AWS Whitepaper

Industrial IoT Architecture Patterns

Industrial IoT Architecture Patterns: AWS Whitepaper

Table of Contents

Industrial IoT Architecture Patterns

Publication date: **December 17, 2021** (*Document history*)

Today, industrial companies want to ingest, store, and analyze IoT data, closer to the point where data is generated to enable predictive maintenance, improve quality control, enhance worker safety, and more. Industrial companies can greatly benefit from the industrial edge's ability to solve for use cases which require low latency, optimized bandwidth utilization, need for offline or autonomous operation, and adherence to regulatory guidelines. This whitepaper outlines the design considerations for industrial IoT architectures which use the industrial edge.

Introduction

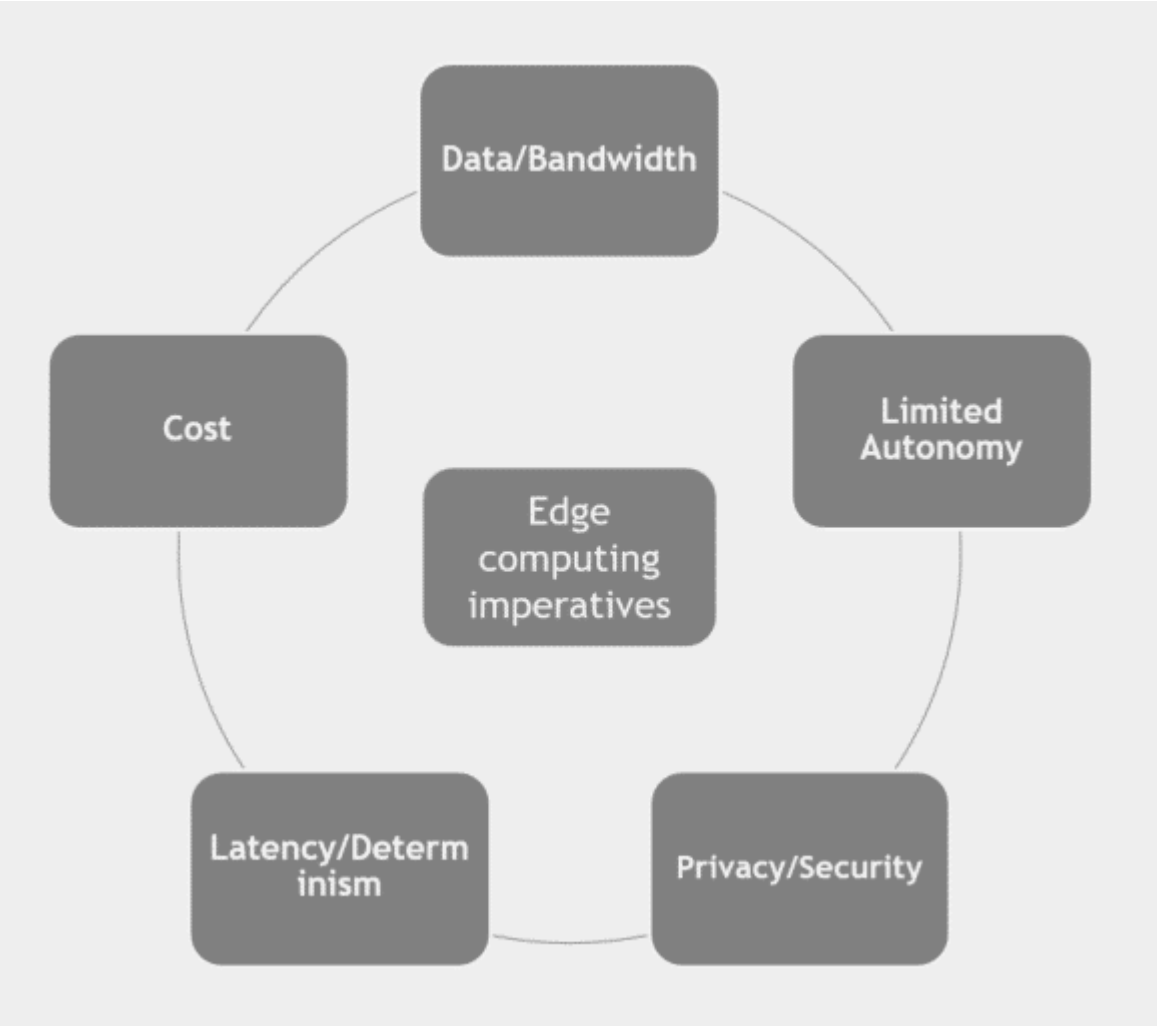
Industrial Edge computing involves hardware and software technologies that enable storage, computing, processing, and networking close to the industrial devices that generates or consumes data within a factory or industrial environment. While Edge computing moves compute closer to the data generation source, the *edge* can range from device hardware to edge gateways to local nodes to cell tower nodes to local data centers. This whitepaper focuses on industrial IoT use cases where edge gateways are placed in a stationary location in an industrial environment and play the role of intermediary between on premise Operational Technology (OT) systems and the cloud and catalog common industrial edge architecture patterns to provide prescriptive guidance to customers implementing industrial IoT systems. Edge gateways fill the critical role of intermediary processing nodes and integrator between industrial assets and the Amazon Web Services (AWS) Cloud, and is heavily influenced by edge computing imperatives. In cases where OT systems are not capable of supporting authentication, authorization, and encryption techniques, the edge gateway can act as a *guardian* to locally interface with these less-capable systems, bridging them to cloud services with strong security patterns. The following section reviews edge computing imperatives.

Edge computing imperatives

Use cases have emerged across various industries that need to combine cloud resources with local processing and storage of data under certain conditions described as edge computing imperatives. Edge computing moves processing and analysis closer to endpoints where data is generated, delivering real-time responsiveness and reducing costs associated with transferring large amounts of information.

Industrial edge computing imperatives are as follows:

- Data volume/bandwidth considerations
- A need for autonomous or disconnected operation
- Privacy, security and data sovereignty concerns
- Low-latency requirements
- Network cost considerations



Edge architecture influenced by edge computing imperatives

Industrial IoT deployments consist of a combination of plant and local Operational Technology (OT), plant/local Information Technology (IT), and remote IT resources which may be in the public cloud or an enterprise datacenter. The benefit of splitting workloads between local and remote processing is to balance the timeliness and high bandwidth of local resources with the scale and elasticity of remote resources. This introduces architectural complexity around managing the different properties of local and remote resources. An Edge Gateway enables you to extend cloud capabilities into industrial environments and help monitor data flows between networks with different properties where direct connectivity between all participants is not recommended. Some of these differences might include **network performance** (latency and throughput), or **security and administrative domain**. For example, it is common in global deployments to use an Edge Gateway between a low-latency local-area network (LAN) and resources on the high-latency wide-area network (WAN). Another example, is an Edge Gateway might mediate between security and

administrative domains such as in a Perdue Enterprise Reference Architecture (PERA), and ANSI/ISA-95 network segmentation.

Industrial edge deployments are heavily influenced by what AWS calls the Three Laws of distributed computing: the **law of physics**, which constrain the latency, throughput and availability of network connectivity; the **law of economics** which determine the cost-effectiveness of transferring ever-increasing volumes of data; and the **law of the land** which regulates how data is handled and where it can be stored.

In addition to understanding the network properties that need to be mediated, and how the three laws effect the deployment, industrial edge architectures must account for the properties and requirements of the data flows involved. In AWS' experience, the most important aspects to consider are the **type** of data flow, the **direction** of data flow, and the **quality** of data flow. While there are many variants within these axes, AWS has identified useful architecture patterns as a starting point for architectural discussions.

This document describes how to navigate the combinations of network properties and data flow types to get the optimal outcomes, subject to the constraints of the Three Laws and provide common industrial edge architecture patterns.

Industrial edge design considerations

Data flow types

There are three primary types of data flows in industrial edge environments: system, telemetry, and object data.

System data is information such as system state, configuration and critical alerts. It is relatively small (less than a gigabyte even in large installations) but it is critical that the data is consistent and highly-available within the local environment, this means that the law of physics prevents this data from residing primarily in the cloud. Changes to the system data should be propagated quickly when the WAN allows it, and must not be dropped if the WAN is down. Most Industrial IoT systems include at least some system data flow, even if the primary purpose of the system is to support other data flows. However, since it is unsafe to have critical control loops depend on WAN connectivity, these flows are low-traffic, such as ANSI/ISA-95 Level 3 and Level 4 (MES, ERP).

System data flows typically support traffic rates up to around 10-100 kb per second, although actual usage is typically much lower. Inbound traffic (such as from the remote resource to the edge gateway) may happen at any time, so polling architectures is not recommended and care must be taken to coordinate with local Network Address Translation (NAT) and firewall policies. Session protocols that originate in the local facility and create a bidirectional connection, such as MQTT or AMQP, are ideal for this type of data. With AWS IoT Greengrass, for example, the MQTT spooler performs this task on system data.

Telemetry data is structured, time-series information about the performance and actions taken by local devices, and is often formatted specifically to fit into existing monitoring and alarming infrastructure. Because these are time-series data, they can grow without bounds and are often much larger than system data. Individual telemetry streams are usually fairly small, in the range of 10s to 100s of kilobytes per second, but the sum of all the telemetry streams in a facility can be quite large. Unlike system data, passing telemetry data to remote resources can tolerate high latencies ranging from seconds to hours in some applications.

In architectures where the source is capable enough and the network segmentation allows, telemetry data can be moved directly from the source to the remote resource after receiving authentication and authorization (often in the form of short-term credentials) from the edge gateway. However, most telemetry sources cannot do this, and some enterprise network security

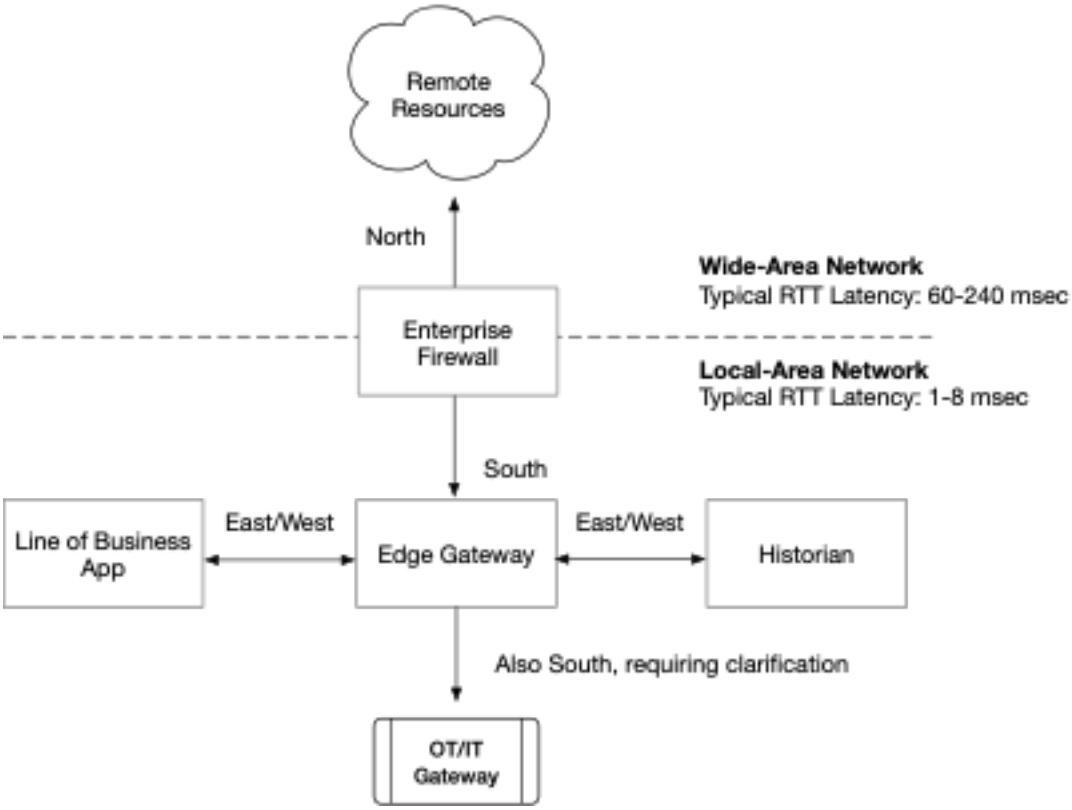
policies disallow it. In those cases, telemetry data can be routed through the edge gateway itself. When doing so, gateways support the higher data rates by taking advantage of lax latency requirements. This allows the gateways to aggregate telemetry data to reduce the number of round-trip transactions to remote resources. This aggregation may include policies on data quality, such as allowing data to be dropped to maintain freshness or queuing data to prevent loss (at the cost of freshness). With AWS IoT Greengrass, for example, the Streams Manager component performs this task.

Object data is unstructured time-series data such as video feeds or analogue sensor streams. The format and contents of this data is specific to the application or sensor, and it is common to apply special algorithms to post-process the data into a more structured metric data. For example, vibration data is a high-resolution audio signal usually analyzed by Fourier or Wavelet analysis, a computationally expensive task. Video and LIDAR sensors are even more data and compute intensive. These object data flows are generally hundreds of Kbs to dozens of Mbs, and in plants with large numbers of sensors the aggregate data bandwidth can be terabits per second. The edge gateway can't process this data, and the WAN can't usually support more than a small fraction of it. As a result, the edge gateway's role in managing object data is as a director: understanding the inventory and purpose of the sensors and using telemetry and system data to inform local processors how to manage it. The edge gateway may also at times direct some useful subset of the object data through the WAN, for example, to provide remote debugging capabilities. The processing application and data flows for object data are not generalized, and few standards exist, meaning custom application development is necessary.

Data flow direction

Directional analogy is used for data flows around the edge gateway, where separate data flows into North-South flows, as shown in the following figure across the boundaries of different networks (such as from LAN to WAN) and East-West flows within the local area network. While an edge gateway is not necessary to manage East-West flows, it is often convenient to have them manage these data flows in cases where they are already in place to support north-south flows. When both are present, it is called a North-South / East-West gateway, which places additional resource load on the edge gateway. As data volume grows, it is common to need multiple gateways with specialized roles, increasing architectural complexity to achieve the correct performance. In the early stages of cloud adoption, it is common for data flows to be primarily *northbound*, meaning that data is moving from the local plant to a remote, centralized data lake for asynchronous processing.

Note: Due to the multi-layer nature of these deployments, the directional analogy can be ambiguous because there are multiple components in each *direction*. It is often necessary therefore to clarify the target, especially for *southbound* data flows.



Data Flow Direction

Data flow quality

Data flow quality has several dimensions: latency, ordering, and loss. In most situations, the Three Laws force difficult tradeoffs in data flow quality. The laws of physics and economics often prevent you from having the ideal latency and ordering properties, and the law of the land may impact whether data loss is acceptable. It is important not to confuse the abstract concept of data flow quality with protocol-level concerns such as MQTT QoS, AMQP QoS, TCP retransmission, TLS retransmission, or OPC-UA metric quality values. It's important to define the abstract data quality needs for the use case, such as data must arrive in order or data loss cannot be tolerated. It's also important to understand the protocols already in place to determine how to achieve the data quality requirements within the constraints of the network protocols.

A key concern for data flow quality is the appropriate behavior in exceptional circumstances, such as long-term network disruption. Because edge gateways are installed at connection points

between networks, it is often not cost-effective or physically feasible to provide fully redundant connections. The Edge Gateway architecture must ensure that system requirements are met even in the face of hours- or days-long disruptions.

Performance considerations

Industrial IoT (IIoT) systems consist of a mix of general-purpose IT components such as relational databases and message brokers, and specialized applications which are purpose-built for a particular task. The performance of specialized applications is recommended rather than attempting the same task with general-purpose components, but assembling general-purpose components is often faster and cheaper. For example, a purpose-build packet-handling pipeline written in C and running in the operating system kernel can easily move 10 GBs on a common Intel-class PC, whereas a general-purpose message broker applied to the same task may only be able to move 1 MBs on the same hardware. The following architecture patterns can be used as-is but it is common to need to modify them to accommodate specialist components for signals-processing, computer vision, machine-learning, or high-data-rate workloads.

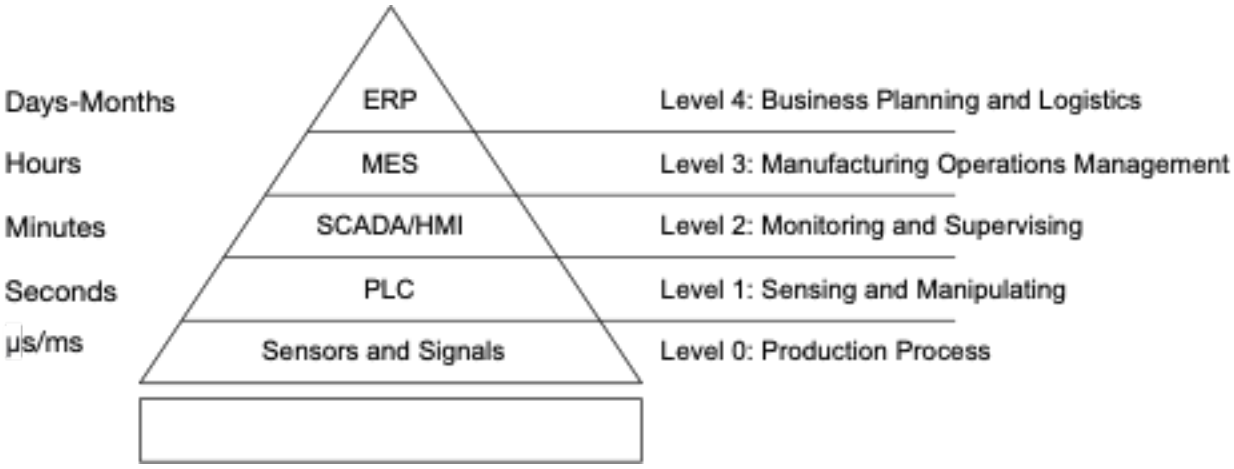
Protocol considerations

IIoT deployments typically implement a number of communications protocols, which creates complexity. In general, consolidating on a small number of protocols helps reduce that complexity. However, the security and performance differences between the LAN and the WAN may require the use of different protocols, especially in cases where the WAN is the public internet. For example, while it is widely deployed inside controlled local-area networks, OPC-UA and OPC-DA are not safe to expose to the internet and when used in a LAN environment, its recommended to use the newer version of OPC with security features enabled and implement strong perimeter security. Another example, synchronous communications such as MQTT in QoS 1 or QoS 2 modes have low performance as network latency increases.

One of the key roles of the edge gateway is to mediate between the protocols that are suitable for each environment. For example, OPC-UA messages need to be proxied through secure tunnel to provide appropriate authentication and flow control. Another example, an edge gateway can perform application-level message delivery validation for messages sent over QoS 0, achieving high performance on high-latency links while still achieving the delivery guarantees of QoS 1.

Mapping to the ISA-95 Model

ANSI/ISA-95 provides a common view of industrial IT systems, which can be helpful for understanding the role of edge gateways and hybrid cloud IIoT environments. Because of the latency and availability characteristics of the WAN, it is not recommended to take a dependency on remote resources for Level 0-1 systems. Critical Level 2 systems should also be local, but additional remote monitoring may be layered on so that operations staff can observe global trends. It is recommended that data delays and gaps for these remote Level 2 systems can cause only observability failures and not production impact.



The ANSI/ISA-95 Pyramid Showing Expected System Response Times

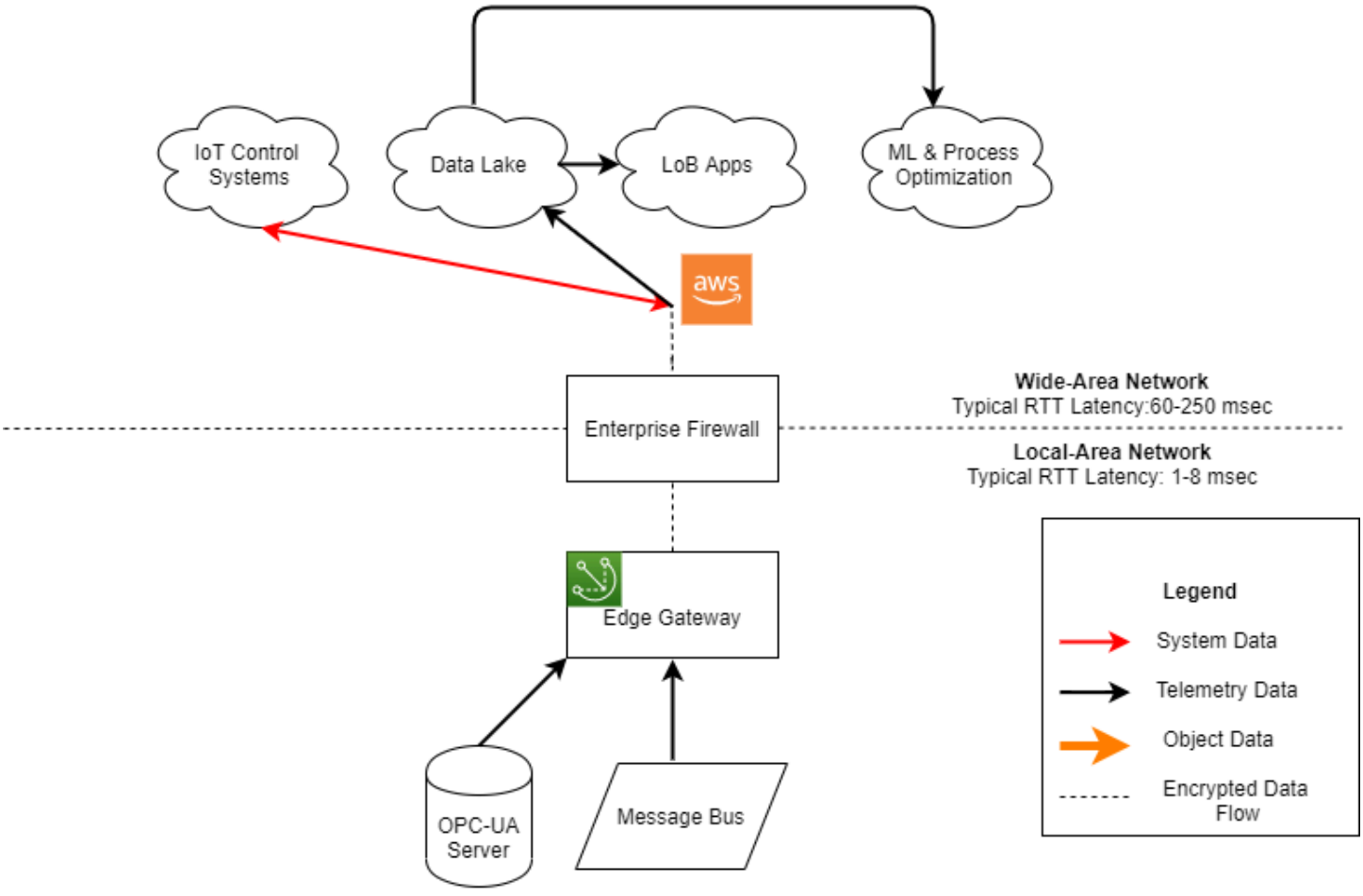
Level 3 and Level 4 systems can benefit from the properties of remote computing, including the ability to aggregate data from multiple sites and the ability to perform compute-intensive analytics tasks.

Edge gateways are generally found in layer 3 and OI/IT Gateways responsible for industrial protocol conversion are often found in layer 2 of the Purdue model.

Common IIoT Edge scenarios and use cases

The most common type of IIoT use case, and the place where most IIoT initiatives start, is with a one-way transfer of telemetry data from the local site to a remote *data lake* where the data from multiple industrial sites is made available for analytics and monitoring workloads.

These workloads include a control channel that is mostly used for managing the edge gateway itself, such as configuring data export and security policies. This system data is real-time when the WAN link is available and uses asynchronous state management through device shadows to guard against link failure. Typically, the telemetry data is aggregated on the edge gateway and sent northbound in batches, with acknowledgement and retry to guard against data loss. The following architecture shows system flow messages up to around 10 per second and telemetry messages up to around 1,000 per second, with enough local storage to avoid data loss for several hours of link failure.



Telemetry export through edge gateway

There are some implementation choices to make for this workload, as the gateway needs to adapt to the specifics of the telemetry flows. Low-traffic telemetry may be sent directly through the system channel and managed by the IoT control systems, though depending on WAN latency this will create performance bottlenecks around 10-40 messages per second. For higher-throughput data flows, the gateway will need to pre-process the data in transit. There are a variety of techniques for this. Where firewall rules allow multiple northbound connections, parallelizing the data flow can overcome the effects of network latency and allow high throughput. Where firewall rules do not allow parallelism, the gateway must aggregate telemetry into batches that will efficiently utilize the high-latency northbound link. AWS IoT Greengrass supports this batching through its Stream Manager component.

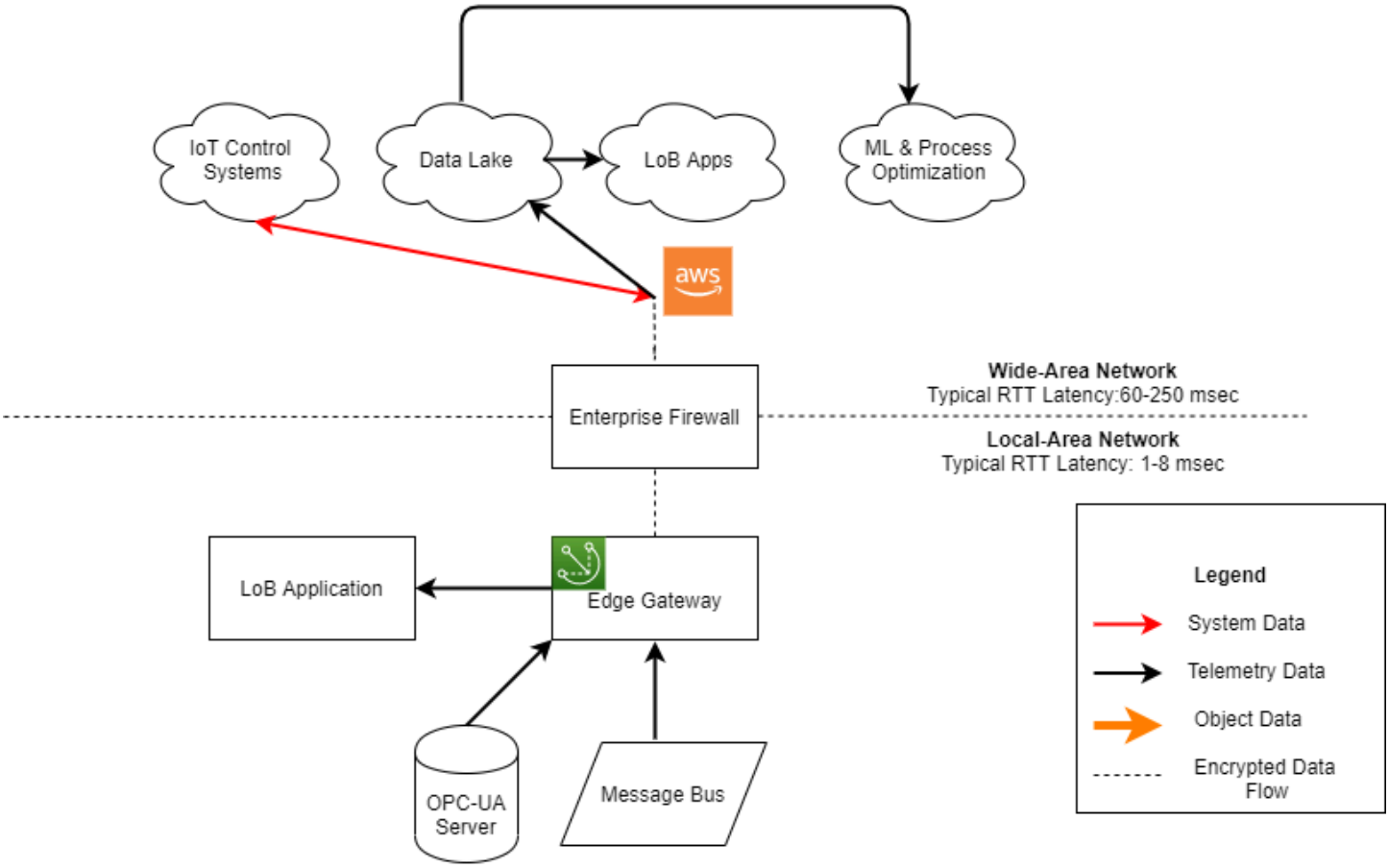
The following sections discuss industrial edge architecture variants for other IIoT use cases.

Topics

Variant 1: Adding East/West Telemetry Routing

In this variant, Line of Business Applications on the local-area network need access to the telemetry data for local display, processing, or persistence. This places an additional load on the edge gateway, as it needs to make routing decisions about where certain data should be routed. It may also require the gateway to add new protocol support depending on the needs of the local applications. In many cases, the local application could access the data without relying on the

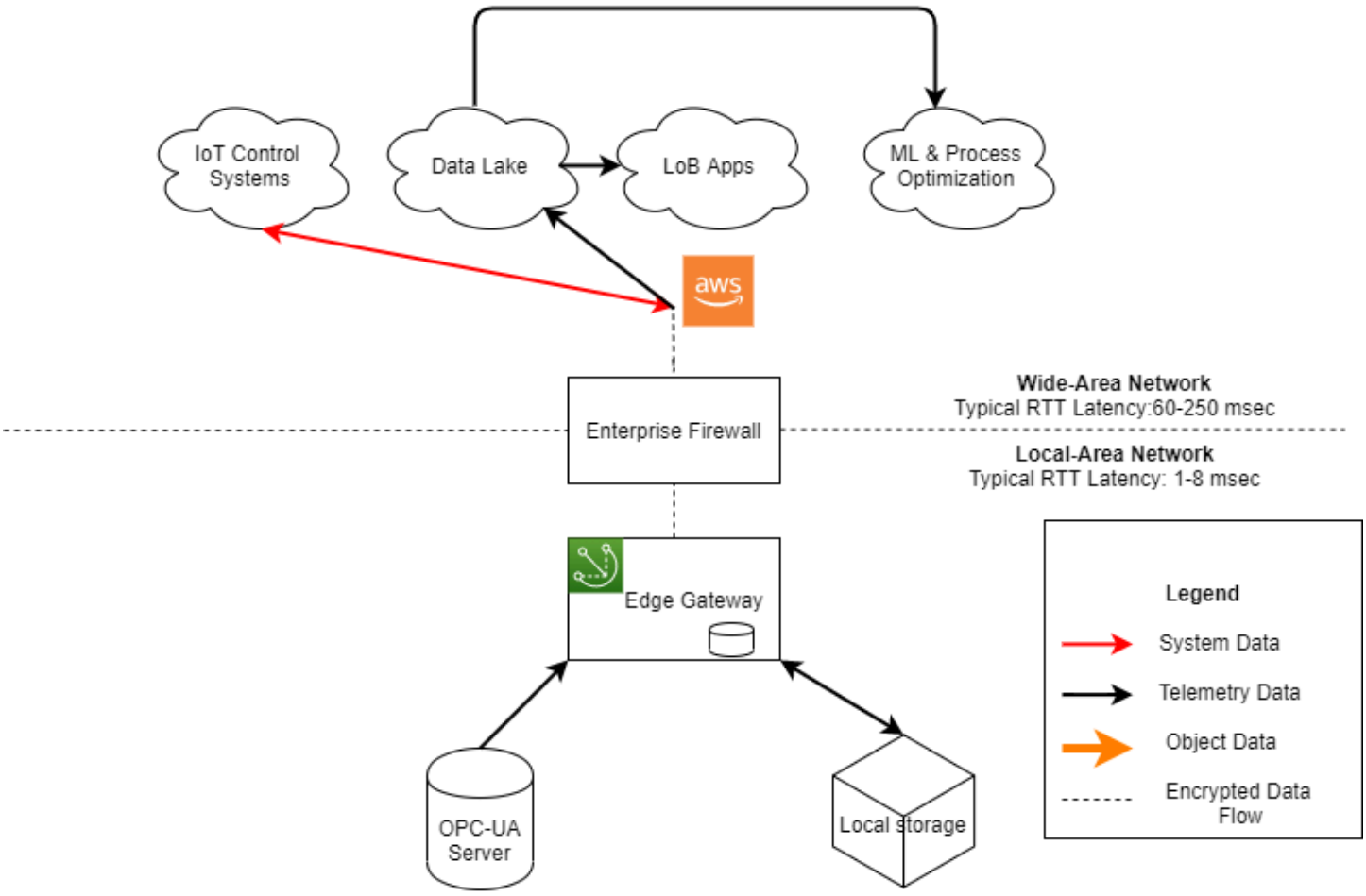
edge gateway, which might simplify the architecture or performance considerations. However, sometimes the gateway is providing local processing of the telemetry or providing access control to the data, making it easier to hook the applications into the gateway data flows.



Telemetry export with local East/West telemetry routing

Variant 2: Critical telemetry

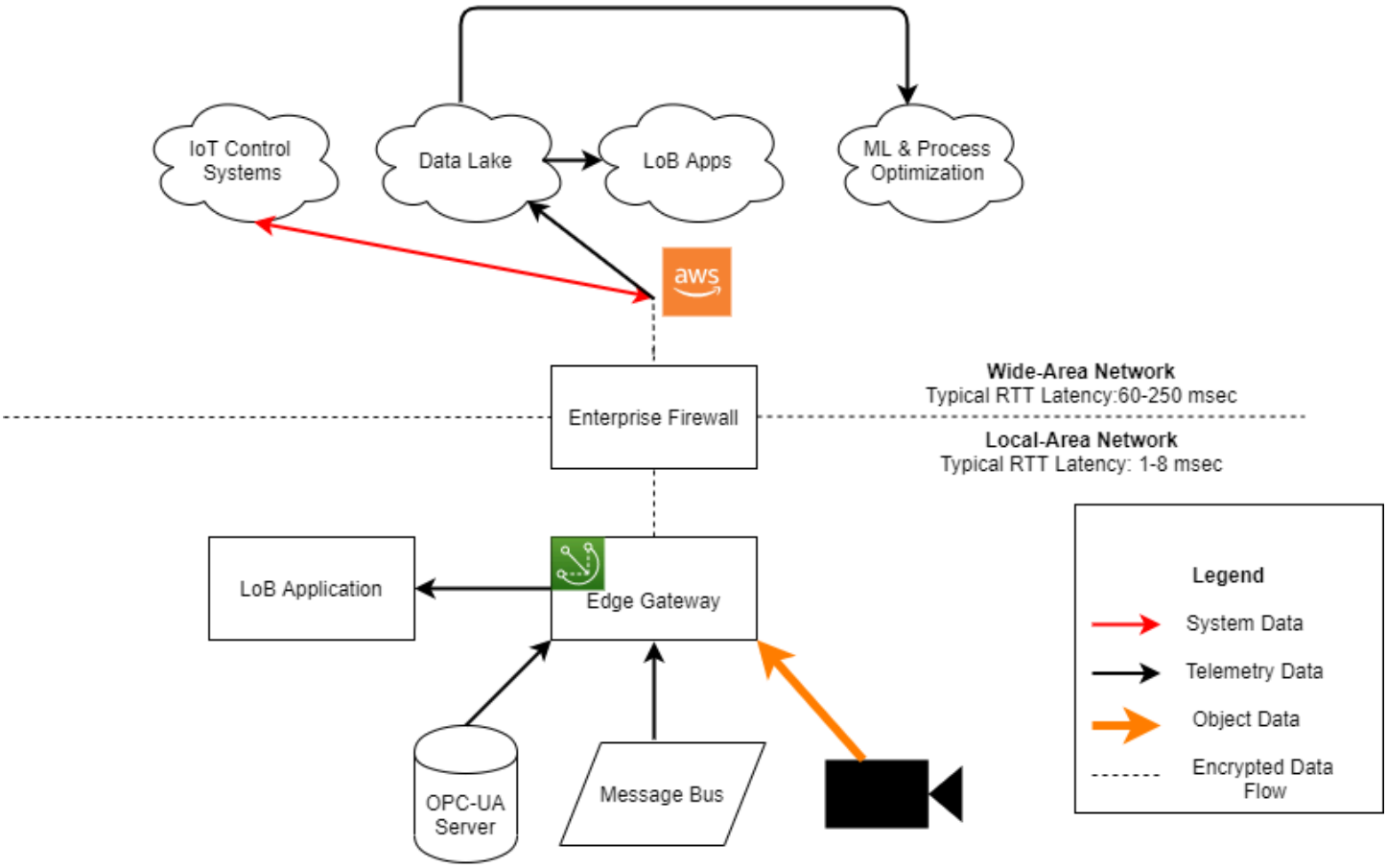
Some workloads need to be regulated or otherwise mission-critical telemetry data must be preserved even in the case of a long-term WAN outage. As a result, a durable buffer must be maintained either on the edge gateway, or in front of it.



Local persistence of critical data

Variant 3: Object Data

In this variant, the application relies on object data to be processed locally and measurements (which is transformed into telemetry data) is sent across the WAN to remote storage. If a large proportion of the object data needs to be exported to remote storage, it is recommended to bypass the edge gateway and have each device connect directly or through a dedicated media proxy. However, the edge gateway must be configured to directly proxy object data streams at critical points, such as to debug an error.



Handling object data

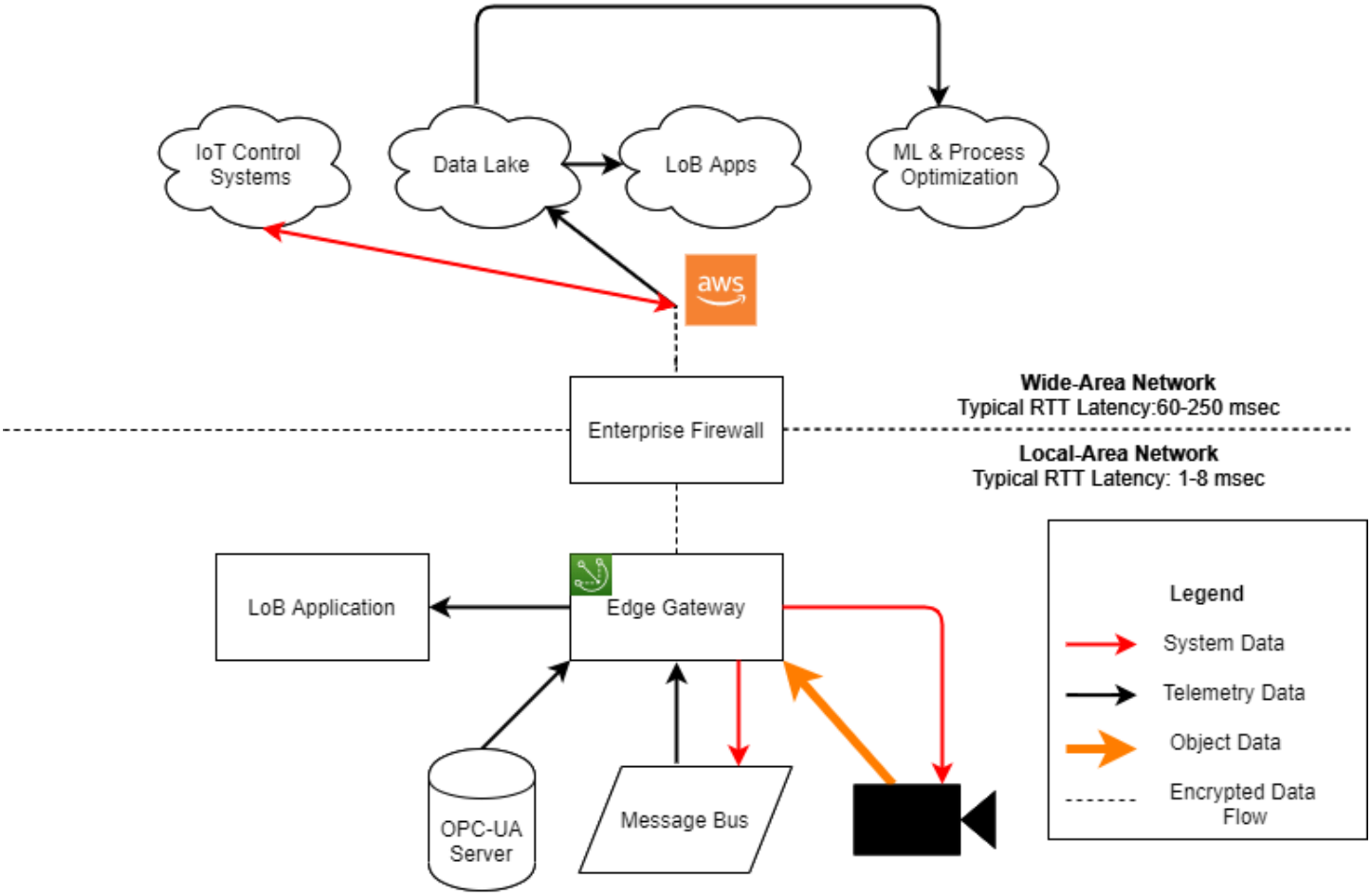
Variant 4: Adding remote control

As IIoT systems mature and use cases expand, industrial enterprises increase the frequency with which they can gain actionable insights from their IoT data. This leads to an increase in demand for automated, low-latency changes in the production environment, rather than requiring manual change-management processes. To support this, the edge gateway takes on an additional role in injecting control and configuration messages to the local message bus or directly to attached pieces of equipment.

The previously described variants of edge gateway deployments can, at most, create data loss of information destined for remote storage and processing. While this is often business critical, it is not safety critical. Interacting with local control flows can create a safety critical or production

critical situation. As a result, extra precautions must be taken when implementing remote control in IIoT systems.

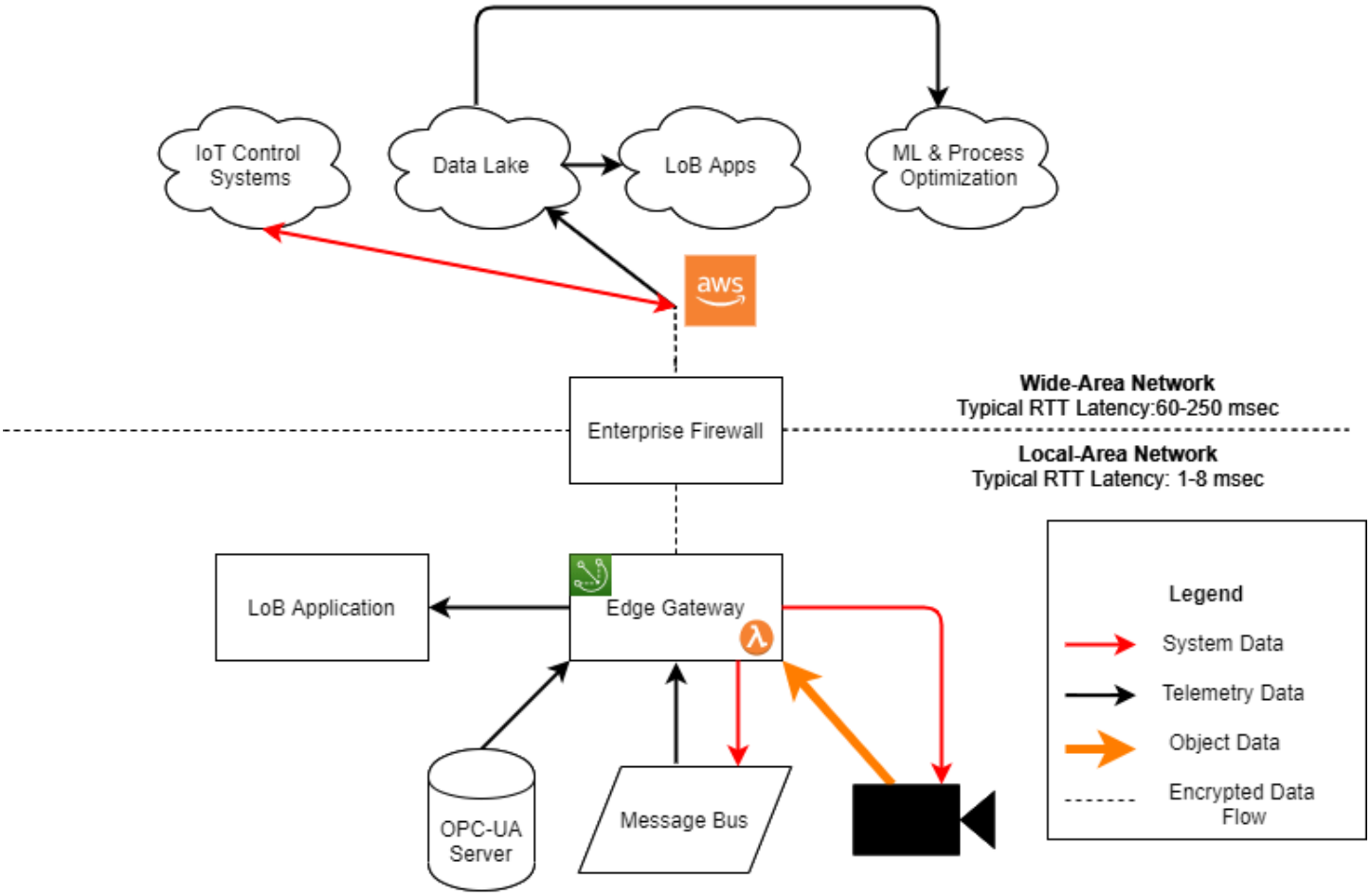
In local deployments, techniques such as exactly once message delivery (such as MQTT QoS 2) are used to reduce risk around critical control flows. These approaches are not safe or recommended for hybrid architectures with WAN data flows. Instead, it is recommended that applications are built to achieve the technical property known as *idempotence*, in which message delays, duplication or misordering do not affect the outcome of the process. AWS IoT Device Shadow provides the architectural building blocks for correctly communicating state transitions in a hybrid architecture, which are then presented as MQTT messages to clients on the local network.



Remote control

Variant 5: Edge analytics/ML

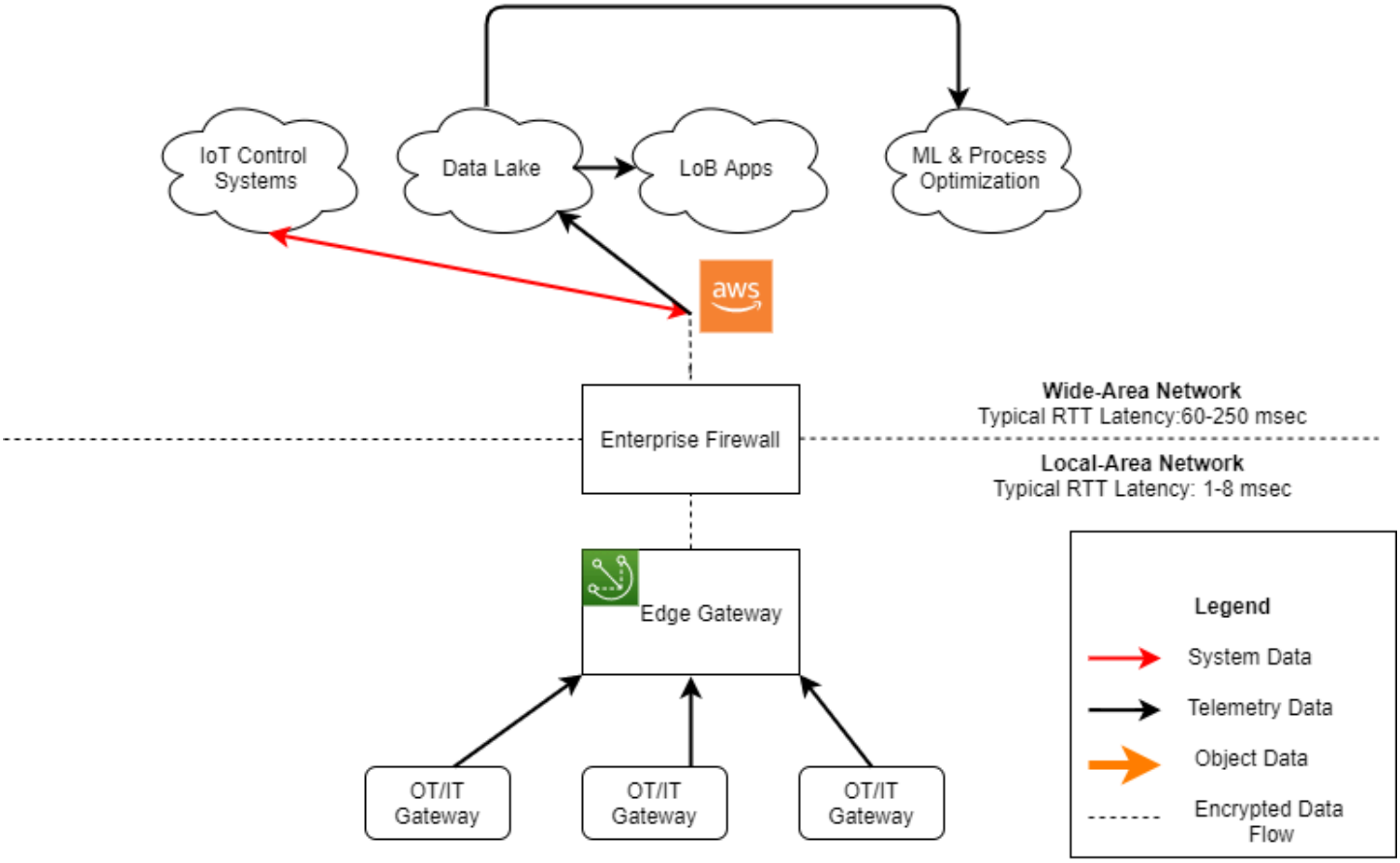
In this variant, in addition to the roles mentioned previously, the edge gateway plays the additional role of processing data locally and running machine learning inference. This variant enables you to deploy machine learning models built and trained in the cloud and run them in the factory on the edge gateway. This is needed for industrial use cases such as inline quality inspection and critical vibration monitoring which requires high volume/high frequency data processing and low latency such that local actions can be taken as soon as an anomaly is detected. An Edge Gateway can send the results and data to the cloud for further analysis and for machine learning (ML) model retraining.



Edge analytics/ML

Variant 6: Nested Gateways

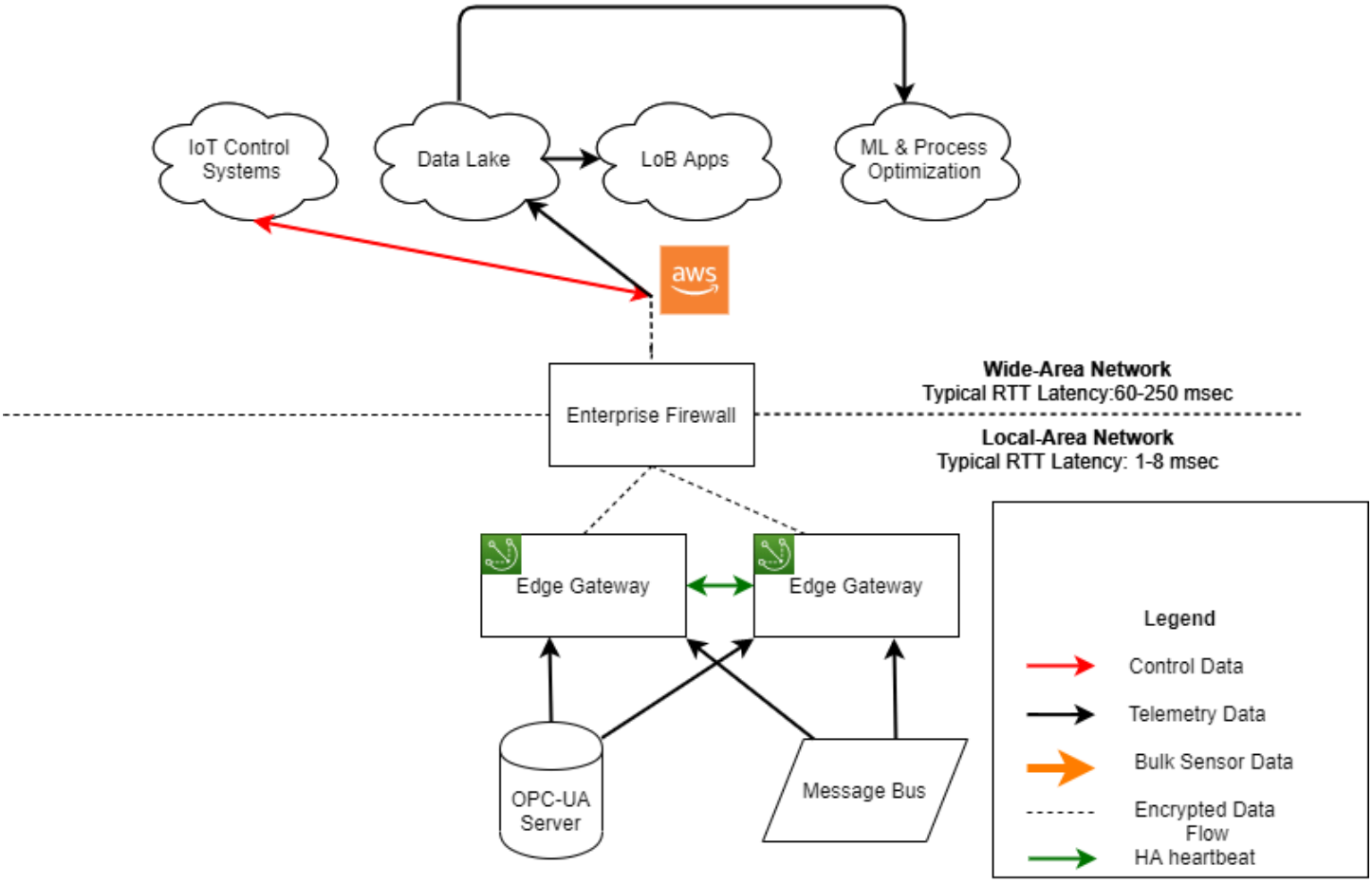
In this variant, either one or many OT/IT Gateways are connected to the edge gateway. The OT/IT Gateway plays the role of connecting to downstream industrial systems and can perform protocol conversion (for example Modbus TCP, Profinet, Ethernet IP to OPC UA or MQTT). With this variant the common task of industrial protocol conversion can be off-loaded from the edge gateway to the OT/IT Gateway. Multiple OT/IT Gateways can be connected to a single edge gateway. Most legacy industrial protocols are insecure and the OT/IT Gateway can play the role of converting an insecure industrial protocol to a secure protocol before connecting to the edge gateway. The OT/IT Gateways are installed in the OT network and as close as possible to the insecure industrial control (ICS/OT) systems. Edge gateways are generally found in layer 3 and OI/IT gateways responsible for industrial protocol conversion are often found in layer 2 of the ANSI/ISA-95 model. Third party tools from PTC Kepware, Inductive Automation Ignition Edge, Matrikon, Softing, and etc. can be used for protocol conversion.



OT/IT gateways connected to edge gateway

Variant 7: Edge Gateway high availability

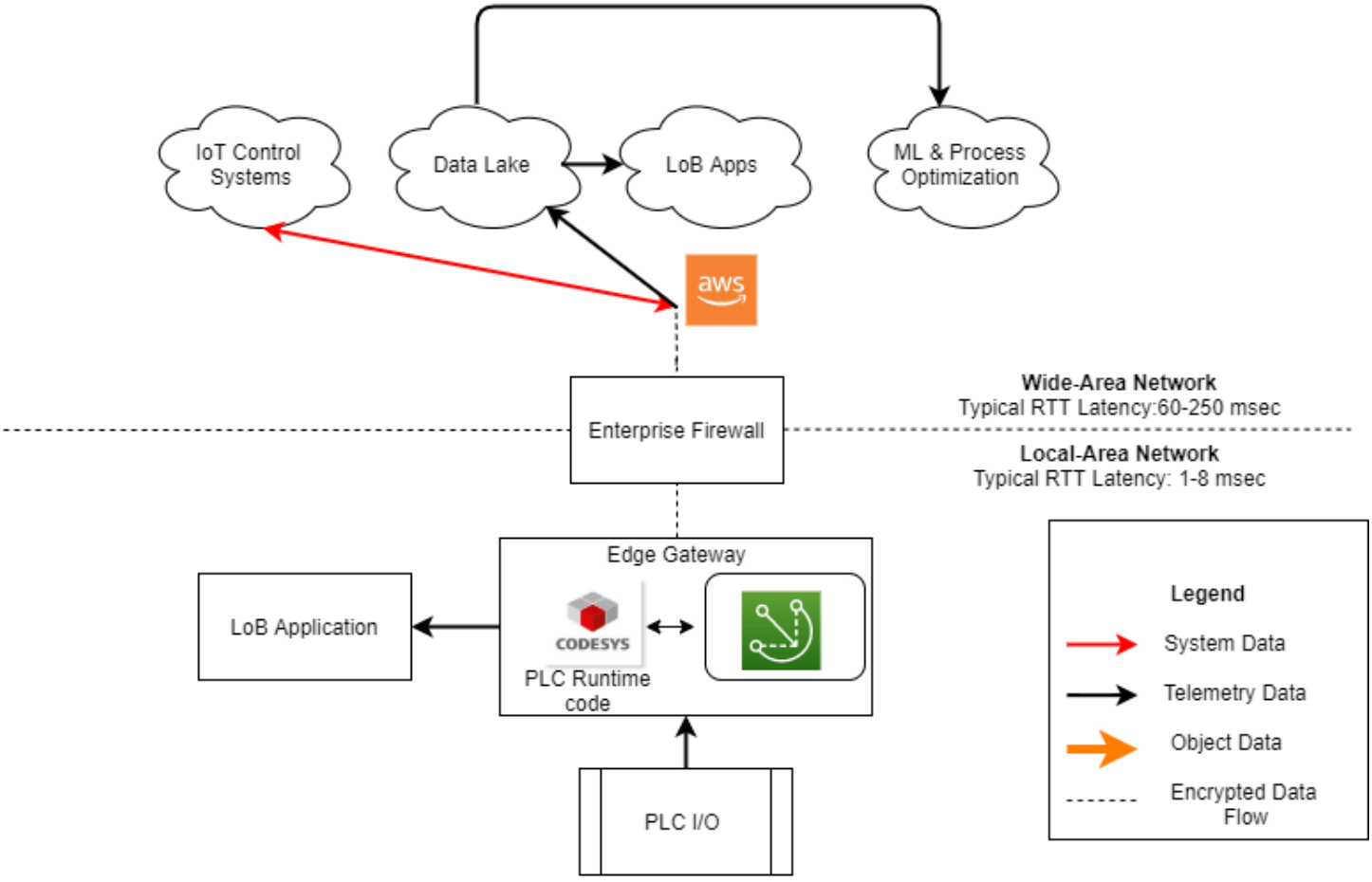
In this variant, the edge gateway is set up in a high availability (HA) configuration to offer redundancy in case of failure (hardware or software) in a single edge gateway and HA could apply to the other variants described in this document. As IIoT edge applications mature, HA in edge gateways will become increasingly important. Partner solution from EdgeIQ offers HA for AWS IoT Greengrass version 1, and provides HA with redundancy and supports additional features such as determining if an edge component or device needs to be restarted or rebooted and this can be achieved in an automated way.



Edge gateway HA example

Variant 8: Edge Gateway running a soft PLC (Replacing traditional PLCs)

The variants discussed so far assume that an industrial automation controller (for example PLC) controls industrial machines and factory equipment. With the demands of Industry 4.0, these proprietary industrial controllers have many limitations such as limited memory and CPU capacity, no support for higher level programming languages, no support for TLS, limited capacity to support IIoT protocols like MQTT, no support for code deployments, etc. In this variant, it is possible to run a software PLC (for example Codesys) on the edge gateway. The edge gateway is connected to downstream industrial systems and input/output PLC modules. In this variant, customers can replace their legacy PLC systems (for example from companies like Rockwell, SIEMENS, Schneider) with a software PLC like Codesys which runs on traditional industrial PC/ edge gateway.

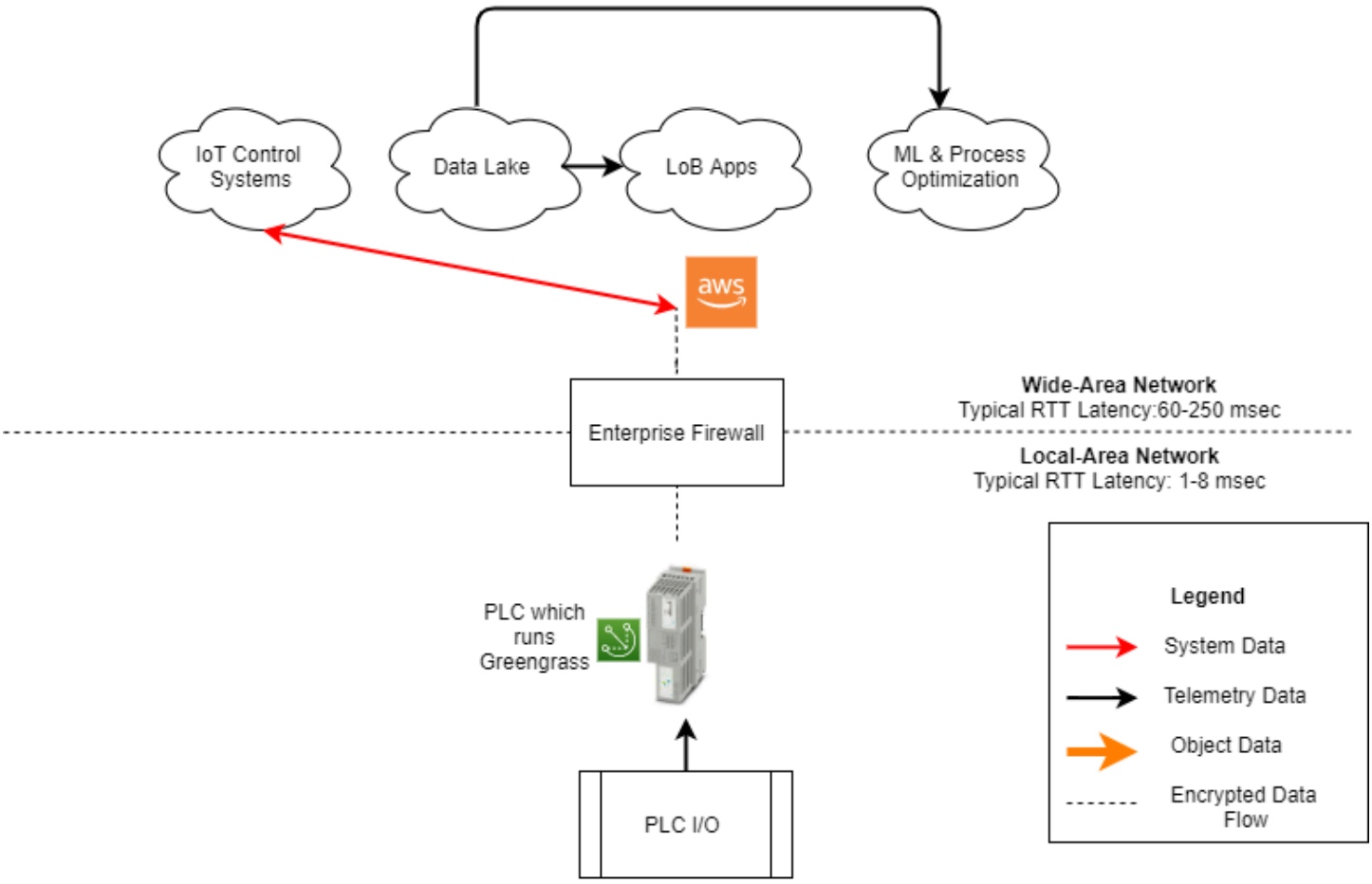


Edge gateway with soft PLC

Edge gateway with soft PLC

Variant 9: Modern PLCs acting as Edge Gateway

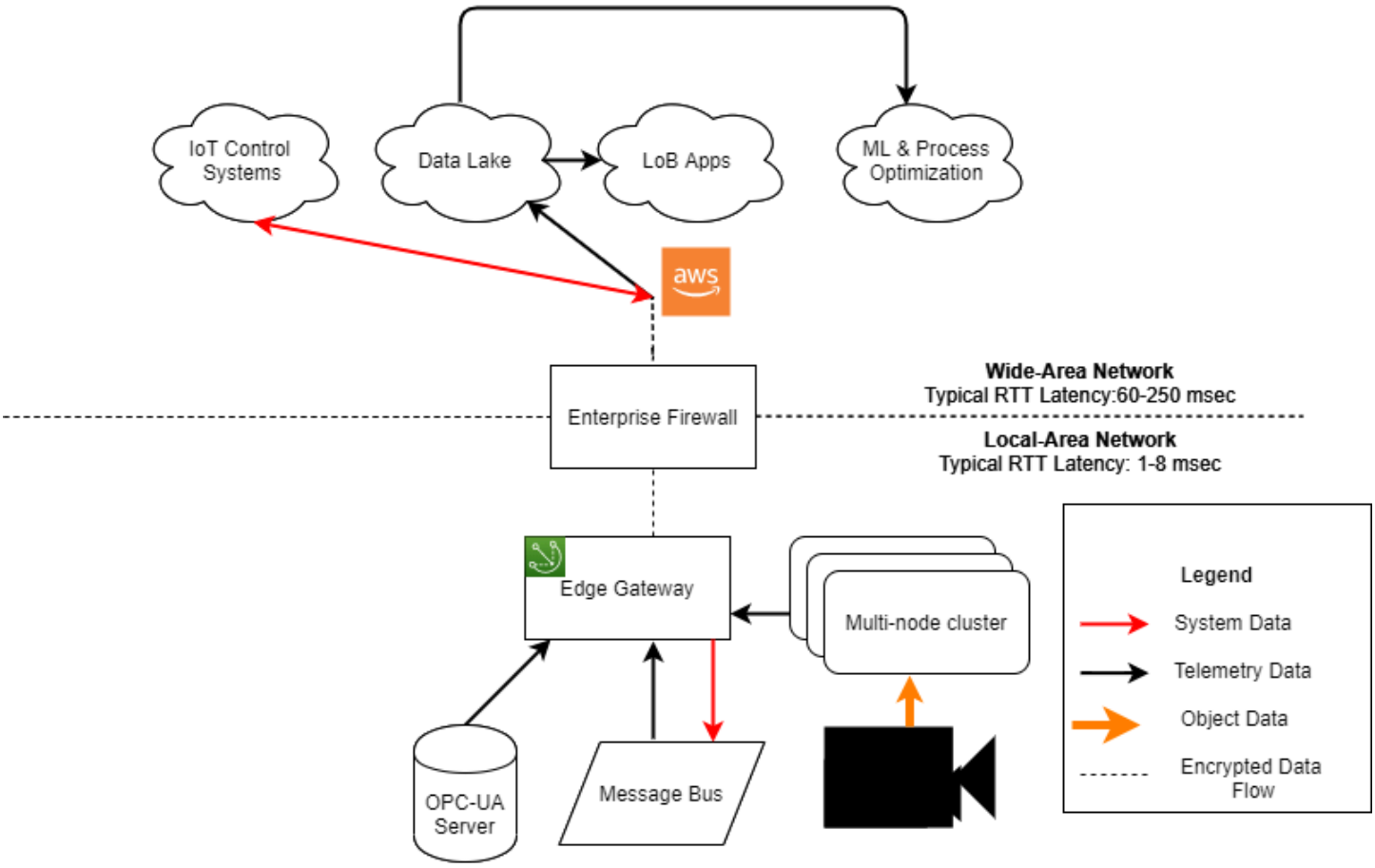
In this variant, modern PLCs based on Linux can act as the edge gateway, can run AWS IoT Greengrass, and connect directly to AWS IoT Core eliminating the need for additional standalone Gateways. Most PLCs are restrictive, proprietary systems that were not designed to support open-source software. As a result, companies have to rely on manufacturers for support, patching, and other technical assistance. Phoenix Contact's PLCnext offers an open Linux environment for edge computing with access to more data through IoT systems and more flexibility with open source code. By leveraging communities like GitHub, PLCnext community, or the various Linux networks, organizations stand to make substantial gains in customizing code to specific applications. Phoenix Contact PLCnext and Beckhoff TwinCAT IoT are just two examples of Linux based PLCs which can run AWS IoT Greengrass and connect directly to AWS IoT core, eliminating the need for additional standalone edge gateways.



PLC acting as an edge gateway

Variant 10: Cluster computing at the edge

Computation intensive applications (for example industrial computer vision) require high computing capacity for data processing and storage. In this variant, clusters can be used at the edge to meet the storage, computation, low latency, high performance, and high bandwidth requirements. The edge computing cluster could be one or many edge servers set up in a cluster configuration and managed and operated like an edge device. For example, in industrial computer vision use cases, one edge server can be connected to cameras on a production line and performing image pre-processing such as image compression, a second edge server doing quality inspection ML inference, and a third edge server performing post processing and sending data to other on-premise systems, edge gateway, and/or the cloud. For use cases which demand low latency and large image file processing, doing this at the edge is more efficient and cost effective.



Cluster computing at the edge

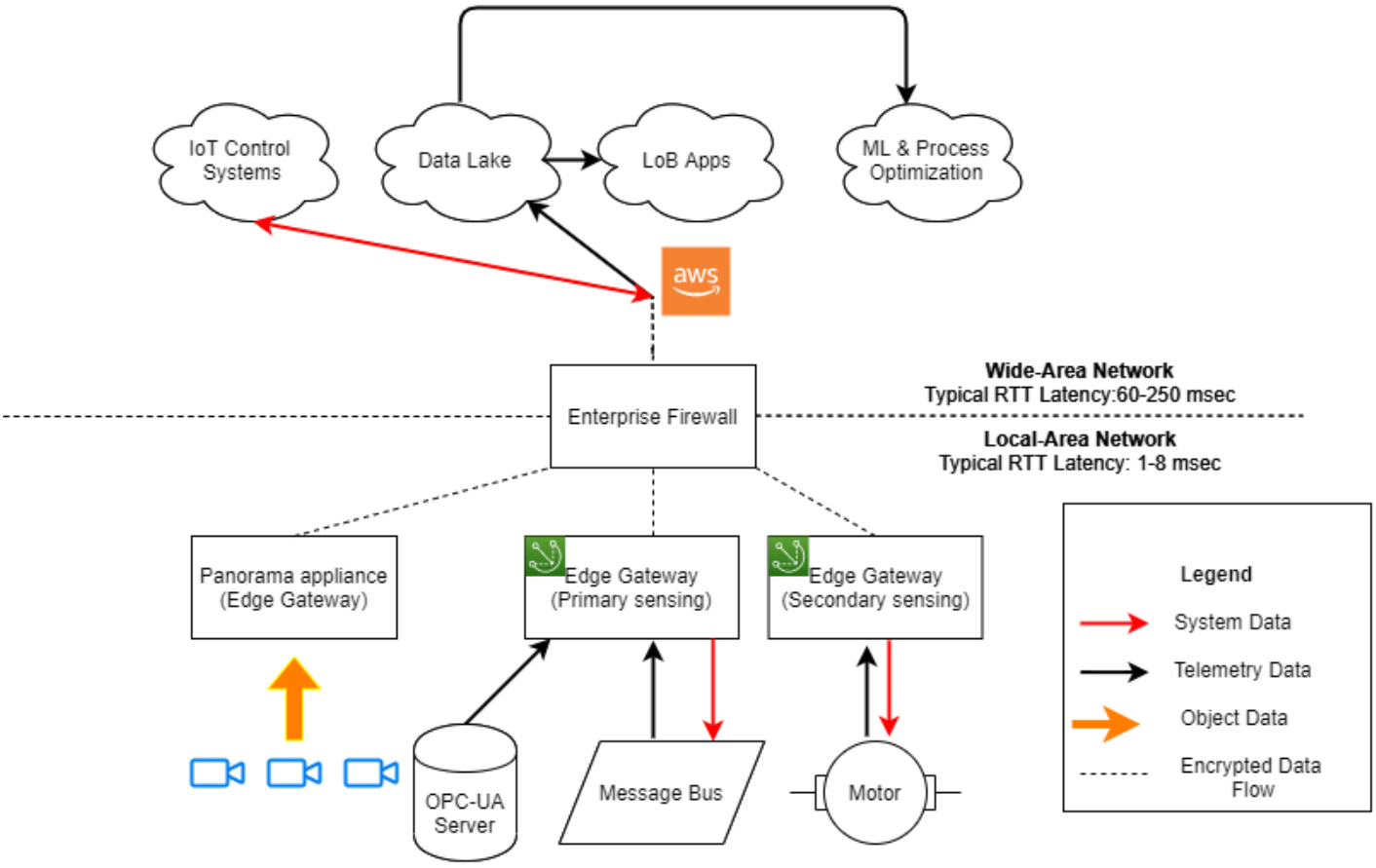
Variant 11: Secondary sensing use case

In this variant, a secondary sensing solution with an integrated edge gateway is added to existing plant infrastructure. This is a non-invasive way to add new sensors (such as temperature and vibration) to critical pieces of industrial equipment (such as rotating equipment: turbines, generators, motors, pumps, compressors, engines, fans, blowers, etc.) without impacting normal operations in a factory and without causing unplanned downtime.

Amazon Monitron, AWS Panorama and CloudRAIL (AWS Partner solution) are three examples of this architecture pattern. With this approach, it is easy to add new sensors to existing critical assets to support IIoT use cases such as asset condition monitoring and predictive maintenance.

The AWS Panorama appliance is a hardware device that you can install on your network to connect to existing cameras within your facility, to run multiple computer vision models on concurrent

video streams. Amazon Monitron is an end-to-end system that uses machine learning (ML) to detect abnormal behavior in industrial machinery, enabling you to implement predictive maintenance and reduce unplanned downtime.



Secondary sensing

Conclusion

Industrial customers can greatly benefit from the industrial edge's unique ability to solve for use cases which require reduced latency, optimized bandwidth utilization, offline or autonomous operation, and adherence to regulatory or security guidelines based on the physical placement of applications and data in an explicit location such as a province or country. Since industrial IoT workloads can be diverse and complex, it is important to understand the use case requirements, customer's industrial landscape, and desired business outcomes before identifying the edge gateway architecture pattern which best fits the project. Selecting the right edge architecture pattern is an important step towards building a secure, high-performing, resilient, reliable, and cost effective industrial IoT solution. While single edge gateway implementations for a proof of concept or pilot project can be straightforward, industrial customers need to plan for implementing such a system at scale, including enterprise-grade security, orchestration, life cycle management, and governance.

Contributors

Contributors to this document include:

- Ryan Dsouza, Principal Solutions Architect, Amazon Web Services
- Brad Behm, Senior Principal Software Engineer, Amazon Web Services

Further reading

For additional information, see:

- AWS Architecture Center
- AWS for the Edge
- Security at the Edge: Core Principles
- AWS at the Edge: A Cloud Without Boundaries whitepaper
- AWS IoT Lens
- Security Best Practices for Manufacturing OT Whitepaper
- Ten security golden rules for Industrial IoT solutions

Document history

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Whitepaper first published.	December 17, 2021

Note

To subscribe to RSS updates, you must have an RSS plug-in enabled for the browser that you are using.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.