Microsoft Azure

Azure IoT Academy

Transforming your business

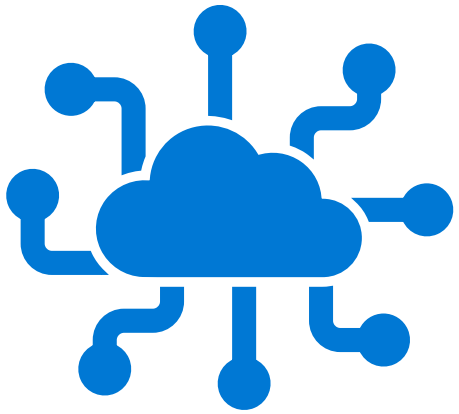
Month 1, Day 2: Core Services

Rebekah Midkiff
Technical Specialist
Microsoft

Alan Blythe
Sr. Technical Specialist
Microsoft

Eric Johnston
Sr. Technical Specialist
Microsoft

IoT Academy Expectations



- *We have a very large audience, so please keep yourself on mute except when called.*
- *Please raise your hand and wait for acknowledgement before unmuting to ask a question.*
- *Use Teams reactions to ease interactions of a large audience*
- *We want this to be interactive so please don't hesitate to let us know if you have a question (comment in chat or raise hand).*
- *If you're stuck on a hands-on lab, we request that you notify us in chat and raise your hand so we can move you to a breakout meeting for assistance.*

IoT Academy Journey

Month 1

- IoT Core Services
- IoT Central
- IoT Hub
- Device Provisioning
- Azure Data Explorer
- Azure Stream Analytics
- Partner Showcase

Month 2

- IoT Edge
- EFLOW
- Azure Digital Twins
- Log Analytics
- Azure Monitor
- Partner Showcase

Month 3

- IoT Security
- Azure Sentinel
- Defender
- Partner Showcase
- Awards Ceremony

Day One, In Review

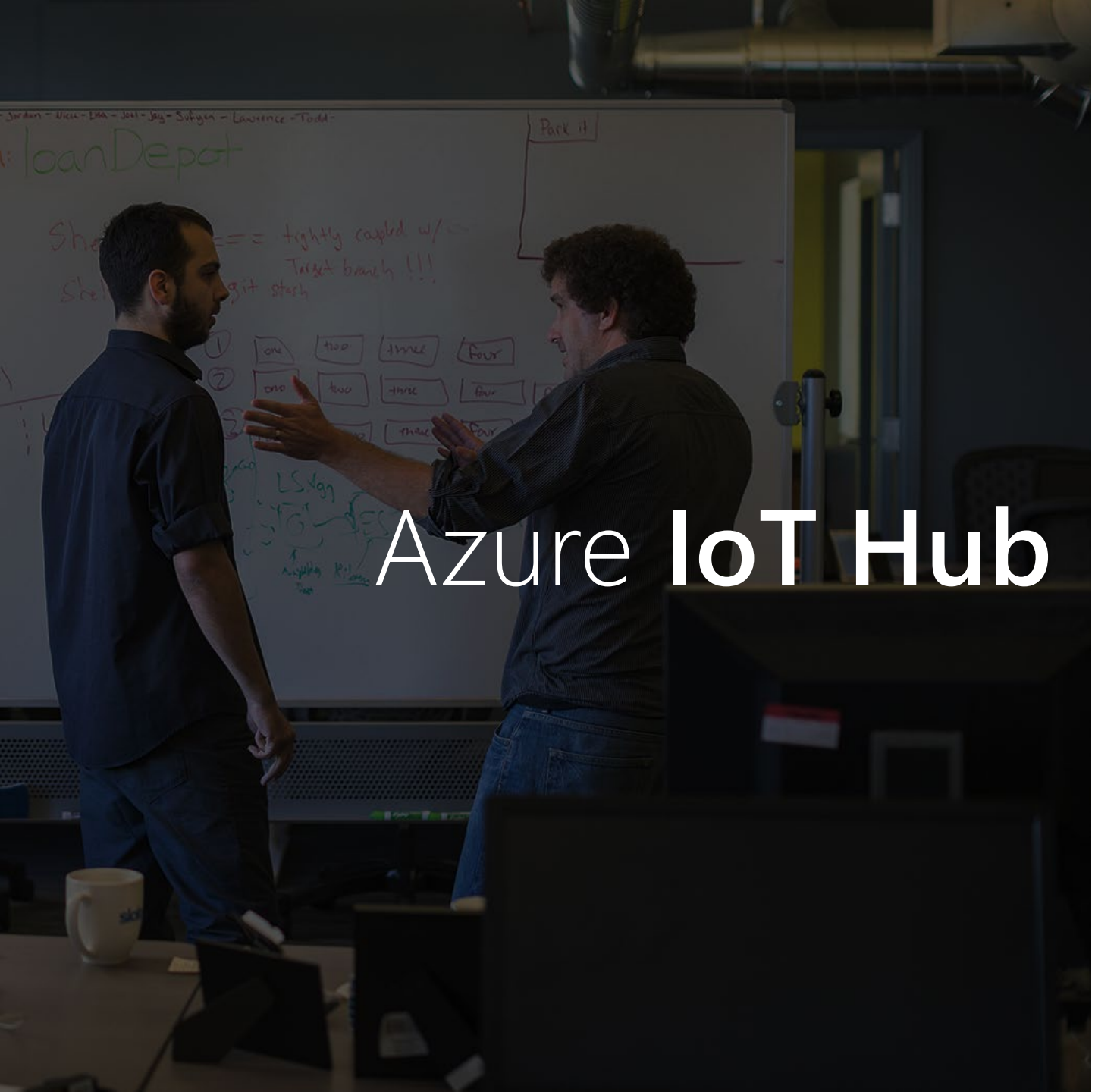
- IoT is a movement, a concept, which embraces distributed computing techniques
- Azure IoT Central
  - aPaaS, an extensible platform
  - A single service, many capabilities, built upon many of the core Azure IoT Services
  - Simple pricing
- What did we learn with our hands-on lab?
  - Familiarity with the Azure Portal
  - Deploying Azure IoT Central
  - Visual Studio Code to simulate an Azure IoT device utilizing the Azure IoT Device SDK for C#
  - Customizing Azure IoT Central with dashboards and visualizations

Day Two Agenda (All times are in EST)

- 10:05am – 10:20am Introduction/Expectations Kickoff - Team
- 10:20am- 12:00pm HOL
- 12:00pm – 12:15pm Coffee Break (Flexible timing)
- 12:15pm – 1:00pm HOL
- 1:00pm - 1:45pm Lunch Break
- 1:45pm - 3:15pm HOL
- 3:15pm - 3:30pm Coffee Break (Flexible timing)
- 3:30pm - 4:00pm HOL

Day Three Agenda (All times are in EST)

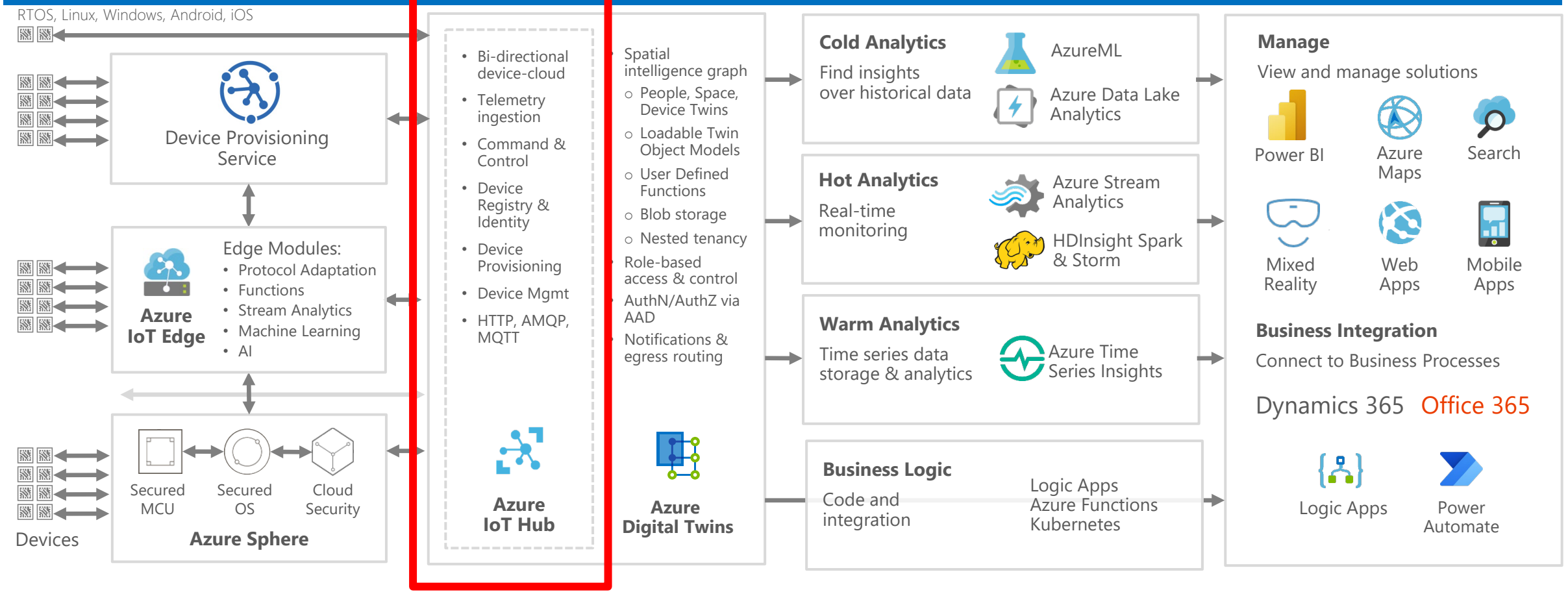
- 10:15am – 10:30am Introduction/Expectations Kickoff - Team
- 10:30am- 12:00pm Partner Showcase
- 12:00pm – 12:15pm Coffee Break
- 12:15pm – 1:15pm Partner Showcase
- 1:15pm - 2:00pm Lunch Break
- 2:00pm - 3:00pm Partner Showcase
- 3:00pm - 3:15pm Coffee Break
- 3:15pm - 3:30pm Close/Recap/Q&A



Azure **IoT Hub**

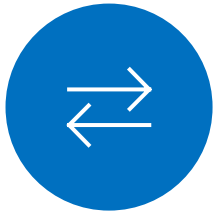
- Establish bi-directional communication with billions of IoT devices
- Enhance security with per device authentication
- Provision devices at scale w/ IoT Hub Device Provisioning Service
- Manage devices at scale with device management
- Multi-language and open source SDKs

Azure IoT End-to-End Platform



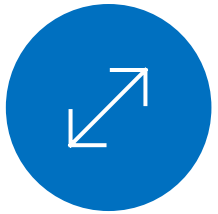
THINGS | INSIGHTS | ACTIONS

Azure IoT Hub



Bi-directional communication

- Millions of devices
- Multi-language, open source SDK
- HTTPS/AMQP/MQTT
- Send telemetry
- Receive commands
- Device management
- Queries and jobs



Enterprise scale and integration

- Billions of messages
- Scale up and down
- Declarative message routes
- File upload
- Web sockets and multiplexing
- Azure monitor
- Azure resource health
- Configuration management

End-to-end security

- Per device authentication
  - Shared Key
  - X.509 certificate
- Per device enable/disable
- TLS security
- IP whitelisting/blacklisting
- All communication device initiated
- Firmware/software updates

Three Authentication Options for Devices

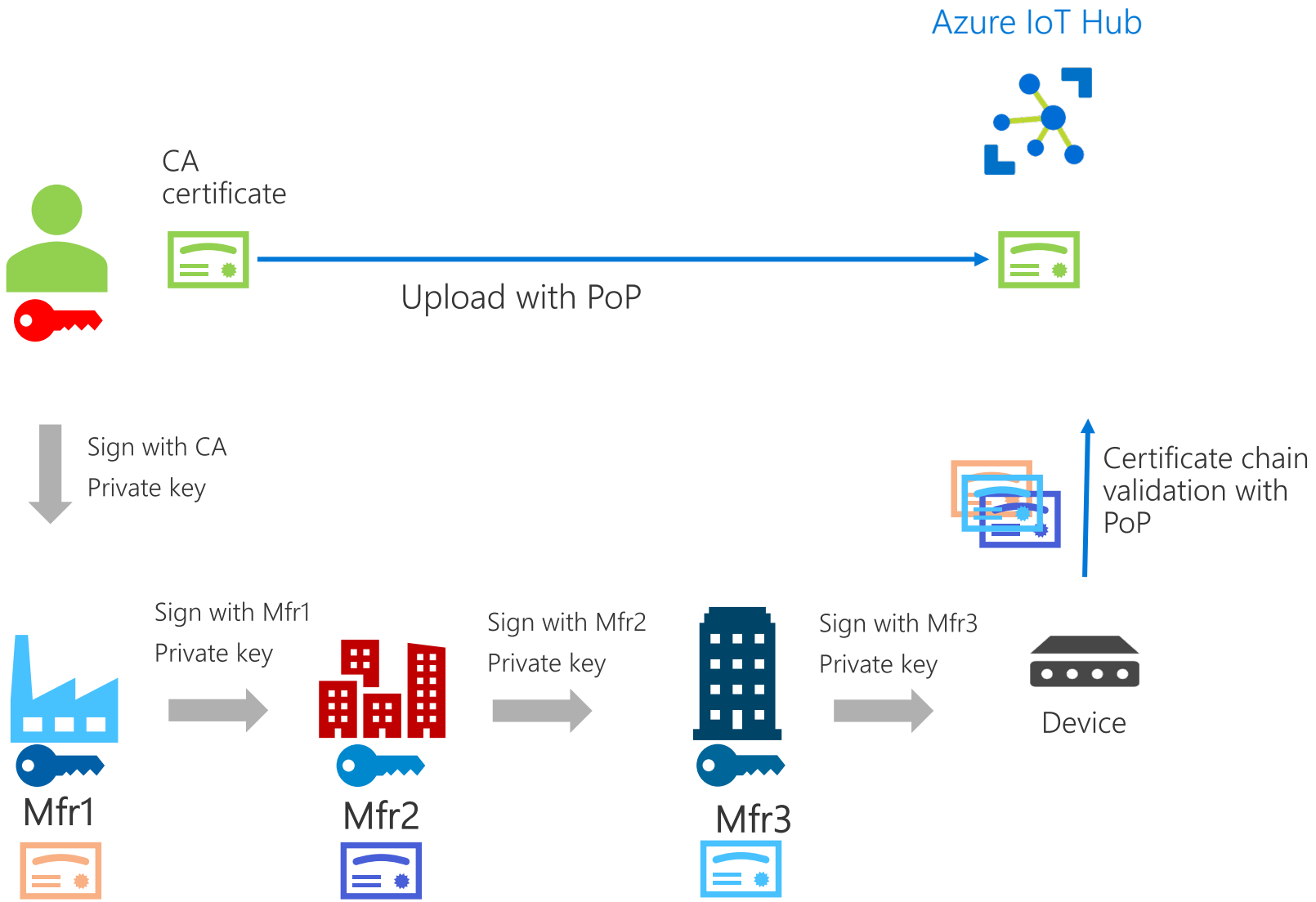
- Shared Access Signature (SAS)
- X.509 Self-signed Certificates
- X.509 Certificate Authority (CA)

Remember: All communications are encrypted using TLS

X.509 CA Setup and Authentication

- Register CA certificate to Hub, entails Proof-of-Possession
- Sign device through manufacturing flow
- IoT Hub uses CA certificate to validate chain and authenticate the device
- Simplifies supply chain by not sharing secrets

[Learn more about X.509 CA and device manufacturing flow]



Azure IoT Hub Device Management

Device Twin

Synchronize the device condition and configuration between cloud and device

Methods

Perform interactive actions on devices

Jobs

Broadcast and schedule device twin changes and methods at scale

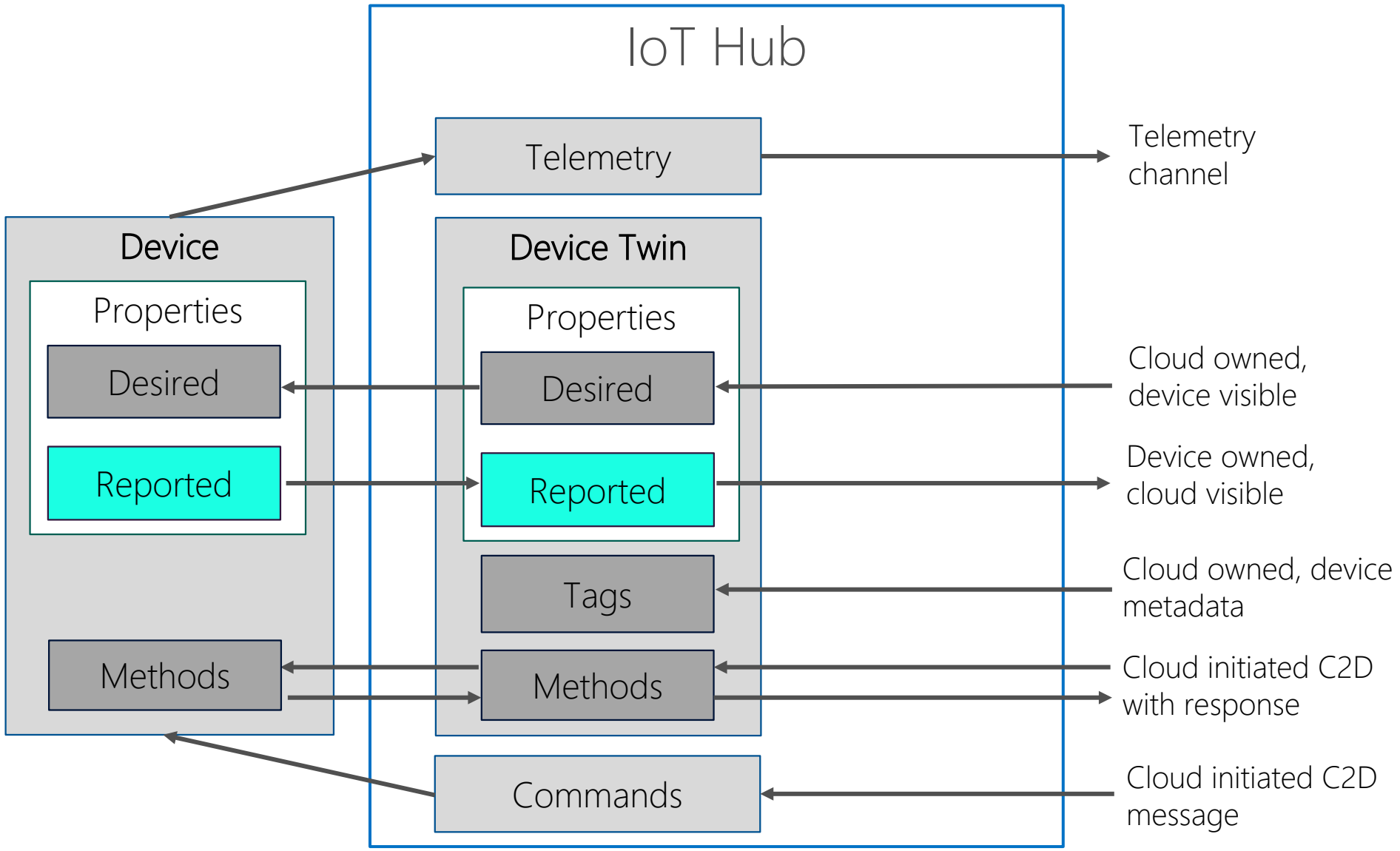
Queries

Dynamic reporting across device twin and jobs to attest device status and health

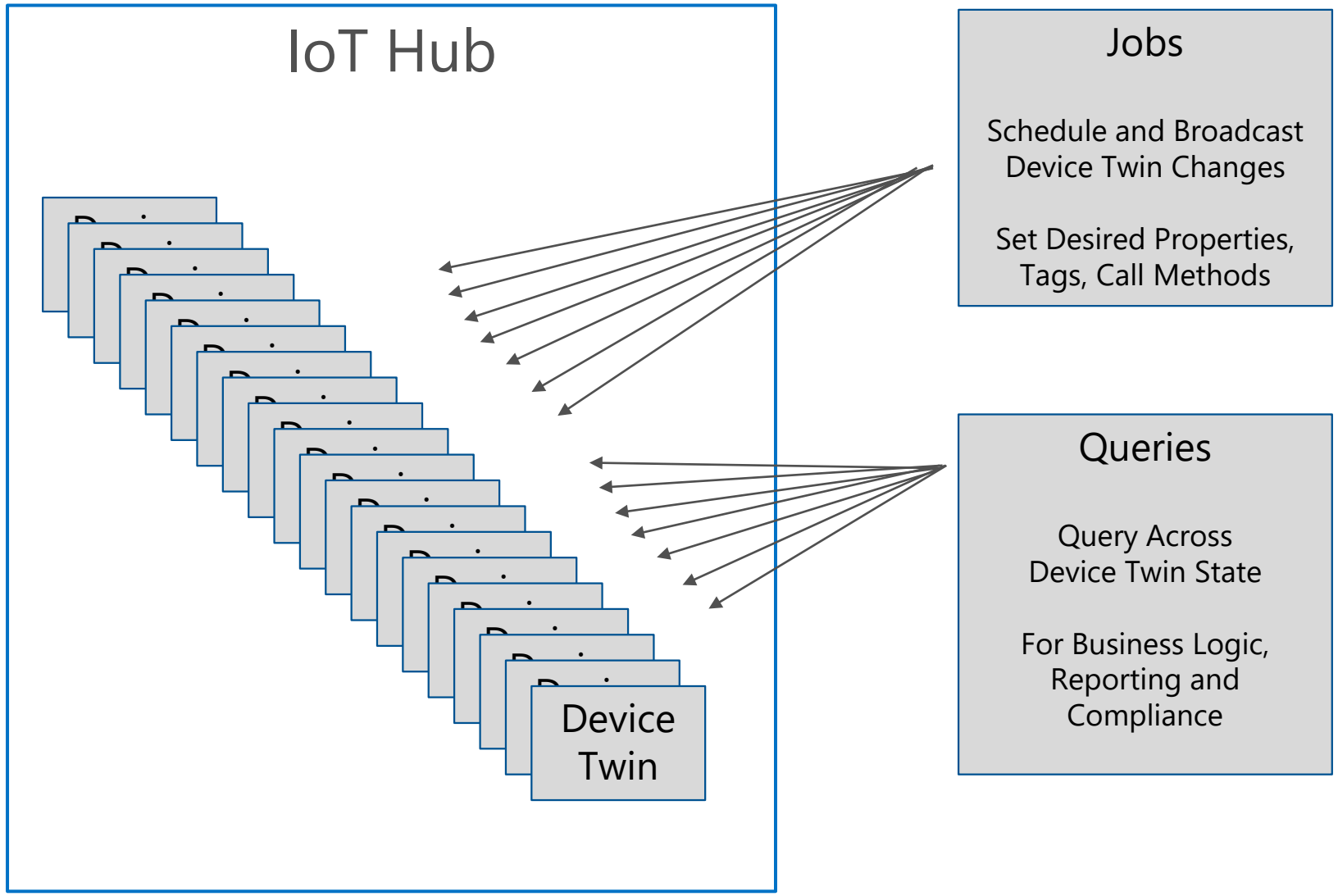
Patterns & Libraries

Get started quickly with the most essential operations supporting a diverse set of platforms

Device Management components



Device Management – at scale



Automatic Device Configuration

- Configure devices based on a set of target ‘conditions’
- Configurations are provided automatically
  - Based on device properties (reported, tags, etc)
  - Based on priority order
  - As devices enter and leave scope
- Three parts
  - Target Condition (e.g. properties.reported.chillerProperties.model='4000x’)
  - Target Content (e.g. properties.desired.chiller-water.temperature=50)
  - Metrics

    System metrics - metrics that are calculated by IoT Hub and cannot be customized by developers. Targeted specifies the number of device twins that match the target condition. Applies specified the number of device twins that have been modified by the configuration, which can include partial modifications in the event that a separate, higher priority configuration also made changes.

    Custom metrics - metrics that have been specified by the developer as queries against device twin reported properties. Up to five custom metrics can be defined per configuration.

Overview of Azure Device Update for IoT Hub

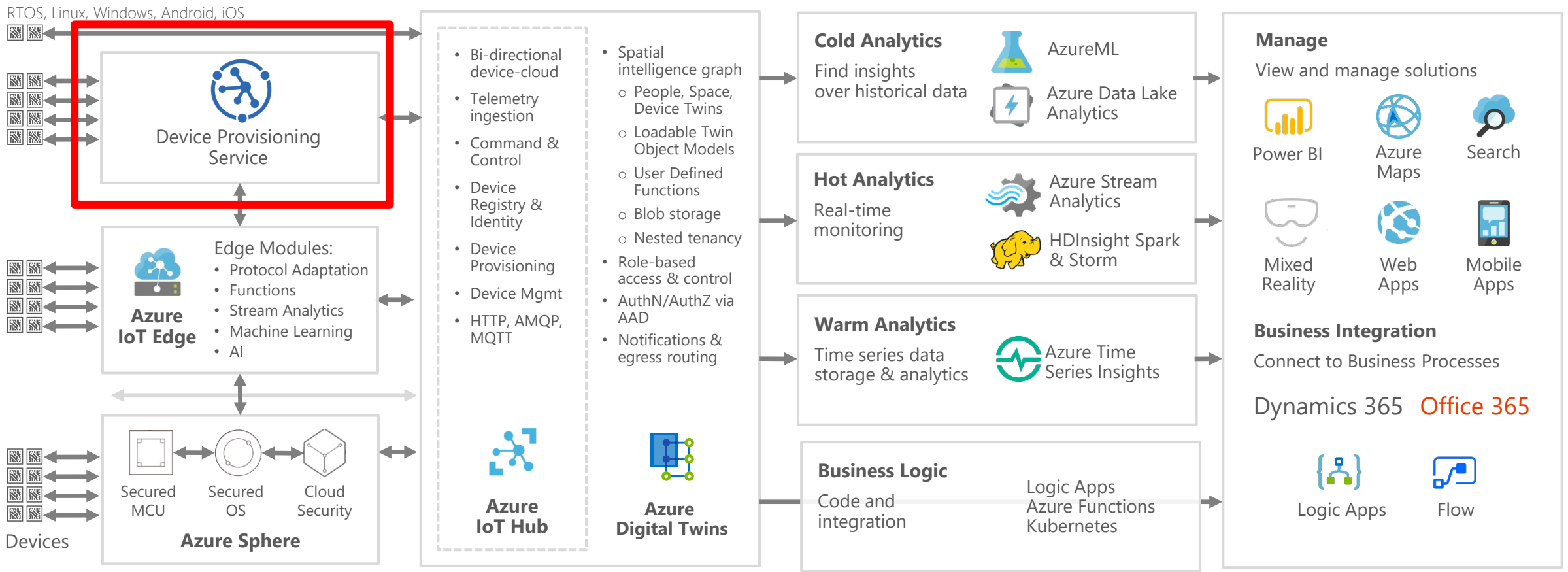
- A secure and flexible way to keep devices up to date using IoT Hub
- Allows importing, scheduling and reporting on updates
- Integrates with IoT Hub and can have a 1 to many [IoT Hub] deployment
- Leverages existing [and requires] Content Delivery Network – like Windows Update
- Based on an Open-Source Agent written in C and expected to be modified
- Sample Agents: Linux [with or without IoT Edge], Yocto and RTOS
- Device Updates can be package (think apt install xxx) or image based

https://docs.microsoft.com/en-us/azure/iot-hub-device-update/device-update-agent-overview

Hands-On Lab

Deploying IoT Hub

Azure IoT End-to-End Platform



Azure IoT Hub Device Provisioning Service

Simplifies provisioning

Supports multiple locations

Easiest way to mass-provision devices

URL stability

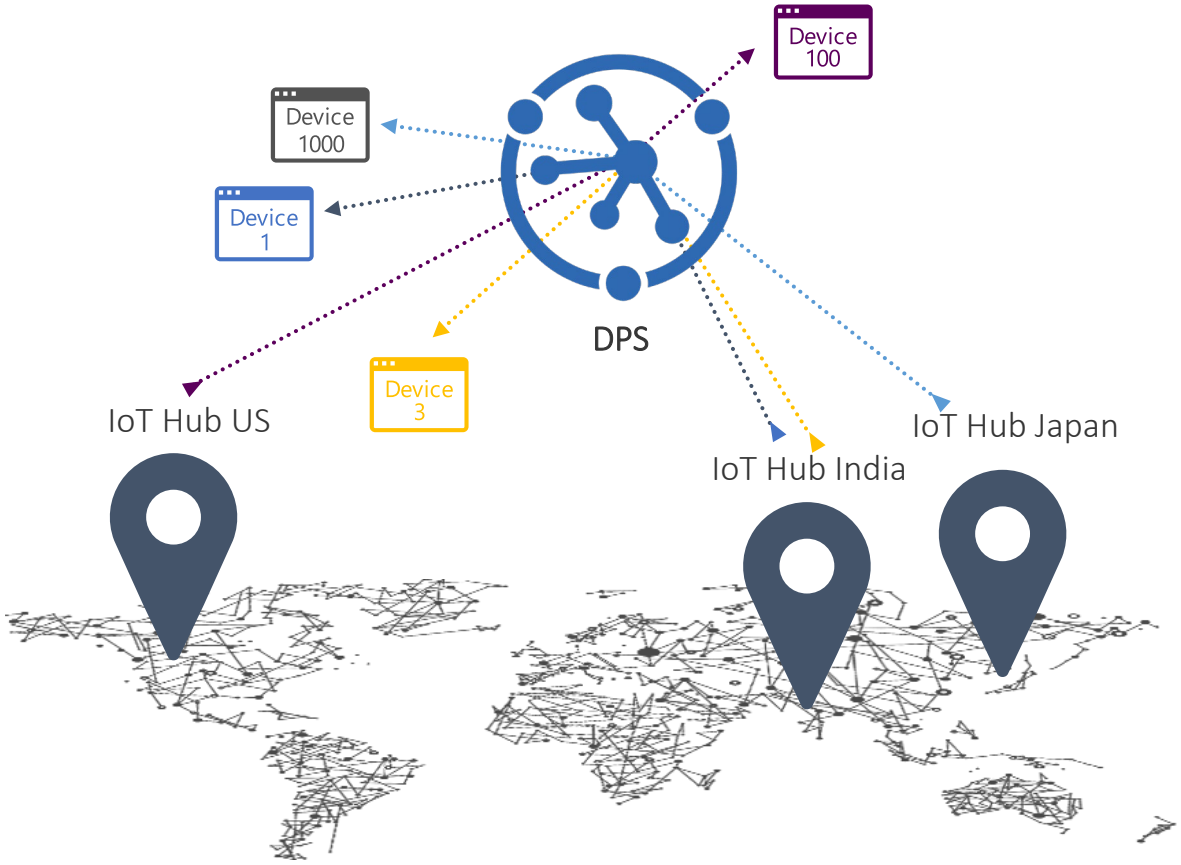
Enhanced security through HSM

For any device compatible with IoT Hub

Remove human error

Minimize manual connection requirements

Multitenancy support



DPS knows exactly which IoT Hub to connect and provision

Goals for device provisioning with Azure IoT

Securely automate the provisioning process

Devices are automatically and securely connected to the IoT Hub service and provisioned with an initial configuration

Multitenancy support

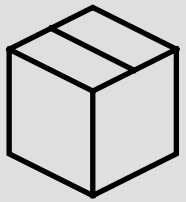
A single DPS can provide service for multiple IoT hubs (in multiple regions)

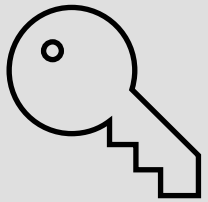
Flexible device assignment

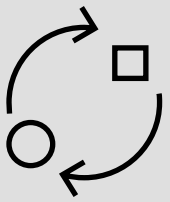
Customers provide rules and logic to assure the right device is attached to the right IoT solution (and associated IoT Hub)

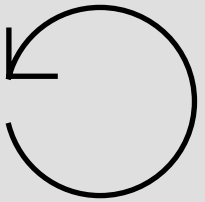
An IoT device's relationship to DPS

Initial setup	Retrieving a key	Rolling a key	Hard reset
Getting the device ready for the first time	For devices with limited or no key storage capabilities	Applicable only for devices which connect via a SAS token	When the device needs to be treated as new in-box









Enrollment list

One-stop shop for everything needed to provision a device

- Attestation information
- Initial configuration
- Additional device info

Support for

- Individual enrollments – good for devices with individual configuration needs
- Enrollment groups – good for lots of devices with the same initial configuration

Updatable throughout the supply chain

Linked IoT hubs

Linking an IoT hub to DPS gives DPS permissions to register devices to the hub

Links can be cross-region or cross-subscription

Allocation policies

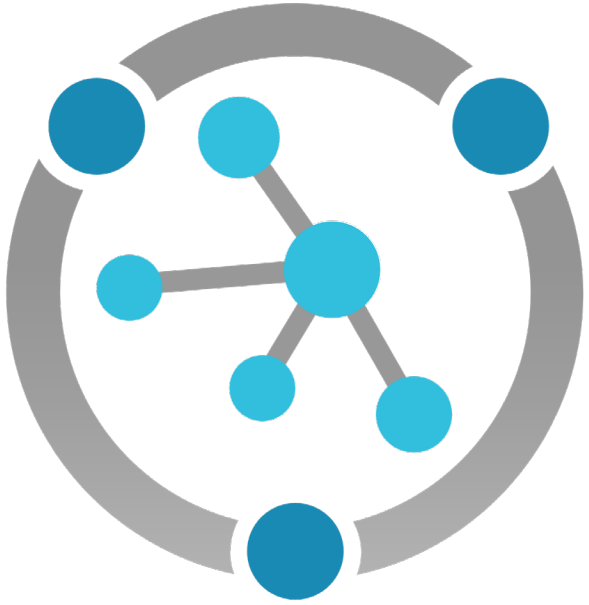
Determines how DPS assigns devices to linked hubs

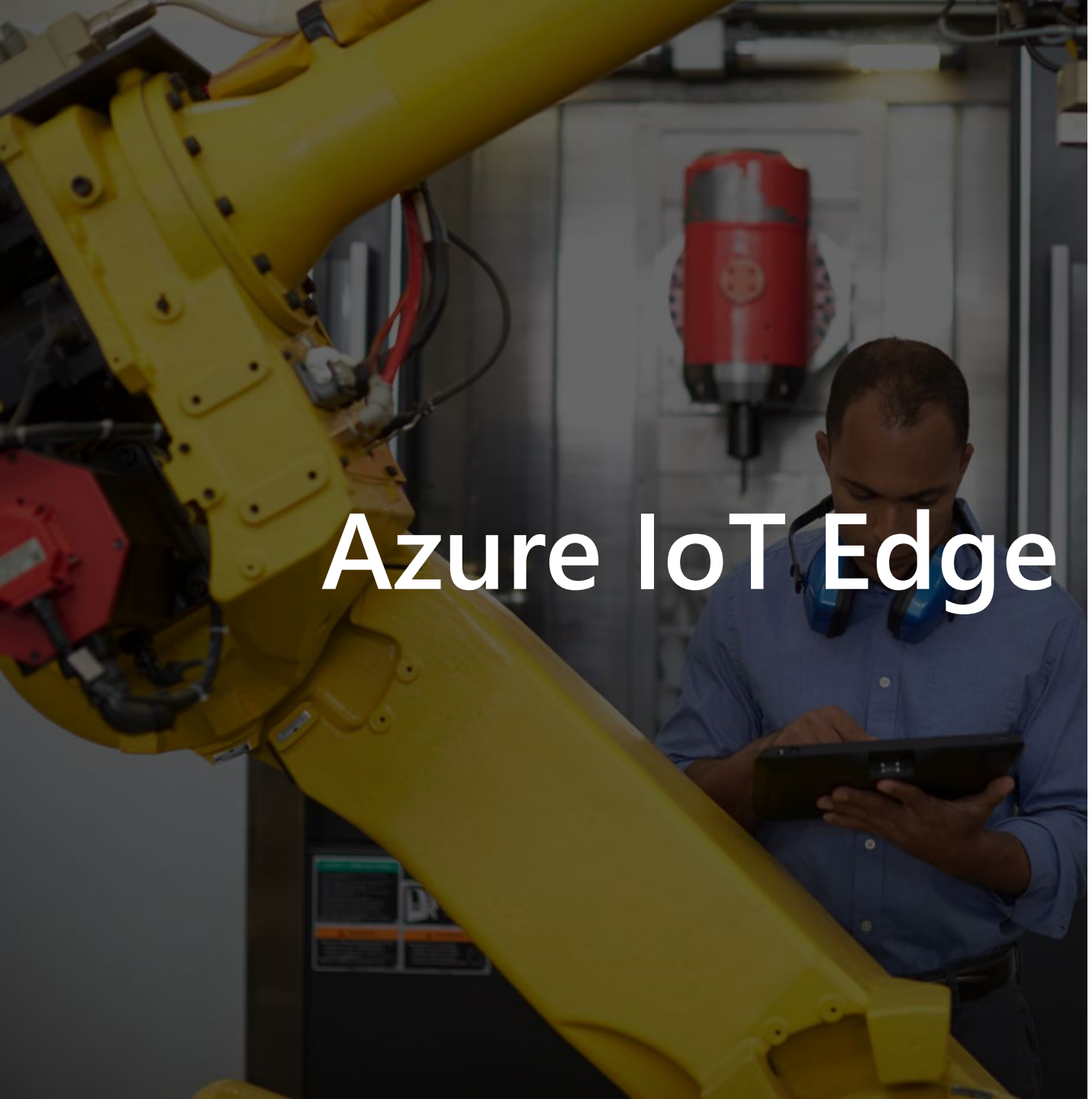
- Evenly weighted distribution
- Lowest latency
- Static configuration via the enrollment list

The allocation policy can be overridden per enrollment entry

Hands-On Lab

Azure IoT Hub Device Provisioning Service



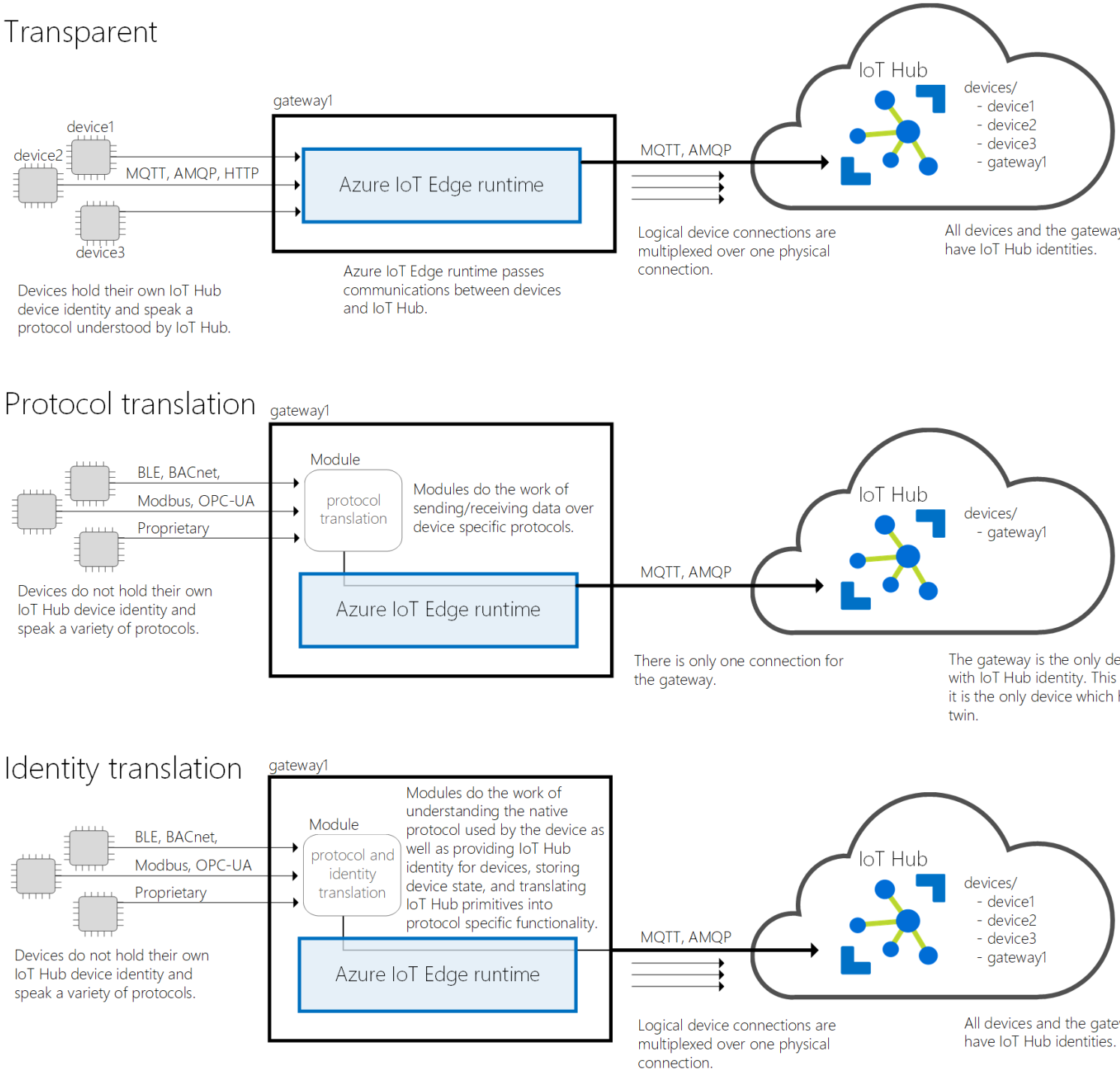


Azure IoT Edge

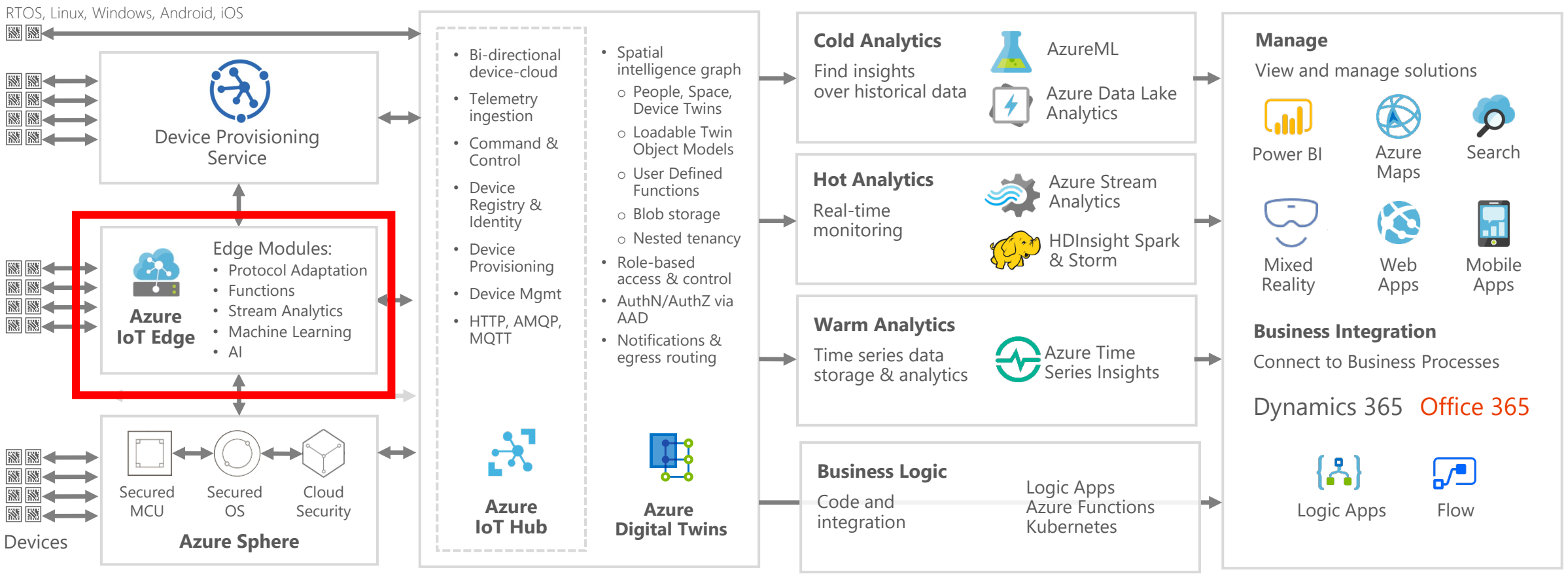
- Move cloud and custom workloads to the edge, securely
- Seamless deployment of AI and advanced analytics
- Configure, update and monitor from the cloud
- Compatible with popular operating systems
- Code symmetry between cloud and edge for easy development and testing
- Secure solution from chipset to cloud

Gateway patterns

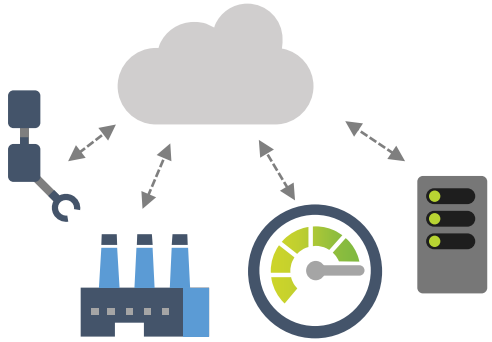
- Transparent
  - IoT Hub-aware devices connect as leaf devices
- Protocol translation
  - Module talks to leaf devices and acts as single device in IoT Hub
- Identity translation
  - Module talks to leaf devices and impersonates/manages identities of leaf devices as unique devices in IoT Hub

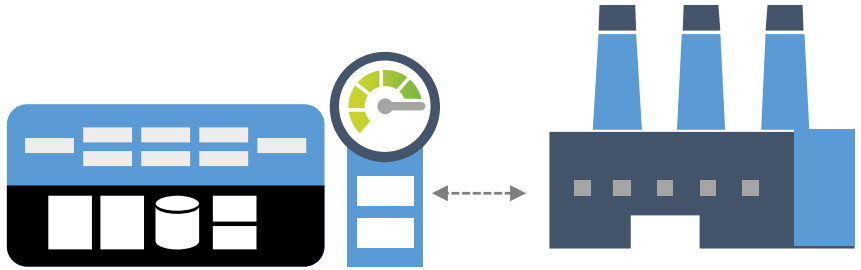


Azure IoT End-to-End Platform



IoT in the Cloud and on the 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

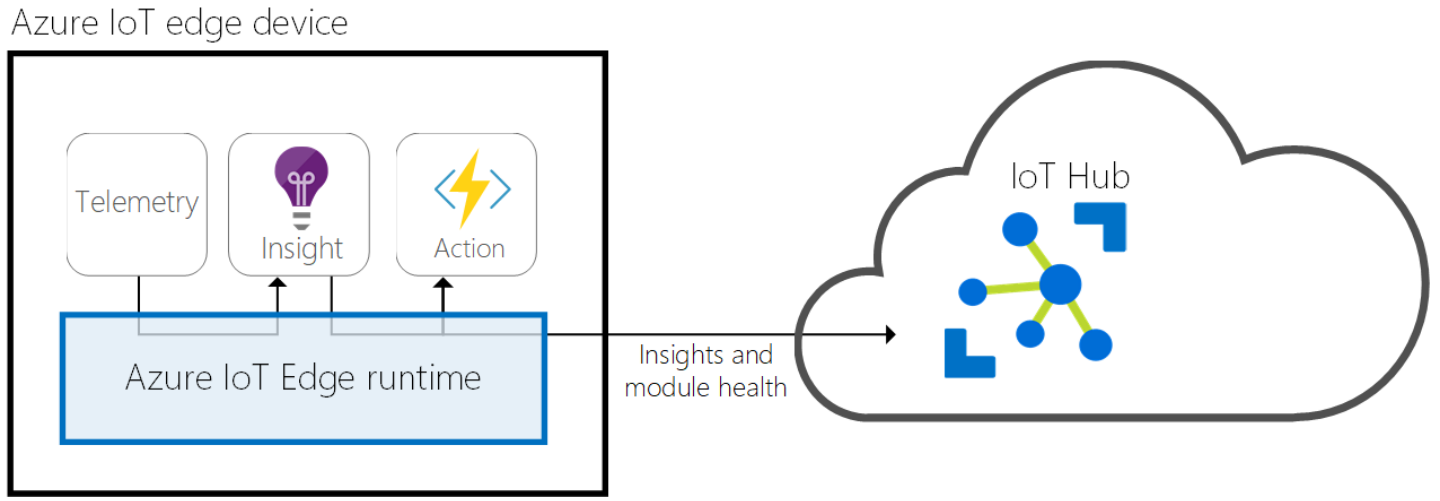
Low latency tight control loops require near real-time response

Protocol translation & data normalization

Privacy of data and protection of IP

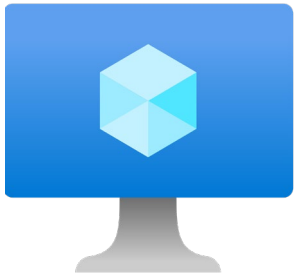
Symmetry

Concept – Azure 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

Azure Virtual Machines

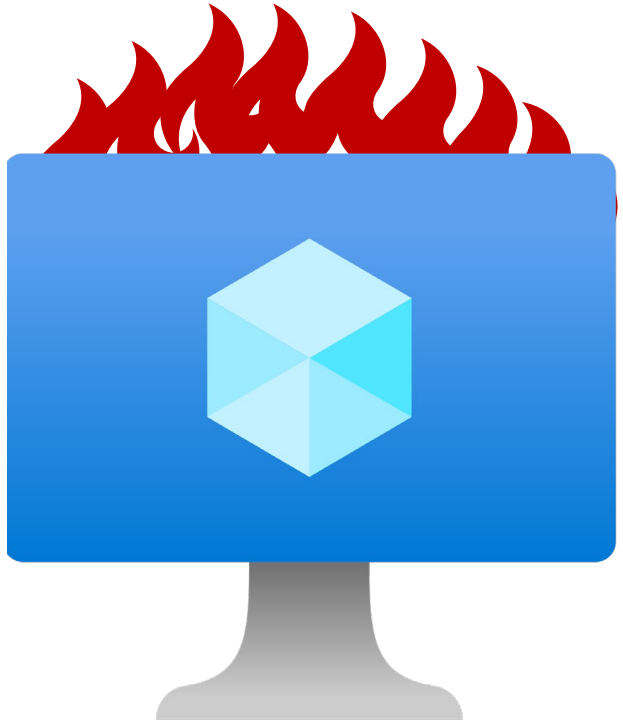


- Virtual Machines (VMs) come in a range of sizes with varying CPU counts, amount of memory, maximum possible disk IOPS, and ephemeral disk sizes (if present).
- We will be using the automatic shutdown feature to ensure our VM doesn't consume too much of our credits.
- VMs are created with either custom images or images from the marketplace. In our case, we will be using a marketplace image of Ubuntu Server 18.04 LTS when creating our VM.

- We will be assigning the VM a Network Security Group (NSG) and adding an inbound rule that will allow connections on SSH (TCP/22) from your public IP address.

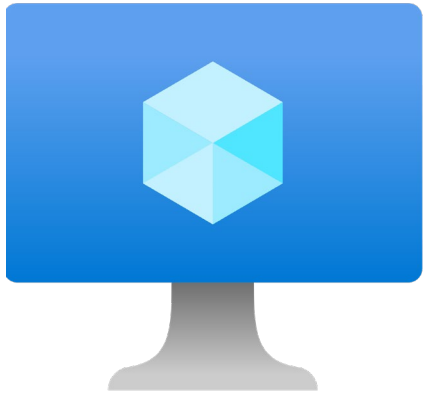
Warning!!!

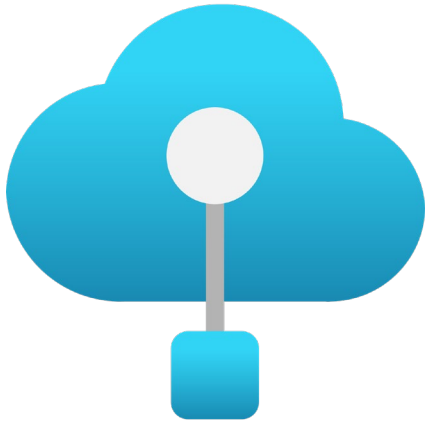
VM Availability is limited!



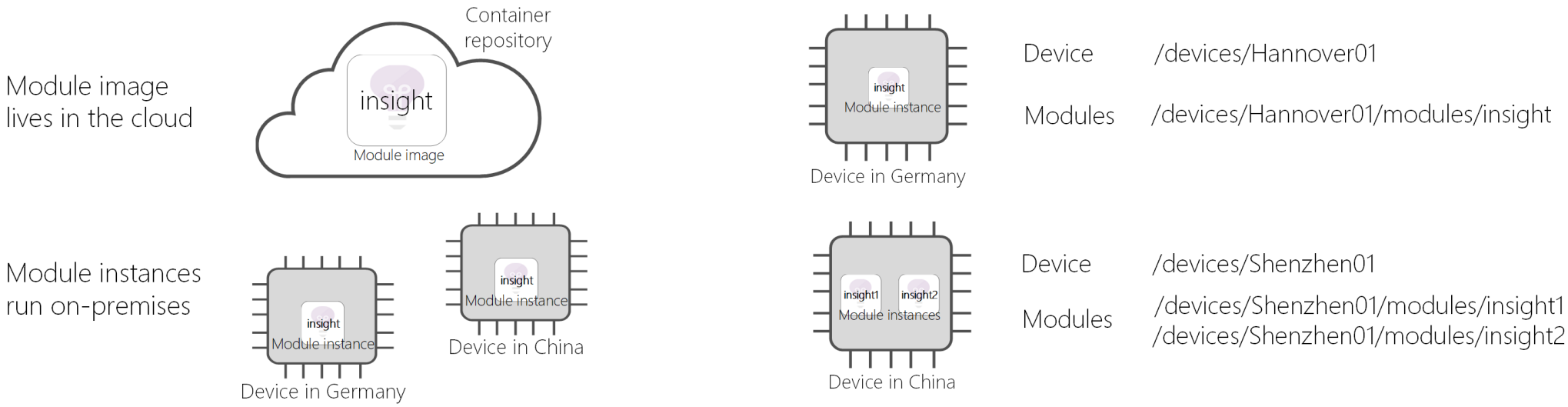
Hands-On Lab

Deploying an Ubuntu Azure VM & Installing the Azure IoT Edge Runtime





Concept – 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)

Concept - Routing

```
FROM <source> WHERE <condition> INTO <sink>
```

Sources – source of messages

```
/messages/modules/{mid}/outputs/{out1}
```

Condition – expression on messages properties/body

```
sensorType = “temp” and alert = true
```

Sinks – destination for messages (endpoints)

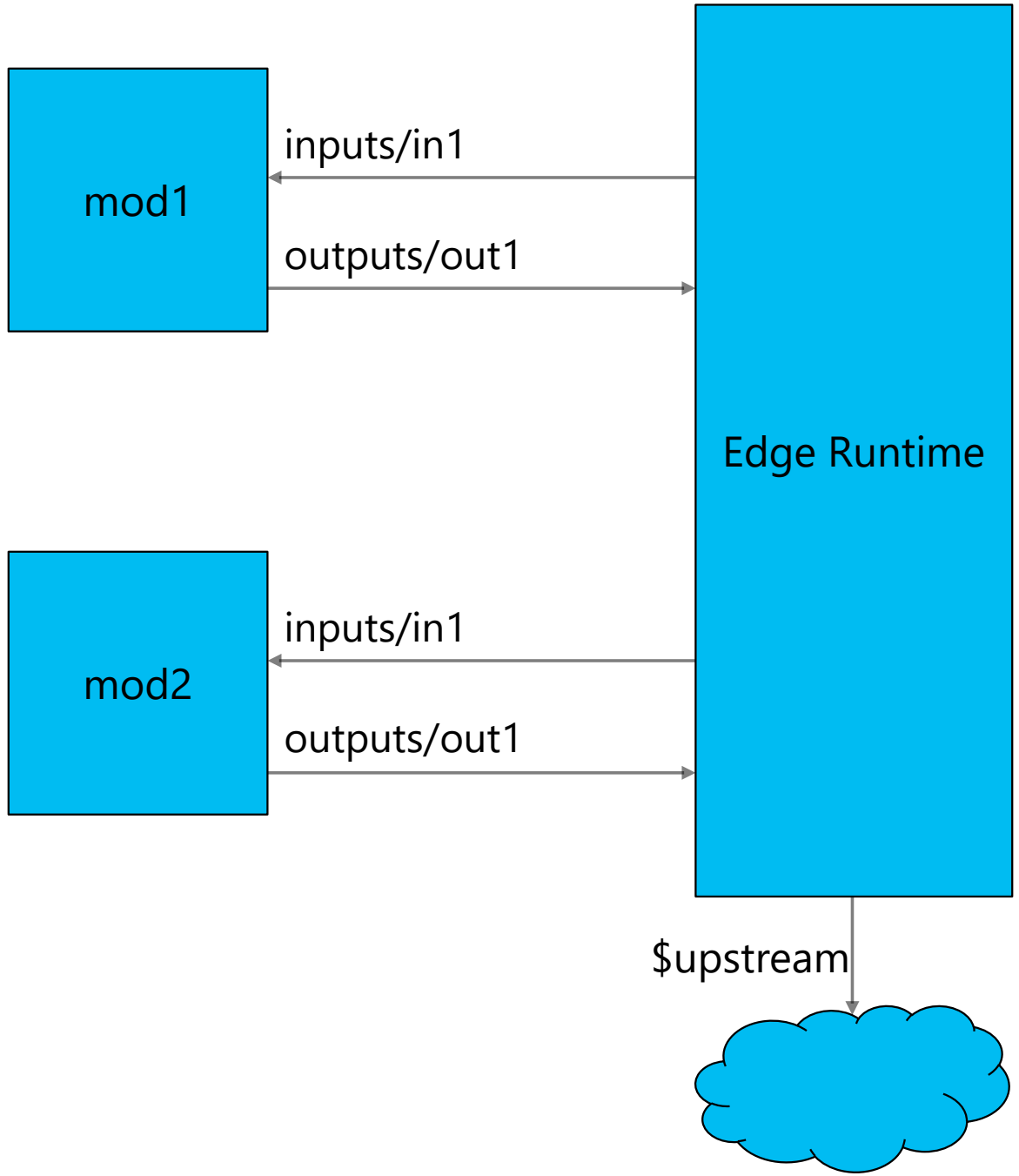
```
$upstream
```

```
brokeredEndpoint(“/modules/{mid}/inputs/{in1}”)
```

For example:

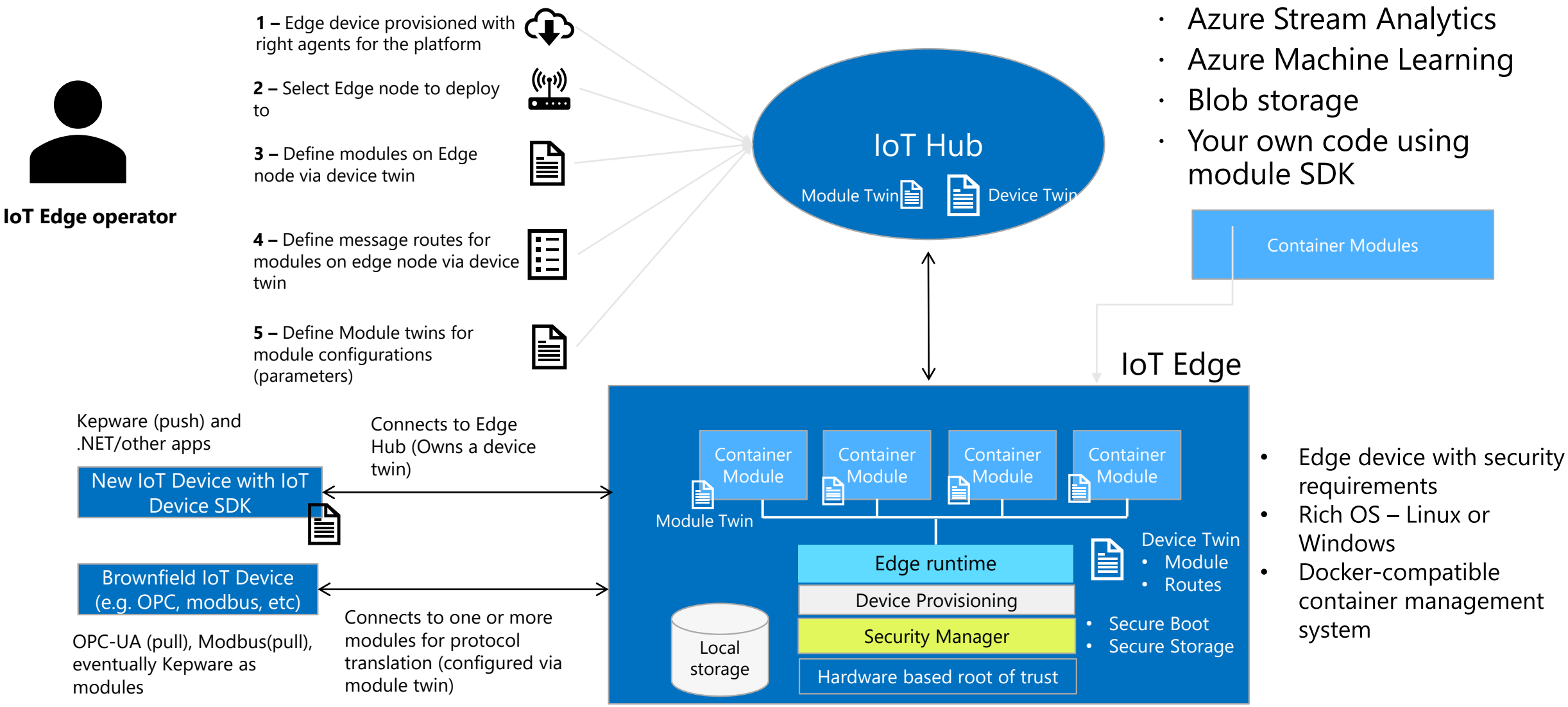
```
FROM /messages/modules/mod1/outputs/*
WHERE sensorType = “temp”
INTO brokeredEndpoint(“/modules/mod2/inputs/in1”)
```

[Query Language]



IoT Edge in action

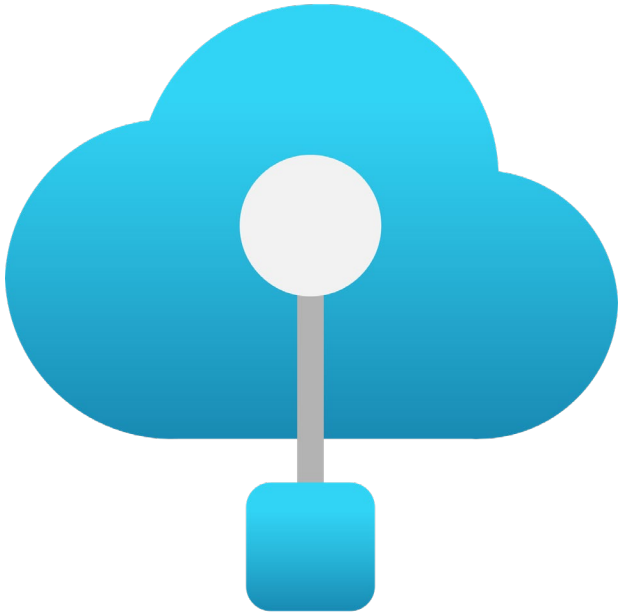
- Container based workloads
- Cognitive Services
- Azure Functions
- Azure Stream Analytics
- Azure Machine Learning
- Blob storage
- Your own code using module SDK



- Edge device with security requirements
- Rich OS – Linux or Windows
- Docker-compatible container management system

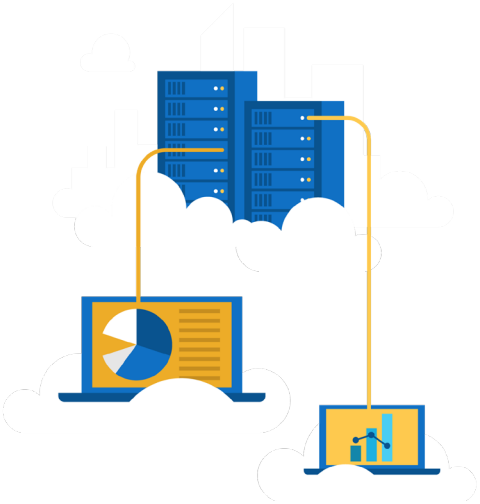
Hands-On Lab

Deploy an IoT Edge Module to Simulate Device Telemetry



Azure IoT Academy

Azure Stream Analytics & Azure Data Explorer





Aspect	Hot path	Warm path	Cold path
Name	Azure Stream Analytics	Azure Data Explorer (Kusto)	Azure Data Lake Analytics (Cosmos)
Latency	Seconds (up to, say, 5)	Minutes (up to, say, 5)	More
Queryable data storage	RAM	Attached (low latency) SSD	HDD or even remote storage
Query frequency	Automated (alerting)	Ad-hoc (human-generated)	Occasional
Max size of intermediate result	Single-node RAM	Cluster RAM	"Infinite" (spilled to HDD)
Recovery from query failures	No	No	Yes (built for batch processing; continue from last checkpoint)
Data analysis	Metrics (TSDB-like)	Text and numbers	Everything you can write a C# function for
Data form	Aggregated	Raw	Raw
Targeted for	Real time data viewing	Ad-hoc data exploration	Programmatic data manipulation

Azure cloud analytics

Azure Stream Analytics
Unlocking Real-time Insights

Time to Insight is Critical

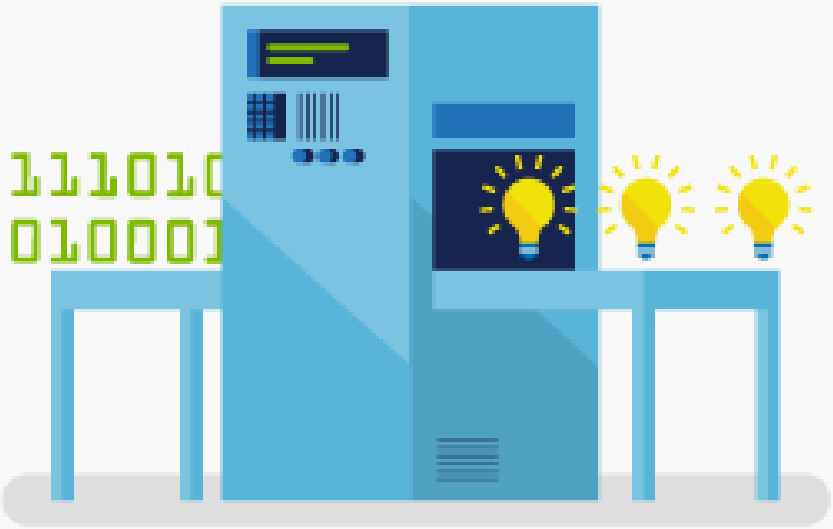
Reducing decision latency can unlock business value

Insights are Perishable

Window of opportunity for insights to be actionable

Ask Questions to Data in Motion

Can't wait for data to get to rest before running computation



Real-time Stream Processing

Simple Event Processing

- Filter
- Transform
- Enrich
- Split
- Route

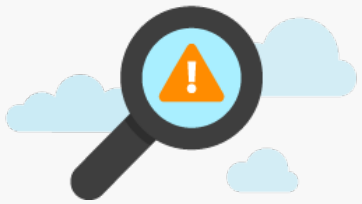
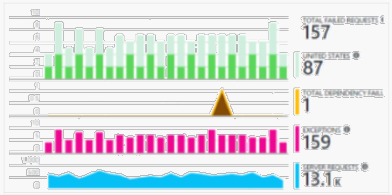
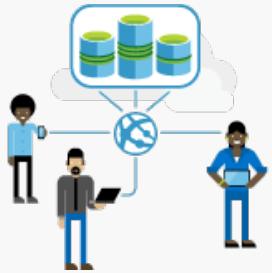
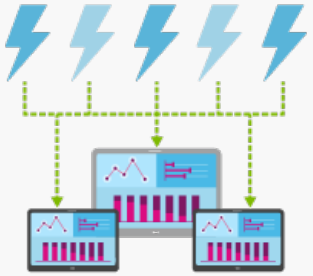
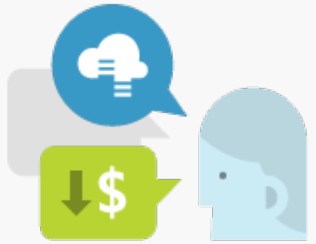
Event Stream Processing

- [Simple event processing] +
- Aggregate
- Rules

Complex Event Processing

- [Event Stream Processing] +
- Pattern detection
- Time windows
- Joins & correlations

Scenario Examples

Real-time Fraud Detection 	Streaming ETL 	Predictive Maintenance 	Call Center Analytics 
IT Infrastructure and Network Monitoring 	Customer Behavior Prediction 	Log Analytics 	Real-time Cross Sell Offers 
Fleet monitoring and Connected Cars 	Real-time Patient Monitoring 	Smart Grid 	Real-time Marketing 

and many more...

Differentiators

Programmer Productivity

Declarative SQL like language
Built-in temporal semantics

Ease of Getting Started

Integrations with sources, sinks, & ML
Build real-time dashboards in minutes

Lowest Total Cost of Ownership(TCO)

Fully managed service
No cluster topology management required
Seamless scalability
Usage based pricing

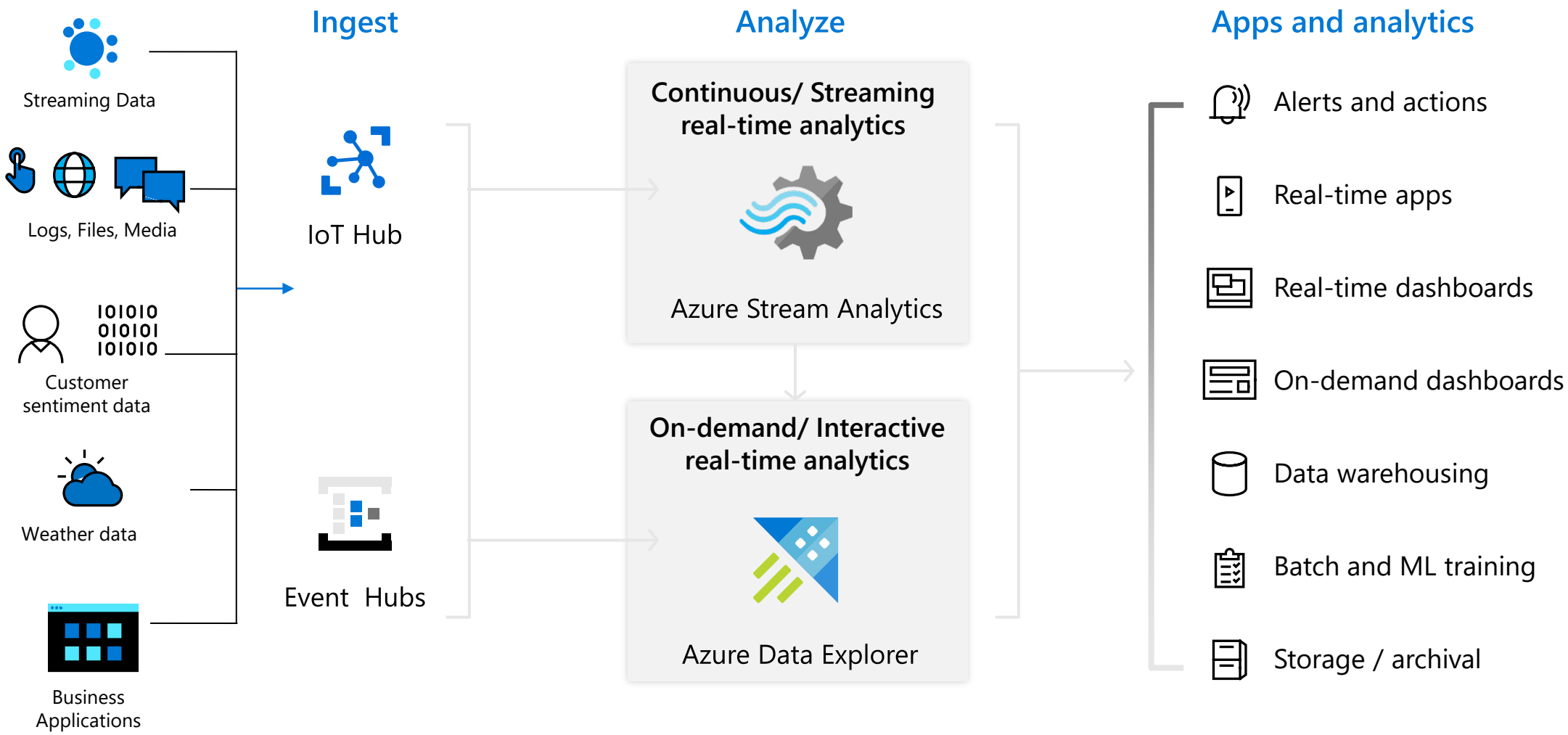
1,915 lines of code with Apache Storm

```
@ApplicationAnnotation(name="WordCountDemo")
public class Application implements StreamingApplication
{
       protected String fileName =
       "com/datatorrent/demos/wordcount/samplefile.txt";
                    private Locality locality = null;

       @Override  public void populateDAG(DAG dag, Configuration
       conf)
       {
              locality = Locality.CONTAINER_LOCAL;
              WordCountInputOperator input =
              dag.addOperator("wordinput", new
              WordCountInputOperator());
              input.setFileName(fileName);
              UniqueCounter<String> wordCount =
              dag.addOperator("count", new
              UniqueCounter<String>());
              dag.addStream("wordinput-count", input.outputPort,
              wordCount.data).setLocality(locality);
              ConsoleOutputOperator consoleOperator =
              dag.addOperator("console", new
              ConsoleOutputOperator());
              dag.addStream("count-console",wordCount.count,
              consoleOperator.input);
       }
}
```

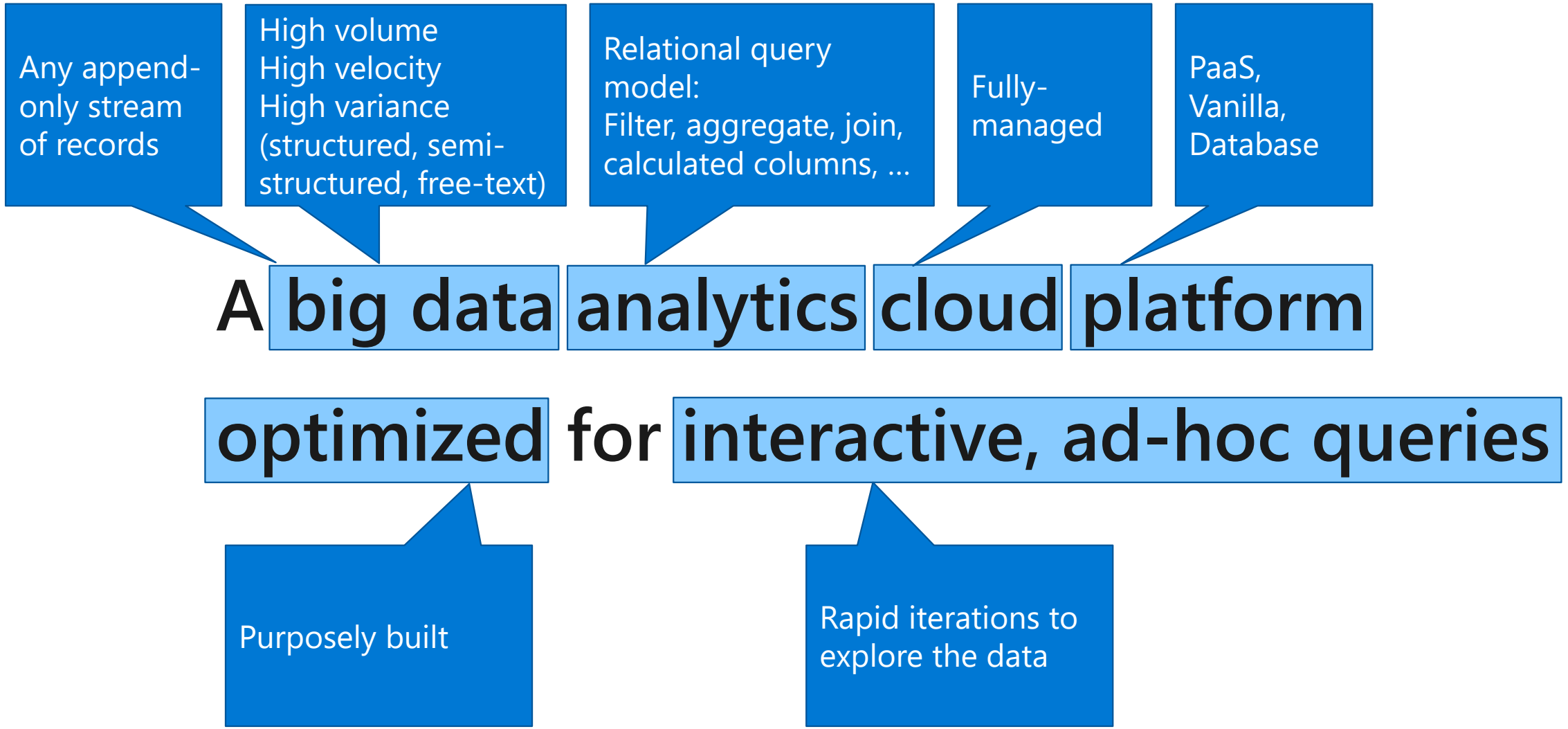
3 lines of SQL in Azure Stream Analytics

```
SELECT Avg(Purchase), ScoreTollId, Count(*)

FROM GameDataStream

GROUP BY TumblingWindows(5, Minute), Score
```

Azure Powered Pattern For Real-Time Analytics



Azure Data Explorer – in a Sentence



Azure Data Explorer
Fast and fully managed data analytics service

Fully managed for efficiency

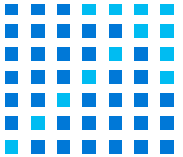
Focus on insights, not the infra-structure for fast time to value

No infrastructure to manage; provision the service, choose the SKU for your workload, and create database.

Optimized for streaming data

Get near-instant insights from fast-flowing data

Scale linearly up to **200 MB per second per node** with highly performant, low latency ingestion**.**



Designed for data exploration

Run ad-hoc queries using the intuitive query language

Returns results from **1 Billion records < 1 second** without modifying the data or metadata

Kusto (kql) basic syntax

```
// Comment
DailyIngestionTableProdNoam
| where Product == 'Teams'
| where EndTime > ago(28d)
| order by Duration
```

- **Table** – Data source
- **Data transformations** – Top to bottom flow through data transformation operators

- **Keyword** – Each Operator has a specific syntax
- **Tabular Operator** – Top to bottom flow through data transformation operators
- **Operator delimiter** – Pipe (|) binds to data transformation steps

Aggregation functions

- `summarize [by]` – Group rows together (see slide on aggregations)

  - `sum()`
  - `avg()`
  - `count()`, `dcount()`
  - `countif()`
  - `max()`, `min()`
  - `percentiles()`
  - Etc. (all here)

Aggregation example

Drink	Price
Juice	6
Juice	8
Juice	9
Milk	3
Milk	4
Milk	5
Milk	3

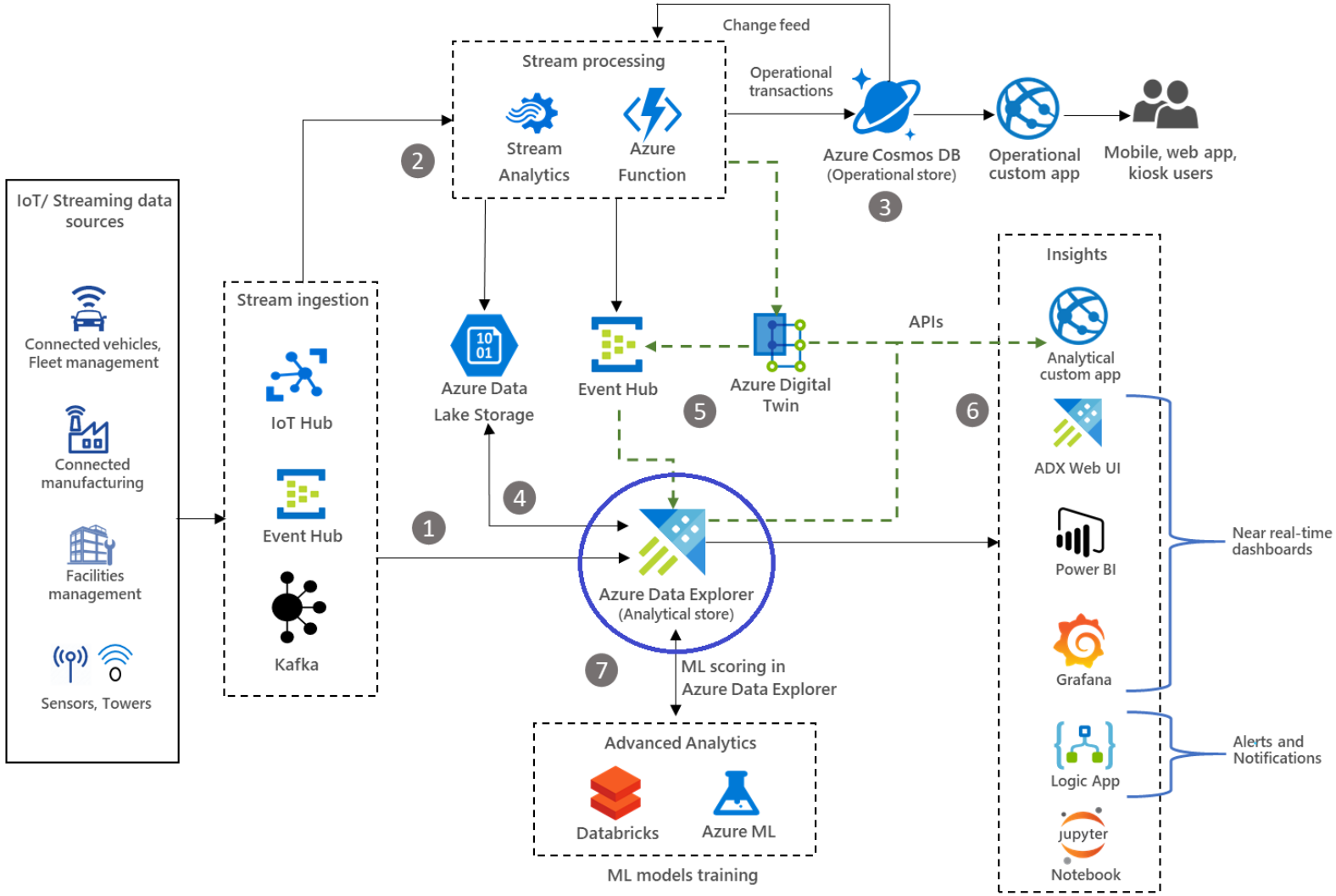
```
| summarize avg_price = avg(Price) by Drink
```

Drink	Average of Price
Juice	7.67
Milk	3.75

```
| summarize count = count() by Drink
```

Drink	Count of Drink
Juice	3
Milk	4

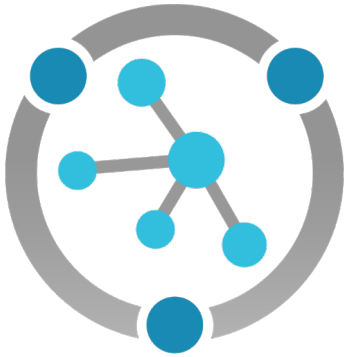
A Closer Look at Solutions Leveraging ASA and ADX



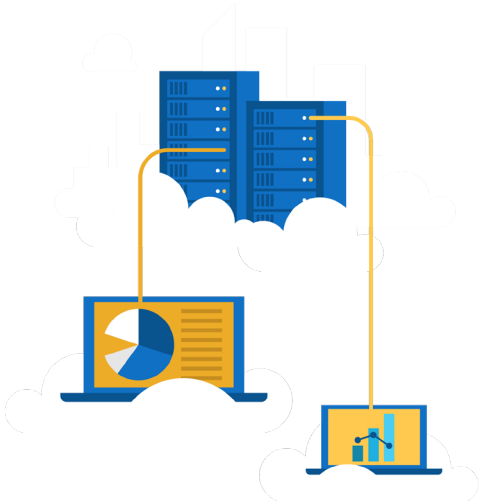
https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/iot-azure-data-explorer

Hands-On Lab

Ingesting Telemetry to IoT Hub
Analyzing Data with Azure Data Explorer
Processing Telemetry with Azure Stream Analytics (ASA)



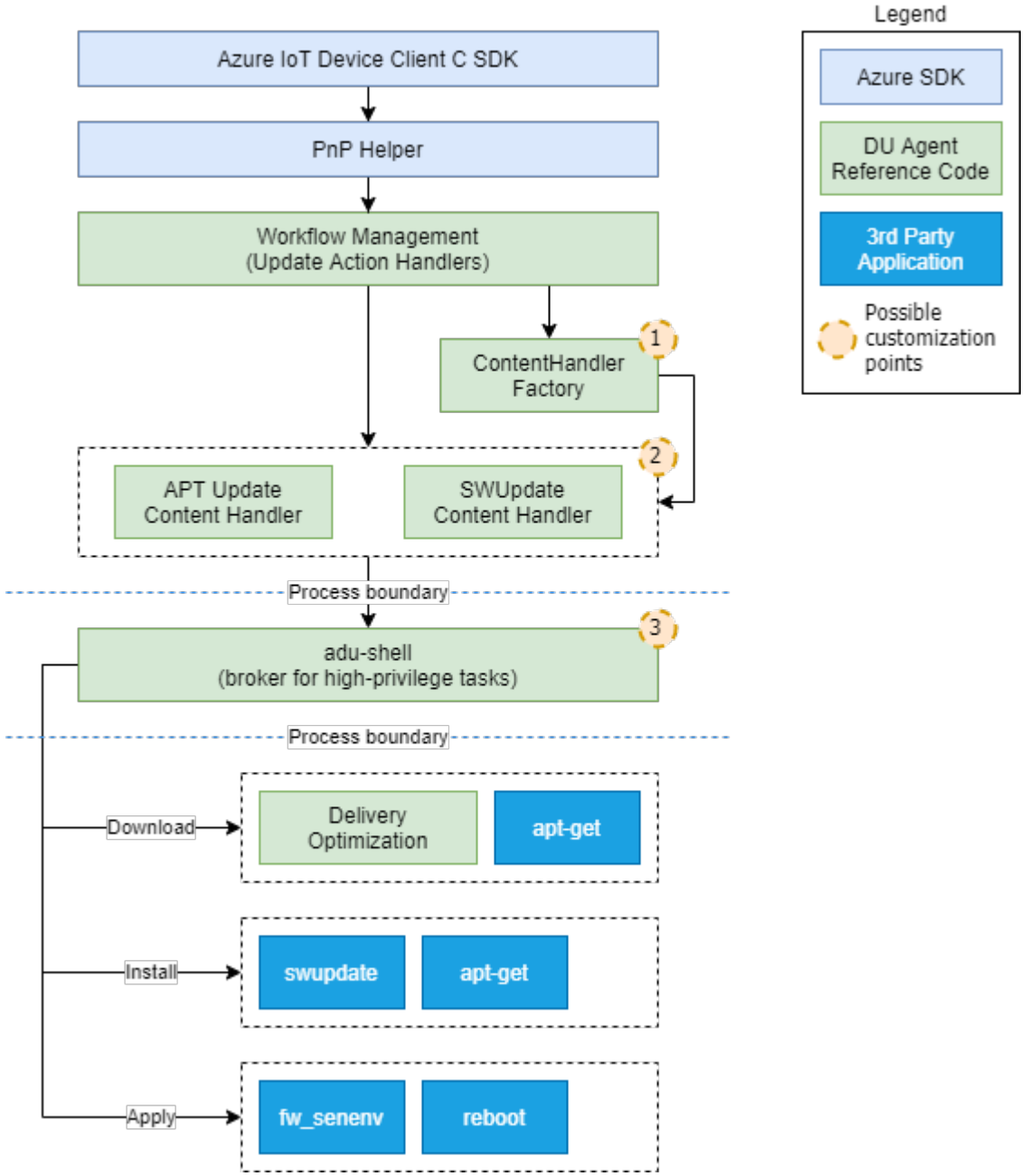
What did we learn?
Send questions to iotacademy@microsoft.com
We start tomorrow at 10:10am ET



Agent Architecture

- https://github.com/Azure/iot-hub-device-update
- APT or Image Handler
  - APT leverages existing package repositories
  - Image delivered via included CDN
  - BASH Scripts
- Delivery Optimization
  - Uses port 80 and can be proxied
- IoT Hub Protocol and supports Nested Edge
- Uses module of device identity in IoT Hub and twin properties
- On Linux, uses the Identity Service

```
apt install aziot-identity-service
apt install deviceupdate-agent
apt install deliveryoptimization-plugin-apt
```



Update Manifest Version 4

- Enables the following features:
  - Multi Step Ordered Execution (MSOE)
  - Multi Component Updating
  - Goal State Deployment
  - Detached Update Manifest
- https://github.com/Azure/iot-hub-device-update/blob/main/docs/agent-reference/update-manifest-v4-schema.md

Multicomponent Updating - Proxy Updates

- Targeting specific update files to different apps/components on the device
- Targeting specific update files to sensors connected to IoT devices over a network protocol (e.g., USB, CANbus etc.).
- https://github.com/Azure/iot-hub-device-update/blob/main/docs/agent-reference/multi-component-updating.md

Multi-Step Ordered Execution (MSOE) Support

- Multi-Step Ordered Execution (MSOE) Support
- Parent and child updates
- Pre and post install steps
- https://github.com/Azure/iot-hub-device-update/blob/main/docs/agent-reference/update-manifest-v4-schema.md

Update Types

- Built in UpdateTypes
  - microsoft/apt:1
  - microsoft/swupdate:1
- Update Content Handler extension and custom UpdateTypes
- https://github.com/Azure/iot-hub-device-update/blob/main/docs/agent-reference/update-manifest-v4-schema.md

Further Reference

- https://github.com/Azure/iot-hub-device-update/tree/main/docs/agent-reference
- https://github.com/Azure/iot-hub-device-update/blob/main/docs/agent-reference/whats-new.md
- https://github.com/Azure/iot-hub-device-update/blob/main/docs/agent-reference/how-to-build-agent-code.md

Azure Stream Analytics Data Flow

- An Azure Stream Analytics *job* consists of an *input*, *query*, and an *output*
- Input – ASA can ingest data from Azure Event Hubs, Azure IoT Hub, or Azure Blob Storage
- Query – ASA uses a SQL-like query language that includes support for filtering, sorting, aggregating, joining, and user-defined functions
- Output – ASA can output to many targets

Direct Methods: Introduction

- *Direct methods* – request from the cloud to a device, executing code directly on the target
- Features
  - Each call targets a single device or module instance
  - Can be used by anyone with appropriate IoT Hub permissions
  - Follow a request-response pattern for immediate feedback
- Lifecycle
  - Called by a back-end application through an HTTPS URL pattern on the IoT Hub
  - Translated to MQTT or AMQP on the device side
  - Reply received from the device sent directly back to the back-end application

Comparing Device Management Approaches

	Direct Method Call	Device Twins	Cloud-to-device messages
Scenario	Requires immediate confirmation	Long-running desired state configuration	One-way notifications.
Data flow	Two-way with immediate response	One-way to the device	One-way to the device
Durability	Disconnected devices are not contacted. The solution back end is notified that the device is not connected.	Property values are preserved in the device twin. Device will read it at next reconnection. Property values are retrievable with the IoT Hub query language.	Messages can be retained by IoT Hub for up to 48 hours.
Targets	Single device using deviceId, or multiple devices using jobs.	Single device using deviceId, or multiple devices using jobs.	Single device by deviceId.
Size	Payload maximum is 128 KB.	Desired properties maximum is 8 KB.	Up to 64 KB messages.
Frequency	High	Medium	Low
Protocol	MQTT or AMQP.	MQTT or AMQP.	MQTT, AMQP, HTTPS