AWS Whitepaper

Security Best Practices for Manufacturing OT

Security Best Practices for Manufacturing OT: AWS Whitepaper

Table of Contents

This whitepaper is for historical reference only. Some content might be outdated and some links might not be available.

Security Best Practices for Manufacturing OT

Publication date: **May 21, 2021** (*Document history and contributors*)

Abstract

New developments in cloud, Internet of Things (IoT), and edge computing have opened the door for traditionally on-premises manufacturing operations technology (OT) workloads to evolve into hybrid workloads. This whitepaper describes security best practices to design, deploy, and architect these on-premises hybrid manufacturing workloads for the AWS Cloud.

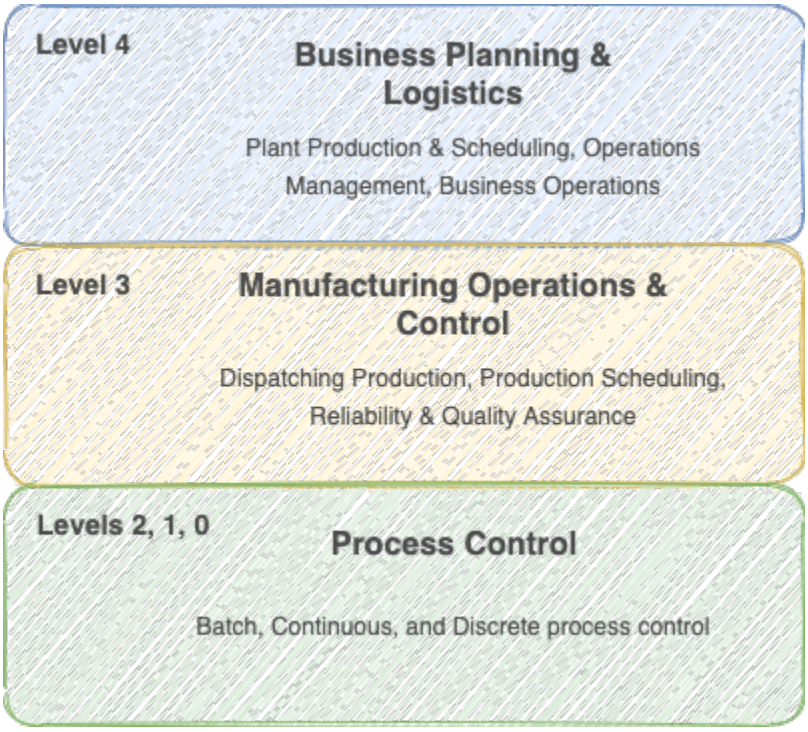
Introduction

Traditionally, manufacturing workloads can be categorized as operation technology (OT) workloads and information technology (IT) workloads. OT workloads support production operations. Enterprise operations are supported by IT workloads.

OT workloads are typically located within factories, because they support operations on the production floor. However, the adoption of cloud, IoT, and edge computing enables OT workloads to transform from on-premises to hybrid workloads, which can take advantage of cloud services.

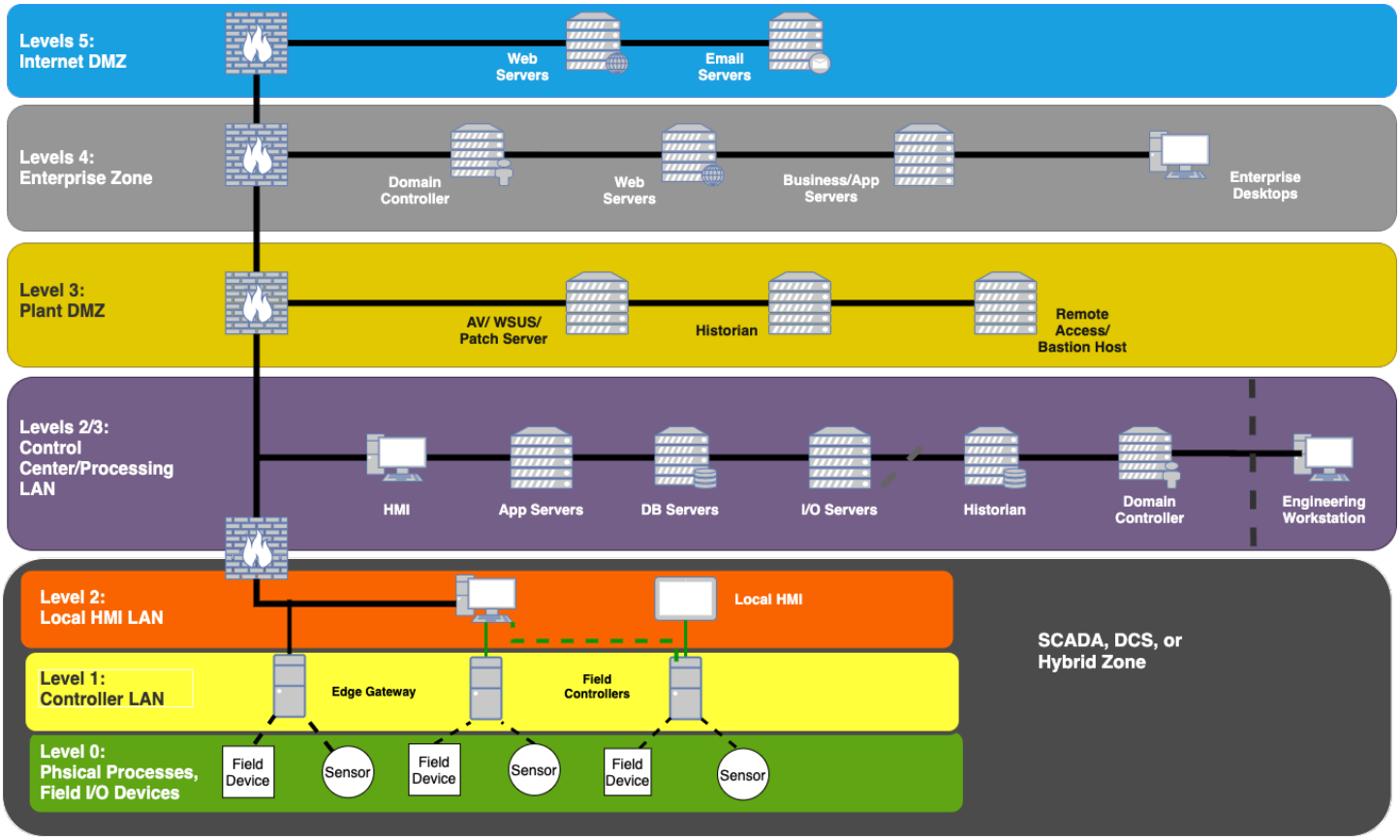
This document describes the security best practices to design, deploy, and architect distributed manufacturing workloads for the AWS Cloud. The focus of this document is securing resources at the industrial edge. The best practices for securing cloud resources are documented in the Security Pillar of the AWS Well-Architected Framework.

The Purdue model, as shown the following figure, is used as the backdrop to define cloud integration points and placement for resources for manufacturing workloads. The Purdue model is a reference model for the manufacturing industry, and is used as the basis for the International Society of Automation ISA-95 standard to define detailed information models for manufacturing and enterprise integration.



Purdue enterprise reference architecture model

Taking the Purdue reference model and applying it to an industrial control network illustrates the distribution of IT and OT functions, as seen in the following figure:



Purdue Model representation of an industrial control network

Levels 4 and 5 are in the IT domain. In most enterprises, the enterprise network boundary to the internet (level 5) is traditionally controlled by the IT organization, along with business operations served by the infrastructure in level 4. The most frequently used connection method between the enterprise and the AWS Cloud is over the internet through the internet DMZ firewall in level 5.

The firewall between levels 3 and 4 is the interface between the corporate data backbone and the local industrial facility. The functions implemented in levels 3 and below are tied to production operations and control.

Levels 2, 1, and 0 form what is sometimes referred to as the Cell / Area zone. Level 2 contains human machine interface (HMI), Supervisory Control and Data Acquisition (SCADA), and Distributed Control System (DCS) used to interact with production control assets (field devices and sensors) in level 0 via logical controllers in level 1.

The emergence of connected sensors and controllers that take advantage of IoT technologies has introduced new gateway devices that can be used with local HMI assets, but are purposely designed to send industrial asset and machine data to the cloud.

Insights for improving operational efficiency are driven from the data generated by services and applications including Manufacturing Execution Systems (MES), SCADA/DCS and Programmable Logic Controllers (PLC) in levels 3, 2 and 1, which is what this document focuses on. Processing this data efficiently is best accomplished by leveraging the availability of on-demand compute resources, unlimited cost-efficient storage, and analytics and Artificial Intelligence/Machine Learning (AI/ML) services in the AWS Cloud.

Connectivity to AWS and AWS services can be achieved with a variety of AWS services, such as AWS Direct Connect, AWS Virtual Private Network (AWS VPN) and AWS Transit Gateway. Depending on the functionality needed at the OT layer, AWS Direct Connect can often provide a level of performance (low predictable latency, high bandwidth) that cannot be achieved by connecting to the cloud over the internet. We refer to connecting these traditionally on-premises OT workloads to the cloud as *hybrid environments*.

Scenarios

These scenarios define common patterns of how AWS services are (or can be) used in manufacturing. They are listed here to help you better understand the security challenges associated with these common usage patterns. The questions that arise from studying these challenges are then addressed in the **Security best practices** section of this document.

Topics

Gaining insights from manufacturing data

Manufacturers embrace the cloud to deliver digital innovation that scales across the enterprise, and want to leverage the cloud to holistically analyze and extract insights from the manufacturing data. In combination, the AWS Cloud and edge services address these use cases by helping manufacturers ingest, structure, and store data from a variety of current and legacy systems and equipment, and create a combined single source of contextual data set. This data allows for holistic analysis and easy consumption to digitally transform and improve business operations. The following figure shows the typical steps to get insights from factory data.

Data to insights

Extracting, structuring, and ingesting data from OT resources to the cloud is the first step to enabling data analysis. AWS has a variety of analytics services in the cloud for processing,

analyzing, and generating insights, but the ingestion stage requires hybrid components and interaction with OT resources. Following are some of the key AWS services to enable data ingestion from an OT environment (levels 1-3) to the cloud. Refer to this Manufacturing on AWS reference architecture diagram for visual representation.

- **AWS IoT Core** — Ingest data from the IoT device via MQTT.
- **AWS IoT Greengrass** — Ingest data from legacy and IoT devices via MQTT, or various inbuilt / custom connectors and AWS Lambda functions.
- **AWS IoT SiteWise** — Collect, organize, and analyze machine data using OPC UA, EtherNet/IP, Modbus, MQTT, or directly via API calls.
- **Amazon Kinesis** — Ingesting streaming data.
- **Amazon CloudWatch** — Ingest logs and infrastructure metrics.
- **AWS Data Sync** — Ingest and sync on-premises file data to Amazon Simple Storage Service (Amazon S3).
- **Storage Gateway** — Serves as a local file server to ingest data to Amazon S3.
- **AWS Transfer for SFTP** — Server as a cloud FTP server to ingest files to Amazon S3.
- **Database Migration Service** — Migrate or sync on-premises databases to the cloud.

Apart from AWS services, third-party integrations and services are also available for data ingestion, providing customers a wide portfolio of options to bring their manufacturing data to the cloud.

While the specific mechanisms for each service are different, typically a component of these services is deployed at the edge (ISA 95 / Purdue model level 3 or below). These components serve as the intermediary to provide services like protocol conversion, secure cloud connectivity, local data transformation, and caching.

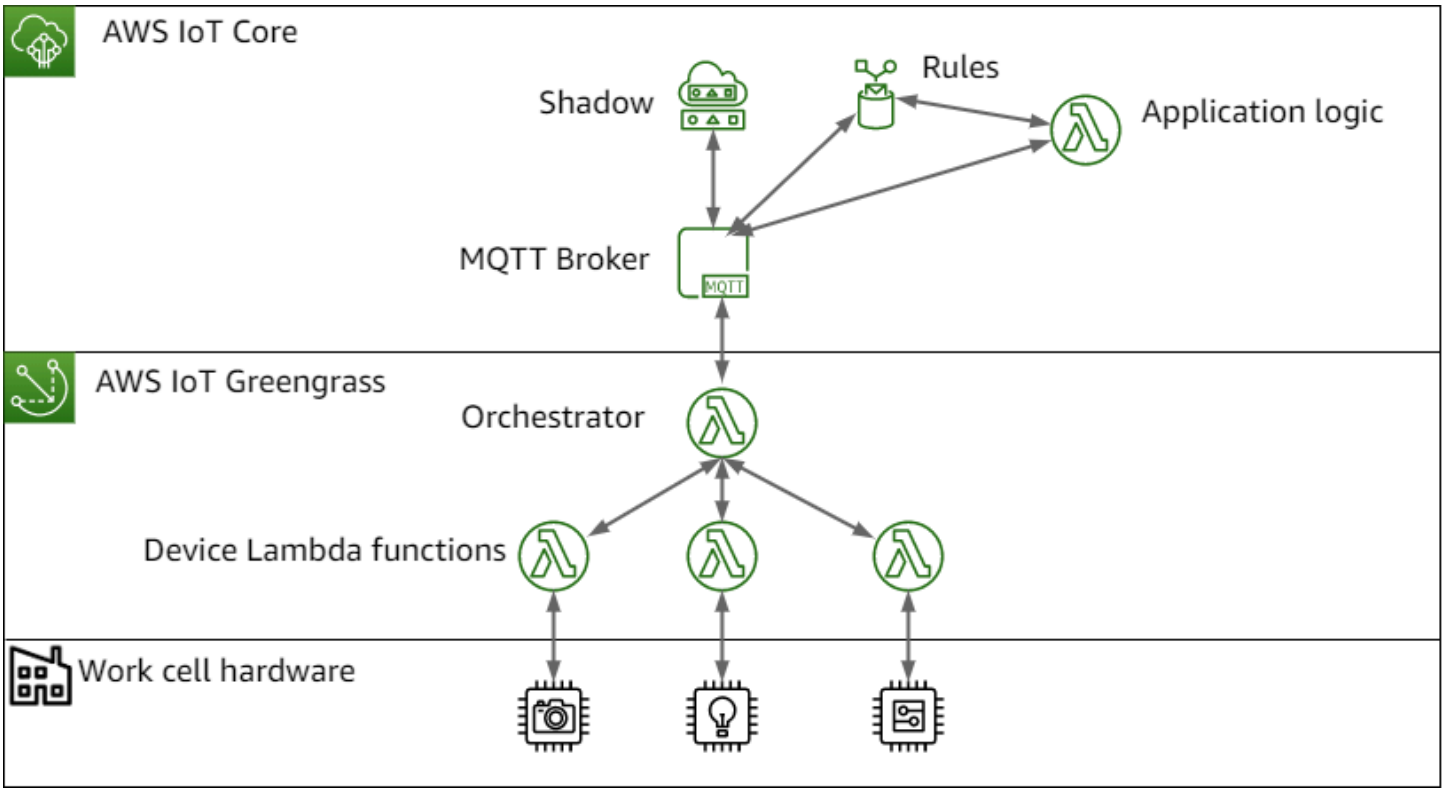
Device control / machine learning inference at edge

Traditionally, the manufacturing industry has relied on PLCs and industrial software like SCADA / DCS / MES running on-premises for device control and process orchestration or automation. The industry is increasingly adopting cloud technologies to augment these local capabilities.

AI/ML at the edge is one such augmentation. AWS provides a set of tools that make AI/ML readily accessible to any organization. Manufacturers can utilize these advanced tools to solve process control challenges. They can train the model in the cloud and deploy it on the edge to leverage ML

for advanced process control. For example, customers can add visual inspection monitored by AI/ML to improve the detection of defects and exceptions.

Process orchestration and control using AWS IoT Greengrass is another way to augment local control capabilities. Lambda functions and microservices running in docker containers can be deployed via AWS IoT Greengrass. AWS IoT Greengrass provides a centralized way to manage and deploy code from the cloud. This allows you the flexibility to manage code at scale, helping to reduce the dependency for on-site expertise and support. Figure 4 represents an example of process orchestration, as demonstrated in the "AWS IoT and Industrial Automation at Amazon" re:Invent session.



Example of process orchestration with AWS IoT Greengrass

FreeRTOS is a real-time operating system (OS) with built-in libraries to establish a secure connection with AWS services and enable over-the-air updates. It is well suited for industrial control tasks, and as an embedded controller in smart industrial sensors, actuators, pumps, and other components.

In this scenario, the cloud-enabled component could exist in Levels 0-3 of the plant networks. With the ability to write back to the controllers and control industrial equipment, this scenario warrants careful security planning and implementation.

Edge computing infrastructure management

A typical manufacturing facility has on-premises computing infrastructure to manage, such as industrial data centers, industrial PCs, and gateways. Managing this infrastructure can be a challenge due to disparate hardware/software, lack of centralized management interface, and no easy way to implement best practices. The responsibility of this infrastructure is shared between OT and IT domains. Customers can leverage the experience of AWS by following the best practices of IT infrastructure management, and by leveraging on-premises management and monitoring services such as AWS Systems Manager and Amazon CloudWatch. These services help manage the on-premises infrastructure at scale, in a similar way as the cloud resources. This removes the barriers to implementing best practices on-premises.

For example, CloudWatch agents can be used to monitor health/usage metrics and logs from edge servers running manufacturing applications. Customers can configure alerts to get notified in case of failures or exceptions. AWS Systems Manager can be used for centralized device management. Customers can collect software inventories, operation system versions, and installed patches. They can automate tasks such as software installation and patch management. This also helps you to maintain your security and compliance requirements, by scanning the instances against specified patch, configuration, and custom policies.

AWS Outposts, on the other hand, provides a fully managed service that extends utility computing to the edge. It is managed from the AWS Management Console, SDK, and API, like any other cloud facility, and is deployed at the customer's premises. It is designed to simplify the management and governance of on-premises infrastructure, and remove barriers to implementing best practices. It utilizes the power of cloud services to augment existing infrastructure, and blurs the boundary between on-premises and cloud.

Integrated manufacturing

Customers experienced with the AWS Cloud for their corporate workloads have expressed that they are eager to leverage a similar experience for all their workloads. AWS for the Edge is a set of services and technologies that have been designed to spread utility computing outside cloud data centers. Utilizing these technologies enables customers to have the same consistent experience across all manufacturing and IT workloads.

AWS for the Edge consists of following software components:

- **FreeRTOS** — An operating system for microcontrollers that enables you to build small, low-power edge devices that connect to AWS IoT.
- **AWS IoT SiteWise** — Easily collect, organize and analyze data from industrial equipment at scale.
- **AWS IoT Greengrass** — Extends AWS to edge devices so they can act locally on the data they generate, while still using the cloud for management, analytics, and storage.
- **Alexa Voice Service (AVS) Integration** — A feature of AWS IoT Core that enables device makers to make any connected device an Alexa built-in device.
- **Amazon Kinesis Video Streams** — Capture, process, and store media streams for playback, analytics, and machine learning.
- **Amazon SageMaker AI Neo** — Train machine learning models once and run them anywhere in the cloud and at the edge.
- **AWS RoboMaker** — Simulate and deploy robotic applications at cloud scale.

AWS for the Edge also offers following options for hardware extensions of the cloud:

- **AWS Snowcone** — Small, portable and rugged, edge computing and transfer device.
- **AWS Snowball Edge** — Rugged, shippable edge computing platform with Amazon EC2 and storage onboard.
- **AWS Outposts** — Run AWS infrastructure and services on premises for a truly consistent hybrid experience.
- **AWS Wavelength** — AWS Wavelength is an AWS infrastructure offering optimized for mobile edge computing applications.
- **AWS Storage Gateway** — AWS Storage Gateway is a hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage.

Cloud computing becomes the preferred platform for the migration and the modernization of Level 4-5 manufacturing applications such as Production Planning, Enterprise Resource Planning (ERP), Product Lifecycle Management (PLM), High-Performance Computing (HPC), Computer-Aided Design (CAD), and industrial data lakes. Edge computing extends modernization to MES and SCADA to Industrial Internet of Things (IIoT) and to the management of proliferating industrial things and industrial computers (IPC).

By connecting their industrial facilities to the rest of the corporation, enterprise manufacturers can get better insight into their operations at global scale, and provide continuous guidance to each

leader and manager accordingly. The bidirectional flow of information generated and consumed by the shop floor enables new levels of collective efficiency that we call integrated manufacturing.

Security principles

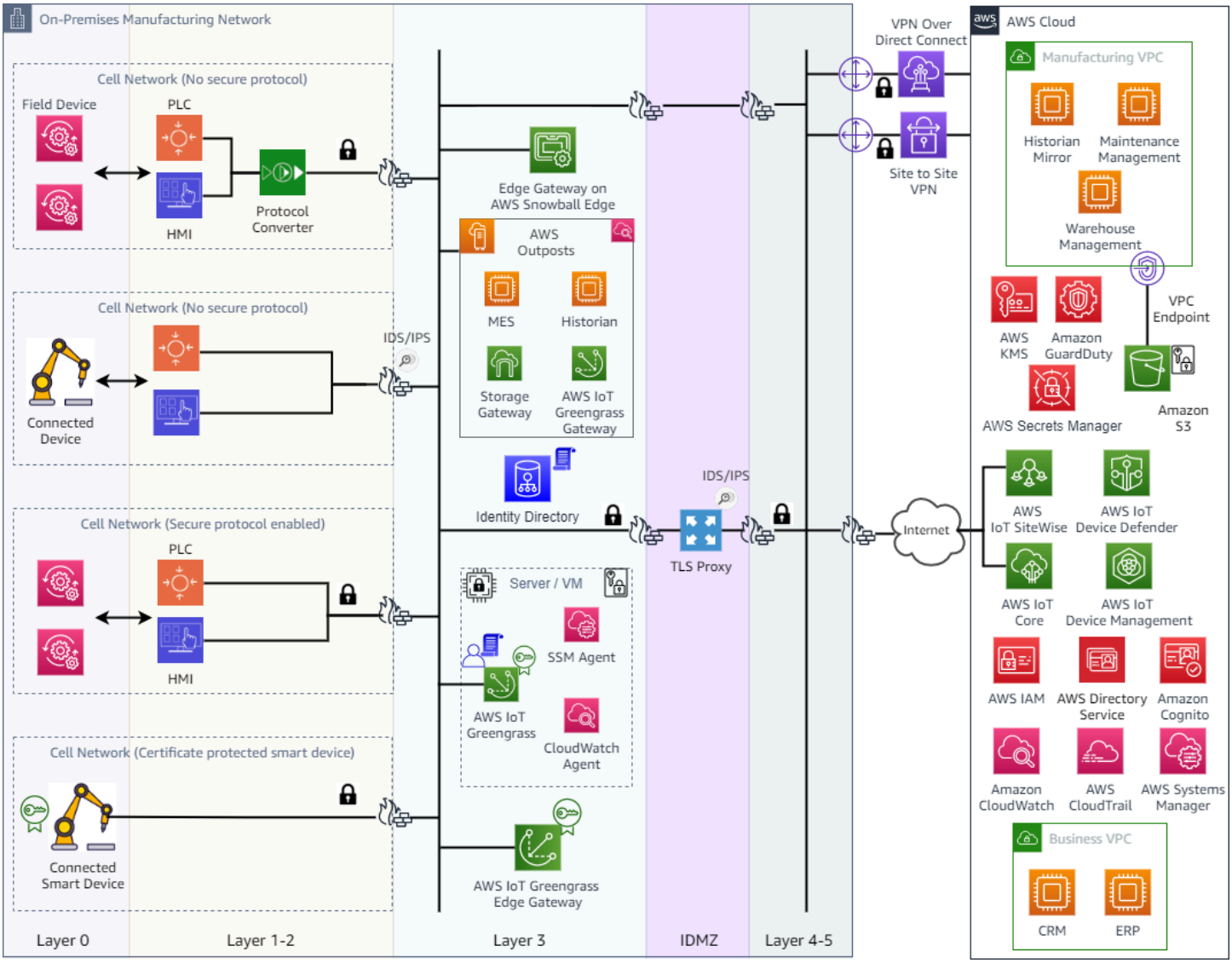
The following key security principles for on-premises OT security are adapted from the Security Pillar design principles of AWS Well Architected Framework, NIST guidelines on ICS cybersecurity, NIST guidelines on zero trust architecture and IEC 62443 standard series. They are adapted and augmented to suit the challenges of the hybrid manufacturing environment. They provide a set of core fundamental guidelines to apply when thinking about the security of the hybrid manufacturing environment.

- **Secure all communications** — Network location alone doesn't imply trust. Historically, OT environments have been air-gapped systems, with perimeter security as the primary defense mechanism for these networks. As such, the resources within the network perimeter are considered "trusted" and don't use any security mechanism. This principle states that all communication, whether it's inside the network perimeter or outside, should be done in the most secure manner possible, providing source authentication and protecting confidentiality and integrity. Application of Zero Trust principles, including existing methods, such as network segmentation and segregation (like cell / zone / area segmentation) can shrink these traditional trust boundaries and reduce the reliance on network location.
- **Enable traceability** — Traceability is key in maintaining and operating secure industrial networks. An enterprise should monitor, alert, and audit actions and changes to the environment in real time. It should collect data about asset inventories (hardware and software), network traffic, access requests, and associated logs and metrics. These data collection systems should be integrated with systems to automatically investigate and take actions. The data should also be analyzed to get insights to improve policy creation and enforcement.
- **Protect data in transit and at rest** — Data should be secured by classifying it into sensitivity levels and using mechanisms, such as encryption, tokenization, and access controls where appropriate. While data classification is not as commonplace in the manufacturing industry (as compared to financial or healthcare industry), the key takeaway is that extra scrutiny may be necessary for certain types of data. Data loss prevention (including backup, redundancy, disaster recovery) is also a part of protecting and securing data.
- **Apply security at all layers** — Apply a defense in-depth approach with multiple security controls. Apply security at all layers (for example, VPC in the AWS Cloud, edge network, OT network, compute instances, operating systems, application, and code).

Security best practices

The following best practices provide guidelines to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies. Manufacturing institutions are expected to maintain a strong cybersecurity posture. The security best practices address the challenges of securing the hybrid manufacturing environment by taking a prescriptive approach, and recommending solutions to each challenge area posed by the usage scenarios.

The following figure shows the reference diagram for manufacturing OT security best practices. This diagram is used as a visual aid in subsequent sections of this document to highlight and describe best practices.



Manufacturing OT security best practices reference diagram

Topics

Secure network connection to the cloud

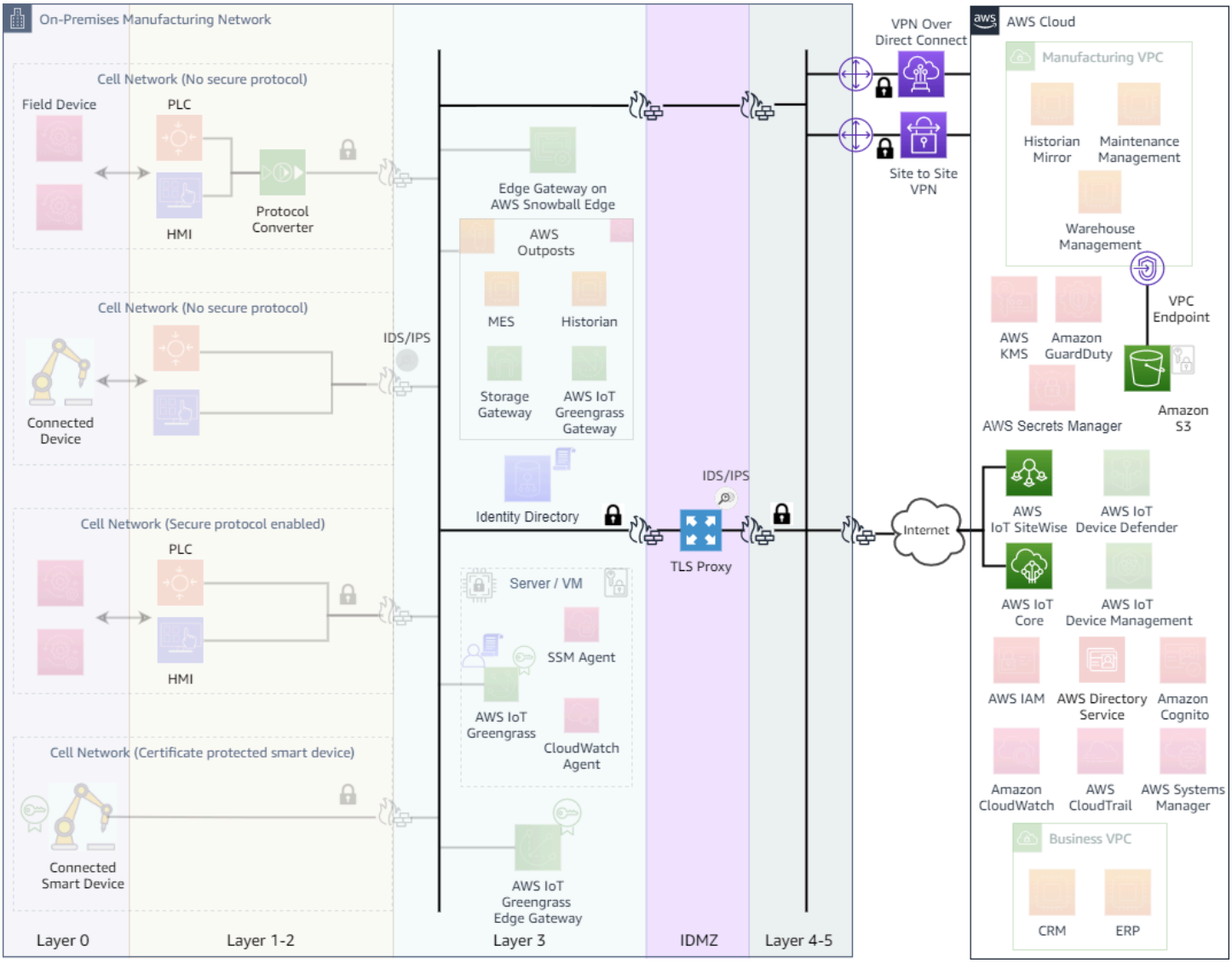
The best practice to manage a secure cloud connection is to keep the network traffic private and encrypted. If the network traffic can't be routed through either a VPN or a private network and one needs to access a cloud service directly over the internet, the traffic must be encrypted and routed through a TLS proxy and an on-premises firewall for added protection. Figure 6 highlights some of these best practices.

- **Establish secure connection with AWS via site-to-site VPN or Direct Connect** — AWS offers multiple ways and design patterns to establish a secure connection to the AWS environment from the manufacturing edge. Establish a secure VPN connection to AWS over the public internet, or set up a dedicated private connection via Direct Connect. Use AWS VPN with Direct Connect to encrypt traffic over Direct Connect.
- **Prefer VPC endpoints or VPC Endpoint Services when possible** — Once a secure connection to AWS has been established via VPN over public internet or Direct Connect, use VPC Endpoints whenever possible. VPC Endpoints enables customers to privately connect to supported regional services without requiring a public IP address. Endpoints also support endpoint policies, which further allow to control and limit access to only the required resources.

VPC Endpoint Services (AWS PrivateLink) enables you to create your own application in your VPC in the cloud and configure it as a VPC Endpoint.

- **Use TLS proxy and a firewall for services connecting to AWS over public internet** — If the VPC Endpoint for the required service is not available, you would have to establish a secure connection over the public internet. The best practice in such scenarios is to route these connections via a TLS proxy and a firewall.

The following figure shows an example of an IoT AWS IoT Greengrass gateway connected to the cloud via a proxy. Using a proxy allows you to inspect and monitor cloud traffic, enabling threat and malware detection. It also allows the security policies to be applied at the network layer. Firewall rules need to be established for HTTPS and MQTT traffic. To sustain the intermittent loss of network connection, the gateway should utilize "store and forward" methods like AWS IoT Greengrass Stream Manager to locally buffer data until the connection is restored.



Secure network connection to the cloud

Secure network connection to local resources

Manufacturing applications running in the AWS Cloud or applications running on an on-premises edge gateway with a connection to the cloud need to access local network resources like PLCs

and field devices. These network resources could also include local computers (HMI / SCADA), file systems, or databases. Manufacturing environments often operate under the assumption of implicit trust of the local network resources. Although an edge gateway or agent software could be part of the local network, it should establish connections with other resources in a secure fashion, assuming they are untrusted. Following are some of these best practices.

- **Use Secure Industrial Protocols** — Historically, Industrial Control Systems (ICSs) have been air-gapped systems (isolated environments), running proprietary control protocols. These ICS protocols have served the challenging needs of the manufacturing industry for decades; however, these protocols were designed assuming all the communications are happening in a trusted environment and hence relied mostly on perimeter security. As a result, ICS protocols didn't typically support the security requirements for encryption, authentication and authorization.

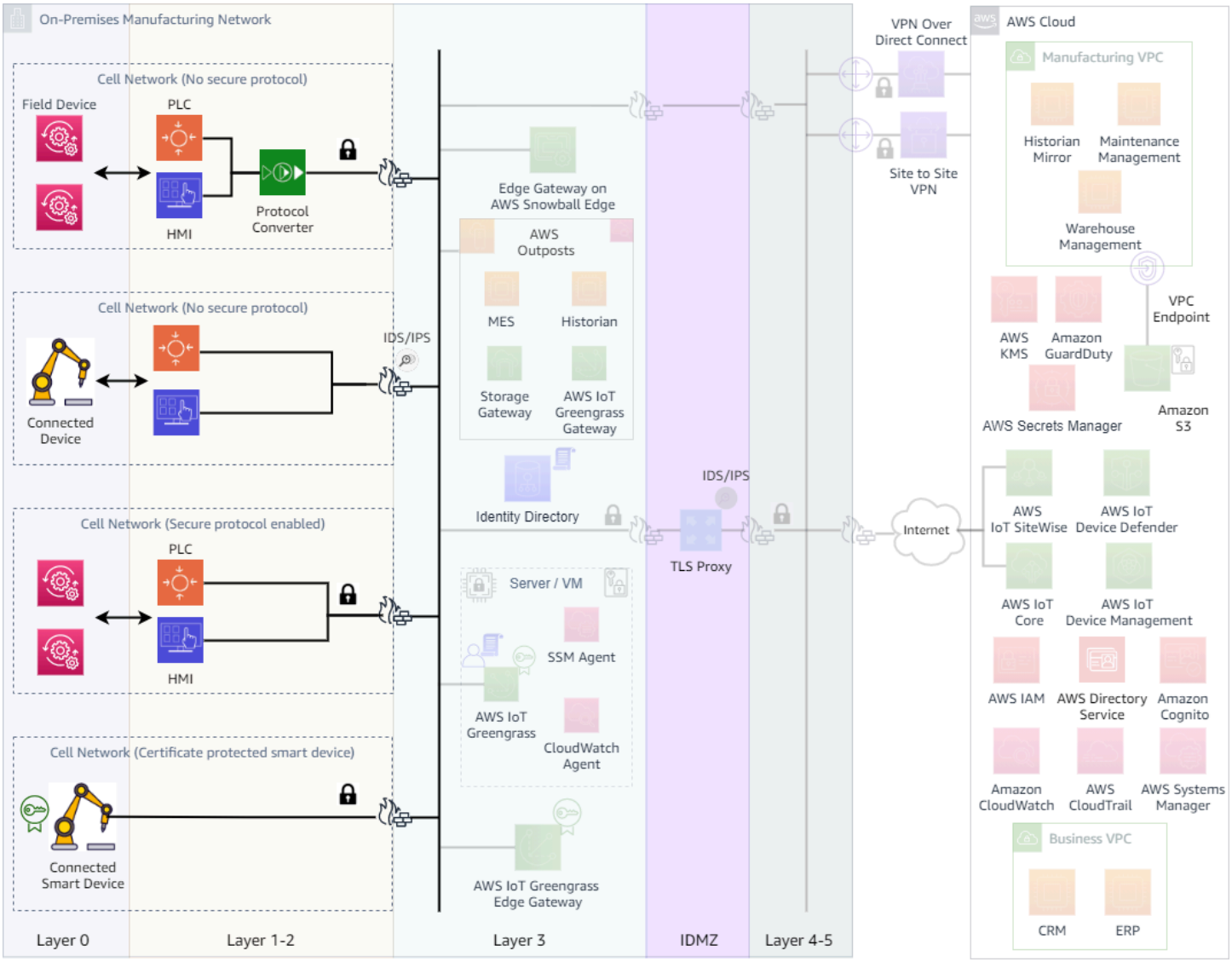
  But amidst the heightened awareness to industrial cybersecurity and the evolution towards smart factory and cloud connected systems, newer versions of some ICS protocols have been developed to support secure communications. Following are some examples of secure versions of existing industrial protocols.

- **CIP Security™** — This is a new method of securing the Common Industrial Protocol (CIP) data at the protocol level. CIP is an industrial protocol supported by hundreds of vendors. CIP Security™ adds specifications for authentication, message integrity verification, and encryption to the CIP protocol, making it secure.
- **Modbus Secure** — This new protocol provides robust protection through the blending of Transport Layer Security (TLS) with the traditional Modbus protocol, a popular industrial protocol. The new protocol leverages X.509 v3 digital certificates for authentication of the server and client. The protocol also supports the transmission of role-based access control information using an X.509 v3 extension to authorize the request of the client.
- **OPC UA** —Open Process Communications (OPC) is an interoperability standard in the industry. OPC UA is the latest iteration of OPC, which is cross platform and secure by design. It offers a combination of an X.509 certificate and user credential-based authentication and authorization schemes. It also offers data encryption in transit. OPC UA specification also allows for server-initiated connections (reverse connect), which allows clients to communicate with servers without opening any inbound firewall ports.

The best practice is to use the secure versions of protocols. If vendor support is not available, consider upgrading or upfitting the existing control system architecture to enable secure protocol support.

- **Tighten trust boundaries** — Secure protocols in the ICS world are fairly new, and vendor support for these protocols varies. If upfitting or upgrading to newer protocols is not an option, consider tightening the trust boundary; for example, limiting the scope and area of unsecure communication. One way to tighten the trust boundary is to place a protocol converter that can translate as well as secure the communications as close to the controller (data source) as possible. Protocol converter PLC modules that reside directly in the control panel, can be an option in this case.

Another recommendation is to functionally segregate the plant into multiple cell/area zones (grouping of ICS devices in a functional area like a machine shop, paint booth, or part assembly). In this scenario, the cell/area zone defines the trust boundary where devices are allowed to communicate unhindered and in real time, but traffic leaving or entering the cell/area zone is subject to inspection, as shown in figure 7. Consider using ICS specialized firewall/inspection products that understand the ICS protocols and can detect anomalous behavior in the control network.



Secure connection to local resources

Secure cloud connected network resources

Cloud connected network resources, such as edge gateways, agent software, and IoT devices, need to be hardened to reduce the risk of inadvertent access. Credentials and permissions to access local resources from cloud connected resources should also be managed to limit the scope of impact of an adverse event. Figure 12 highlights some of these best practices.

- **Harden cloud connected compute resources** — While specific hardening guidelines are dependent on the edge gateway's operating system, general guidelines to harden and securely configure an OS include:

- Remove unnecessary service, applications, and network protocols.
- Configure OS user authentication (remove unneeded accounts, disable non-interactive accounts, configure automatic time synchronization).
- Configure resource controls appropriately (allow access to only needed resources).
- Install and configure additional security control (anti-malware, intrusion detection, host-based firewalls).

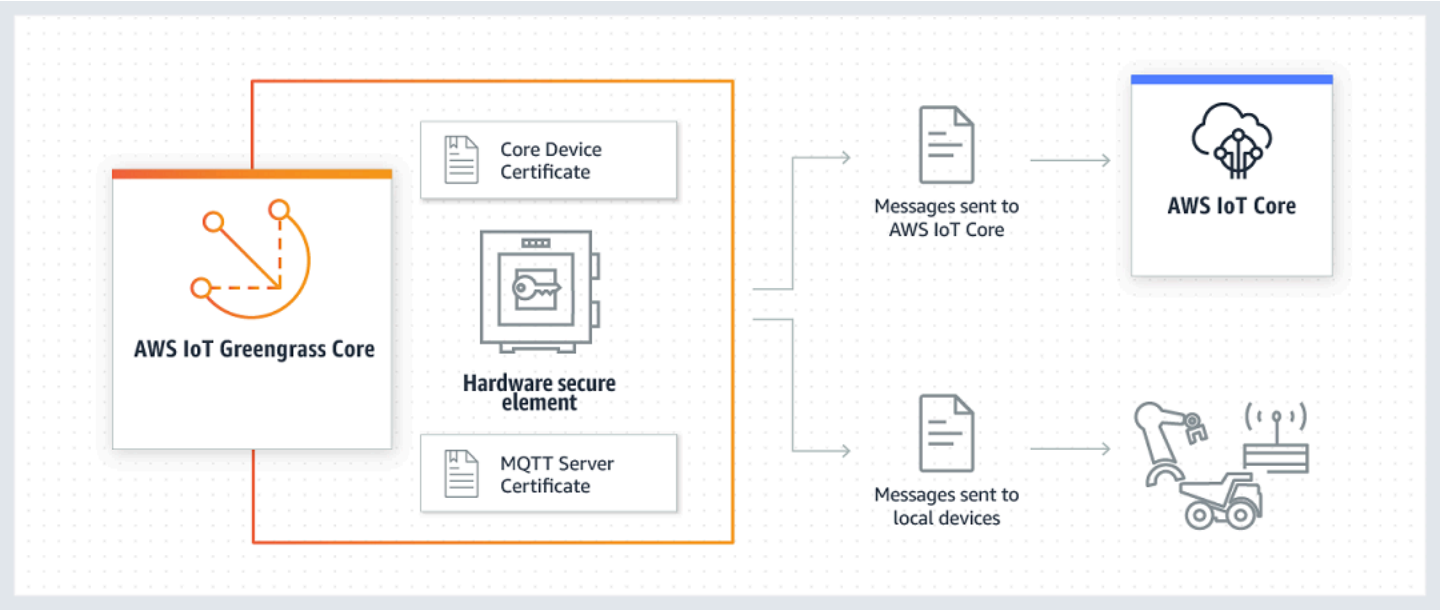
Access to unnecessary hardware ports such as USB and serial should also be disabled using both physical and software means.

When edge gateway is purchased from the vendor, you may not have direct access to the OS; consult the vendor's documentation to ensure the vendor has taken appropriate steps to harden the underlying OS.

- **Use hardware security features like TPM to secure devices** — Leverage hardware security features like Trusted Platform Module (TPM) whenever possible. A TPM is a cryptographic processor present on most commercial PCs and servers. Ubiquitous in nature, it can be used for a wide variety of use cases, such as storing keys for VPN access and encryption keys for hard disks, or preventing dictionary attacks to retrieve private keys.

AWS IoT Greengrass, for example, supports the use of hardware security modules (HSM) for secure storage and offloading of private keys (see Figure 8). Private keys can be securely stored on hardware modules, such as HSMs, Trusted Platform Modules (TPM), or other cryptographic elements. Search for devices that are qualified for this feature in the AWS Partner Device Catalog.

On a standard installation, AWS IoT Greengrass uses two private keys. One key is used by the AWS IoT client (IoT client) component during the Transport Layer Security (TLS) handshake when a AWS IoT Greengrass core connects to AWS IoT Core. (This key is also referred to as the core private key.) The other key is used by the local MQTT server, which enables AWS IoT Greengrass devices to communicate with the AWS IoT Greengrass core. Hardware security can be used for both components using shared or separate private keys. For more information, see Provisioning Practices for AWS IoT Greengrass Hardware Security.



Hardware security architecture for AWS IoT Greengrass

- **Plan and manage security lifecycle of devices** — Planning the device and solution security lifecycle at design time reduces business risk and provides an opportunity to perform upfront infrastructure security analysis.

One way to approach the device security lifecycle is through supply chain analysis. A large number of suppliers can be involved in the supply chain of devices, whether directly or indirectly. To maximize solution lifetime and reliability, ensure that you are receiving authentic components.

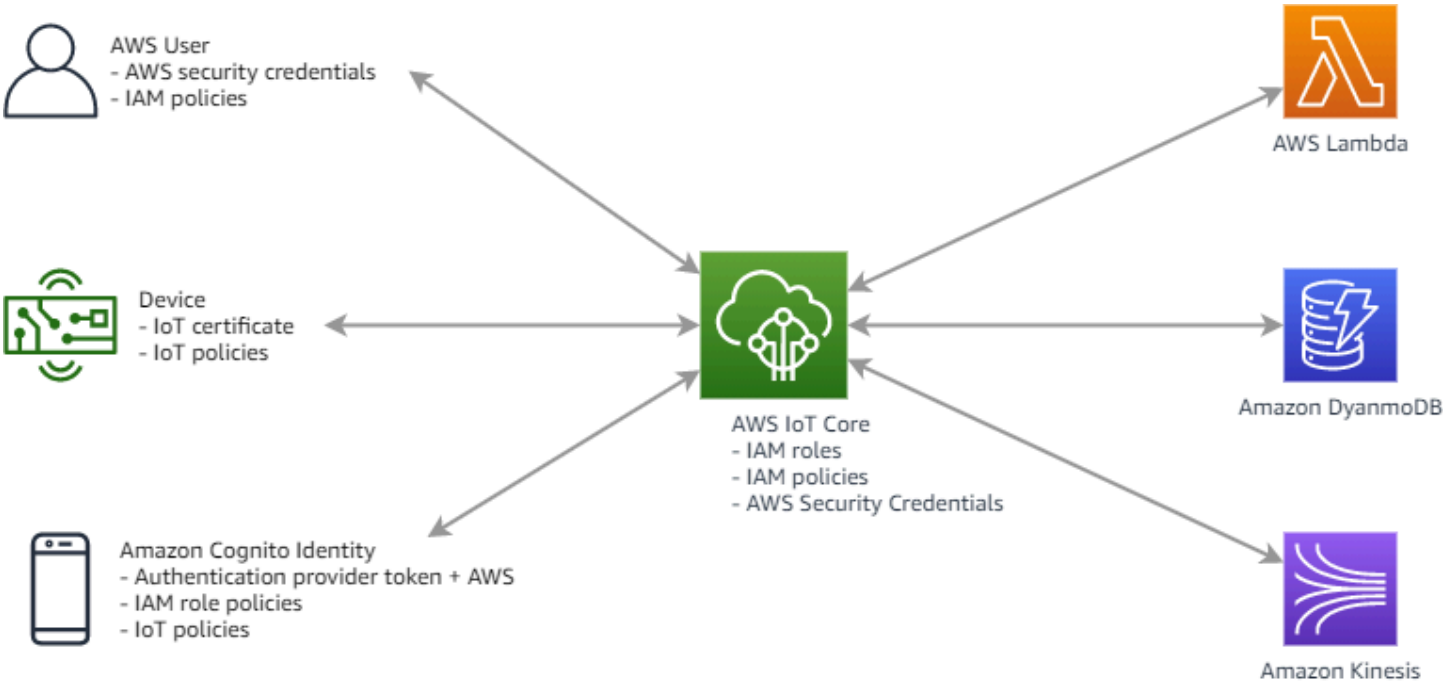
Software is also a part of the supply chain. Analyze each software provider in the supply chain to determine if it offers support and how it delivers patches. Have a plan to validate firmware, patches, or any other software, to ensure their authenticity and validity.

- **Prefer IAM roles / device certificates over IAM credentials** — Edge software requires AWS credentials to access AWS resources. Depending on the service, the device can use IAM roles, X.509 certificates, or even custom authentication methods. Avoid using hard coded long-term credentials and prioritize using IAM roles or X.509 certificates for authentication.

In case of AWS IoT, devices can connect using X.509 certificates or Amazon Cognito identities over a secure TLS connection (see Figure 10). During research and development, and for some applications that make API calls or use WebSockets, authentication can also be done using IAM users and groups or custom authentication tokens. When using custom authentication, a custom

authorizer is responsible for authenticating devices and granting or denying access permissions specified for devices using AWS IoT or IAM policies.

Unique identities should be assigned to each device and permissions should be managed for each device or group of devices. If device certificate or static credentials are used, those credentials should be rotated as appropriate, given the current best practices.



AWS IoT authentication and authorization

- **Implement certificate rotation for AWS IoT, AWS IoT Greengrass core and AWS IoT Greengrass aware devices** — X.509 certificates used by AWS IoT, AWS IoT Greengrass Core, and AWS IoT Greengrass aware devices provide stronger client authentication over other schemes, such as sign-in credentials or bearer tokens, because the private key never leaves the device. The clock on the device is used to verify that a server certificate is still valid and not expired, therefore it is important to maintain accurate time on the device.

Each device or client should be given a unique certificate to enable fine-grained client management actions, including certificate revocation. Devices and clients should also support rotation and replacement of certificates to help ensure smooth operation as certificates expire or if they are inadvertently disclosed.

Rotate certificates under the following circumstances:

- Just before the certificate expires
- Based on violations detected by AWS IoT Device Defender
- In case of inadvertent disclosure

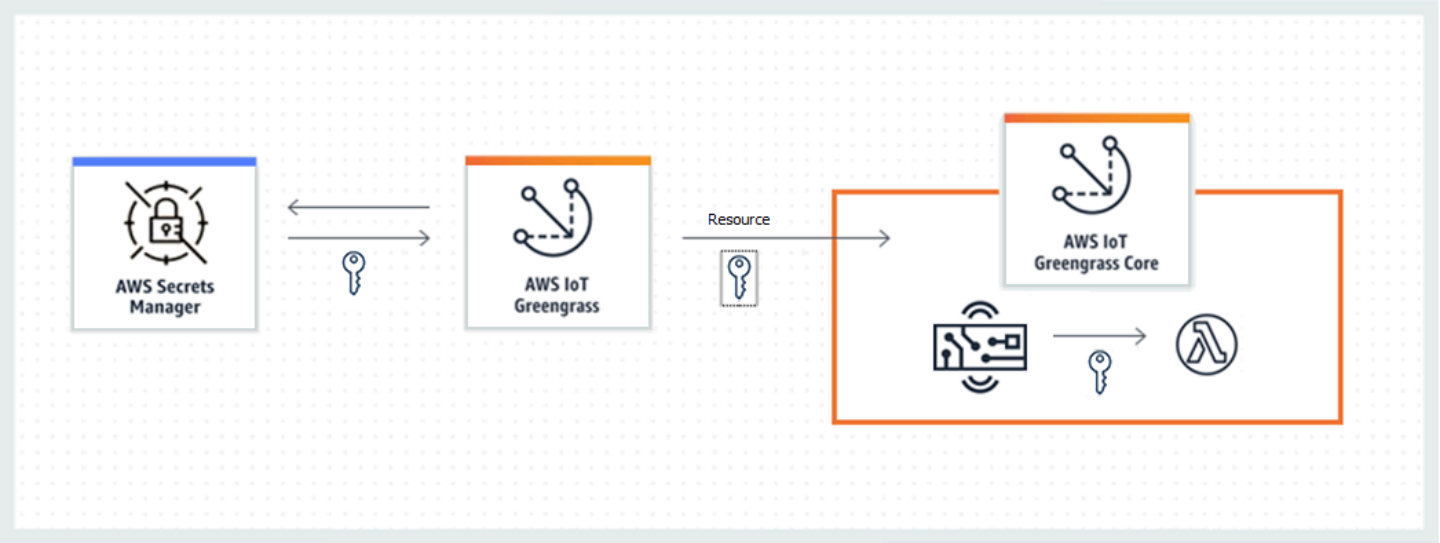
For example, AWS IoT Device Defender audit can be used to identify if a device certificate is expiring soon. This is reported to an SNS topic that triggers a Lambda function that schedules an IoT job to rotate the certificate.

AWS IoT Greengrass-connected devices use the local MQTT server certificate for mutual authentication with the AWS IoT Greengrass core device. By default, this certificate expires in seven days. This limited period is based on security best practices. The MQTT server certificate is signed by the group CA certificate, which is stored in the cloud. The expiration can be set for up to 30 days directly, or even longer duration by contacting AWS Support. More frequent rotation requires more frequent cloud connection. Less frequent rotation can pose security concerns. Certificates can also be rotated manually. Customers should create and follow a certificate rotation policy suiting to their needs.

- **Avoid hardcoding or storing local credentials** — The edge software may need credentials to access local resources like databases, OPC UA servers. Instead of storing and managing credentials locally, store credentials within a secure vault store such as AWS Secrets Manager. AWS Secret Manager is service to securely store and manage secrets in the cloud. Code can retrieve the secrets via an API call. AWS Secrets Manager can also be set to automatically update and rotate credentials. The secrets are encrypted with the AWS Key Management Service (KMS) key of choice, and administrators can explicitly grant access to these secrets with granular IAM policies for individual roles or users.

AWS IoT Greengrass offers built in integration with Secrets Manager. AWS IoT Greengrass extends Secrets Manager to AWS IoT Greengrass core devices, so connectors and Lambda functions can use local secrets to interact with services and applications. For example, the Twilio Notifications connector uses a locally stored authentication token. To integrate a secret into an AWS IoT Greengrass group, create a group resource that references the Secrets Manager secret. This secret resource references the cloud secret by ARN. To learn how to create, manage, and use secret resources, see Working with secret resources. AWS IoT Greengrass encrypts secrets while in transit and at rest. During group deployment, AWS IoT Greengrass fetches the secret from Secrets Manager and creates a local, encrypted copy on the AWS IoT Greengrass Core. After rotating the cloud secrets in Secrets Manager, redeploy the group to propagate the updated values to the core.

Figure 10 shows the high-level process of deploying a secret to the core. Secrets are encrypted in transit and at rest.

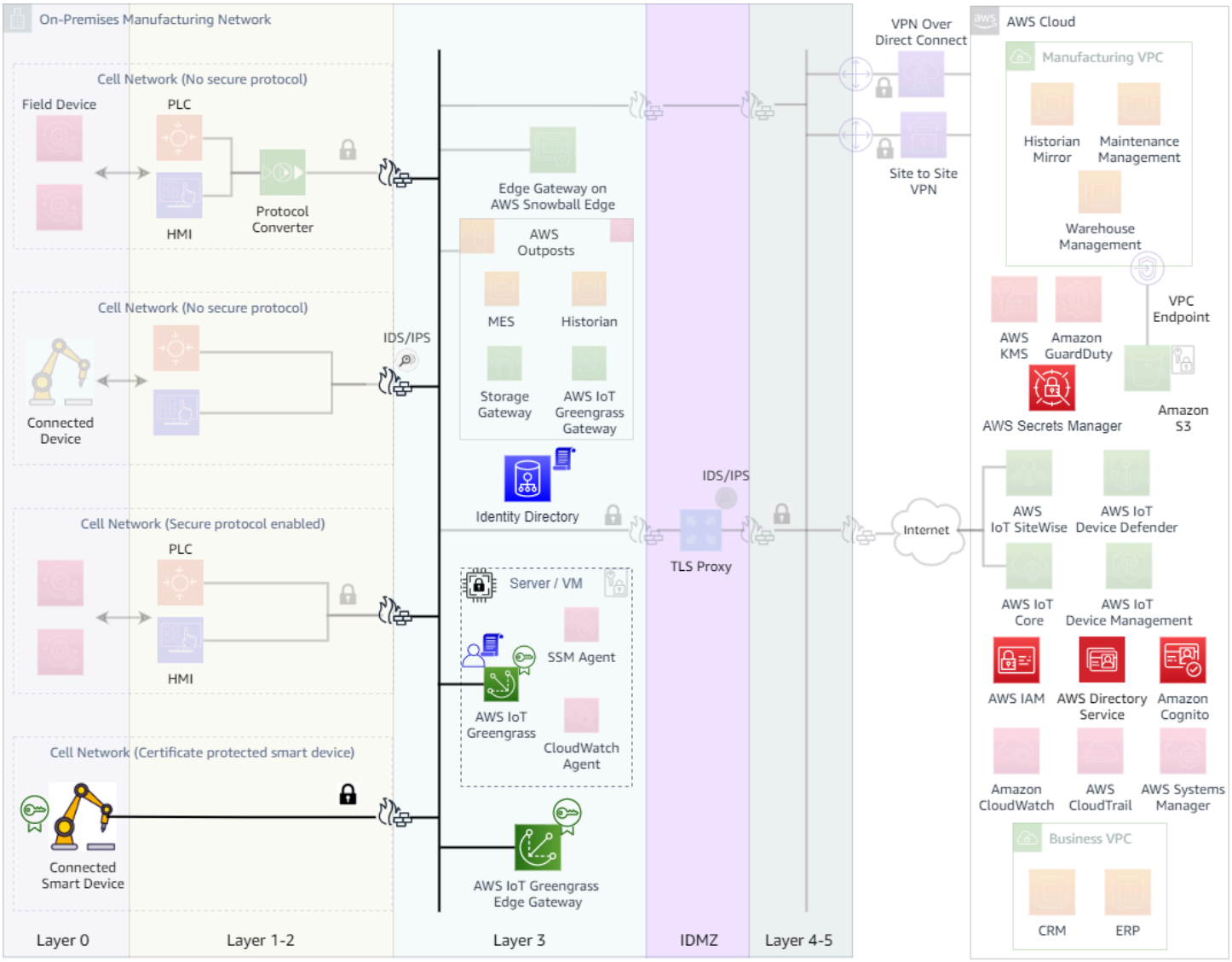


Deploying a secret to AWS IoT Greengrass Core

- **Ensure least privilege access controls for "edge gateways" and "agent software" accessing local and AWS resources** — Edge gateways and agent software should be configured to allow only the required access to both local and AWS resources. For edge gateways, access to local resources should be controlled via firewall on the southbound side to limit access only the needed local resources (for example, access to only required PLCs / OPC servers, IP addresses, and protocols). OS / Active Directory permissions should also be used to prevent access to local network resources like file servers.

Agent software is often installed on the host machine, generating the data to be collected. The agent software access to the host machine resources should be restricted using the operating systems access controls. The agent software daemon (or service) should run under its own account with only the necessary permissions. Just like edge gateway, firewalls and operating system/active directory permissions should to be used to prevent access to local network resources.

Access to AWS resources should be controlled with appropriate IAM policies attached to the edge gateway, or the agent software AWS identity/role. AWS Systems Manager, along with AWS Config, can be used to gain visibility and track changes to OS configurations, system-level updates, installed applications, network configuration, and more.



Secure cloud connected network resources

Securely manage and access computing resources

Keeping computing resources up to-date, securely accessing them for configuration and management, and automatically deploying changes can be challenging. This issue is exacerbated by disparate hardware and software systems used for compute, making it hard to consistently apply best practices. It often leads to more open permissions and more security exposure than needed (for example, a traditional approach managing an edge gateway remotely would typically open RDP or SSH ports and/or a VPN solution, increasing the security risk for the gateway). AWS provides options to securely manage existing compute resources (AWS System Manager), IoT resources (IoT Device Management, AWS IoT Greengrass) and also provides a fully managed

infrastructure service (AWS Outposts) to make it easy to consistently apply best practices to all resources. Figure 13 highlights some of these best practices.

- **Manage and monitor on-premises resources with Systems Manager** — AWS Systems Manager is an AWS service that you can use to view and control your computing resources both on-premises and on AWS. Using the Systems Manager console, you can view operational data from all managed instances and automate operational tasks across your managed resources. Systems Manager helps you maintain security and compliance by scanning your managed instances and reporting on (or taking corrective action on) any policy violations it detects.

You can install AWS Systems Manager Agent (SSM Agent) on on-premises infrastructure and configure it to connect to AWS Systems Manager service in your AWS account. SSM Agent communicate with the AWS services over HTTPS port 443 and don't require any inbound open ports for connectivity.

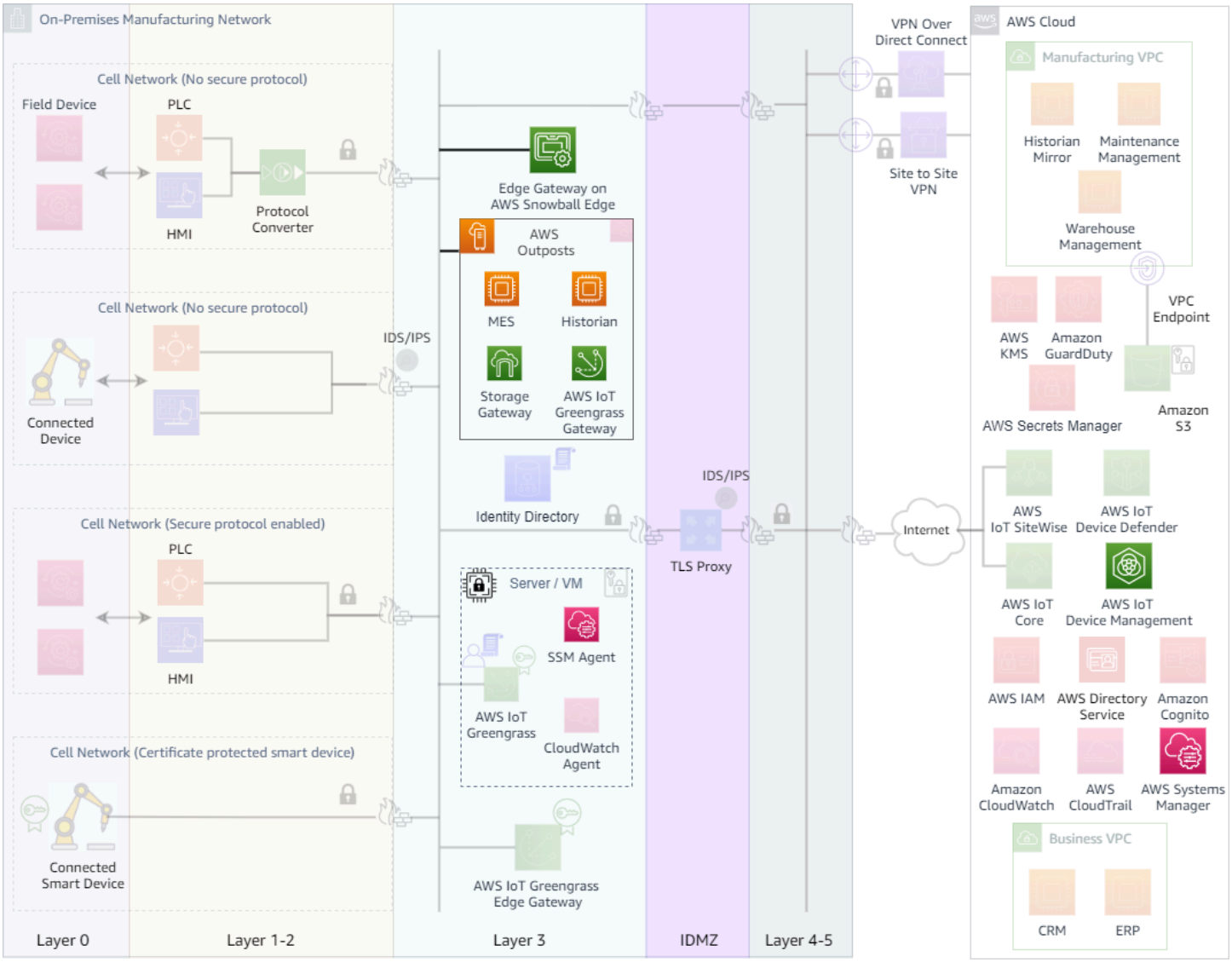
Session Manager is a fully managed AWS Systems Manager capability that lets you manage your EC2 instances, on-premises instances such as edge gateways, and virtual machines (VMs) through an interactive, one-click, browser-based shell, or through the AWS CLI. Session Manager provides secure and auditable instance management without the need to open inbound ports, maintain bastion hosts, or manage SSH keys. Session Manager also makes it easy to comply with corporate policies that require controlled access to instances, strict security practices, and fully auditable logs with instance access details, while still providing end users with simple one-click cross-platform access to your managed instances.

- **Use AWS provided on-premises infrastructure solutions to simplify management and monitoring** — AWS provides solutions for a hybrid cloud environment enabling consistent experiences across AWS and on-premises environments. AWS Outposts is a fully managed hybrid solution that extends the AWS cloud to the on-premises environment, bringing the same AWS infrastructure, services, APIs, management tools, support, and operating model as the AWS Cloud. AWS Outposts can be securely managed from the cloud. It can be used to run a wide variety of traditional on-premises manufacturing applications (SCADA/MES) along with edge applications. It provides a secure and consistent experience of managing and accessing on-premises resources in a similar way to AWS Cloud resources. It also makes it easy to leverage AWS services (such as CloudWatch and Systems Manager) for continuous monitoring and management.

The AWS Snow Family provides highly secure portable devices to collect and process data at the edge. They are designed to operate offline and offer localized management, monitoring and task automation features with AWS OpsHub application. AWS Snow Family also offers security features such as security groups, and local IAM users, roles and policies. It allows customers to implement security via code and also allows them to reason about permissions in a similar manner as they would in a full cloud environment.

- **For IoT devices use secure tunneling for AWS IoT device management** AWS IoT devices can use secure tunneling to establish bidirectional communication to remote devices over a secure connection that is managed by AWS IoT. Secure tunneling does not require updates to your existing inbound firewall rule, so customers can keep the same security level provided by firewall rules at a remote site. The access permissions for the tunnel can be managed in the cloud with IAM permission policies, offering customers a consistent way to manage access.

For example, suppose a sensor device located at a factory a few hundred miles away is having trouble measuring the factory temperature. You can use secure tunneling to open and quickly start a session to that sensor device. After you have identified the problem (for example, a bad configuration file), you can reset the file and restart the sensor device through the same session. Compared to a more traditional troubleshooting (for example, sending a technician to the factory to investigate the sensor device), secure tunneling decreases incident response and recovery time and operational costs.



Securely manage and access computing resources

Continuously monitor network traffic and resources

Security doesn't end with architecting and configuring resources just once. Continuous monitoring to detect changes and malicious behavior is a key to keeping a network secure in the long run. Automation is a key benefit of cloud—the ability to script for thresholding and remediation, so the monitor > detection > action cycle can take place without human intervention. Monitoring should also be expansive, including multiple sources of information such as network traffic, application logs, and operating system logs. (With cloud, you can easily do analytics on your security analytics.) Figure 14 highlights some of these best practices.

- **Maintain a digital asset inventory, monitor and analyze network traffic** — A key component in maintaining a secure ICS network is to be able to identify maintain and control the inventory of both hardware and software assets in the industrial network. After establishing the networked assets inventory, a network interaction baseline mapping all device connections should be created and continuously monitored for any deviations. Local network traffic should be monitored and analyzed using network analysis.

Specialized OT network analysis tools can help create the hardware asset inventory by passively monitoring network traffic. They can also provide deeper insights by analyzing industrial protocols and providing information on specific data and commands exchanged among network devices. Automated rules to send alerts on deviation from baseline should also be configured. Apart from proprietary tools, open-source tools like Zeek can provide such capabilities, to gain a comprehensive view of the network interaction within the plant. AWS Systems Manager can complement these capabilities by providing an automated way to gather software inventory from managed resources.

On the AWS Cloud, turn on Amazon GuardDuty to continuously detect threats, malicious activities and unauthorized behavior. GuardDuty is a "one-switch" shop that uses machine learning, anomaly detection, and integrated threat intelligence to identify and prioritize potential threats. GuardDuty analyzes tens of billions of events across multiple AWS data sources such as AWS CloudTrail, Amazon VPC Flow Logs, and DNS logs. By integrating with AWS CloudWatch Events, GuardDuty alerts are actionable, easy to aggregate across multiple accounts, and straightforward to push into existing event management and workflow systems.

- **Collect local application, operating systems and infrastructure logs and metrics** — Application, operating system, infrastructure logs, and metrics are an important source of information, not only in managing and detecting security threats, but also in troubleshooting and early alerting on application issues. In Industrial Control Systems (ICS), these logs typically stay local and are only analyzed when troubleshooting. AWS services such as CloudWatch and Kinesis can be used to collect logs into a central place. Services like AWS Glue, Amazon EMR, or Amazon OpenSearch Service can be used to analyze the log data at scale and to create automated rules for alerting on any detected malicious behavior. For example, SCADA / MES systems application logs and host server logs can be collected using a CloudWatch agent and sent to CloudWatch and OpenSearch for search and analysis. CloudWatch events and alarms can also be configured to detect anomalous conditions.

Hardware/server performance metrics can provide indicators (like sudden high CPU/network usage) of malicious behaviors and should be continuously collected, monitored and analyzed. Amazon CloudWatch is again a key service to use to collect and monitor performance metrics. A CloudWatch agent can be used with on on-premises servers/virtual machines to collect metrics directly.

Metrics and logs can also be forwarded to the cloud via an edge gateway. The edge gateway can be configured for real-time analysis and detection, providing customers the ability to detect threats on-premises. Third party AWS partner products provide another option for collecting this data in this manner.

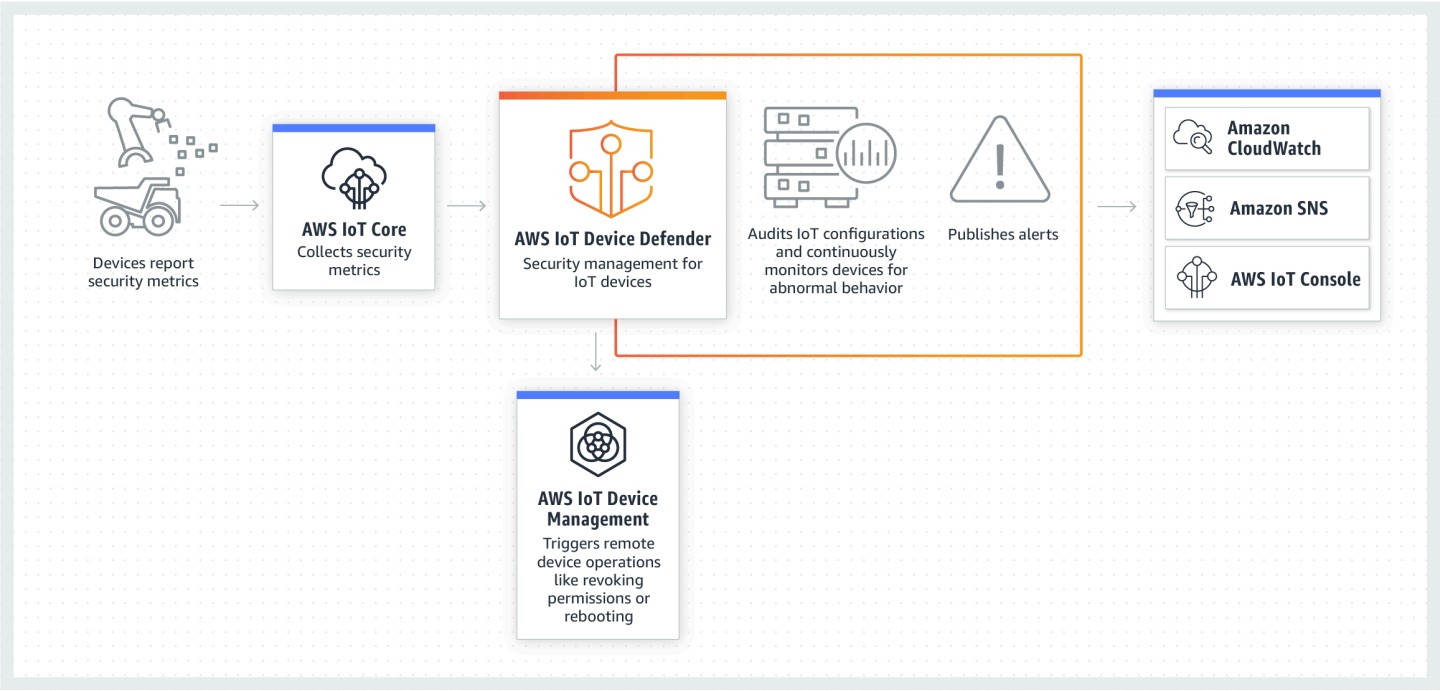
Use of AWS-provided solutions for on-premises infrastructure can further simplify this performance and log data gathering by providing built in mechanisms and deeper integration with cloud services. AWS Outposts, for example, provides built in integration with CloudWatch, CloudTrail, and VPC Flow Logs for monitoring and analysis.

- **Use AWS IoT Device Defender to audit and monitor IoT devices** — AWS IoT Device Defender is a fully managed service that helps you secure your fleet of IoT devices. AWS IoT Device Defender continuously audits IoT configurations to make sure that they aren't deviating from security best practices. A configuration is a set of technical controls you set to help keep information secure when devices are communicating with each other and the cloud. AWS IoT Device Defender makes it easy to maintain and enforce IoT configurations, such as ensuring device identity, authenticating and authorizing devices, and encrypting device data. AWS IoT Device Defender continuously audits the IoT configurations (a full list of audit checks is available in the AWS IoT Defender developer guide) on your devices against a set of predefined security best practices. AWS IoT Device Defender sends an alert if there are any gaps in your IoT configuration that might create a security risk, such as identity certificates being shared across multiple devices, or a device with a revoked identity certificate trying to connect to AWS IoT Core.

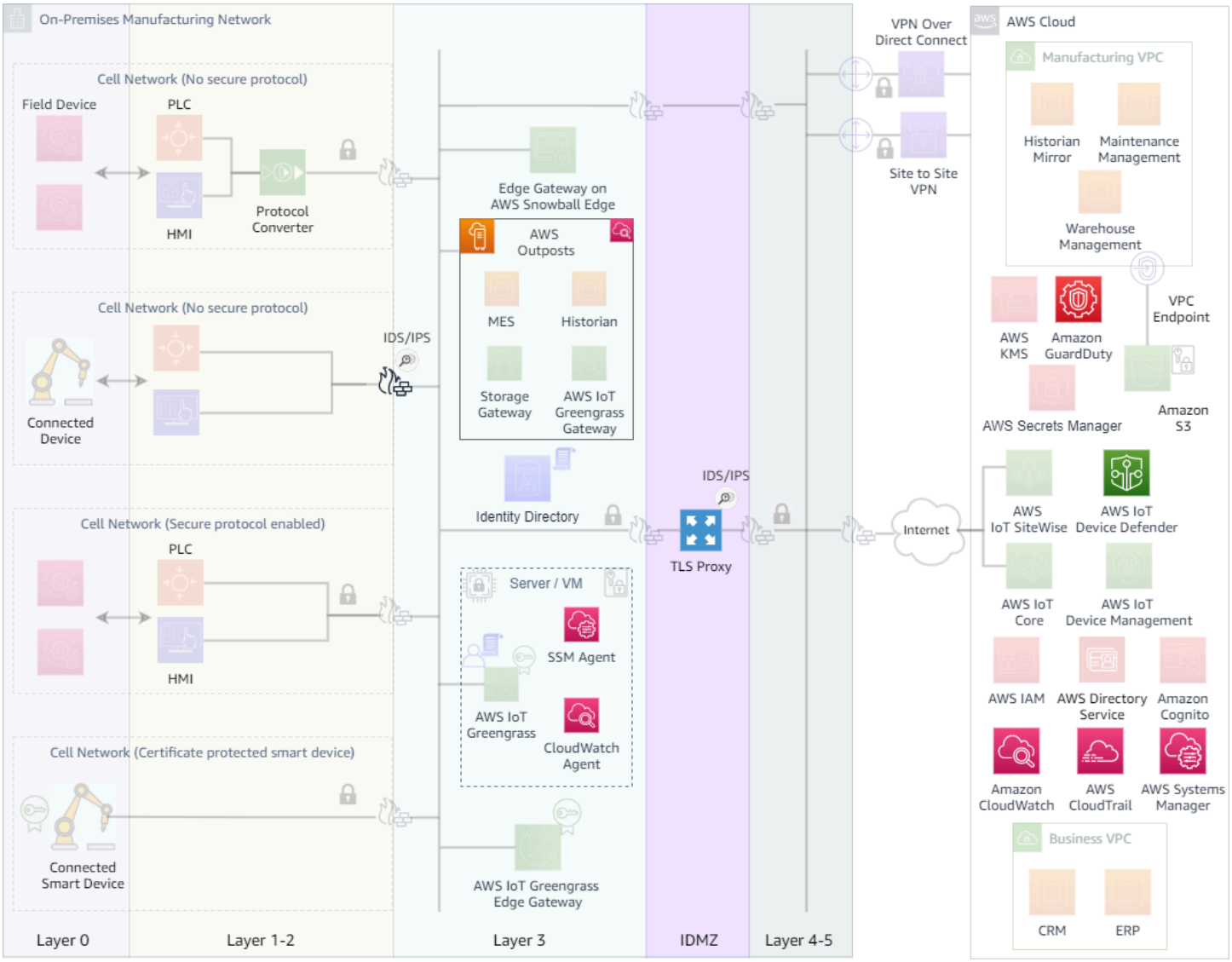
AWS IoT Device Defender can also continuously monitor security metrics from devices and AWS IoT Core for deviations that are defined as appropriate behavior for each device. If a deviation occurs, AWS IoT Device Defender sends out an alert to act to remediate the issue (as shown in Figure 13). For example, traffic spikes in outbound traffic might indicate that a device is participating in a DDoS attack. AWS IoT Greengrass and FreeRTOS automatically integrates with AWS IoT Device Defender to provide security metrics from the devices for evaluation.

AWS IoT Device Defender can send alerts to the AWS IoT Console, Amazon CloudWatch, and Amazon SNS. AWS IoT Device Management can be used to take mitigating actions based on the alert, such as pushing security fixes.

Refer to the "Elevate your IoT security with AWS multi-layered security approach" re:Invent talk for the principles of IoT defense in depth, and a demonstration of AWS IoT Device Defender capabilities.



AWS IoT Device Defender



Continuously monitoring network traffic and resources

Secure manufacturing data

Securing and protecting manufacturing data demands a holistic approach. Some key pillars to protect data within OT and cloud environments are: encryption in transit, encryption at rest, access controls, data classification, and monitoring/auditing data access. The following figure highlights some of these best practices.

- **Encrypt data in transit and at rest** — Examine the data flow patterns to ensure data is encrypted in transit. Follow the recommendations of securing the connections to the cloud and to network resources as discussed in the sections above. Use secure protocols for network

resources and choose strongest possible encryption. Ensure the data is also encrypted at rest; for example, use disk encryption with computing resources like edge gateways.

The encryption techniques for cloud resources may vary based on each AWS service. Consider service documentation to ensure encryption in transit and rest is properly configured for the service. For example, the following figure depicts data stored in S3 bucket is encrypted at rest and is encrypted in transit.

- **Apply access controls using least privilege principle and monitor/audit data access** — Apply data access controls of OT resources on both data sources and data consumers. On data sources, the access controls should be applied at application, OS and Active Directory level to restrict access to known entities and only for the needed resources. Data consumers should also be configured with a unique identity and should only be allowed to communicate and access the intended data sources. If data consumers are AWS services (Kinesis Streams, S3 buckets), IAM access policy and resource policy can be used.

Access controls should also be applied at the connectivity layer using security appliances such as firewalls or data diodes (unidirectional network devices).

Physical security of network resources should be considered when planning access controls for OT. Physical access to computing hardware, network infrastructure and security appliances should be strictly controlled and limited to only a few approved personnel.

Control measures to manage end of lifecycle for compute and storage should also be considered. Ensure the data is sanitized as per NIST media sanitization guidelines at the end of the lifecycle.

Follow the monitoring best practices guidelines discussed in this document, and ensure that data usage and data access monitoring is part of the overall monitoring plan. Configure automated rules to analyze monitoring data in real time, to send alerts or take automated actions.

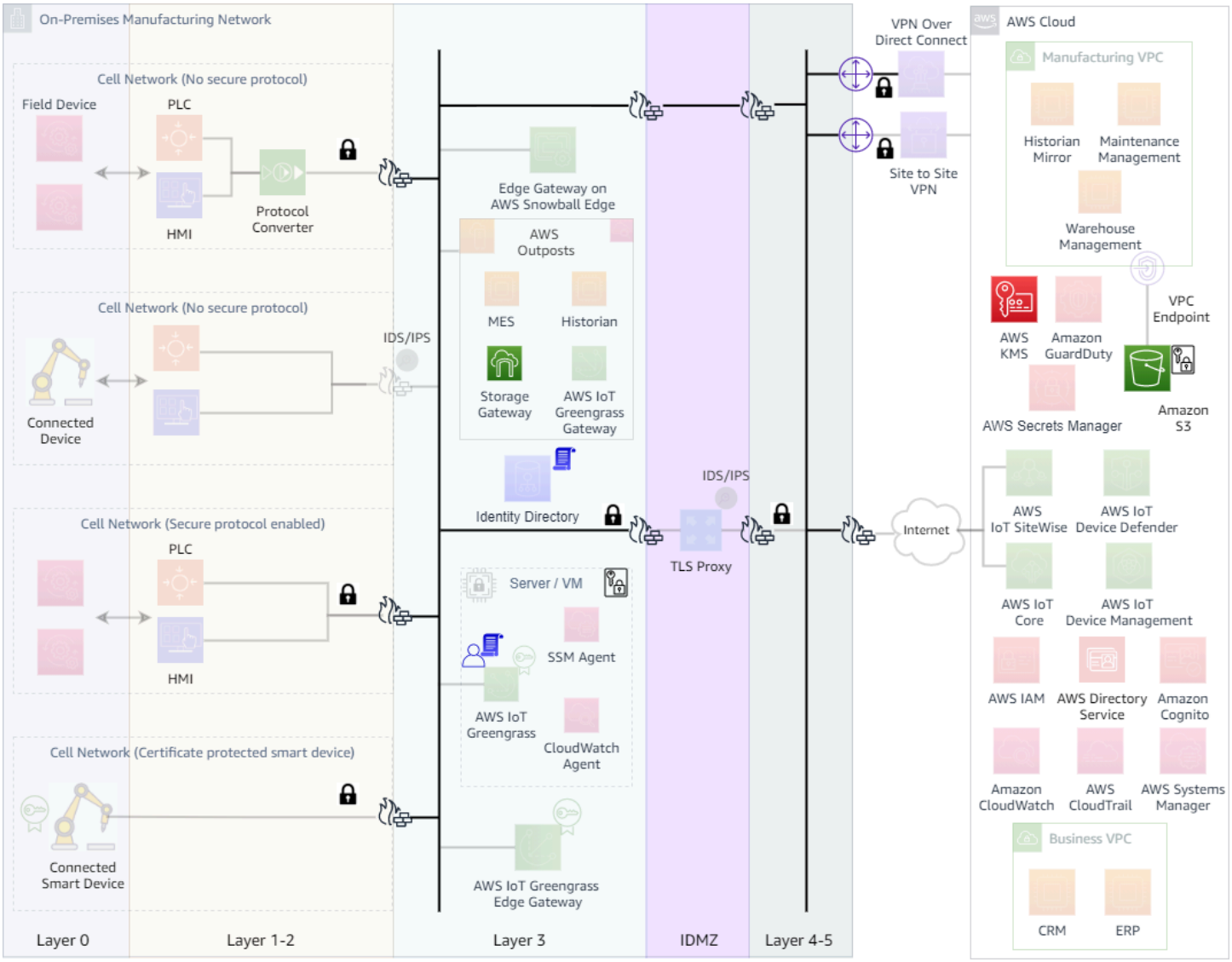
For resources in the cloud, physical security of resources in the cloud is the responsibility of AWS in the shared responsibility model. However, the customer has the responsibility of securing connection to the cloud and securing the workloads running in the cloud. AWS offers various access controls and monitoring/auditing mechanisms to help customers secure their workloads running in the cloud. These mechanisms may vary by each service; use the service documentation to ensure access controls are properly applied for cloud resources.

- **Use resiliency features of the edge and agent software to prevent real time data loss** — Data loss prevention is also an important element to consider in data security. Design the data collection with resiliency in mind. Use features such as "store and forward" in edge and agent software to protect against intermittent data loss. AWS IoT Greengrass, for example, provides stream manager to locally store data streams in case of intermittent connectivity. It also provides other resiliency features such persistent sessions with AWS IoT core and message queue for cloud targets.
- **Use cloud for backup and business continuity** — Ensuring effective data backup and disaster recovery strategies is critical to ensure data and systems can be available even in the case of an adverse event at the local facility.

Cloud is an attractive target for backup and restore use cases as it provides virtually unlimited, durable, and cost-effective storage. Amazon S3 and Amazon S3 Glacier are the primary targets for storing backup data. They provide highly durable, flexible, scalable, and secure service to store backup data. Many third-party services offer built-in cloud connectors to send data to the cloud.

Ensuring business continuity requires a more careful consideration on how quickly and efficiently the systems can be restored in case of system failures, server corruptions, or cyber-attacks. With AWS, CloudEndure Disaster Recovery can be used to minimize downtime and data loss for on-premises physical and virtual servers, and databases. CloudEndure can continuously replicate machines into a low-cost staging area in AWS Cloud. It also provides fail-back capability to on-premise after the failover.

For manufacturing, some low latency workloads may not be able run in cloud even for a shorter duration during disruption. As such disaster recovery plans should include consideration for running on-premises workload on a separate on-premises infrastructure (such as AWS Snowball Edge Edge or AWS Outposts).



Secure manufacturing data

Conclusion and further reading

In summary, this document provided the best practices to architect and manage manufacturing OT environments for on-premises hybrid workloads using the AWS Cloud. We started with understanding the traditional network architecture using the industry standard Purdue model. We then outlined how AWS Edge and AWS Cloud services contribute to modernization of the traditional operation model. We discussed common usage patterns in the scenarios. We highlighted how the IT/OT convergence introduces new security challenges and established fundamental security principles to deal with these new challenges. We then provided prescriptive guidance and best practices on these specific challenges such as securing the network connection to the cloud, and securely accessing, managing and continuously monitoring industrial assets and OT network resources.

Further reading

For additional information, see:

- Hybrid Connectivity Whitepaper
- Security Pillar – AWS Well Architected Framework
- IoT Lens – AWS Well Architected Framework
- NIST Special Publication - Guide to Industrial Control System Security
- NIST Guidelines – Zero Trust Architecture
- AWS User Guide to Support Compliance with North American Electric Reliability Corporation (NERC) Critical Infrastructure Protection (CIP) Standards

Document history and contributors

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

Change	Description	Date
Initial publication	First published.	May 21, 2021

Note

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

Contributors

Contributors to this document include:

- Nishant Saini, Solutions Architect, Amazon Web Services
- Russell de Pina, Solutions Architect, Amazon Web Services
- Ryan Dsouza, Solutions Architect, Amazon Web Services
- Bernard Paques, Solutions Architect, Amazon Web Services
- Steve Blackwell, Tech Leader, Manufacturing, Amazon Web Services

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.