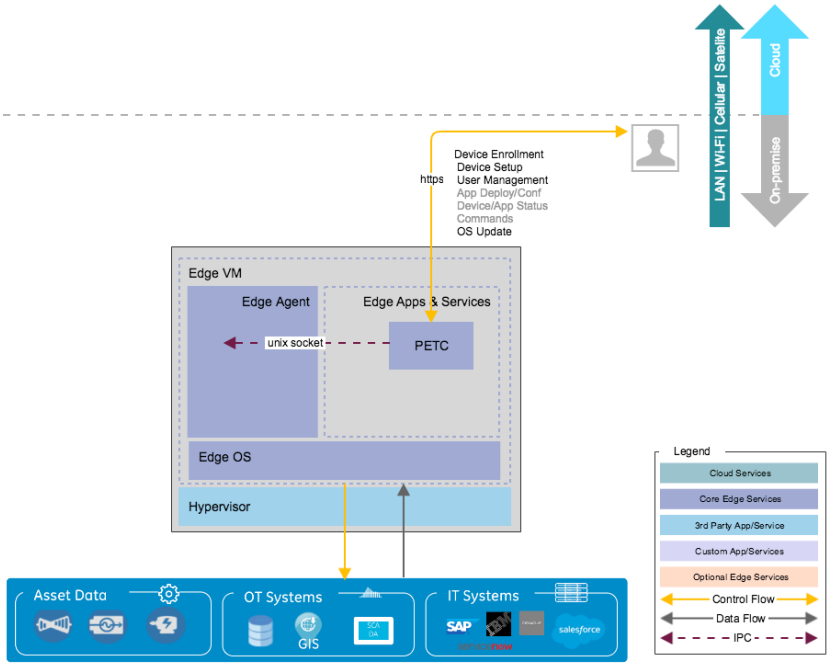
- **Application Configuration and Management** – enables user to upload, maintain, edit, and deploy applications, bills of materials, and configuration packages
- **Command** – sends commands to Edge Agent and applications that are running on the device
- **Device Management** – stores, retrieves, and updates device metadata such as device ID, name, model, and attributes
- **Scheduler** – provides an endpoint for scheduling tasks for devices based on time and priority
- **Alert Management** – generates events from sources that need attention
- **Statistics** – stores and retrieves device resource usage and status history
- **Health Monitoring** – displays the internal health status of Edge Manager to enable deep monitoring of service status
- **API** – provides a single entry point for API requests that are then routed to the backend service
- **User Management** – create users and assign roles

Predix Edge Technician Console Architecture

Predix Edge Technician Console Architecture

The Predix Edge Technician Console (PETC) runs as a system container in Edge OS. It interfaces with the same core services as Edge Agent to accomplish platform management functions.

Figure 6. Predix Edge Technician Console



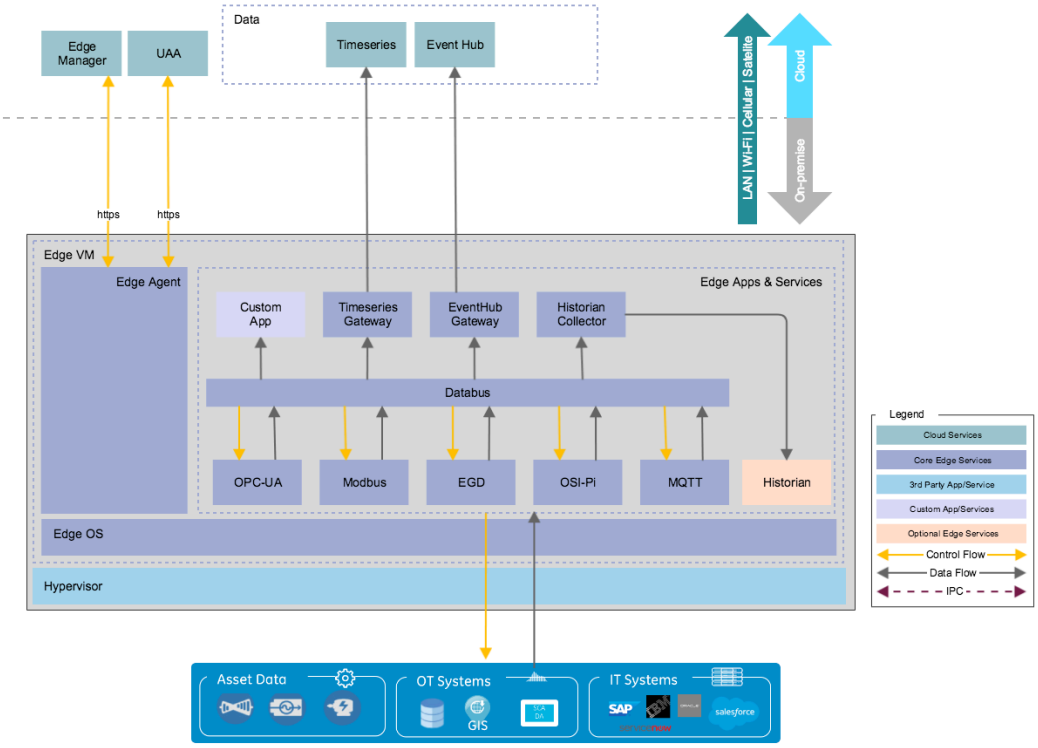
PETC allows users to locally set up and manage their devices and for each device to update underlying OS and manage users. In the future, PETC will support application/VM deployment and configuration, device and application status, execute commands and provide a UI container for application extensibility.

Predix Edge Application Components

Edge Application Components

The following diagram depicts the core components available for applications in Edge and how they interact with each other over the Edge Broker.

Figure 7. Edge Application Components



Edge Application Concepts

- All applications are deployed as Docker containers.
- Any development language can be used so long as it can
    - be deployed as a Docker container in a Linux environment;
    - communicate to MQTT. Most modern languages you would consider have MQTT libraries.
- Each application communicates with other applications by publishing and subscribing to messages on the Edge Broker.

For example:

1. The OPC-UA Protocol Adapter is configured to retrieve tag data and publish it to the broker on a topic named **opcua_data**.
2. Your custom app running as a container subscribes to the **opcua_data** topic, manipulates the data in some way and publishes the results back to the broker on the topic **timeseries_data**.
3. The Time Series Cloud Gateway application subscribes to the **timeseries_data** topic and sends the data to Time Series.

You can use the same “pub/sub” data flow to route data to other applications such as Event Hub Cloud Gateway, Historian, or other custom applications.

For more information on the applications provided with Edge, refer to:

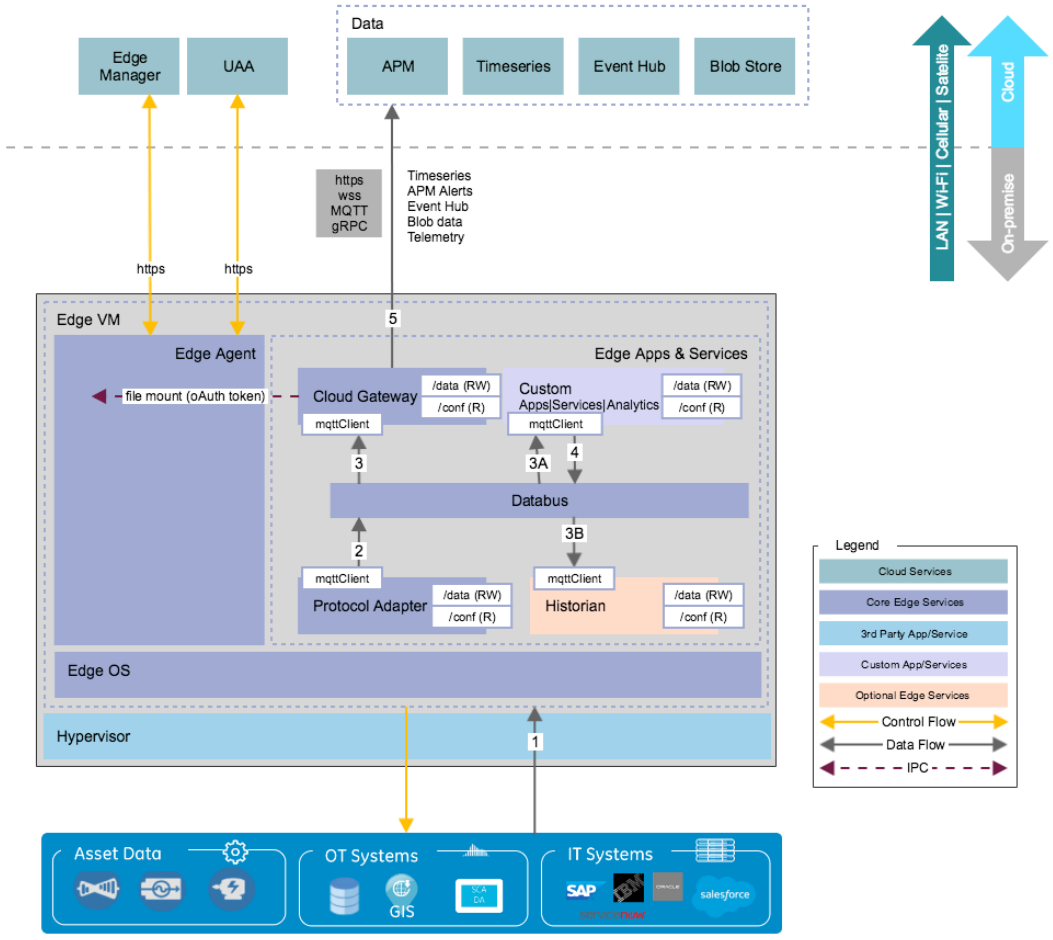
- Protocol Adapters
- Cloud Gateways
- Historian

Predix Edge to Cloud Data Flow

Edge to Cloud Data Flow

The following diagram depicts the data flow from an industrial asset to Edge and onto Cloud Data Services.

Figure 8. Edge to Cloud Data Flow



Edge to Cloud Data Flow

The data flows from edge to cloud as follows:

1. Telemetry data from the asset is retrieved or received from the asset. Depending on the protocol, the data could be broadcast or the protocol adapter would request the data from the asset.
2. The Protocol Translator transforms data to be sent to Time Series and publishes to the Databus.

3. The Cloud Gateway subscribes to data from the Protocol Adapter and sends it to Digital-hosted Cloud Applications and Services.
    a. A custom app can also subscribe to data from the Protocol Translator for post-processing.
    b. Data from the Protocol Translator can also be stored locally in Historian.
4. A custom application can publish data back to the Databus after processing for cloud or local ingestion.
5. Finally, the Cloud Gateway publishes data to Digital-hosted Cloud Applications and Services. Data is held in a store & forward buffer to handle intermittent connectivity.