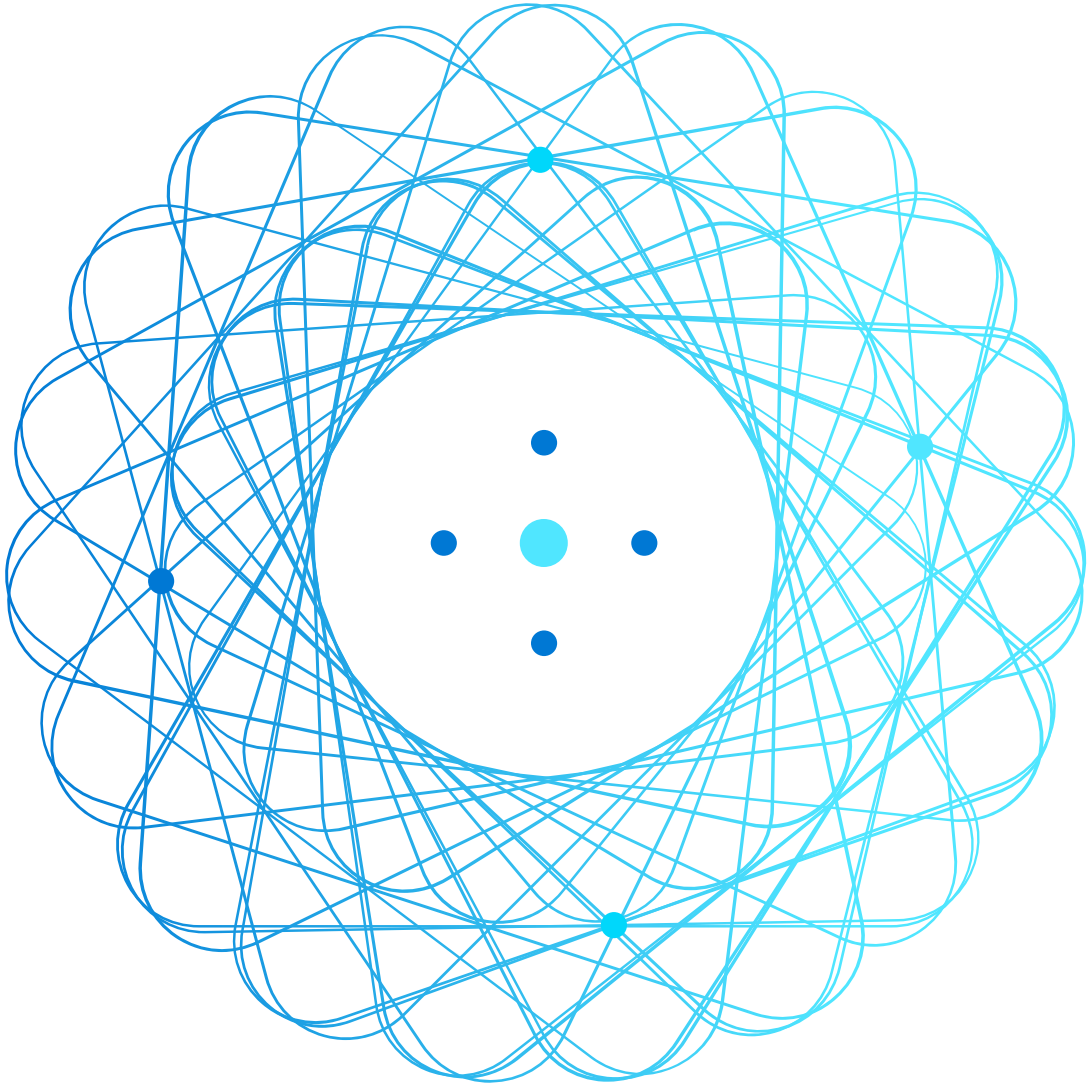
Microsoft Azure

AZ-220T01
Module 06:
Azure IoT Edge Deployment Process

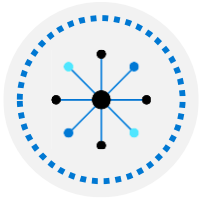


Lesson 1: Learning objectives

Module 6 – Learning objectives

Describe the difference between an IoT device and an IoT Edge device

Configure an IoT Edge device



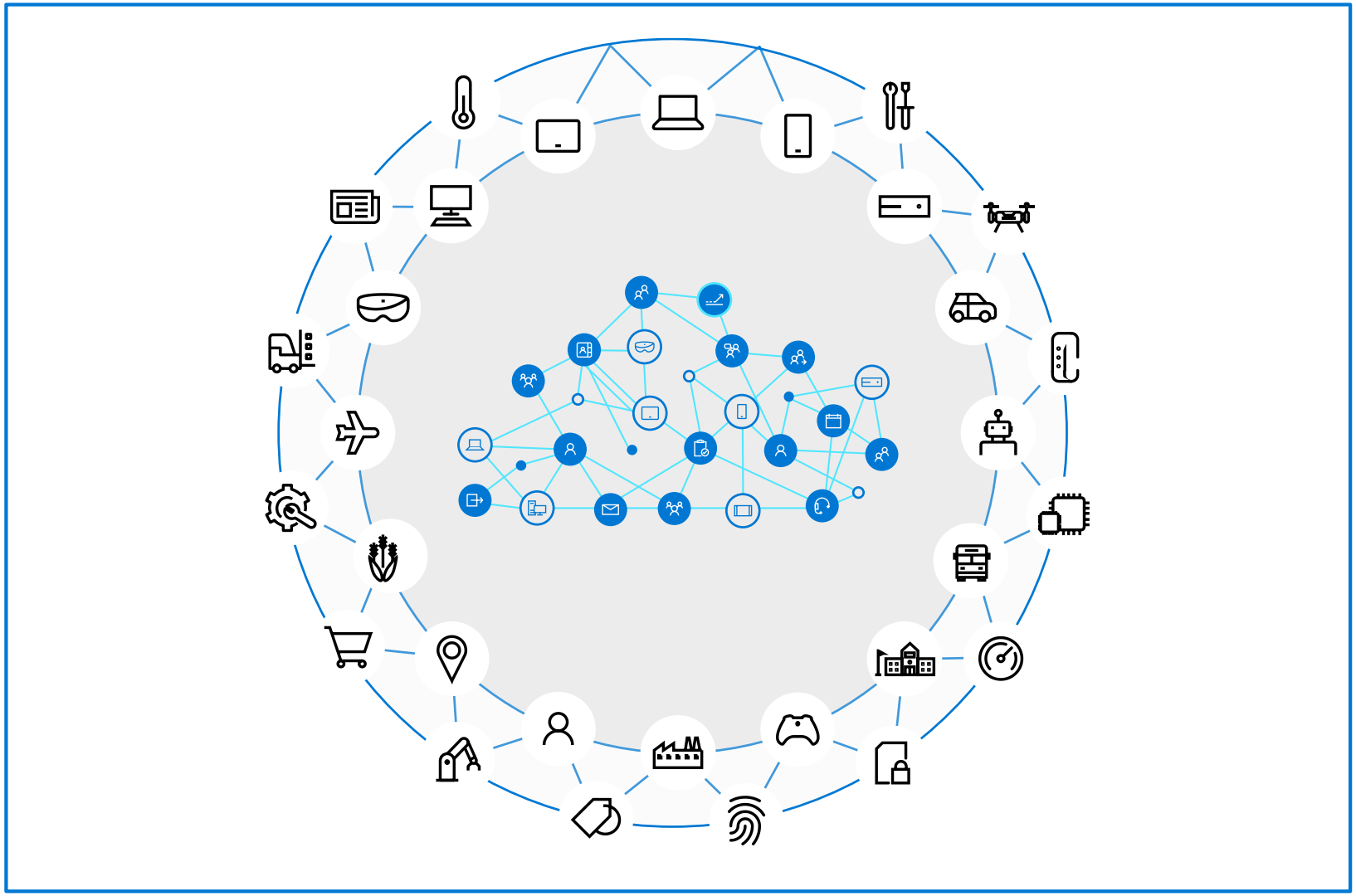
Implement an IoT Edge deployment using a deployment manifest

Configure an IoT Edge device as a gateway device

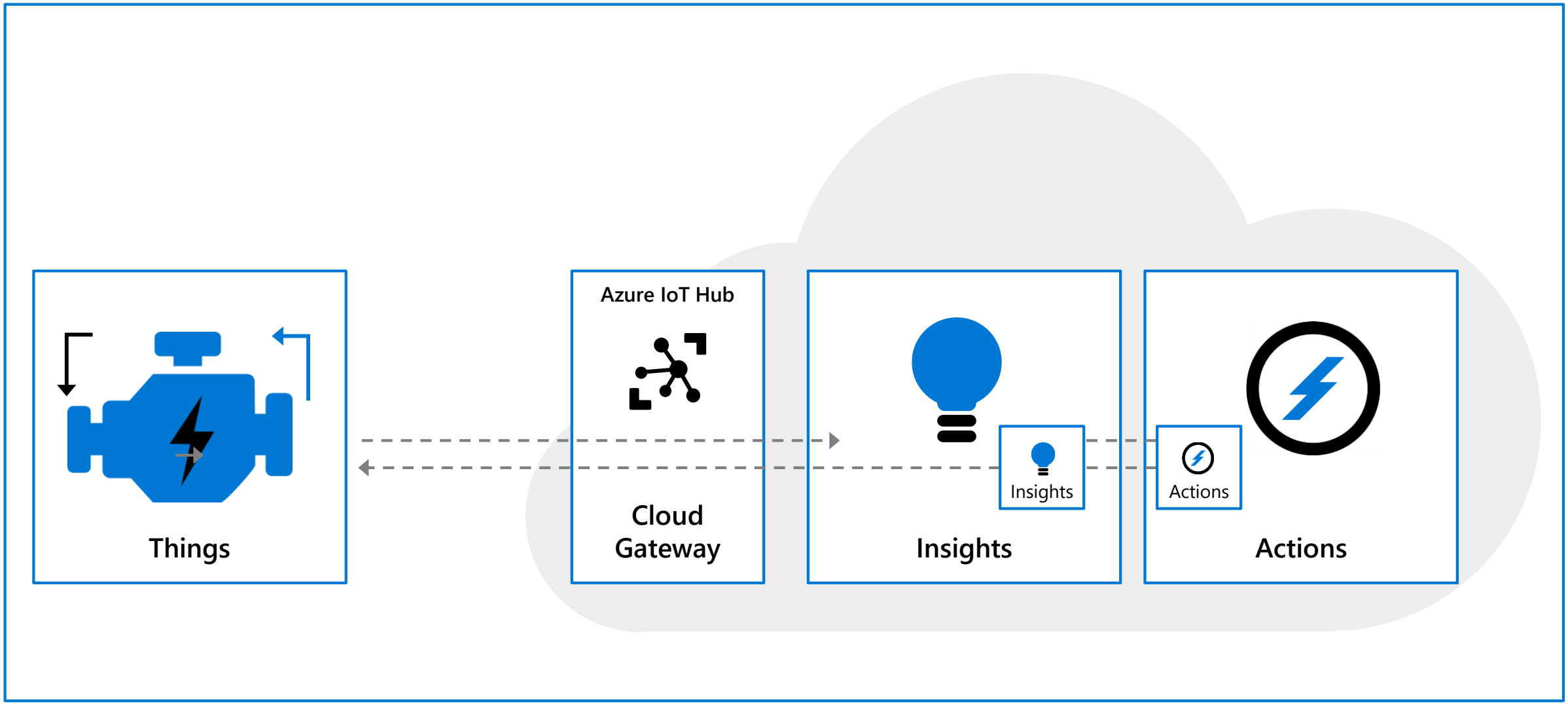
Lesson 2: Introduction to Azure IoT Edge

Cloud and IoT Edge intelligence

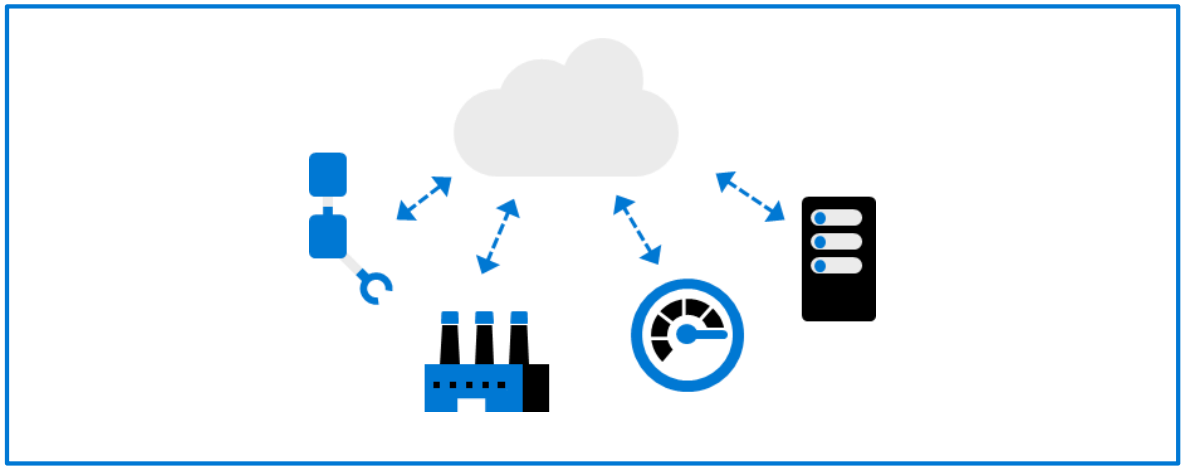
There is a natural balance in IoT between the cloud and the IoT Edge

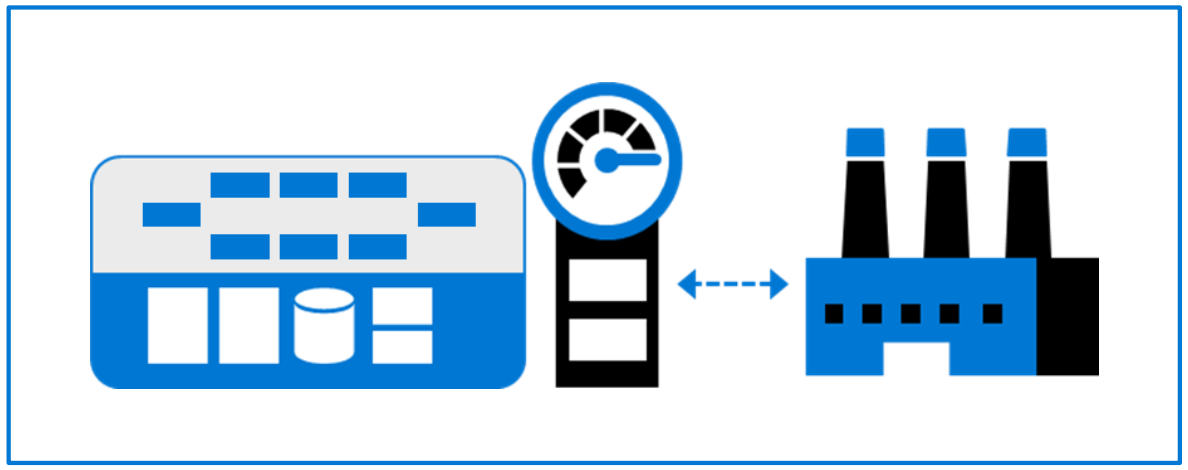


IoT application pattern + IoT Edge



IoT in the Cloud and on the IIoT Edge





IoT in the Cloud:

Remote monitoring and management

Merging remote data from multiple IoT devices

Infinite compute and storage to train machine learning and other advanced AI tools

IoT on the Edge:

Offline operations (short and long term)

Privacy of data and protection of IP

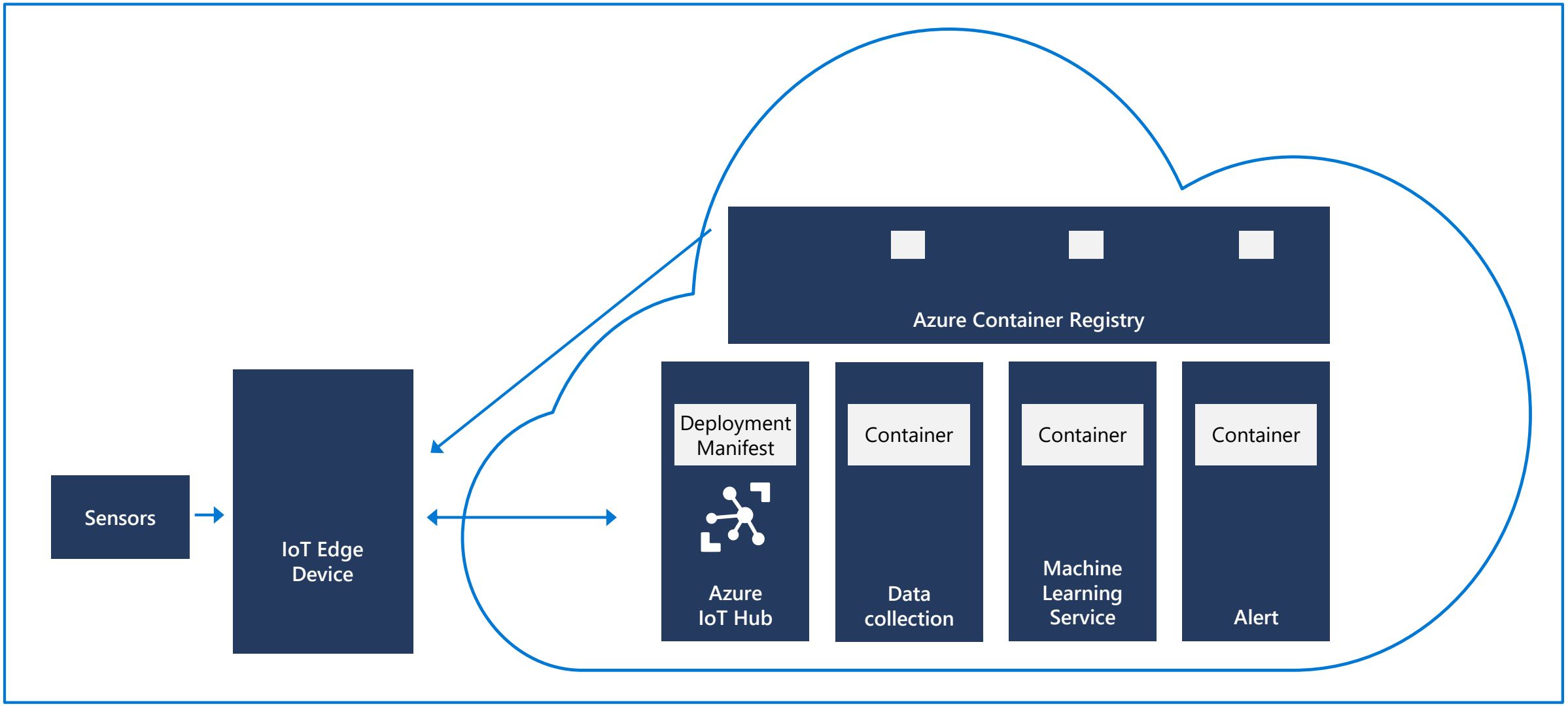
Pre-process data on prem – E.g. video streams

Low latency tight control loops require near real-time response

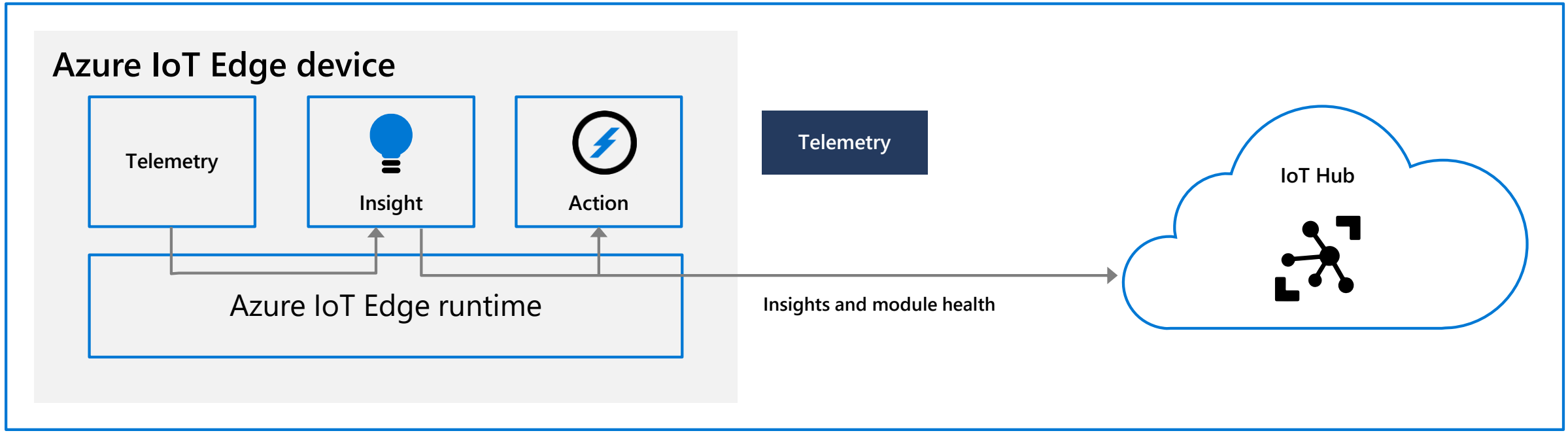
Protocol translation & data normalization

Consistency

Bringing compute to where the data is



IoT Edge runtime



Installs and updates workloads on the device

Maintains Azure IoT Edge security standards on the device

Ensures that IoT Edge modules are always running

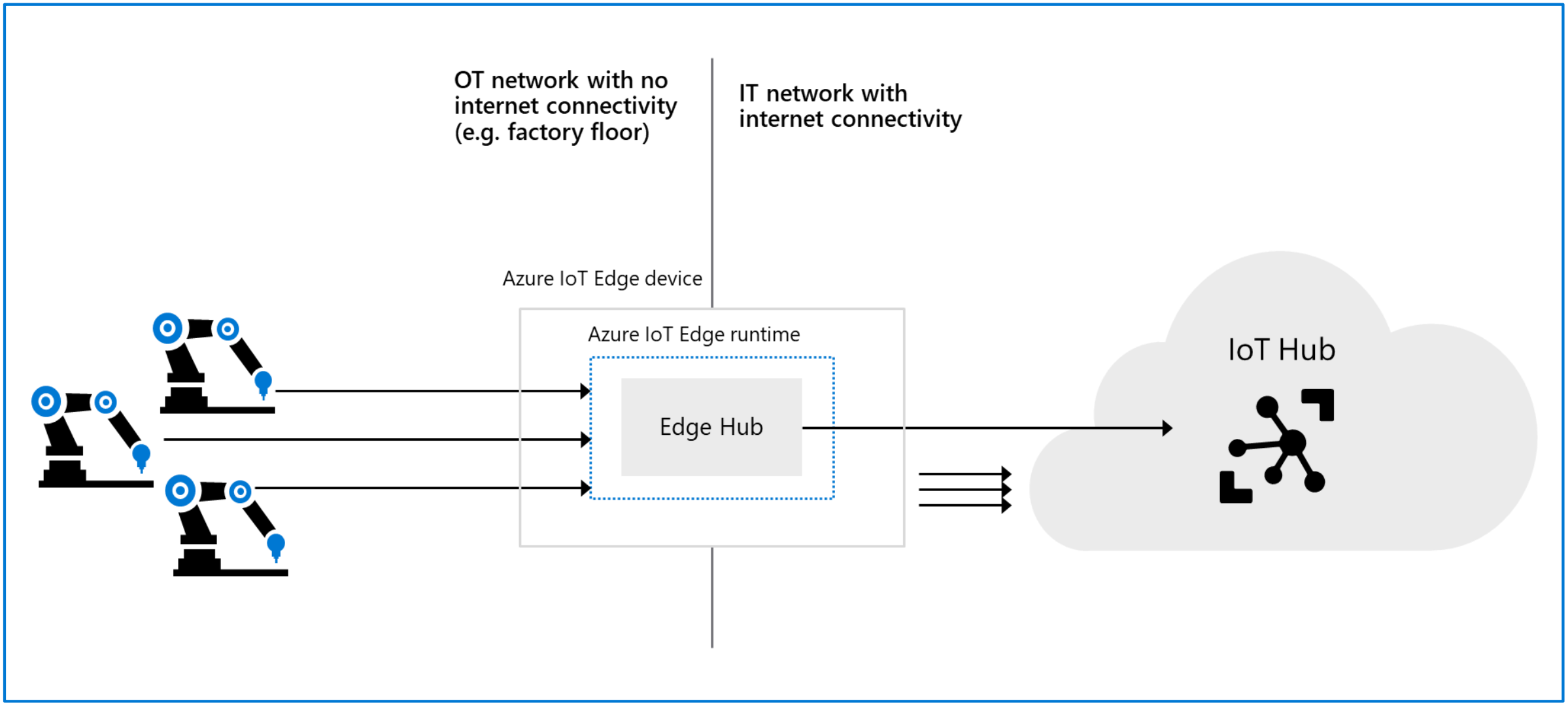
Reports module health to the cloud for remote monitoring

Facilitates communication between downstream leaf devices and the IoT Edge device

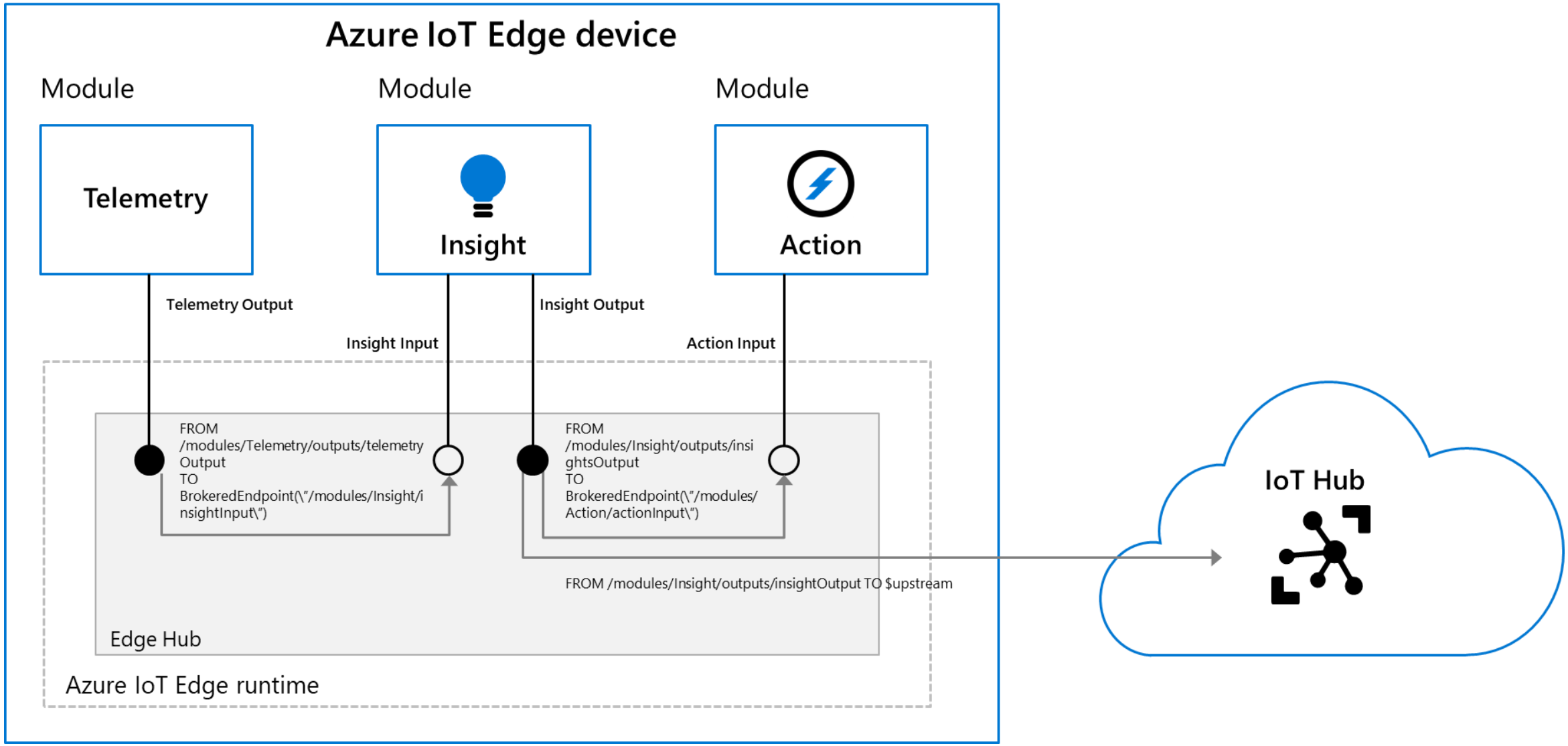
Facilitates communication between modules on the IoT Edge device

Facilitates communication between the IoT Edge device and the cloud

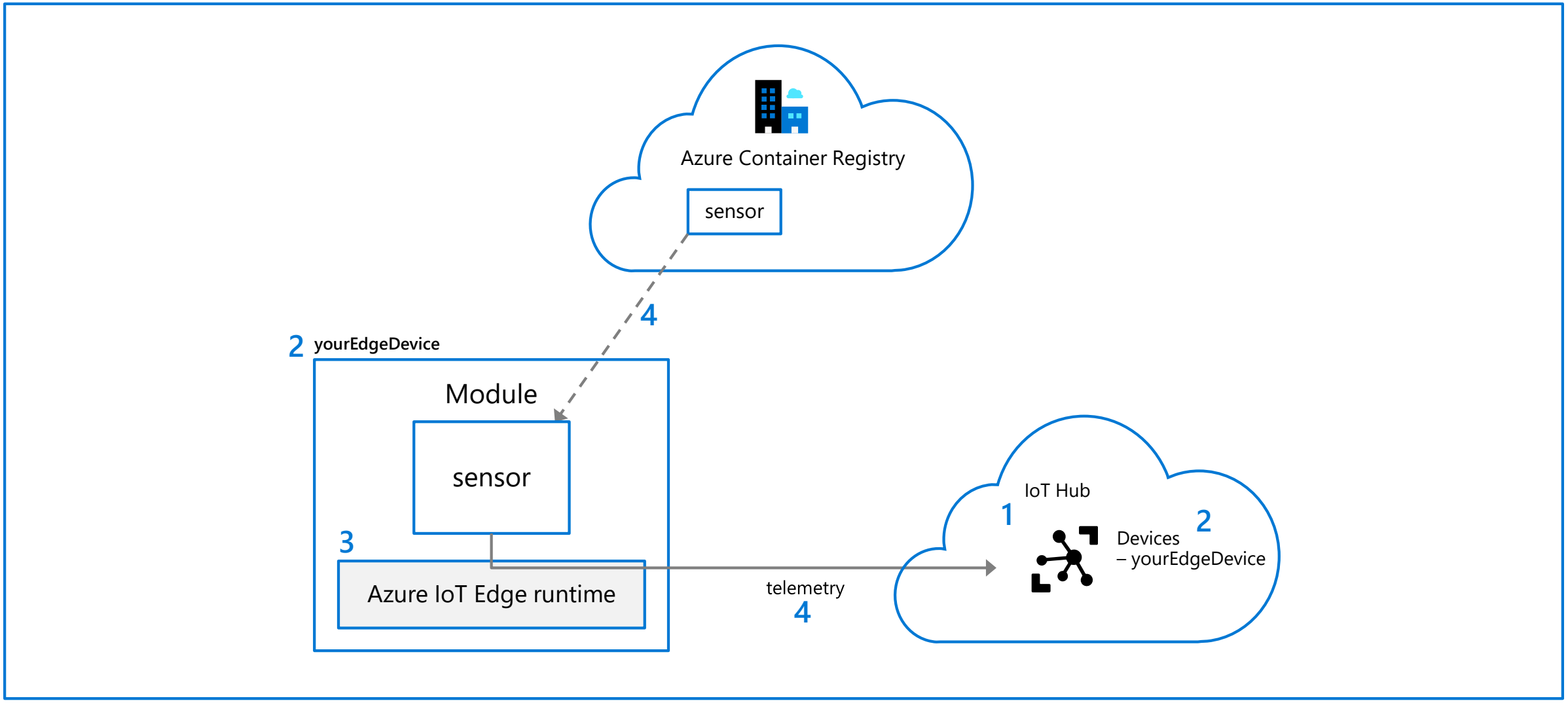
IoT Edge Hub: Offline Support



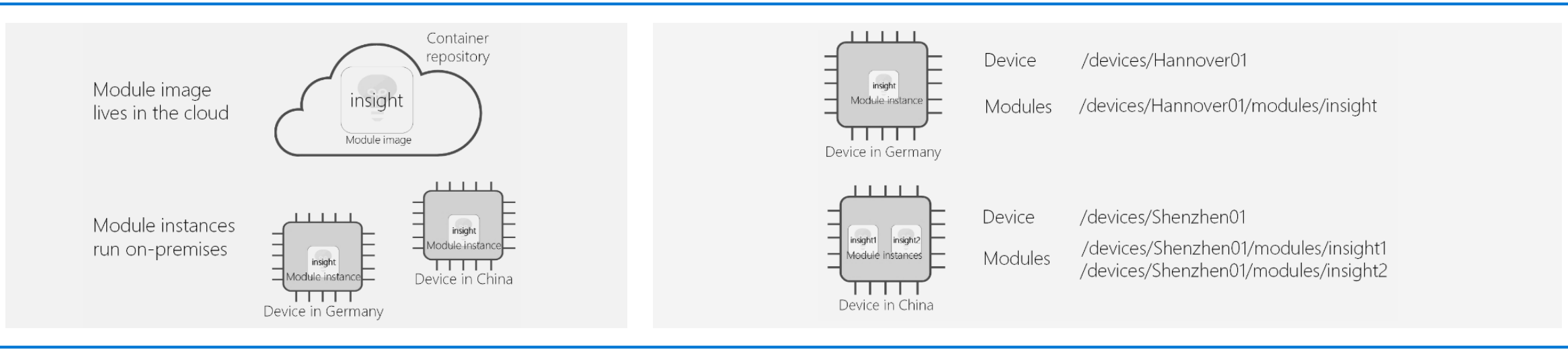
IoT Edge Hub: Module communications



IoT Edge agent



IoT Edge module



A **module image** is a package containing the software that defines a module

A **module instance** is the specific unit of computation running the module image on an IoT Edge device. The module instance is started by the IoT Edge runtime

A **module identity** is a piece of information (including security credentials) stored in IoT Hub, that is associated to each module instance

A **module twin** is a JSON document stored in IoT Hub, that contains state information for a module instance, including metadata, configurations, and conditions

SDKs to develop custom modules in multiple languages (C#, C, Python, Java, Node.JS)

Azure IoT Edge module on Azure marketplace

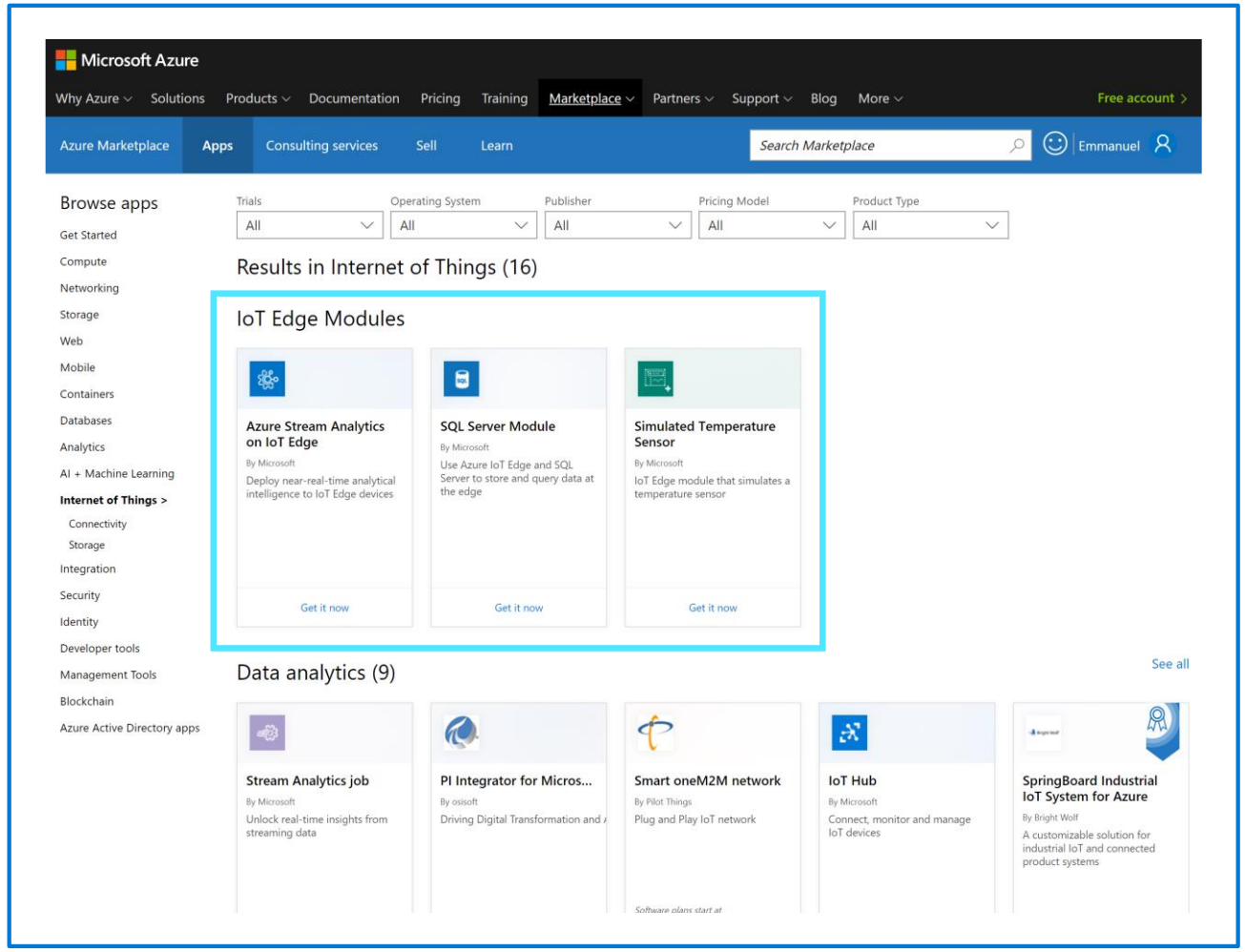
Customers:

Save development effort: Discover and integrate certified pre-built modules

Partners:

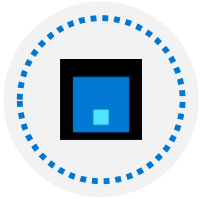
Showcase with wide reach: Certified modules gives IoT customers peace of mind for their project

Work with a leader in IoT: Market with Microsoft, and collaborate with other Microsoft IoT partners



Module twin properties of IoT Edge runtime modules

Desired and reported properties

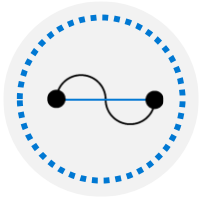


EdgeAgent properties:

Container runtime information

Container registry information, including credentials

List of runtime modules



EdgeHub properties:

Routing

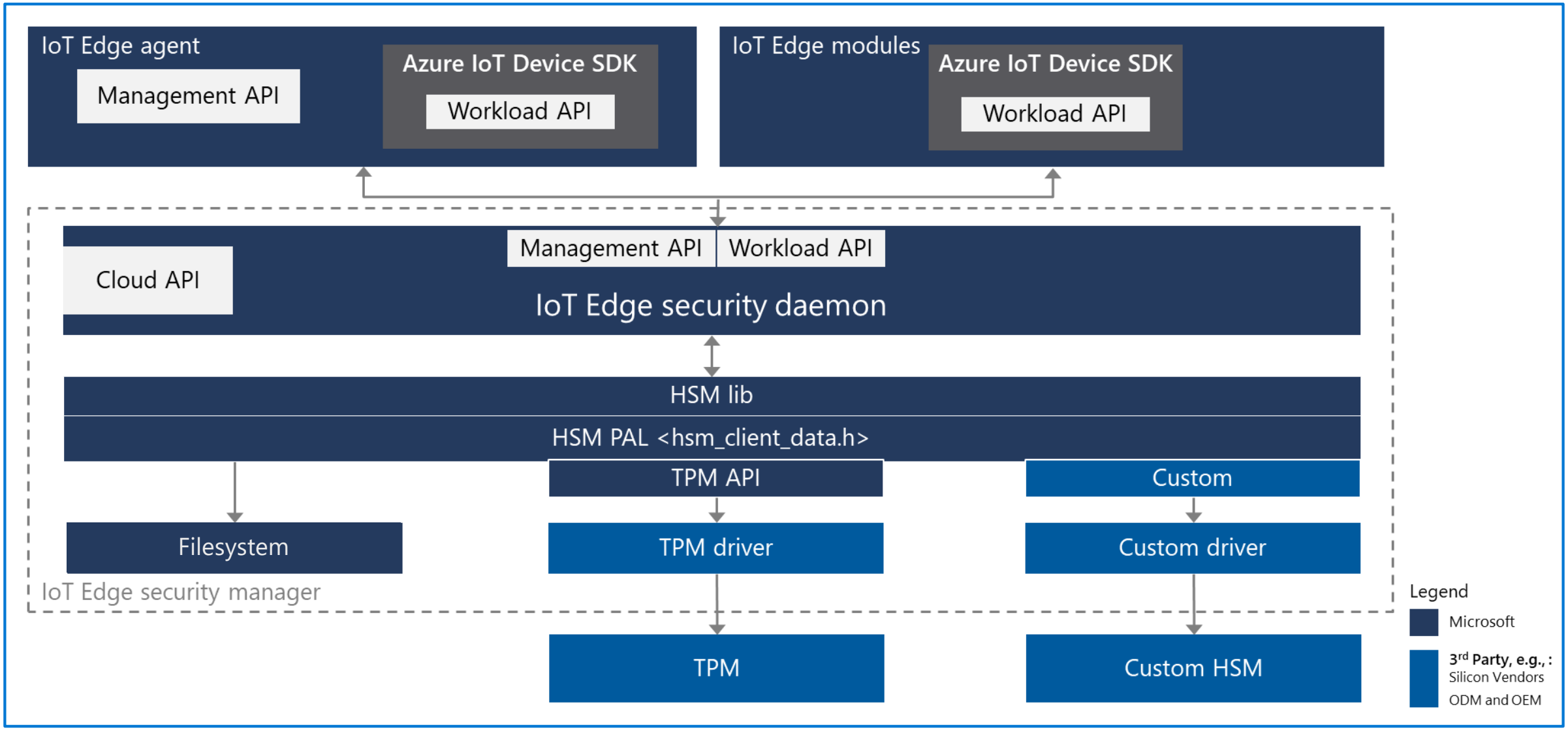
Azure IoT Edge security

Principles and goals:

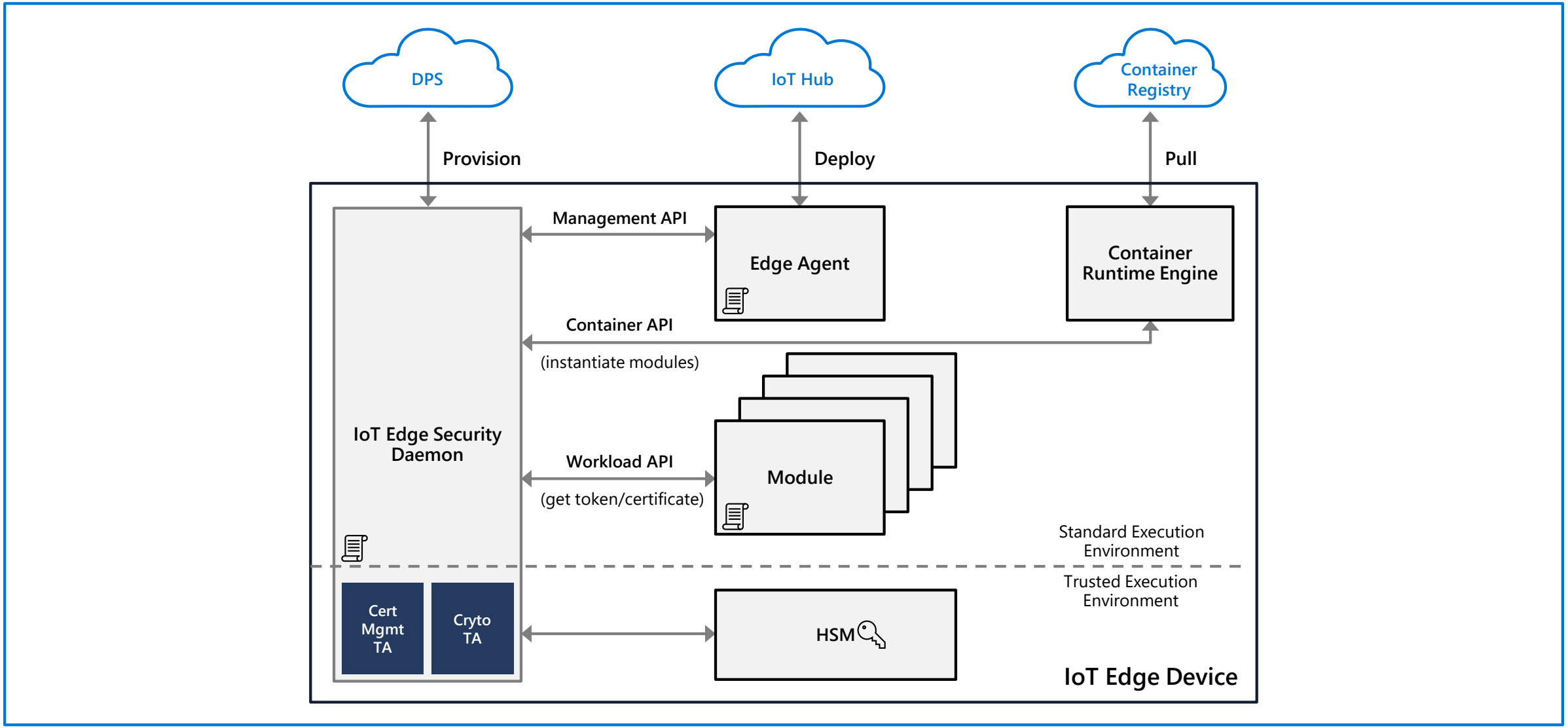
Standardized protocols	Secure technology isolation from app developer	Availability of technology

Protected general computing	Application execution with runtime integrity checking
Secure execution environment	Privileged executions and systems resource access control
Secure boot/updates	Bootstrapping and recovery
Hardware root of trust	Trust anchor

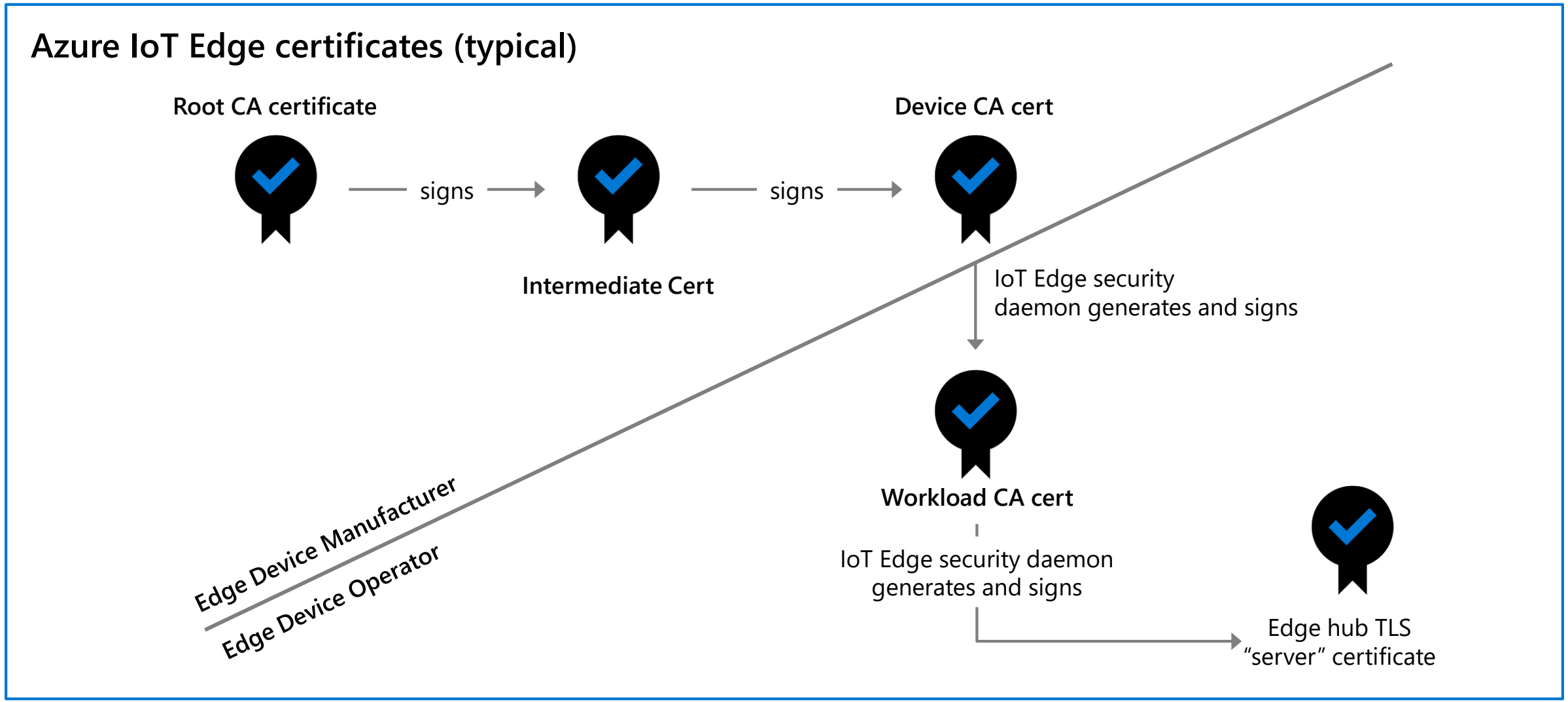
Azure IoT Edge security manager



Azure IoT Edge security daemon

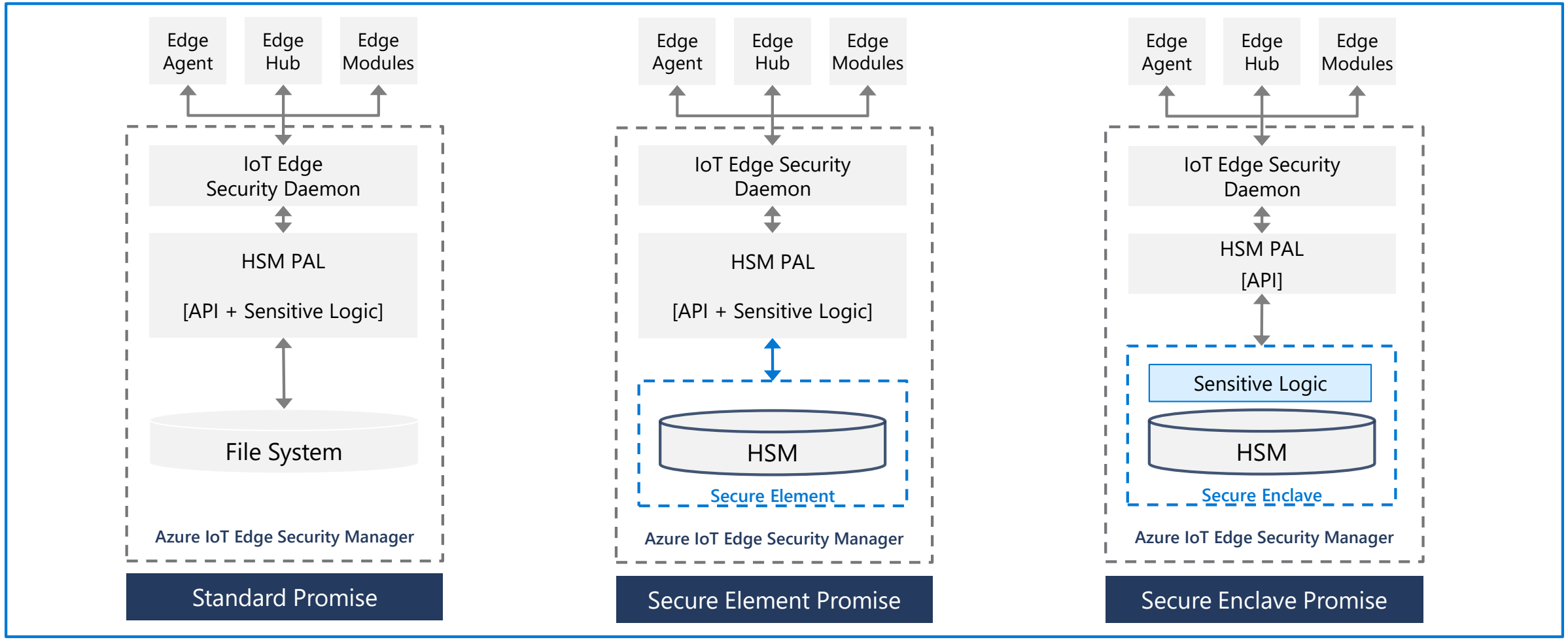


How Azure IoT Edge uses certificates



Azure IoT Edge device security promises

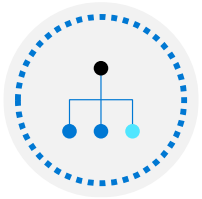
What is the maximum protection you can expect if the device fell into wrong custody?



HSM PAL = Hardware Secure Module Platform Abstraction Layer

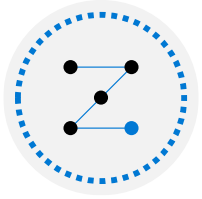
Lesson 3: Edge deployment process

Introduction to IoT Edge deployment



Deployments can be *manual* or *automatic*

Manual deployments are configured per-device



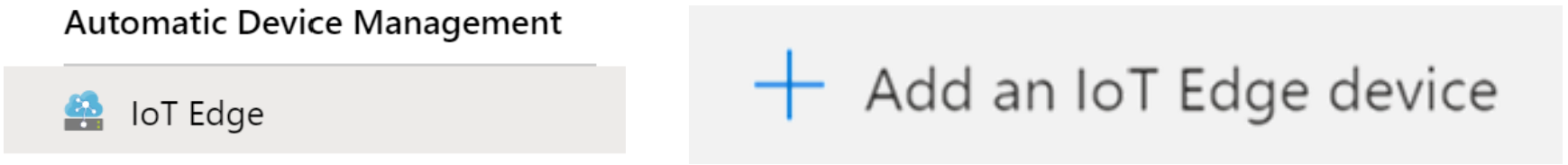
Automatic deployments are configured based on targeting device tags

Adding an IoT Edge device to IoT Hub

CLI: Same as adding a regular device, with an extra setting

```
az iot hub device-identity create – device-id myEdgeDevice --hub-name {hub_name}
--edge-enabled true
```

Portal: Similar process as adding a device, but in a different area



Registering IoT Edge devices through DPS

CLI: Same as adding a regular device, with an extra setting

```
az iot dps enrollment create … --edge-enabled true
```

Portal: Same as adding a regular device, with an extra setting

Deployment manifest

Deployment manifests follow this structure

```
{
    "modulesContent": {
        "$edgeAgent": { // required
            "properties.desired": {
                // desired properties of the Edge agent
                // includes the image URIs of all modules
                // includes container registry credentials
            }
        },
        "$edgeHub": { //required
            "properties.desired": {
                // desired properties of the Edge hub
                // includes the routing information between modules, and to IoT Hub
            }
        },
        "module1": {  // optional
            "properties.desired": {
                // desired properties of module1
            }
        },
        "module2": {  // optional
            "properties.desired": {
                // desired properties of module2
            }
        },
        ...
    }
}
```

Deployment manifest

The $edgeAgent properties follow this structure

```
"$edgeAgent": {
    "properties.desired": {
        "schemaVersion": "1.0",
        "runtime": {
            "settings":{
                "registryCredentials":{
                      // give the edge agent access to
                      // container images that aren't public
                    }
                }
            }
        },
        "systemModules": {
            "edgeAgent": {
                // configuration and management details
            },
            "edgeHub": {
                // configuration and management details
            }
        },
        "modules": {
            "module1": { // optional
                // configuration and management details
            },
            "module2": { // optional
                // configuration and management details
            }
        }
    }
},
```

Deployment manifest

Routes are declared in the $edgeHub desired properties with the following syntax

```
"$edgeHub": {
    "properties.desired": {
        "routes": {
            "route1": "FROM <source> WHERE <condition> INTO <sink>",
            "route2": "FROM <source> WHERE <condition> INTO <sink>"
        },
    }
}
```

Phased rollout

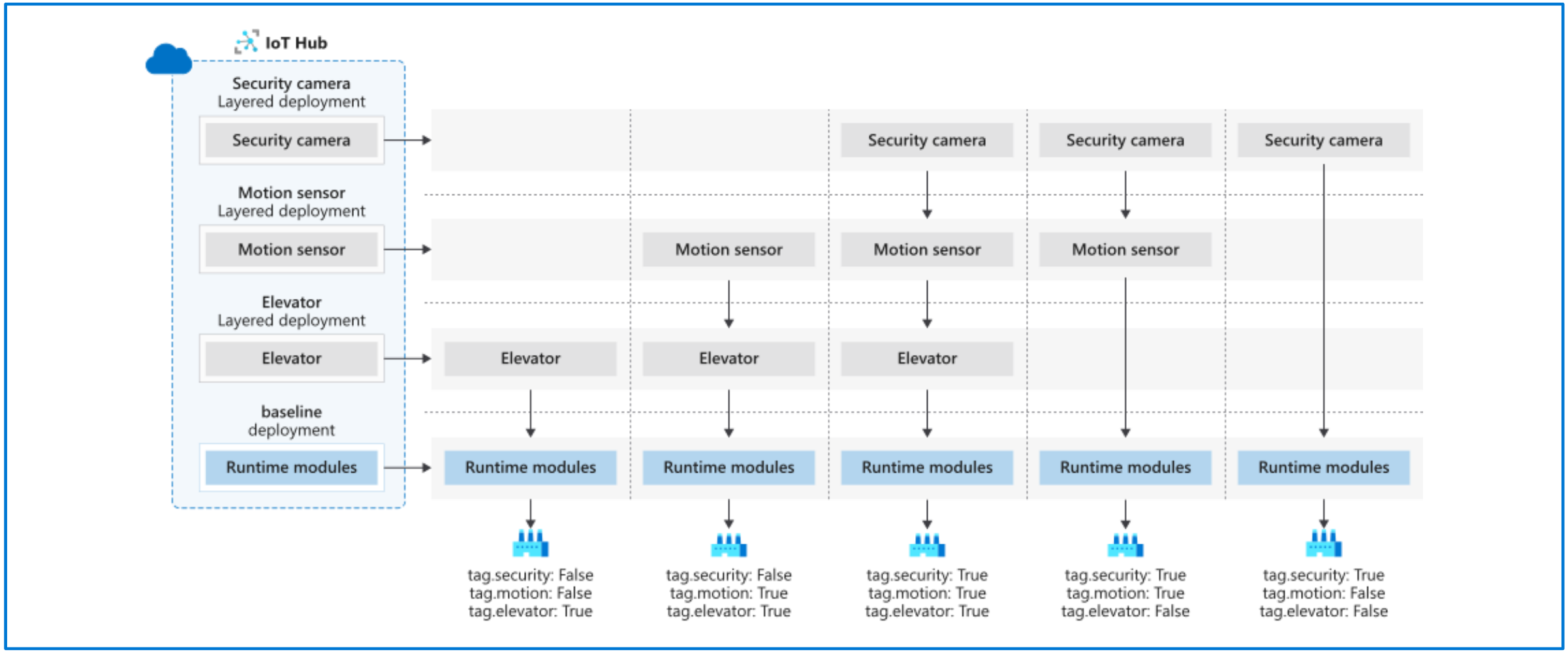
A *phased rollout* is an overall process whereby an operator deploys changes to a broadening set of IoT Edge devices

Deployment is based on targeting... examples:

```
deviceId = 'linuxprod1'
tags.environment = 'prod'
tags.environment = 'prod' AND tags.location = 'westus'
tags.environment = 'prod' OR tags.location = 'westus'
tags.operator = 'John' AND tags.environment = 'prod' NOT deviceId = 'linuxprod1'
properties.reported.devicemodel = '4000x'
```

Layered deployment

A *layered deployment* allows for minimizing manifest duplication



Layered properties example

```
"SimulatedTemperatureSensor": {
  "properties.desired": {
    "SendData": true,
     "SendInterval": 5
  }
}
```

```
“SimulatedTemperatureSensor": {
  "properties.desired.layeredProperties": {
    "StopAfterCount": 1000
  }
}
```

```
"properties": {
  "desired": {
    "SendData": true,
    "SendInterval": 5,
    "layeredProperties": {
      "StopAfterCount": 1000
    }
  }
}
```

Rollback

Deleting a deployment doesn't remove the configuration from the device... so rollback means doing a new deployment that has the previous configuration

Perform rollbacks in the following sequence:

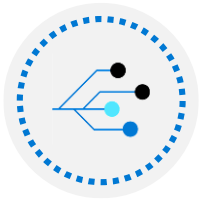
Modify or remove the target condition expression of the deployment you wish to roll back so that the devices no longer meet the targeting condition

Confirm that a second deployment is also targeted at the same device set. If the goal of the rollback is to remove all modules, the second deployment should not include any modules

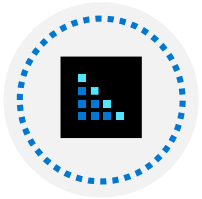
Verify that the rollback succeeded by viewing the deployment status

Deployment checklist

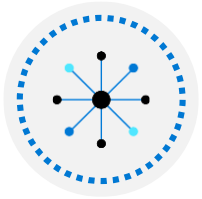
Device configuration



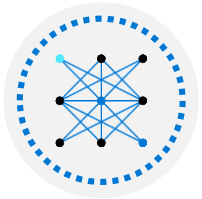
Deployment



Container management



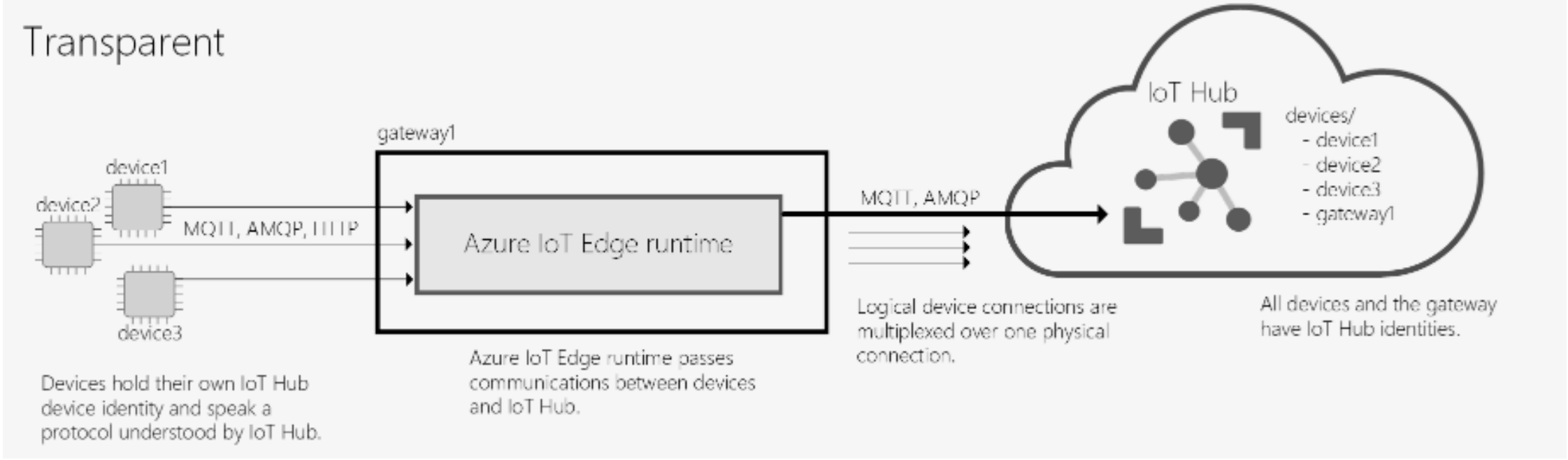
Networking



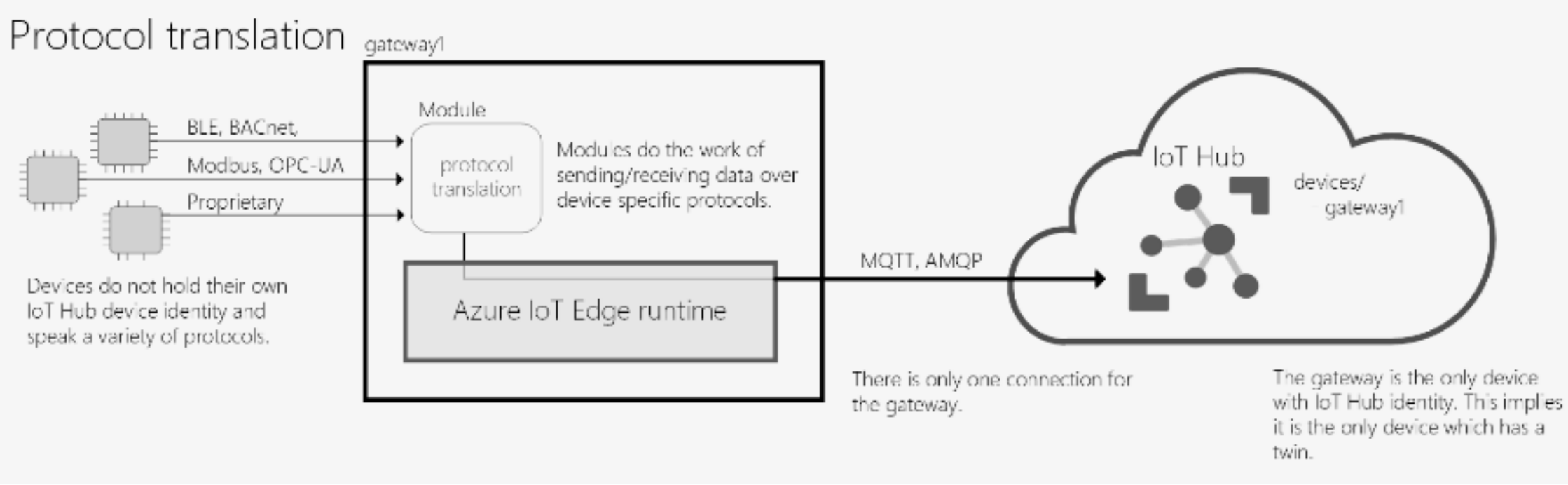
Solution management

Lesson 4: Edge gateway devices

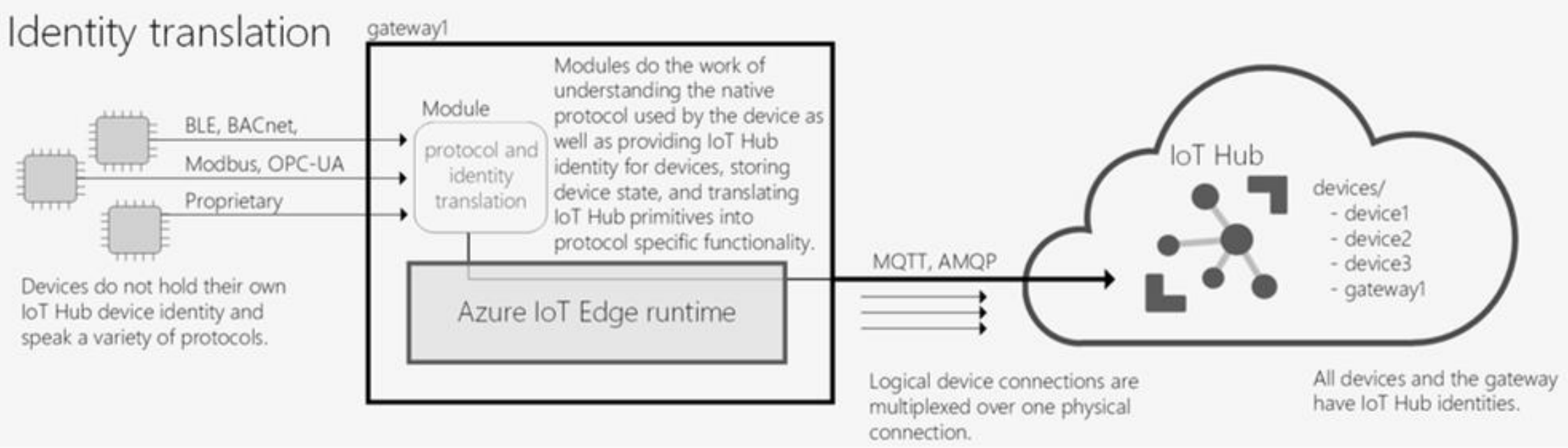
Azure IoT Edge gateway patterns: Transparent



Azure IoT Edge gateway patterns: Protocol Translation



Azure IoT Edge gateway patterns: Identity Translation



Azure IoT Edge as a gateway: Pattern comparison

	Transparent gateway	Protocol translation	Identity translation
Identities stored in the IoT Hub identity registry	Identities of all connected devices	Only the identity of the gateway device	Identities of all connected devices
Device twin	Each connected device has its own device twin	Only the gateway has a device and module twins	Each connected device has its own device twin
Direct methods and cloud-to- device messages	The cloud can address each connected device individually	The cloud can only address the gateway device	The cloud can address each connected device individually
IoT Hub throttles and quotas	Apply to each device	Apply to the gateway device	Apply to each device

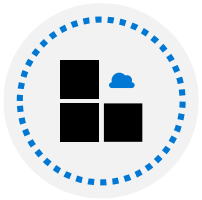
Authenticate a downstream device

Symmetric key authentication

X.509 authentication:

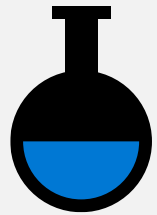
Self-signed certificate with a per-device thumbprint

CA-issued by a trusted CA (root or intermediate)

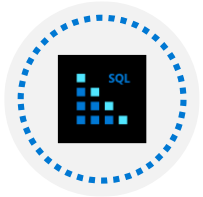


Tied to the device identity in IoT Hub, just like a device without a gateway

Lesson 5: Module labs



Module 6 labs

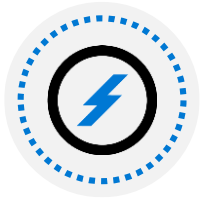


Lab 11: Introduction to Azure IoT Edge

You will create an IoT Edge device identity in IoT Hub using Azure CLI, and connect the IoT Edge Device to IoT Hub

You will add an Edge Module to the Edge Device

You will deploy Azure Stream Analytics as an IoT Edge Module



Lab 12: Setup an IoT Edge Gateway

You will generate and configure IoT Edge Device CA Certificates

You will create an IoT Edge device identity in IoT Hub using Azure portal, setup the IoT Edge gateway hostname, and connect an IoT Edge gateway device to IoT Hub

You will open IoT Edge gateway device ports for communication and create the downstream device identity in IoT Hub

You will connect a downstream device to IoT Edge gateway and verify event flow

Lesson 6: Module 6 review questions

Module review: Question 6.1

Which of the following choices describes the purpose of the IoT Edge hub?

Answer A:
The IoT Edge hub is responsible for deploying and monitoring the edge modules for the IoT Edge device.

Answer B:
The IoT Edge hub is responsible for managing communication between modules and with downstream devices.

Answer C:
The IoT Edge hub is responsible for configuring device settings when the Edge device is configured as an Edge gateway.

Module review: Question 6.2

What is the purpose of the IoT Edge agent?

Answer A:
The IoT Edge agent is responsible for deploying and monitoring the Edge modules for the IoT Edge device.

Answer B:
The IoT Edge agent is responsible for managing communication between modules and with downstream devices.

Answer C:
The IoT Edge agent is responsible for configuring device settings when the Edge device is configured as an Edge gateway.

Module review: Question 6.3

Which of the following answer choices describes a required component for an Azure IoT Edge implementation?

Answer A:
IoT Edge modules

Answer B:
IoT Edge gateway

Answer C:
Linux OS

Module review: Question 6.4

What is an IoT Edge deployment manifest used for?

Answer A:
It tells your device which modules to install and how to configure them to work together.

Answer B:
It tells your device to forward authentication requests to IoT Hub when a device first tries to connect.

Answer C:
It is a report that an IoT Edge device sends to IoT Hub indicating the status of instantiated modules.

Module review: Question 6.5

Which of the following choices is a step in the high-level deployment process for IoT Edge modules?

Answer A:
Remove the device identities from the IoT Hub registry.

Answer B:
Reprovision the devices after configuration.

Answer C:
Retrieve the status of the devices after configuration.

Module review: Question 6.6

A developer wants to connect devices that are not IP-enabled to an IoT hub using an IoT Edge gateway device. The developer wants only the IoT Edge gateway device to have an identity in IoT Hub.

Which IoT Edge gateway pattern should the developer use?

Answer A:
Transparent.

Answer B:
Protocol translation.

Answer C:
Identity translation.

Module review: Question 6.7

A developer wants to connect devices that are not IP-enabled to an IoT hub using an IoT Edge gateway device. The developer wants each device to appear as a separate device in IoT Hub.

Which IoT Edge gateway pattern should the developer use?

Answer A:
Transparent.

Answer B:
Protocol translation.

Answer C:
Identity translation.