

Methven  
Canterbury, New Zealand

# Azure IoT Cloud Workshop

Eric van Uum

Microsoft GBB WE – IoT Technology Specialist  
[eric.vanuum@microsoft.com](mailto:eric.vanuum@microsoft.com)



Weather

51 °



27 MPH  
Wind  
43%  
Humidity

SUN	MON	TUE	WED	THU	FRI	SAT
54°   38°	55°   40°	58°   32°	54°   30°	55°   36°	59°   42°	58°   44°



# Topics guideline

Title	Content
Overview	Welcome and introductions Microsoft Cloud Vision Internet of Things Overview
Azure IoT solutions	Azure SaaS and PaaS solutions
Azure IoT Hub Deep Dive	Azure IoT Reference Architecture and typical Architecture Examples from the real world IoT Hub Deep Dive (SDKs, device provisioning, routing, security & operations, IoT Edge)
Azure for IoT	Azure Services used in IoT
Advanced Analytics and Visualization	Getting from device data to insights with Azure Data and Machine Learning solutions
Hands-on	Hands-on Lab
Closing	Wrap up, next steps and Q&A

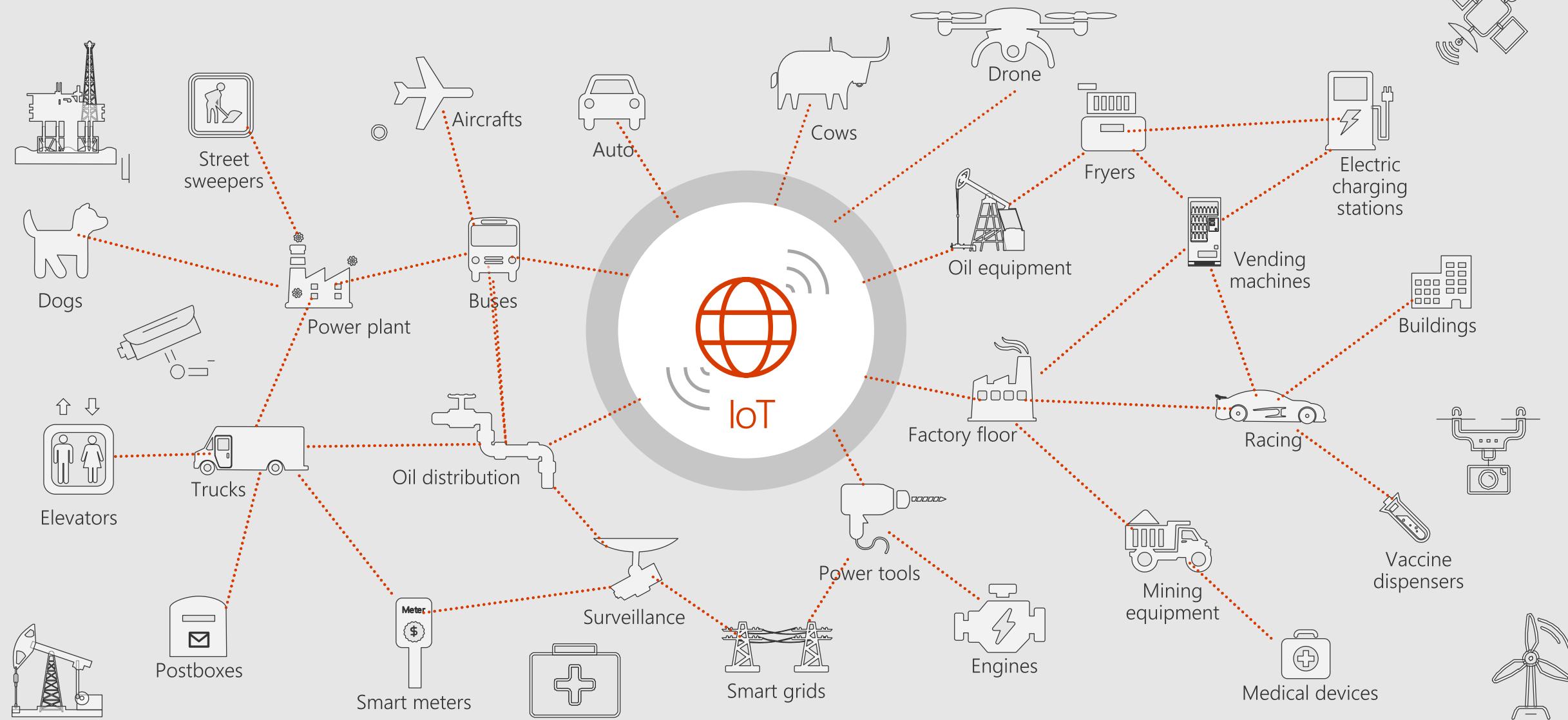
# Introduction

## *Microsoft is simplifying IoT*

# IoT Solutions Have a Common Pattern



# The IoT Pattern is Broadly Applicable Across Industries



# IoT projects can be complex



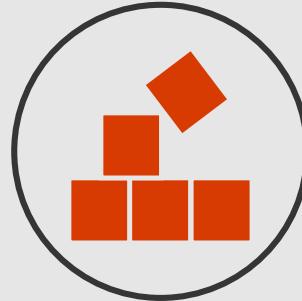
Difficult to maintain cohesive **security**



Time-consuming to get started



Incompatible with existing infrastructure



Challenging to scale over time

# Microsoft is simplifying IoT

Easier to build secure, scalable solutions from device to cloud

Easier to provision devices at scale

Easier to manage devices at scale

Easier to find insights from your IoT devices

Easier to infuse devices with intelligence

Easier to benefit from IoT

# Microsoft is simplifying IoT

## Azure IoT Suite

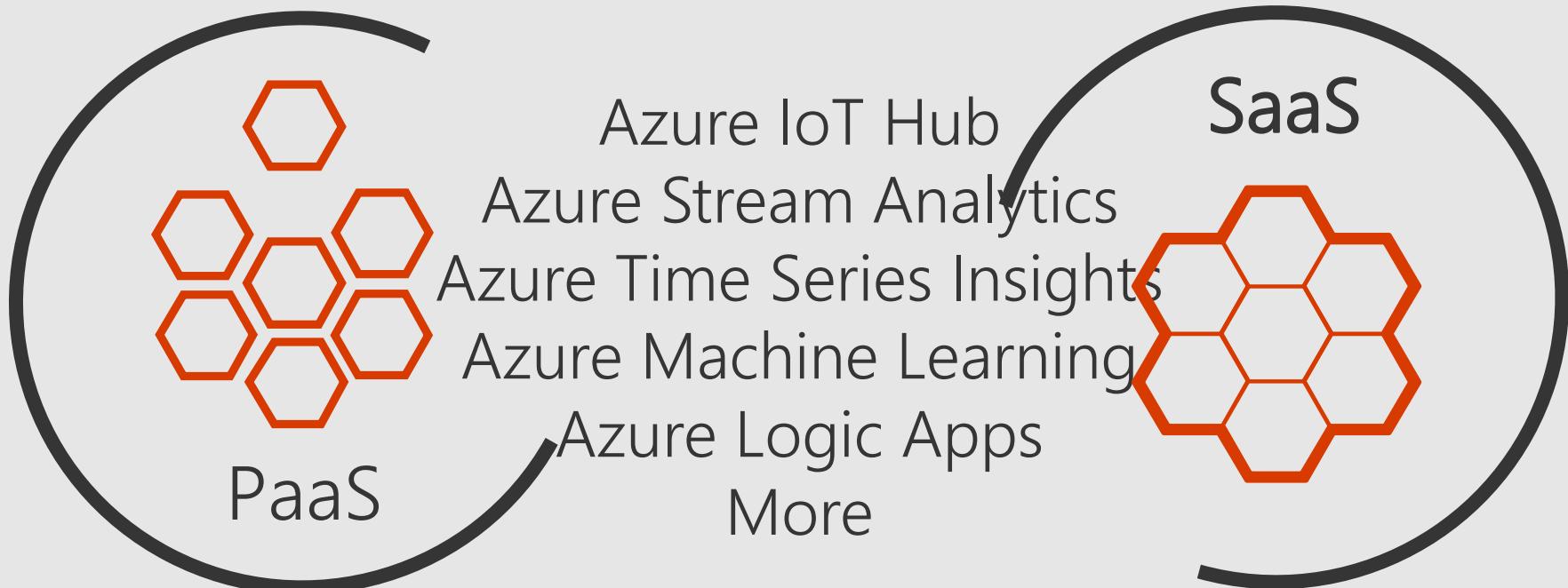
Preconfigured solutions for common IoT scenarios



Remote Monitoring | Predictive Maintenance | Connected Factory

## Microsoft IoT Central

Fully managed IoT SaaS  
No cloud solution expertise required



# Choosing Between Approaches



Azure IoT Suite

**Primary usage**

Custom solutions that need maximum flexibility



Microsoft IoT  
Central

Straightforward IoT solutions that don't require deep service customization

# Azure IoT Suite



Device Connectivity & Management



Data Ingestion and Command & Control



Stream Processing & Predictive Analytics



Workflow Automation and Integration



Dashboards and Visualization



Preconfigured Solutions

Remote monitoring

Predictive maintenance

Connected factory



# Microsoft IoT Central



Device Connectivity & Management



Telemetry Ingestion and Command & Control



Monitoring Rules & Triggered Actions



User roles and permissions



Dashboards, Visualization & Insights



Fully Hosted & Managed by Microsoft



# Microsoft is simplifying IoT

## Azure IoT Suite

Preconfigured solutions for common IoT scenarios



Remote Monitoring | Predictive Maintenance | Connected Factory

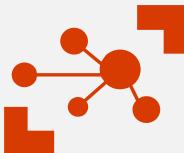
## Microsoft IoT Central

Fully managed IoT SaaS  
No cloud solution expertise required



## Azure IoT Hub

IoT cloud gateway, secure, bi-directional communication with billions of devices sending trillions of messages



## Azure IoT Hub Device Provisioning Service

Fully managed service for securely provisioning devices at scale



# Azure IoT Hub



## Bi-directional communication

Millions of Devices

Multi-language, open source SDKs

HTTPS/AMQPS/MQTTs

Send Telemetry

Receive Commands

Device Management

Device Twins

Queries & Jobs



## Enterprise scale & integration

Billions of messages

Scale up and down

Declarative Message Routes

File Upload

WebSockets & Multiplexing

Azure Monitor

Azure Resource Health

Configuration Management



## End-to-End Security

Per Device Certificates

Per Device Enable/Disable

TLS Security

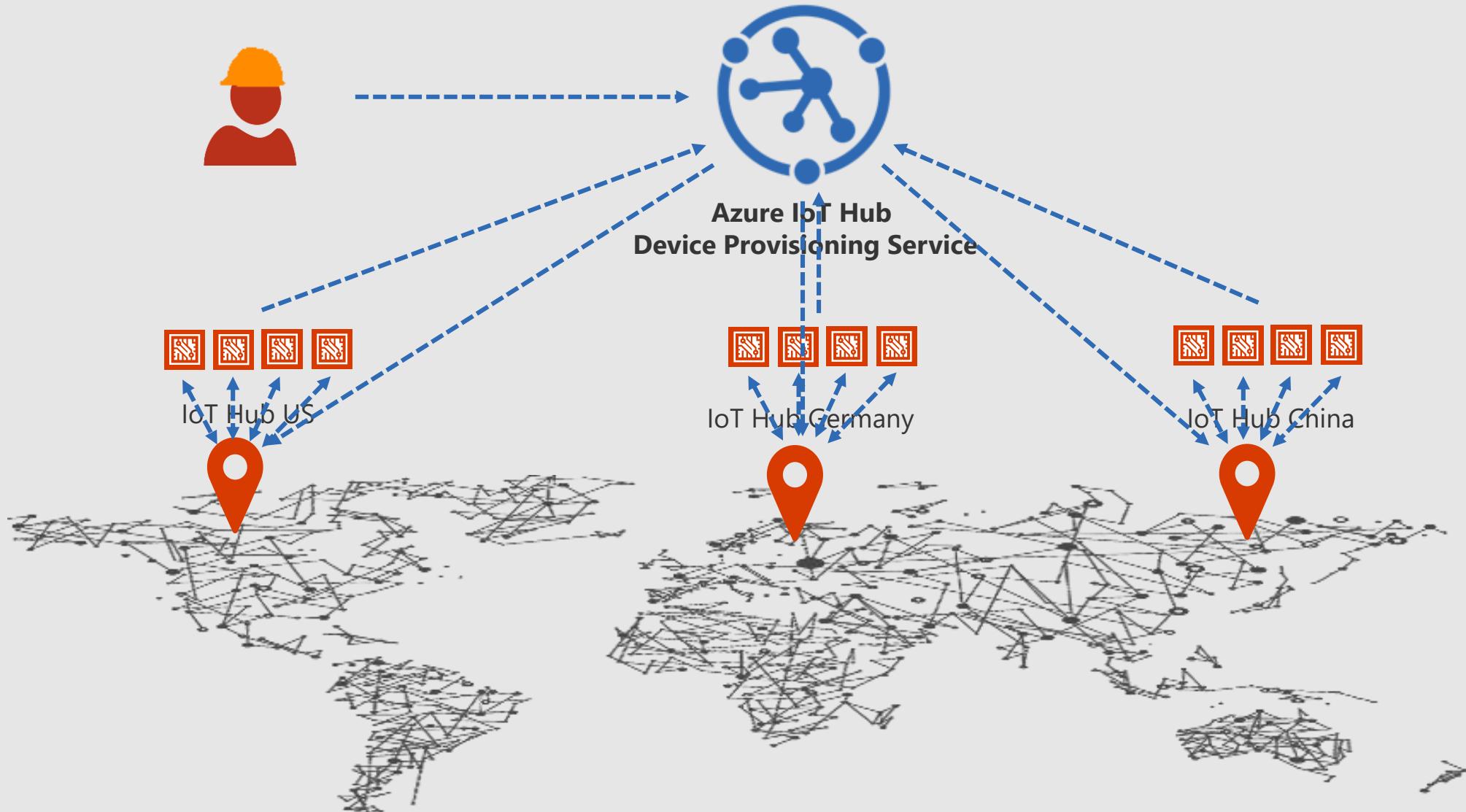
X.509 Support

IP Whitelisting/Blacklisting

Shared Access Policies

Firmware/Software Updates

# Azure IoT Hub Device Provisioning Service



# Microsoft is simplifying IoT

## Azure IoT Suite

Preconfigured solutions for common IoT scenarios



Remote Monitoring | Predictive Maintenance | Connected Factory

## Microsoft IoT Central

Fully managed IoT SaaS  
No cloud solution expertise required



## Azure IoT Hub

IoT cloud gateway, secure, bi-directional communication with billions of devices sending trillions of messages



## Azure IoT Hub Device Provisioning Service

Fully managed service for securely provisioning devices at scale



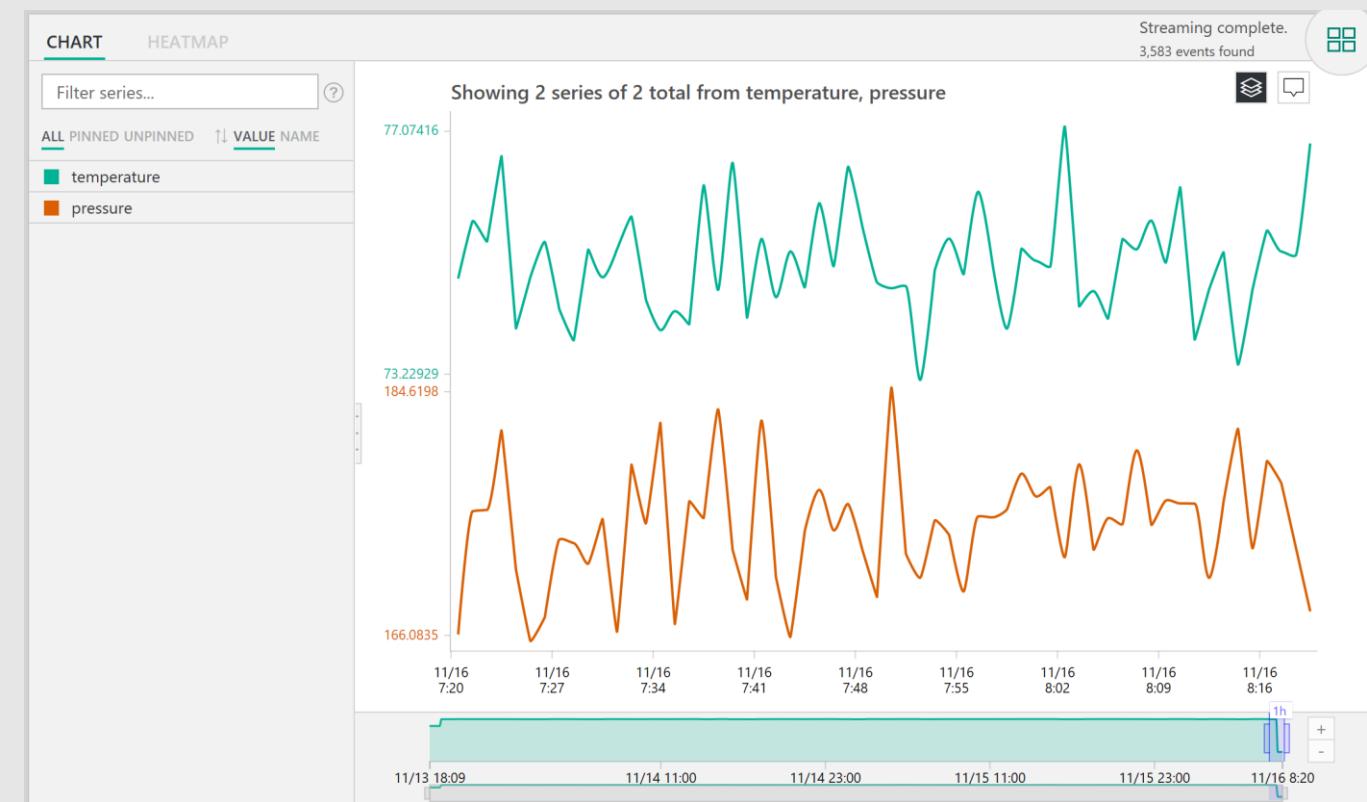
## Azure Time Series Insights

Explore and analyze time series data fast, and at scale with a fully managed offering



# Azure Time Series Insights

- IoT scale time-series data store
- Schema-less store, just send data
- Easy IoT Hub connection
- Store, query and visualize billions of events
- Simple and fast navigation



# Microsoft is simplifying IoT

## Azure IoT Suite

Preconfigured solutions for common IoT scenarios



Remote Monitoring | Predictive Maintenance | Connected Factory

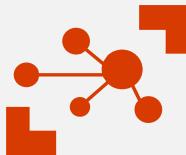
## Microsoft IoT Central

Fully managed IoT SaaS  
No cloud solution expertise required



## Azure IoT Hub

IoT cloud gateway, secure, bi-directional communication with billions of devices sending trillions of messages



## Azure IoT Hub Device Provisioning Service

Fully managed service for securely provisioning devices at scale



## Azure Time Series Insights

Explore and analyze time series data fast, and at scale with a fully managed offering



## Azure IoT Edge

Securely distribute cloud intelligence locally, and at scale



# Waves of Innovation

Cloud

Globally available, unlimited compute resources

IoT

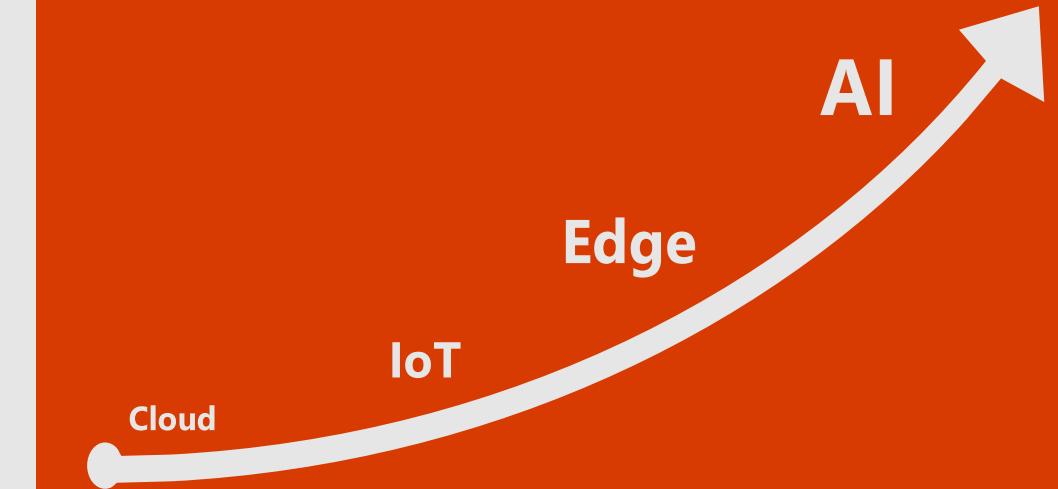
Harnessing signals from sensors and devices, managed centrally by the cloud

Edge

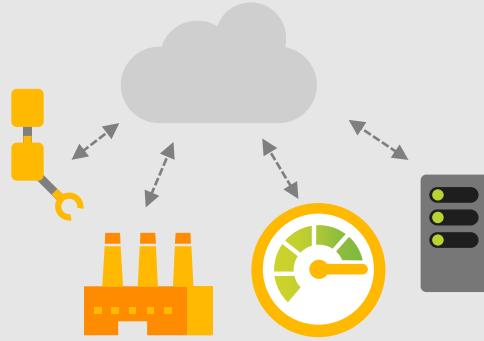
Intelligence offloaded from the cloud to IoT devices

AI

Breakthrough intelligence capabilities, in the cloud and on the edge



# 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

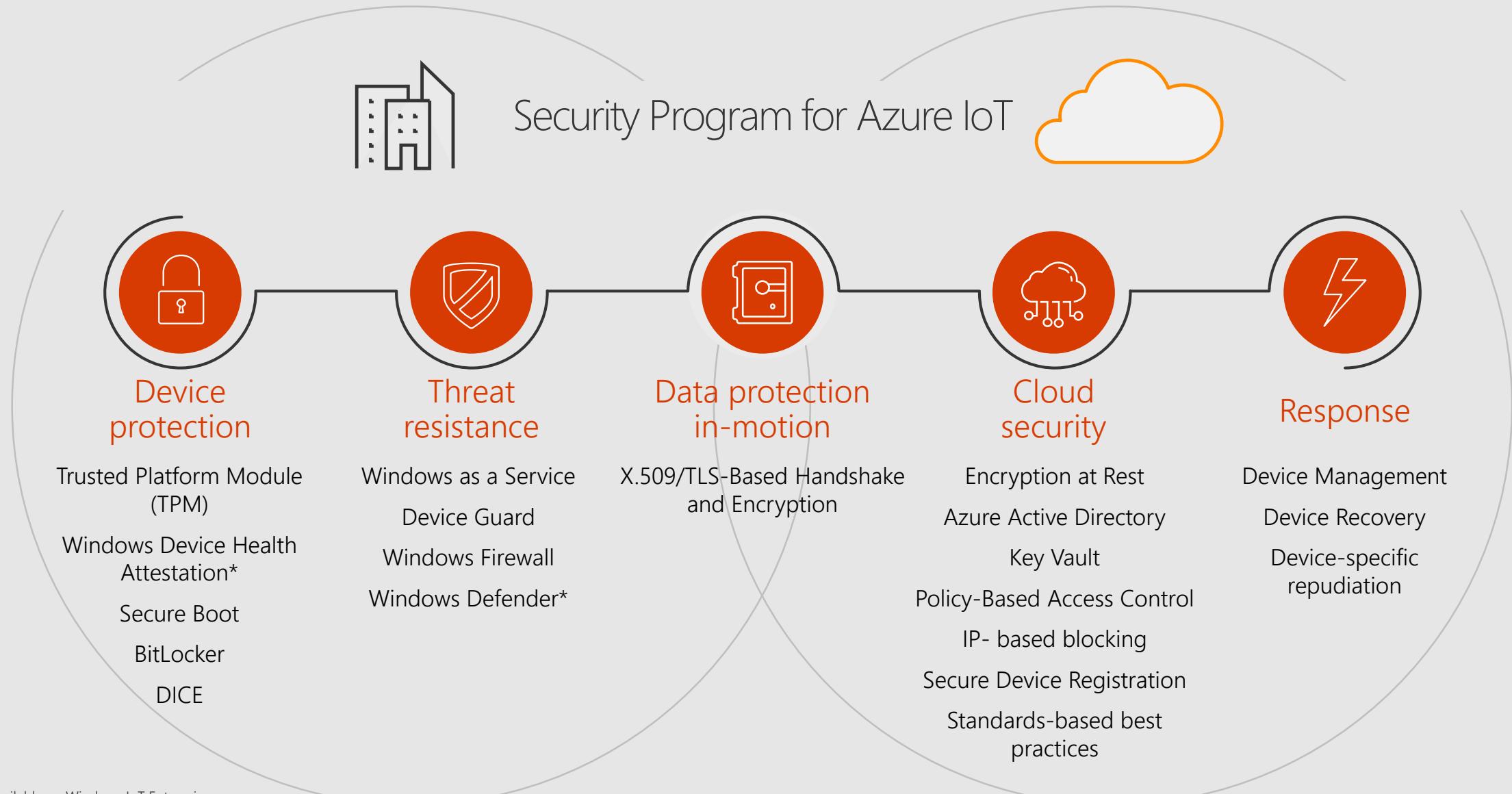
Low latency tight control loops require near real-time response

Protocol translation & data normalization

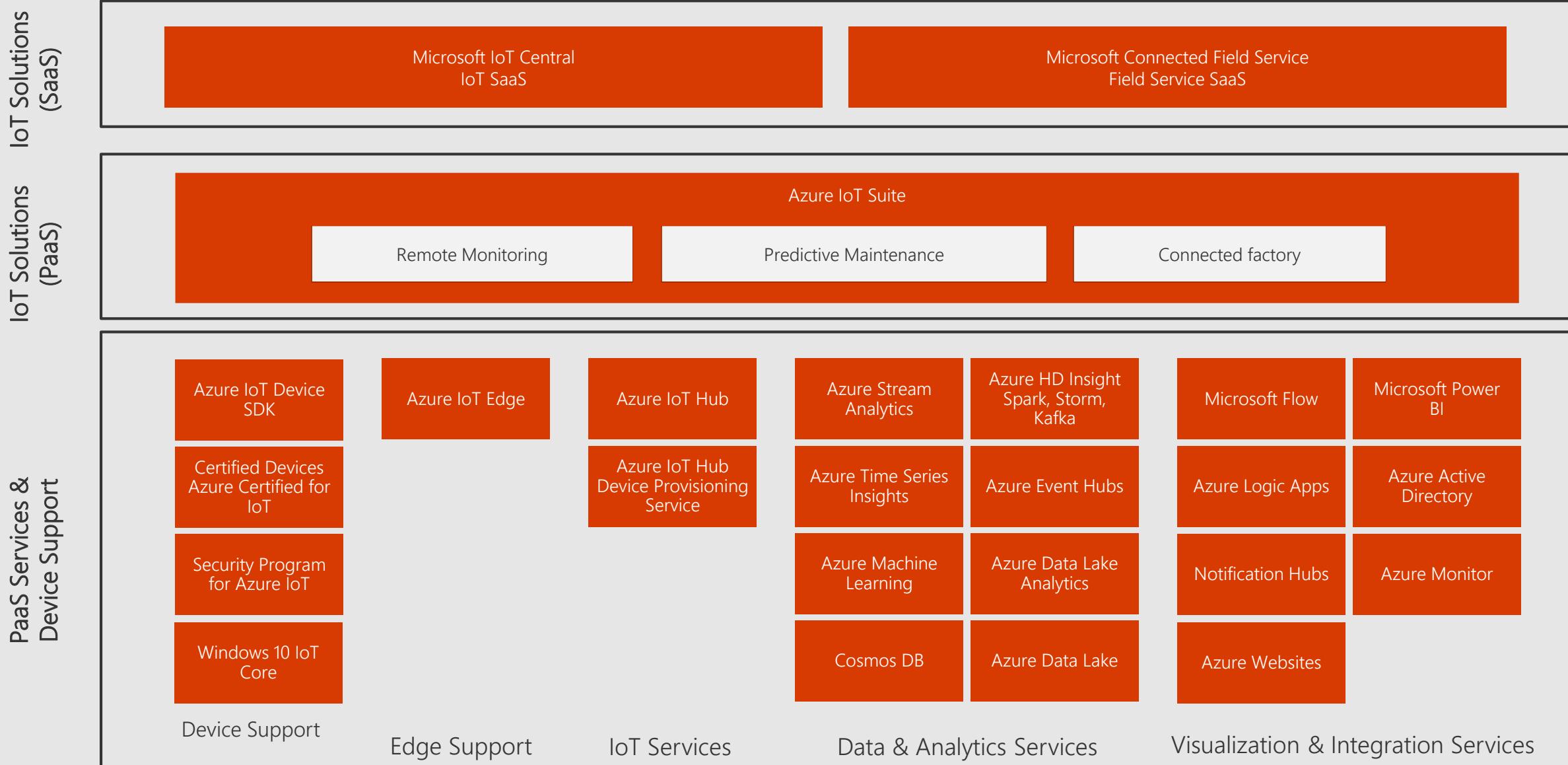
Privacy of data and protection of IP

Symmetry

# Device to Cloud Security

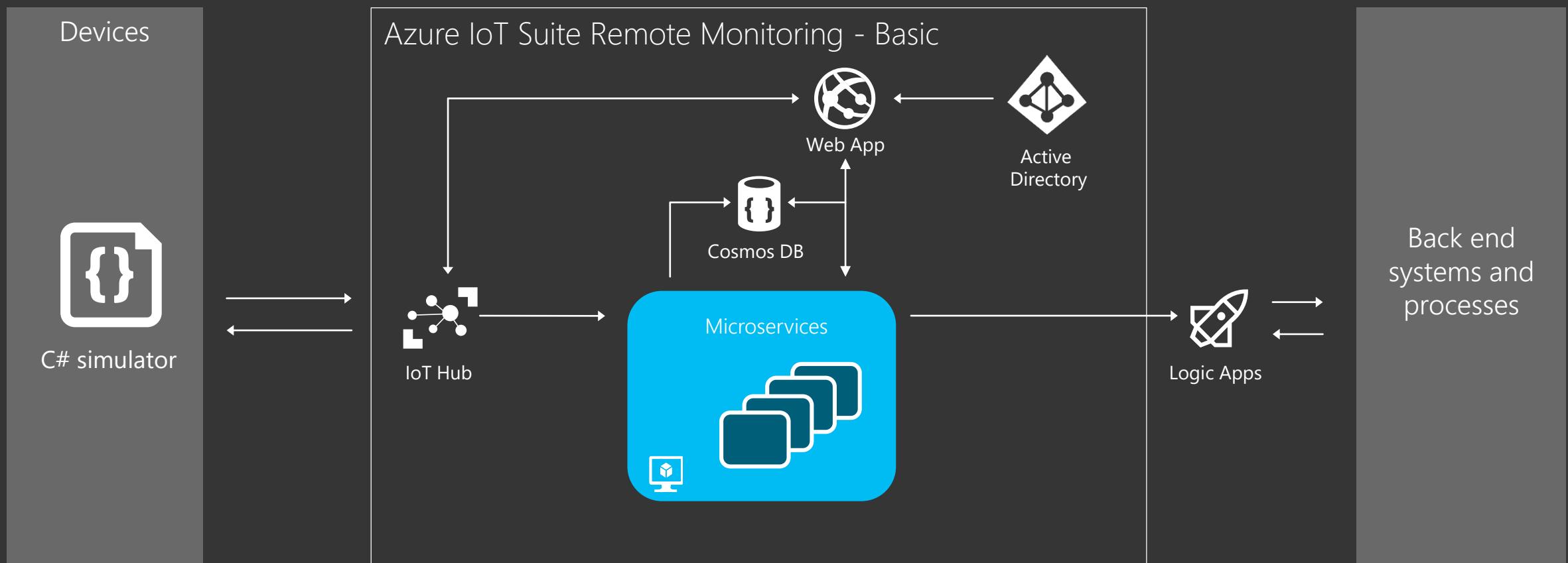


# A Comprehensive set of offerings for IoT

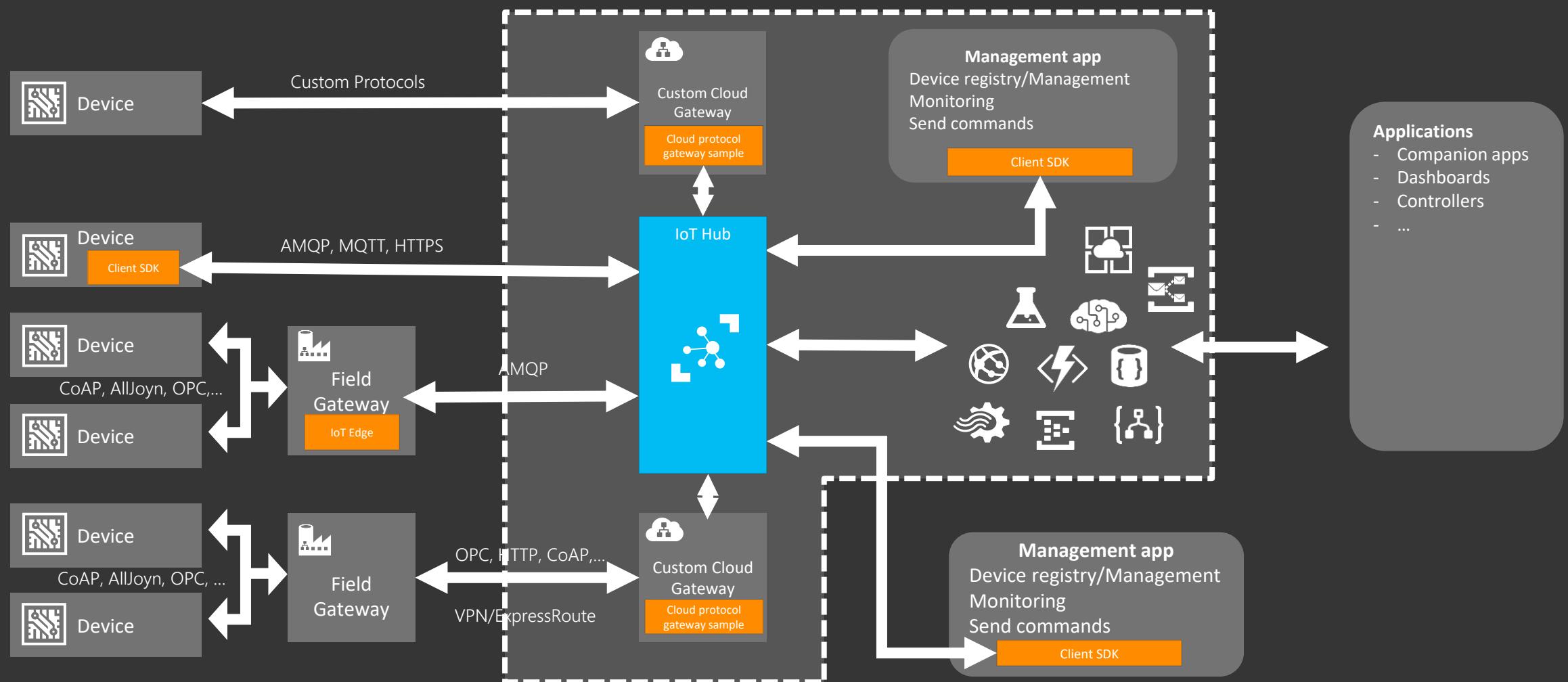


# Get Started Developing with Azure IoT Suite

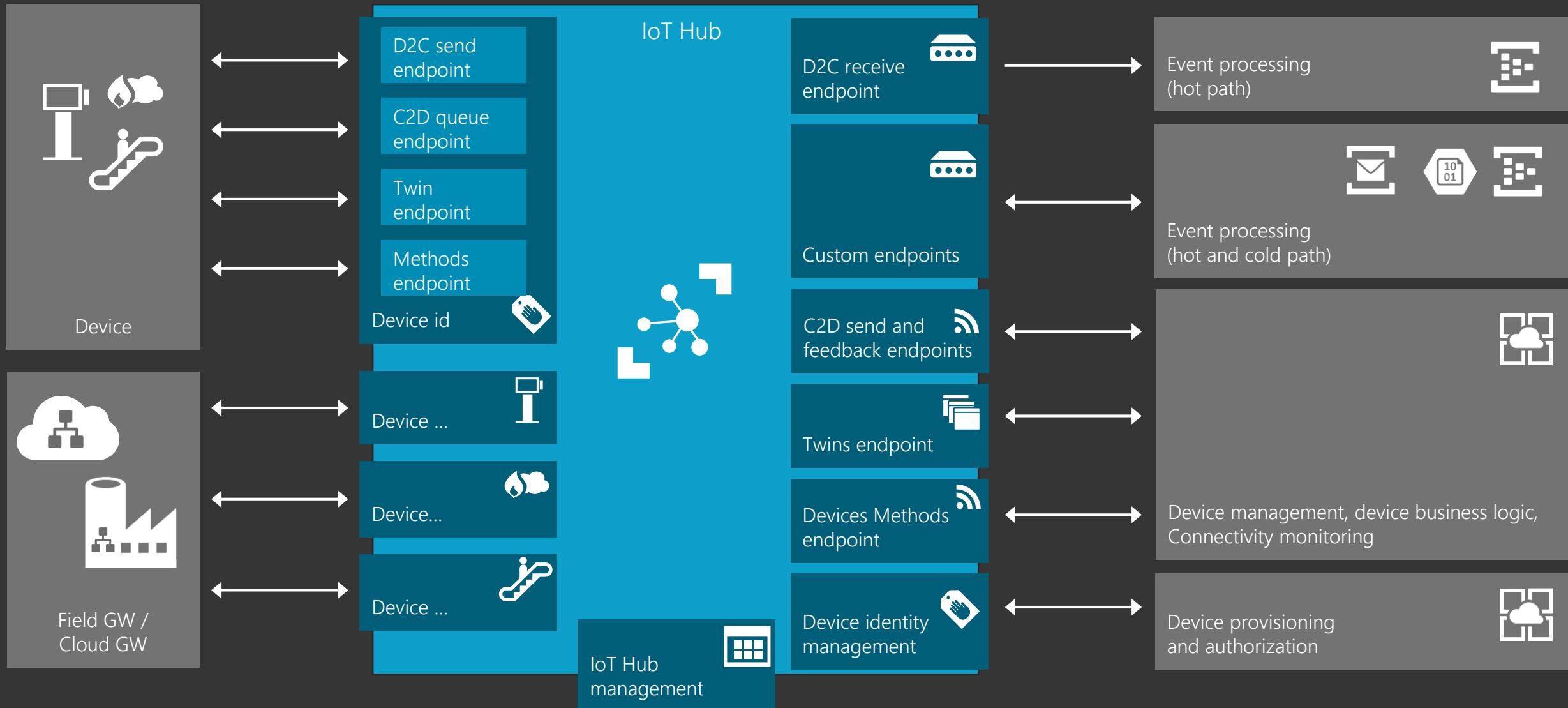
# Example of an IoT solution



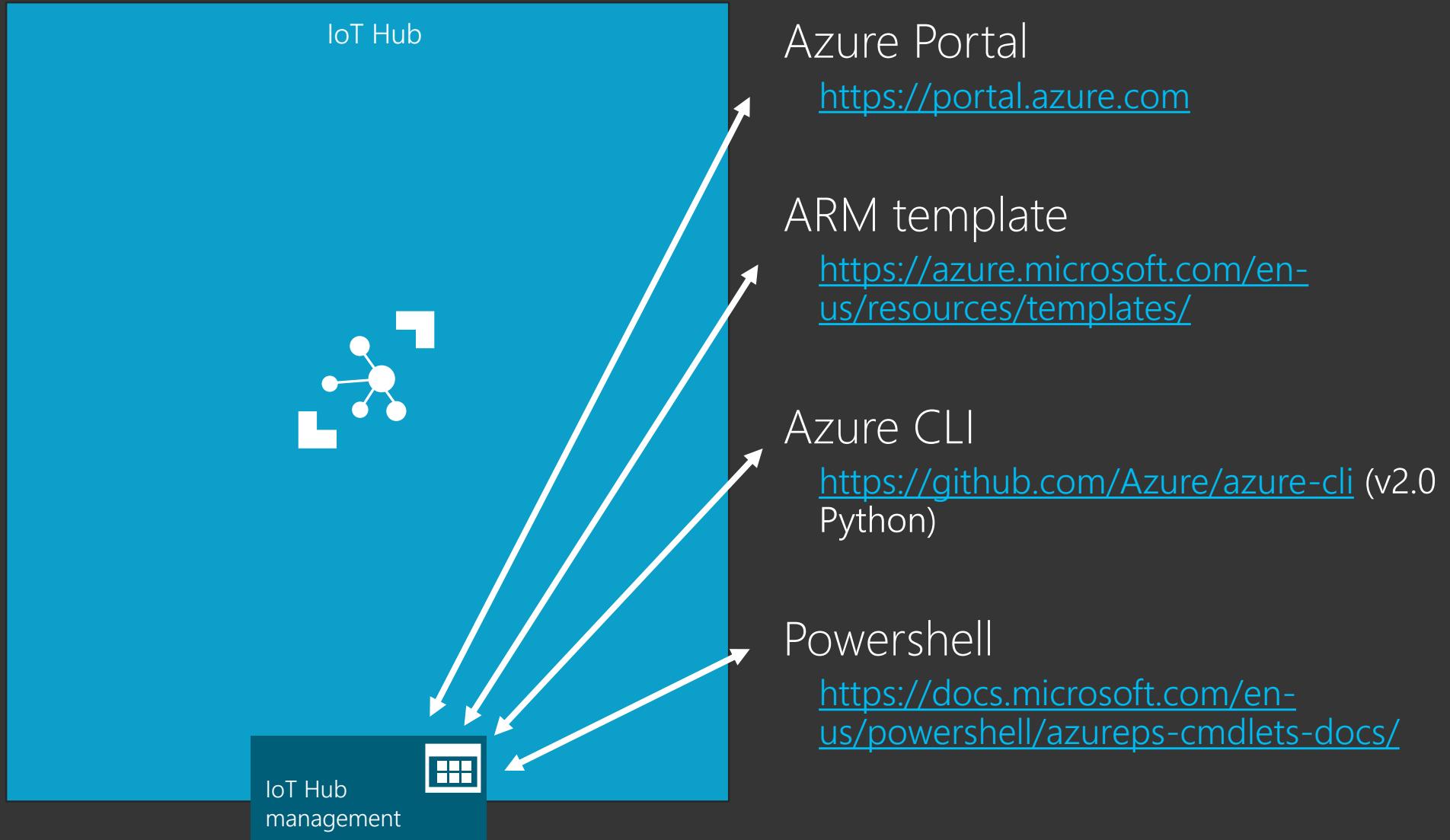
# Azure IoT Hub



# IoT Hub ABC



# Pick your favorite to create a hub



All devices

LAST REFRESHED | 09:10:23 09.26.2017

Manage filters

## Dashboard

Devices

Rules &amp; Actions

Maintenance

## Device locations

MORE

# All devices

DEVICES

2.

Critical

2.

Warnings

10

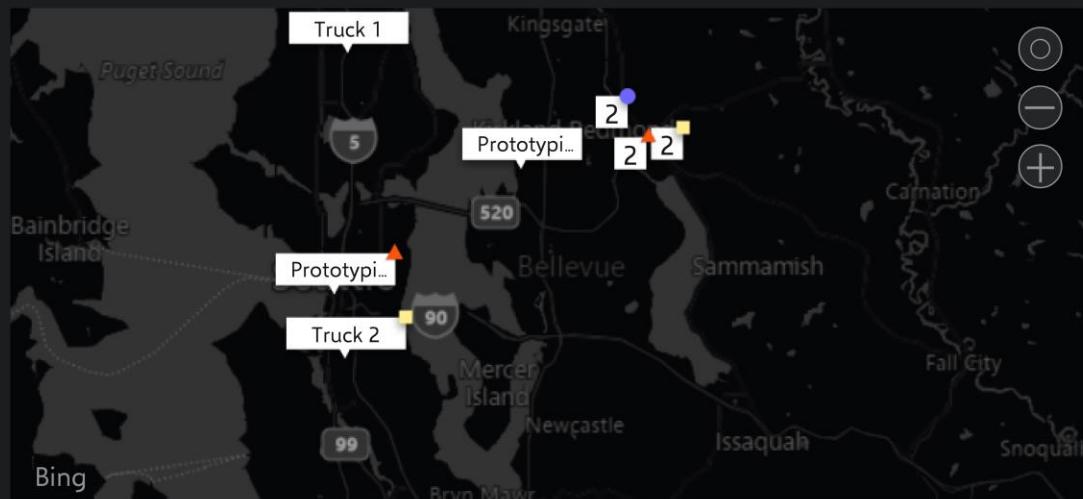
10

0

Total

Online

Offline



## Telemetry

MORE

Pressure [4]

Humidity [2]

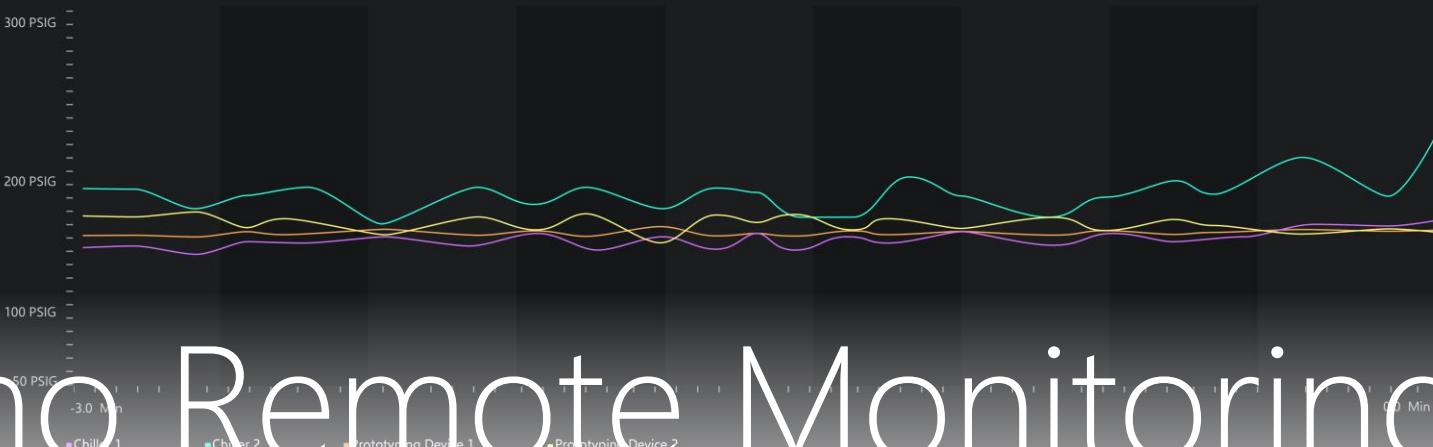
Temperature [6]

Movement [2]

Vibration [4]

Speed [3]

Tank fuel lev



# Demo Remote Monitoring

## System alarms

Last week

RULE NAME	SEVERITY	LAST INCIDENT	OPEN OCCURRENCES	EXPLORE ALARM
High Pressure	▲ Critical	08:10:04 09.26.20	1	...
Low Fuel	● Info	08:56:45 09.26.20	1	...
High Temp	▲ Critical	05:56:33 09.25.20	1	...
Hot Truck	■ Warning	11:20:10 09.25.20	1	...
Stopped Elevator	■ Warning	23:13:23 09.24.20	1	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

## System KPI

Last month

Top rules triggered



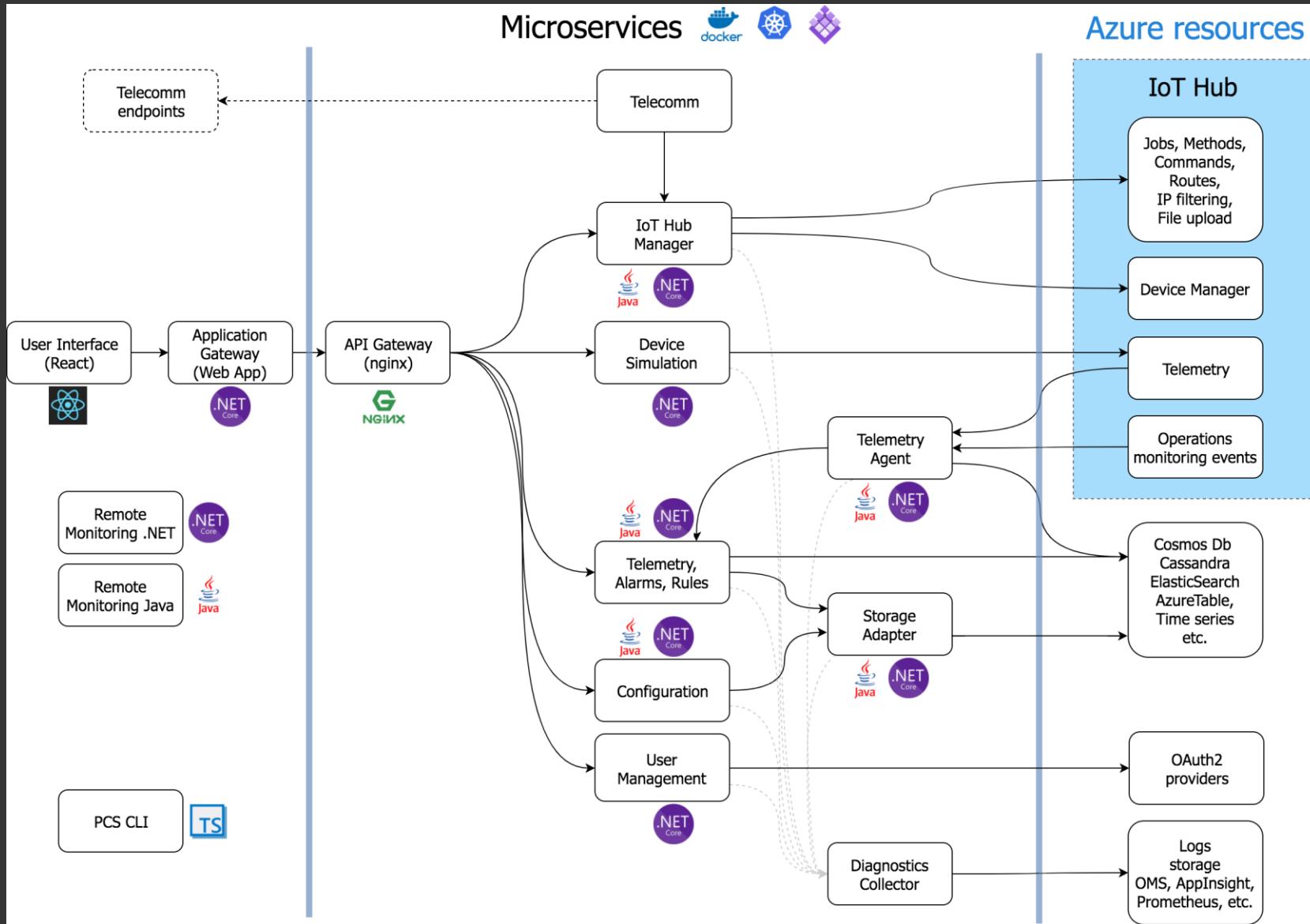
Alarms by device type



Critical alarms



# PCS Remote Monitoring (preview)

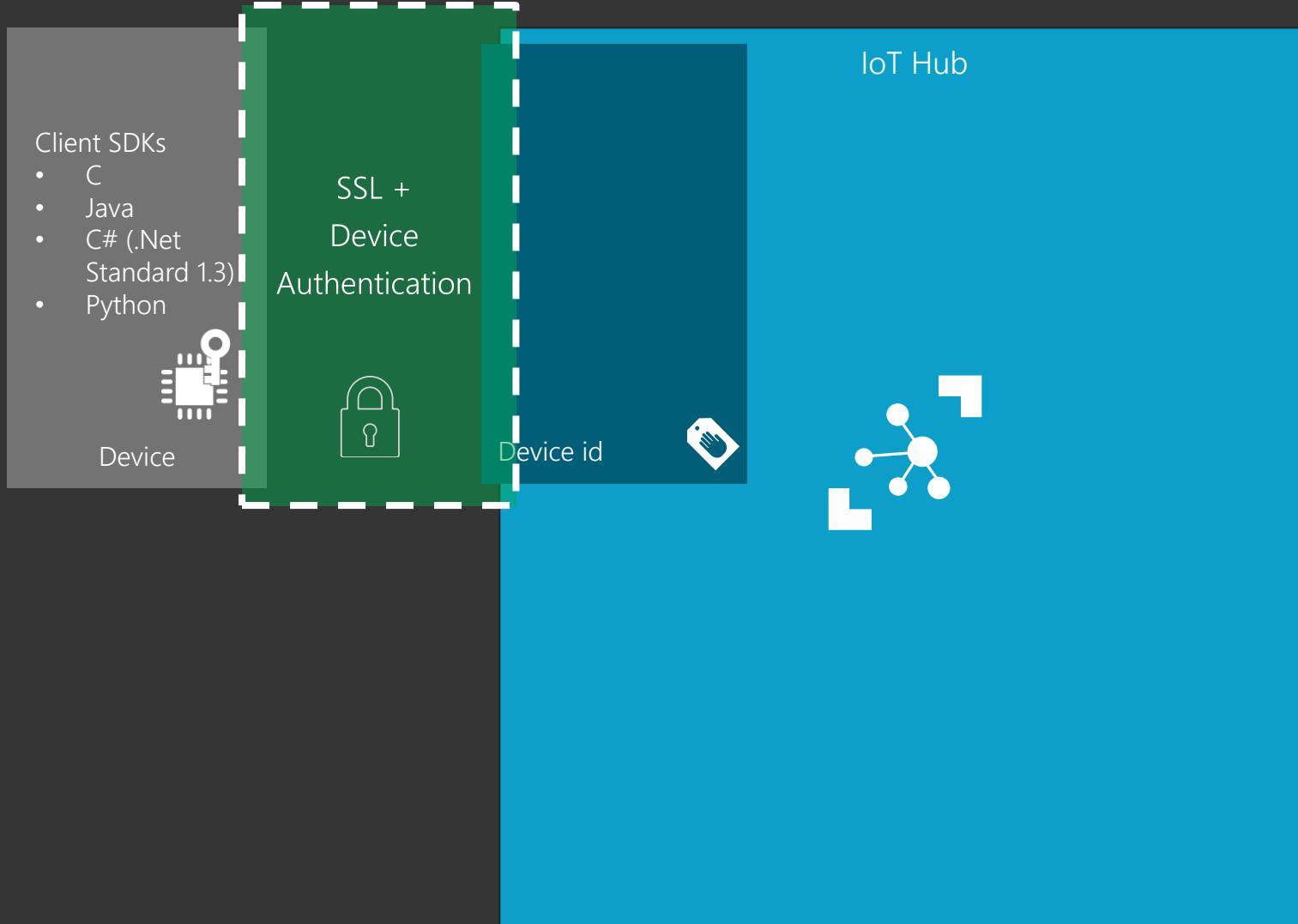


Lab 1:  
*Deploy remote monitoring solution and do a walk-through*

# Lab 1 Steps

1. Go to <http://www.azureiotsuite.com> and create a new RM solution (preview). *[Provision the solution in North Europe.]*
2. Download the open source code from github:  
<https://github.com/Azure/azure-iot-pcs-remote-monitoring-dotnet> and extract it onto your computer.
3. Do a walk-through of the deployed RM on the Azure portal.
4. Open the solution in Visual Studio (Code) and do a walk-through.

# Security first!



Private Key -> SAS Token

x.509 Certificate

- Self Signed
- CA

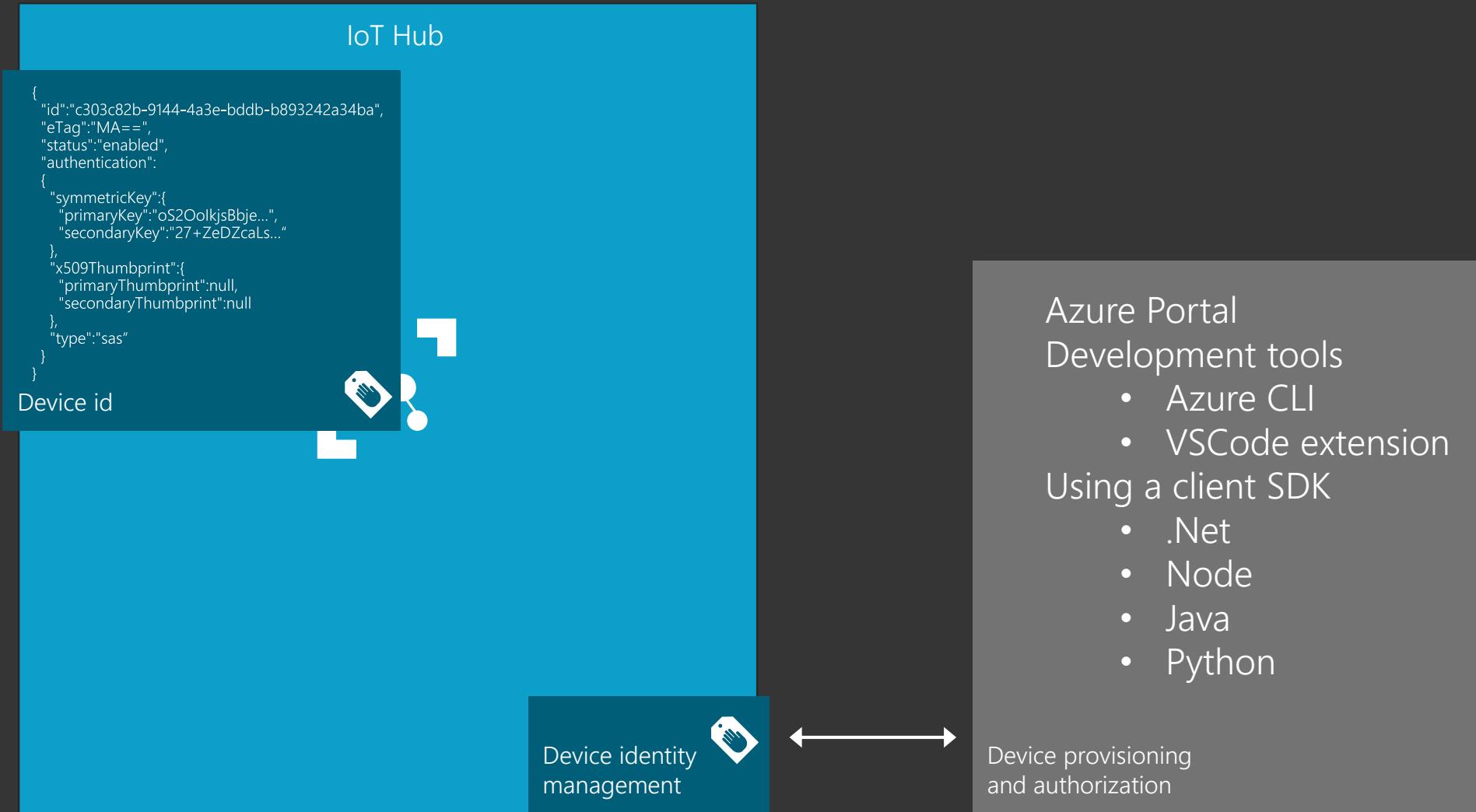
# Device registry

Unique id for each device

Unique credentials for authentication

- Private Key/SAS Token
- X.509 Certificates

Device Twin



# Demo

Creating and managing device identities

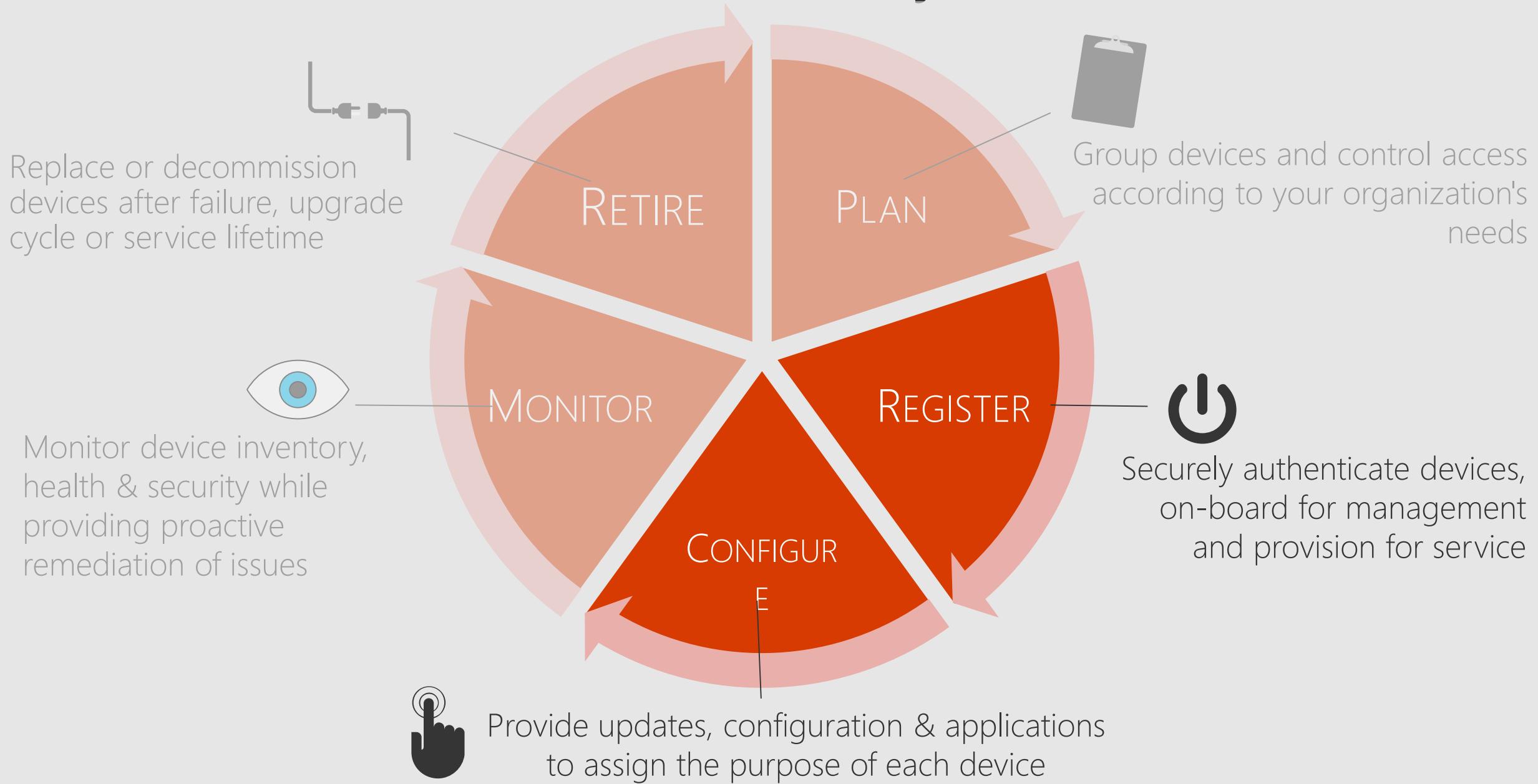
# Lab 2:

*Create a physical device and connect it to the  
Remote Monitoring solution*

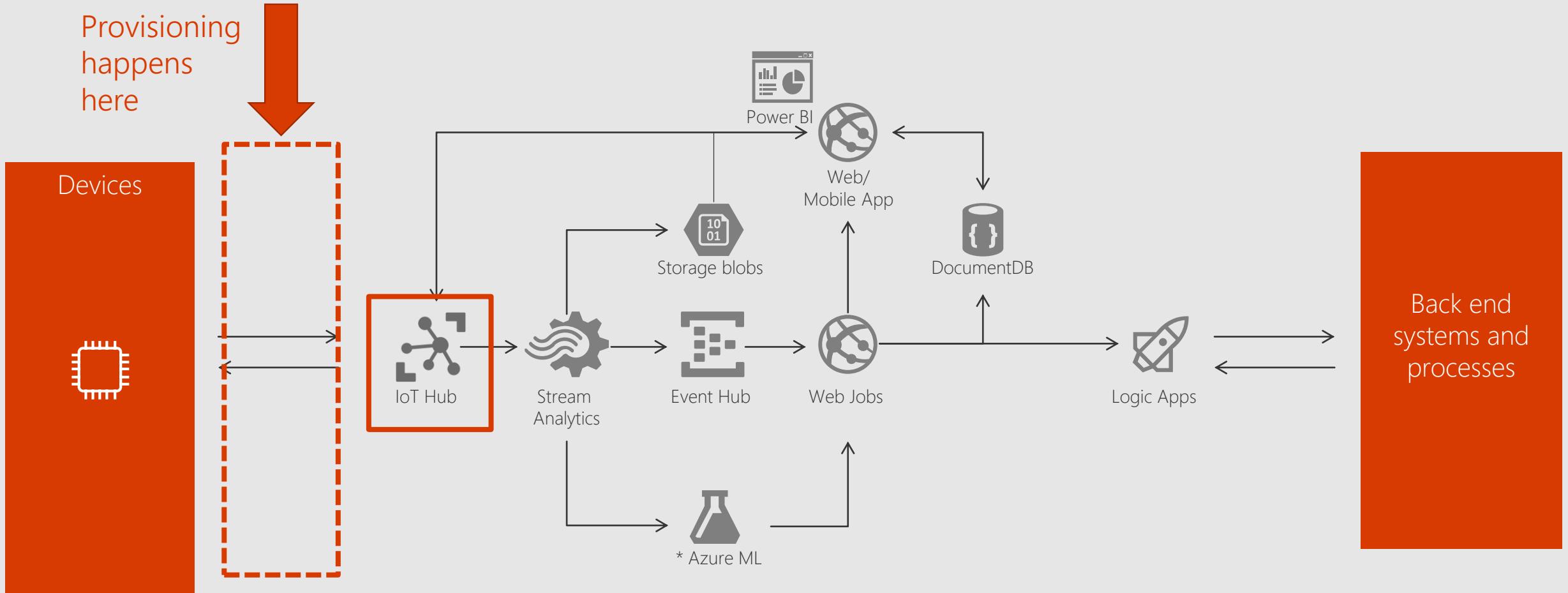
# Lab 2 Steps

1. Add a new physical device in the Remote Monitoring solution.
2. Create the physical device [Ubuntu VM, with nodejs & npm]:  
<https://docs.microsoft.com/en-us/azure/iot-suite/iot-suite-connecting-devices-node>
3. Run the physical device and see the result in the Remote Monitoring solution.
4. Use iothub-explorer [<https://github.com/azure/iothub-explorer>] to see the events sent to IoT-Hub
5. *Extra*: customize the Remote Monitoring solution and deploy it using CLI and Docker Hub: <https://github.com/Azure/azure-iot-pcs-remote-monitoring-dotnet/wiki/Developer-Reference-Guide#customization>.

# IoT Device Lifecycle



# Quick orientation



# What is provisioning?



# Why provisioning is hard today



Solutions must have per-device revocable access



Provisioning is a manual process



Initial configuration can become irrelevant between manufacturing and deployment



Mergers, acquisitions, and bankruptcies can orphan devices



Device supply chains are complex

# Azure IoT Hub Device Provisioning Service

Simplify with zero touch provisioning

Supports multiple locations

Easiest way to mass-provision devices

URL stability

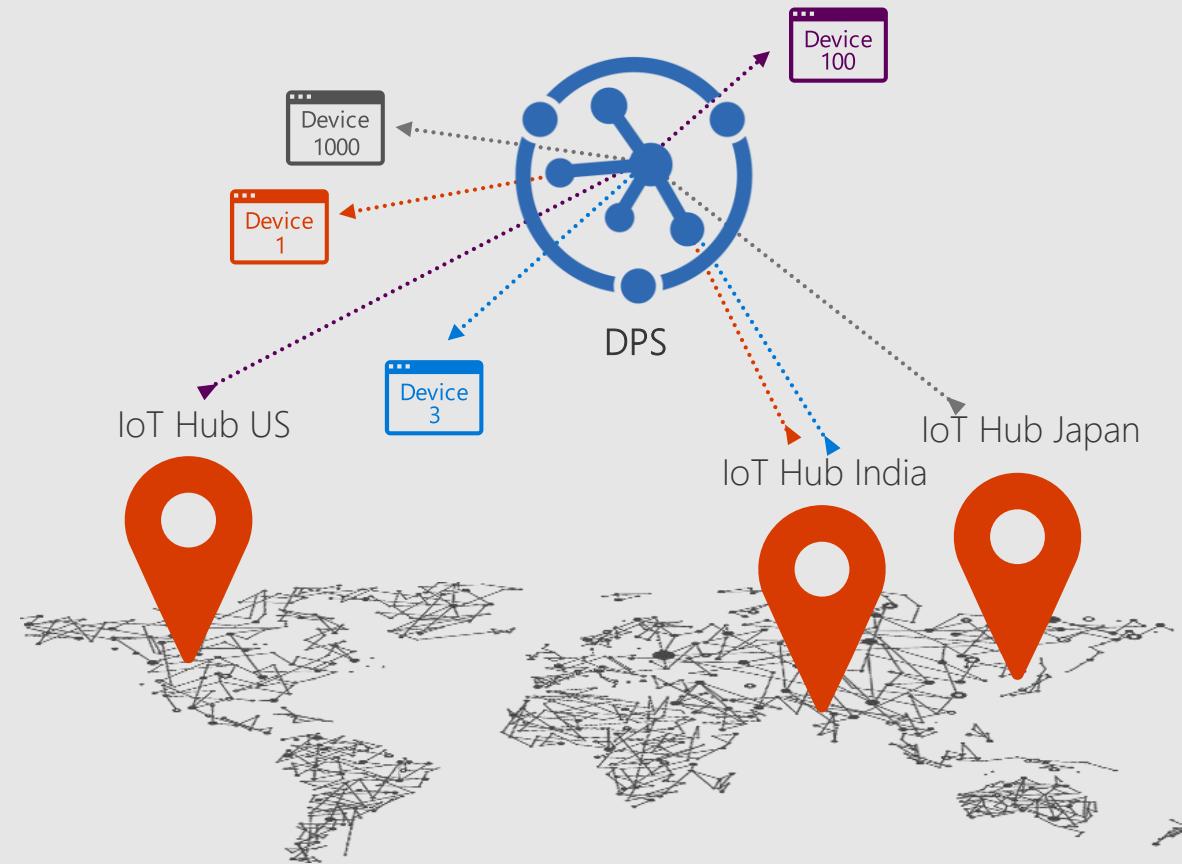
Enhanced security through HSM

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

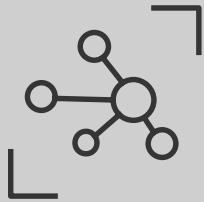
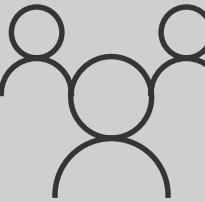
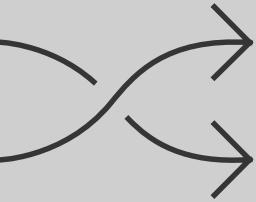
A single DPS can provide service for multiple IoT hubs (in multiple regions)

## Flexible device assignment

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



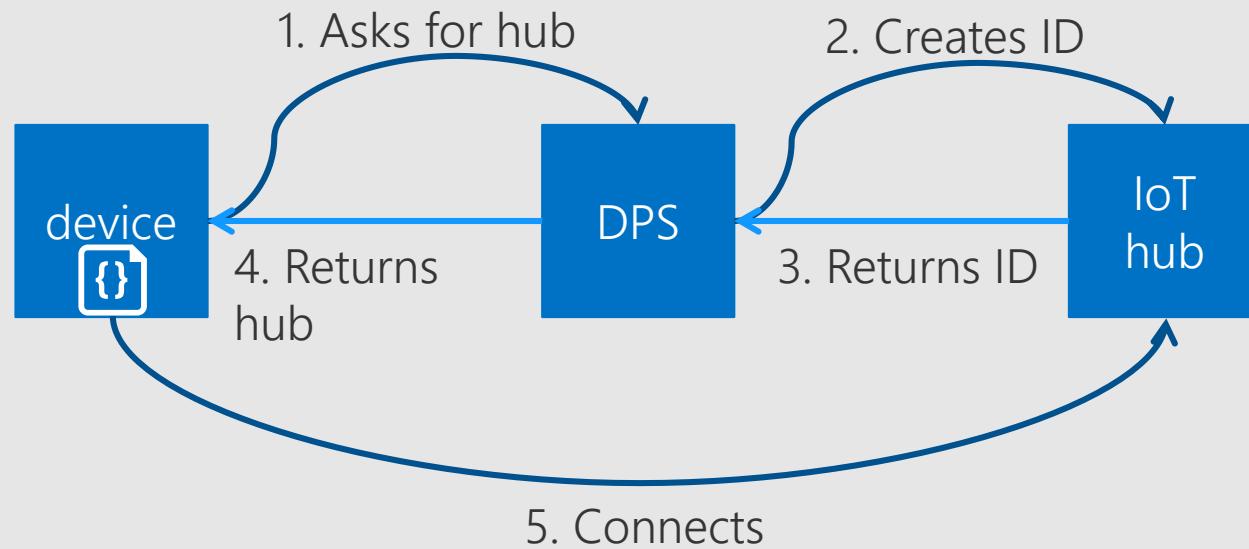
# A selection of scenarios

Initial connection	Load balancing	Ownership based	Location based	Re-provisioning
Zero-touch provisioning to a single IoT solution 	Across multiple hubs 	Connecting devices to their owner's IoT solution based on sales transaction data 	Connecting a device to the IoT hub with the lowest latency 	Based on a change in the device, e.g. change of ownership 

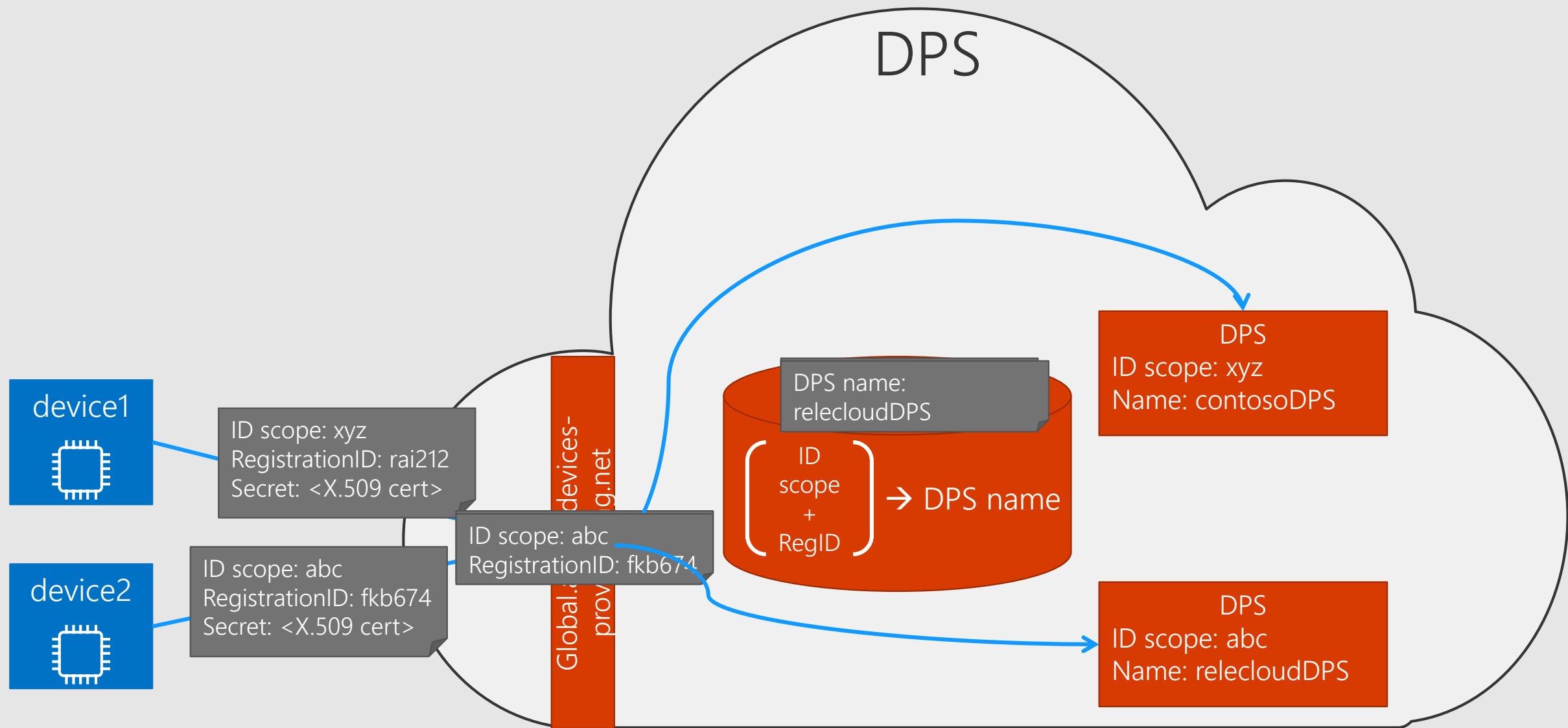
# An IoT device's relationship to DPS

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

# High level provisioning



# Using a global device endpoint



# Demo

Device Provisioning Service

# Lab 3:

## *Individual Device Provisioning*

# Lab 3 Steps

1. Clone the Azure IoT SDK C to your PC:

```
git clone https://github.com/Azure/azure-iot-sdk-c.git --recursive
```

2. Create the Visual Studio solution for the provisioning client (you need cmake on your PC).

```
cd azure-iot-sdk-c
```

```
mkdir cmake
```

```
cd cmake
```

```
cmake -Duse_prov_client:BOOL=ON ..
```

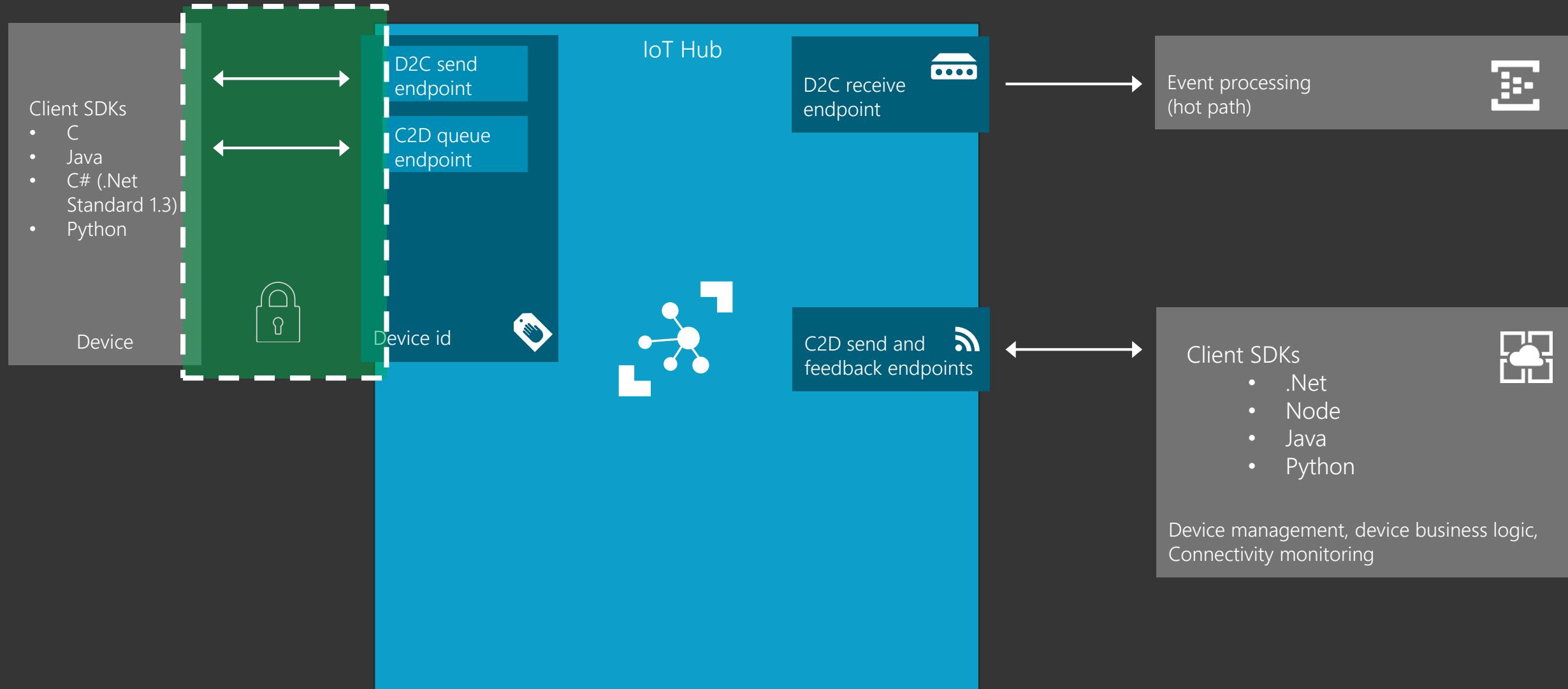
3. Create the X.509 Cert file

*Use: <https://docs.microsoft.com/en-us/azure/iot-dps/quick-create-simulated-device-x509#create-a-device-enrollment-entry-in-the-device-provisioning-service>*

4. Simulate first boot sequence for the device

*Use: <https://docs.microsoft.com/en-us/azure/iot-dps/quick-create-simulated-device-x509#simulate-first-boot-sequence-for-the-device>*

# Send/receive data



# Structured data

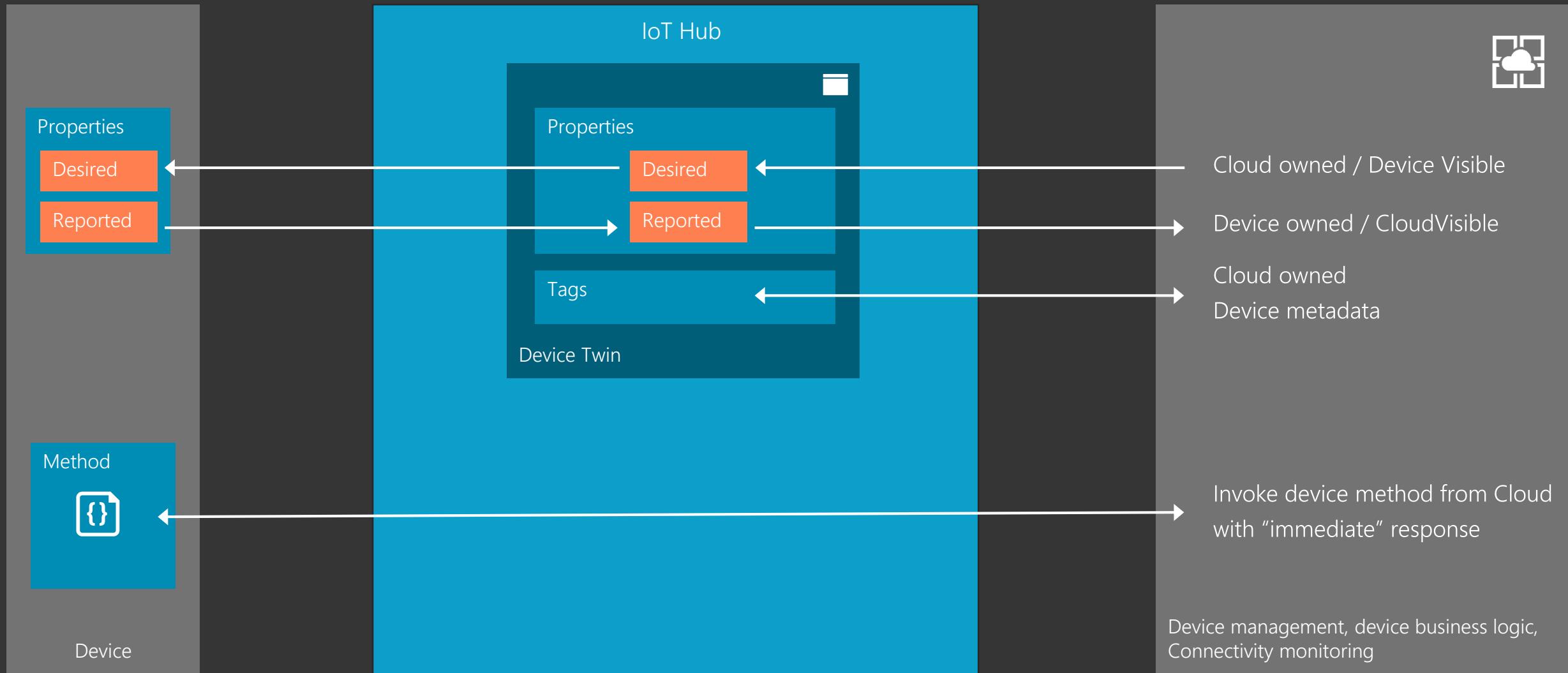
## Device Twins

- Last known state of device
- Desired state configuration
- Group devices

## Device Direct Methods

- Invoke method on device from Cloud
- “Immediate” response

# Structured data



# Demo

Device Twin and Methods

# Lab 4:

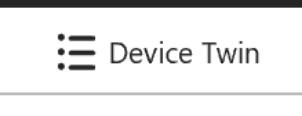
## *Working with device twins.*

# Lab 4 Steps

1. Open the Azure Portal and browse to your IoT Hub
2. Select a simulated device in the Device Explorer



3. Open the device twin



4. Add latitude & longitude to the desired properties

```
6      "desired": {  
7          "Latitude": 40.343432,  
8          "Longitude": 8.3334,  
9          "$metadata": {
```

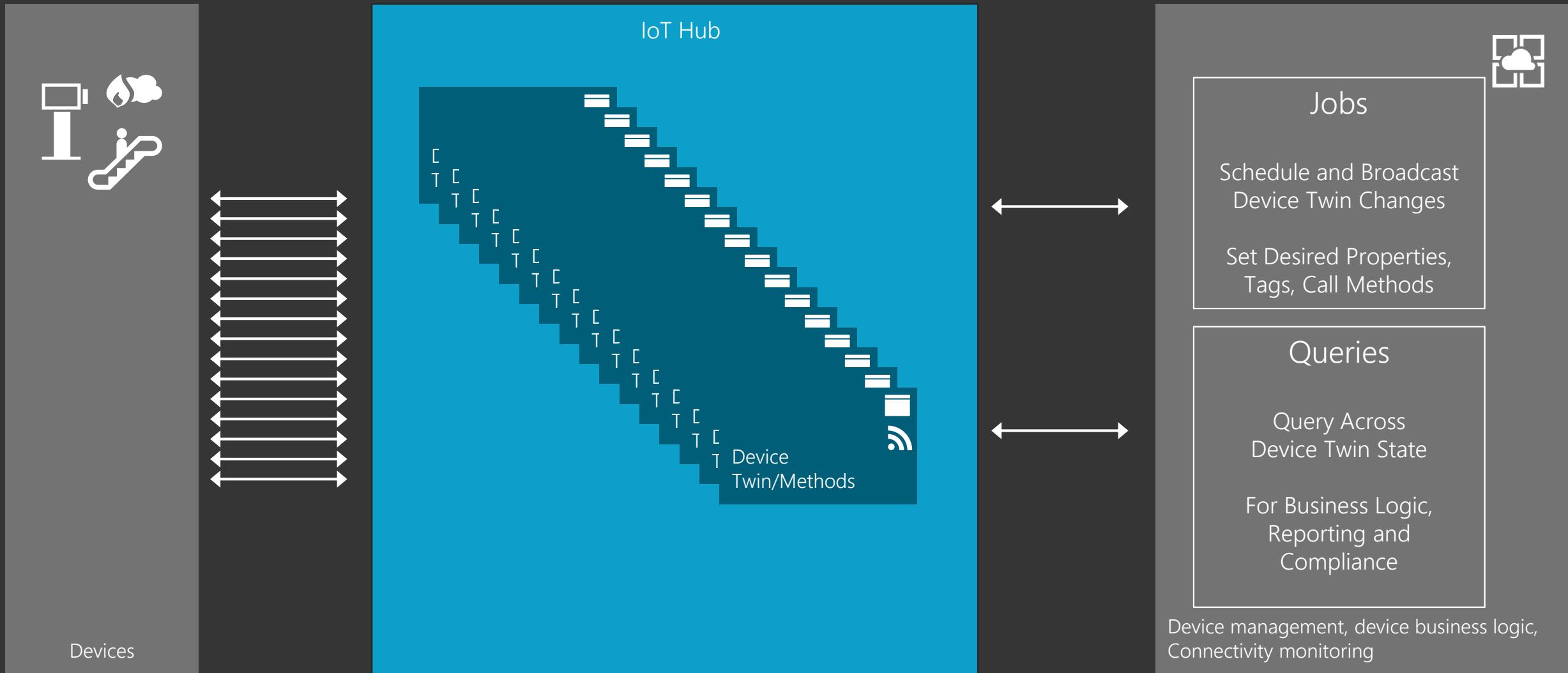
# Lab 4 Steps [2]

5. Open the node.js file of your device in bash
6. Add code in your device to react to changes of the device twin

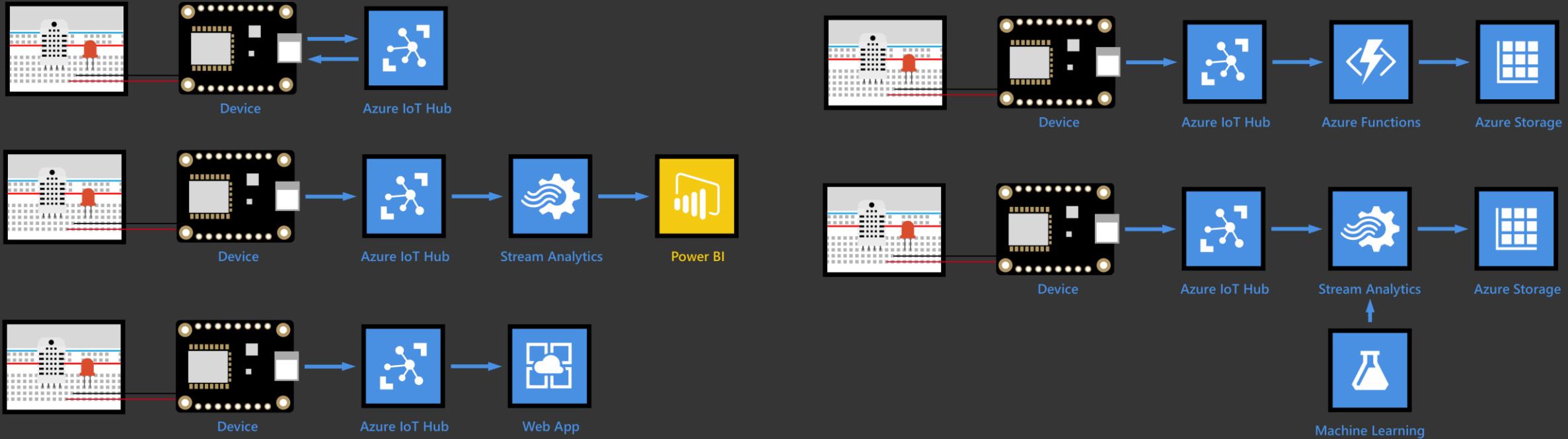
```
twin.on('properties.desired', function (delta) {  
    // Handle desired properties set by solution  
    console.log('Received new desired properties:');  
    console.log(JSON.stringify(delta));  
    // update location  
    reportedProperties.Latitude = delta.Latitude;  
    reportedProperties.Longitude = delta.Longitude;  
    // Send updated properties  
    twin.properties.reported.update(reportedProperties, function (err) {  
        if (err) throw err;  
        console.log('twin state reported');  
    });  
});
```

7. Start your device and change the values in the device twin
8. Open the Remote Monitoring Solution to see the change

# Jobs and Queries

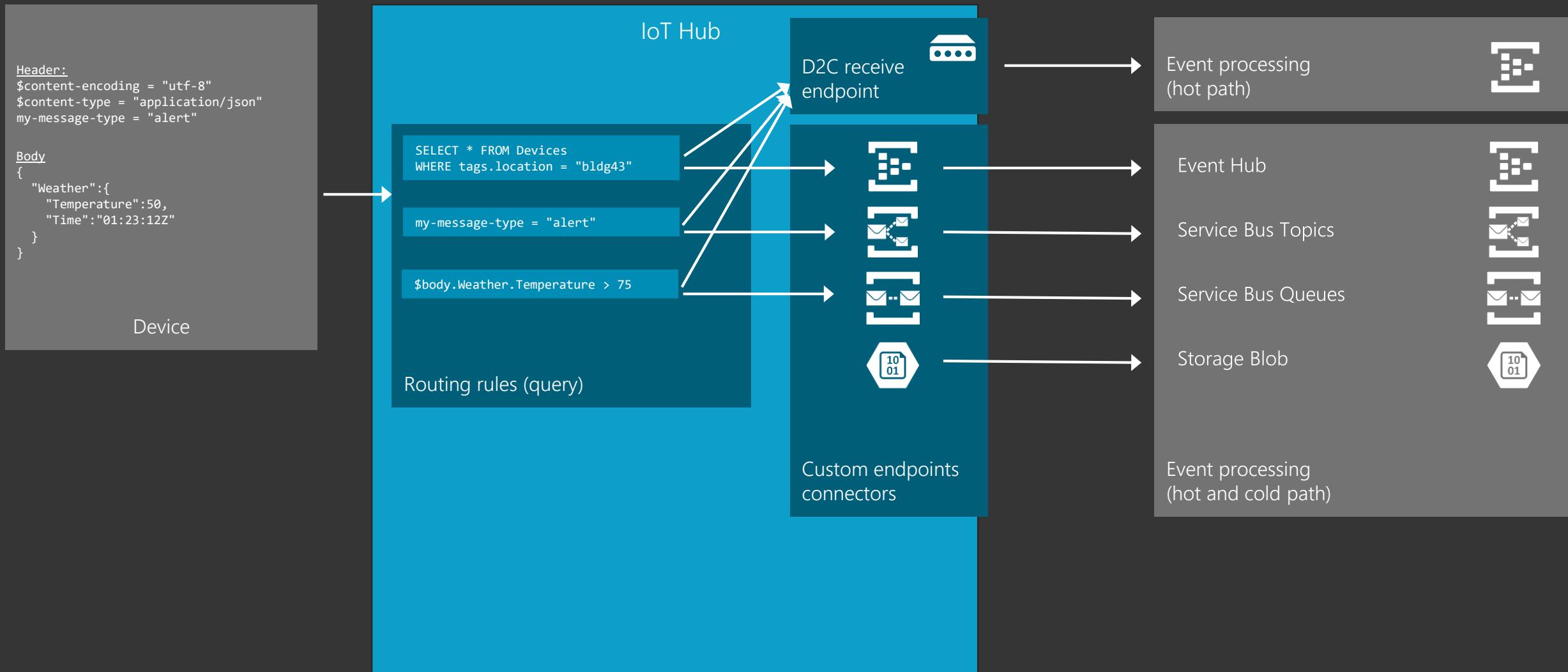


# What's next now that I have data flowing in?



Learn more: <https://aka.ms/azureiotgetstarted>

# Routing telemetry data



# Demo

Simple routing

# Lab 5:

## *Creating a IoT Hub Job.*

# Lab 5 Steps

1. Adjust your “physical” device and add a direct method “ReactOnJob”
  - Add “ReactOnJob” to SupportedMethods
  - Add the client.onDeviceMethod for “ReactOnJob”
  - Implement the defined method to update a property.
2. Create a nodejs job “jobService”
  - Adjust the steps in <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-node-node-schedule-jobs#schedule-jobs-for-calling-a-direct-method-and-updating-a-device-twins-properties> to represent your device and direct method.
3. Run the physical device and device job on 2 separate bash windows to see the output.
4. Use <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-node-node-schedule-jobs> as your guideline

# D2C scenario recommendations

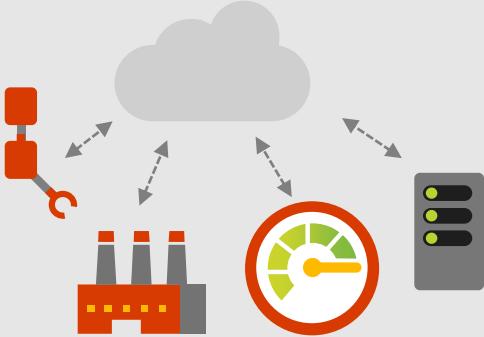
	D2C message	Twin's Reported Properties	File Upload
Scenario	Telemetry and alerts (time series, read sequential)	Synchronizing long-running workflows, such as configuration and software updates.	Large media files. (cold storage)
Size	Up to 256KB messages	Maximum reported properties size is 8KB.	Maximum file size supported by Azure Blob Storage.
Frequency	High	Medium	Low

# C2D scenario recommendations

	Direct methods	Twin's Desired Properties	C2D messages
Scenario	Commands that require immediate confirmation	Long-running commands	One-way notifications to the device
Size	8KB requests / 8KB responses.	Maximum 8KB	64KB
Frequency	High	Medium	Low

# Azure IoT Edge principles & concepts

# 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

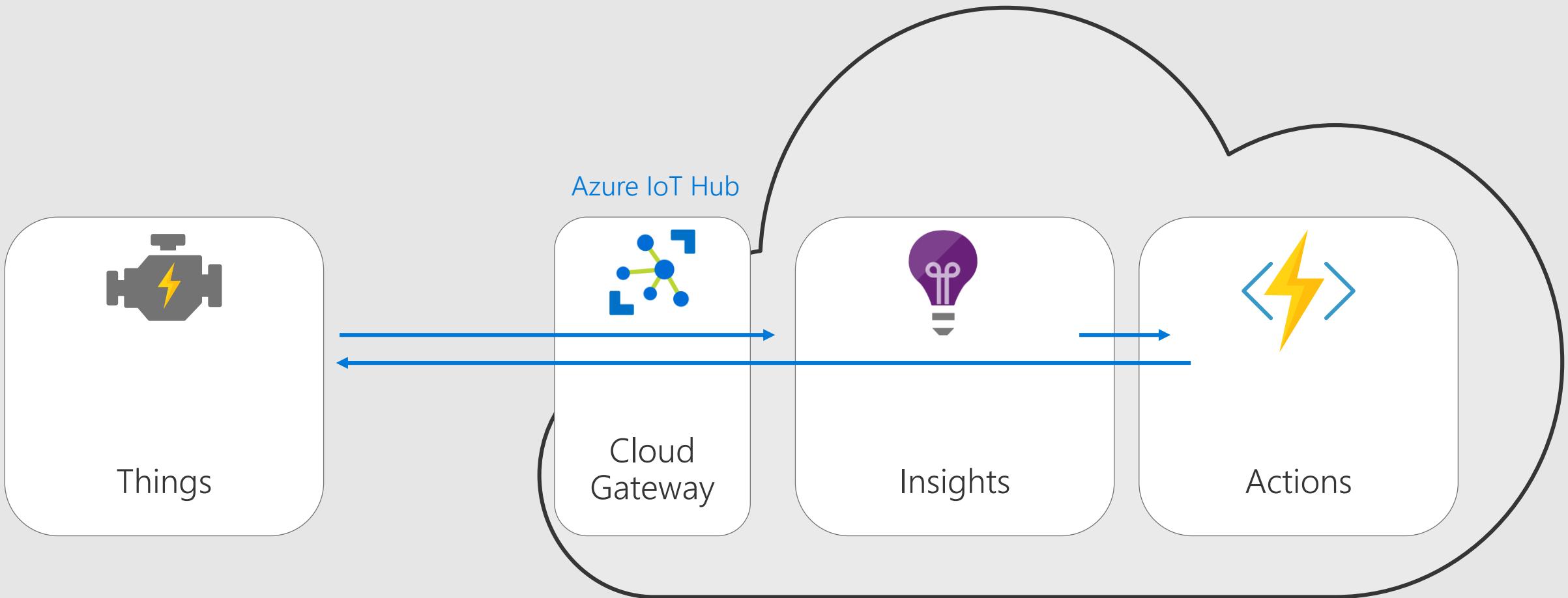
Low latency tight control loops require near real-time response

Protocol translation & data normalization

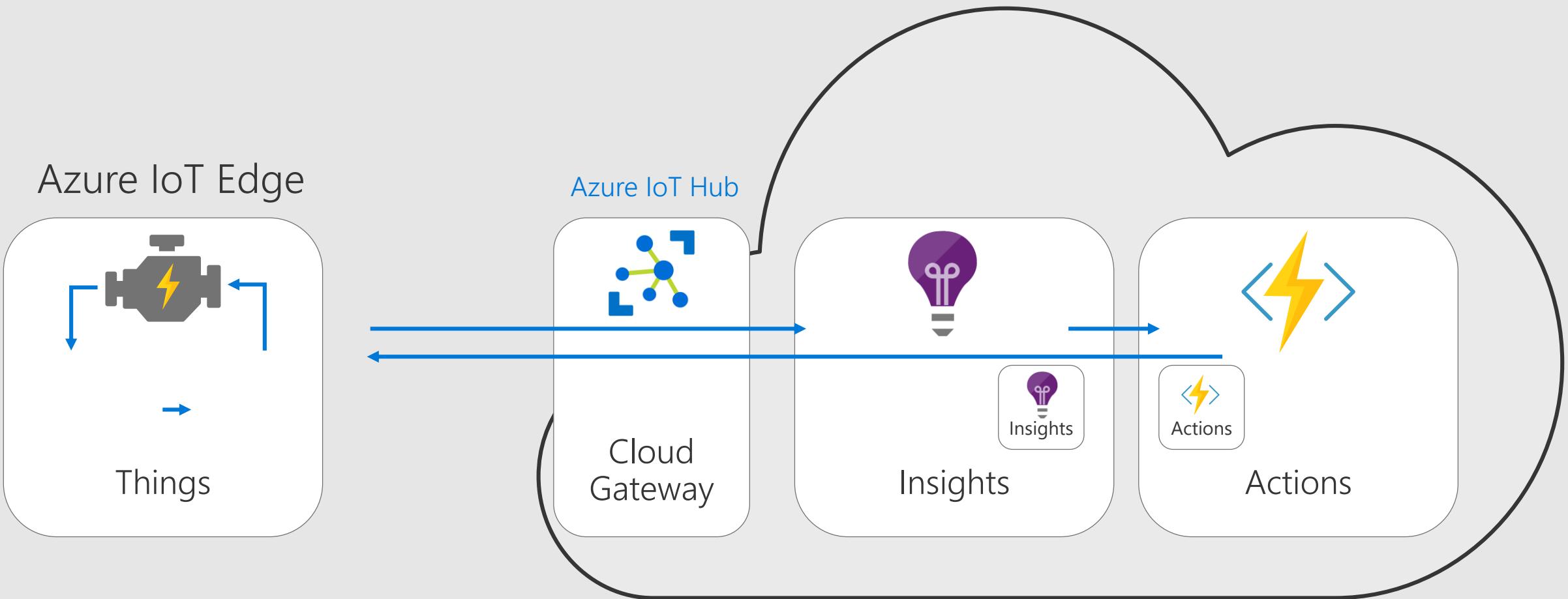
Privacy of data and protection of IP

Symmetry

# IoT Pattern



# IoT Pattern + Edge



# Azure IoT Edge Design Principles

## Secure

Provides a secure connection to the Azure IoT Edge, update software/firmware/configuration remotely, collect state and telemetry and monitor security of the device

## Cloud managed

Enables rich management of Azure IoT Edge from Azure provide a complete solution instead of just an SDK

## Cross-platform

Enables Azure IoT Edge to target the most popular edge operating systems, such as Windows and Linux

## Portable

Enables Dev/Test of edge workloads in the cloud with later deployment to the edge as part of a continuous integration / continuous deployment pipeline

## Extensible

Enables seamless deployment of advanced capabilities such as AI from Microsoft, and any third party, today and tomorrow



# Security

## Principles and Goals

- Cross Platform (Linux, Windows, multiple architectures)
- 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

# Concepts – Edge Runtime

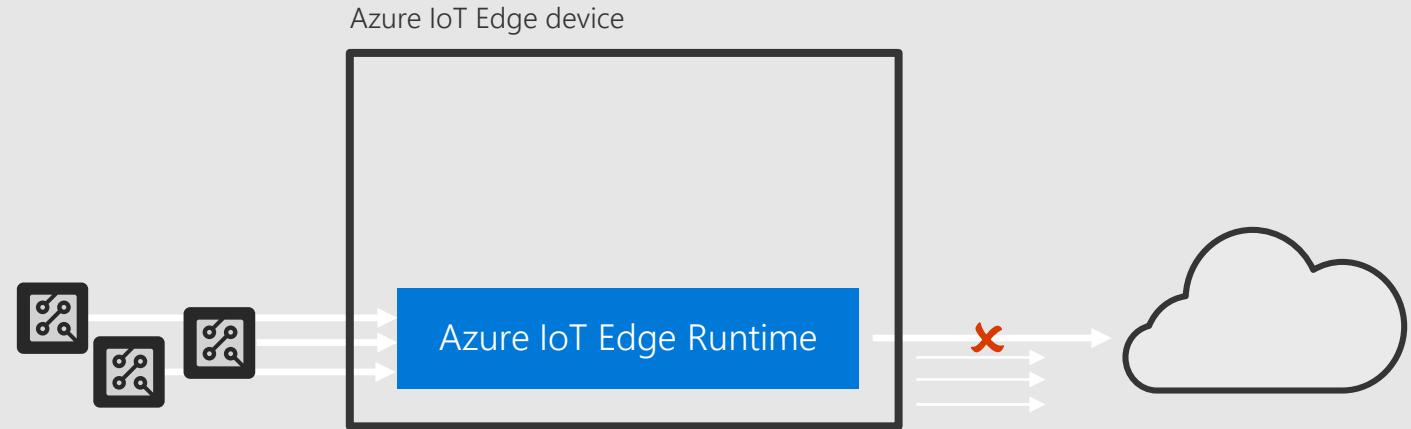
Edge Runtime provides fundamental services

Security

Multiplexing

Store and forward (Offline)

Management for devices otherwise isolated from internet



# Concepts – Modules

Edge Runtime manages modules

Modules add capabilities to the runtime

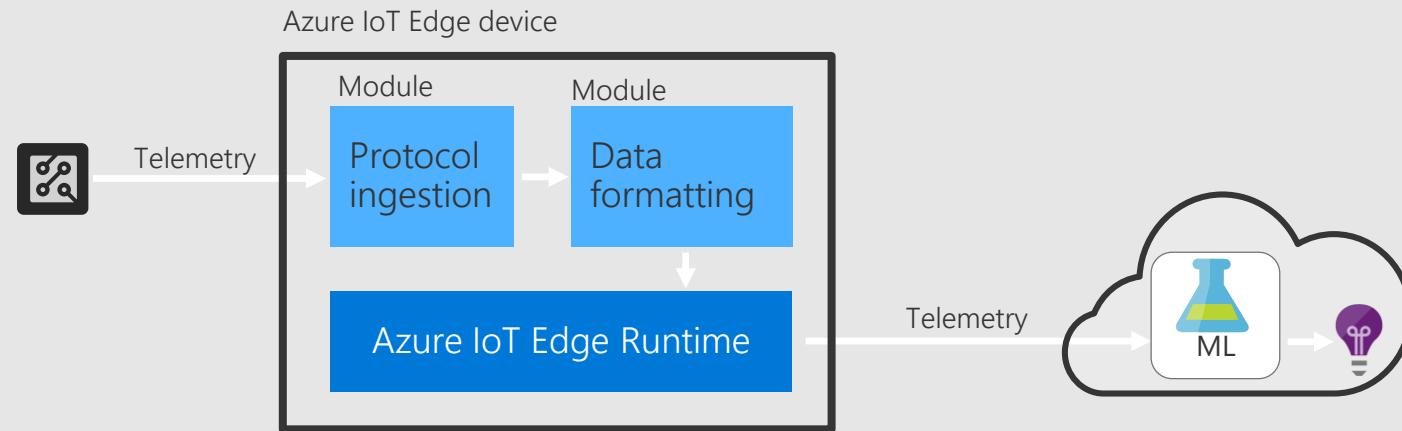
Each module performs an action

Chain of modules can be thought of as a data processing pipeline, solving an end to end scenario

Modules are Docker containers

Custom modules can be written in the language of your choice

Scenario: Find insights in the cloud from telemetry sent by a device that does not speak an internet ready protocol.



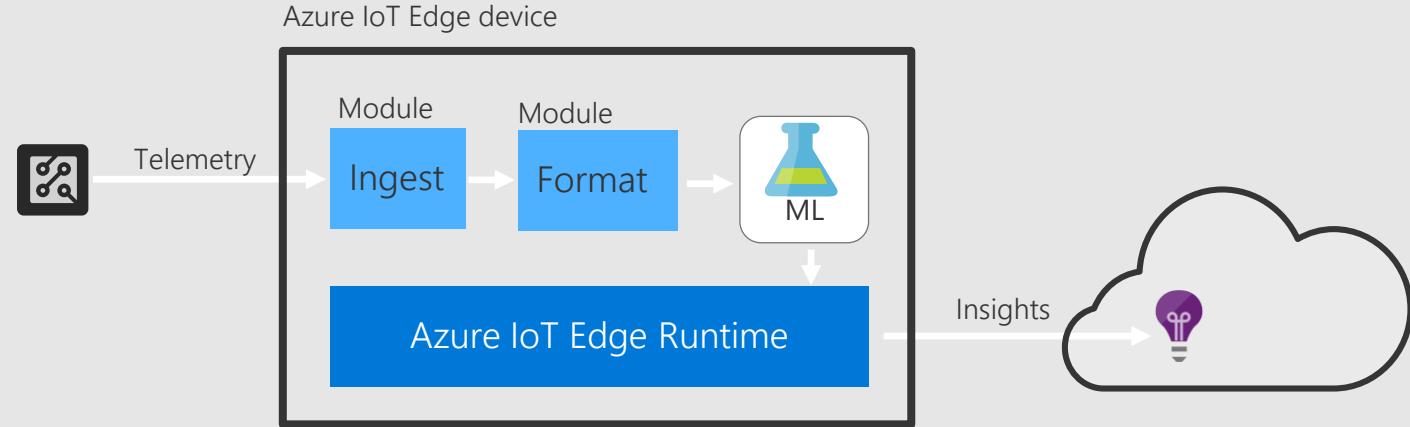
# Concepts – Cloud off load

Modular architecture for cloud offload

Azure services provide AI modules provide true edge analytics

Ecosystem for 3<sup>rd</sup> party edge services

Scenario: Find insights locally from telemetry data and only send insights to the cloud.



# Concepts – Cloud configurability and Monitoring

Runtime + IoT Hub provide full control of device lifecycle

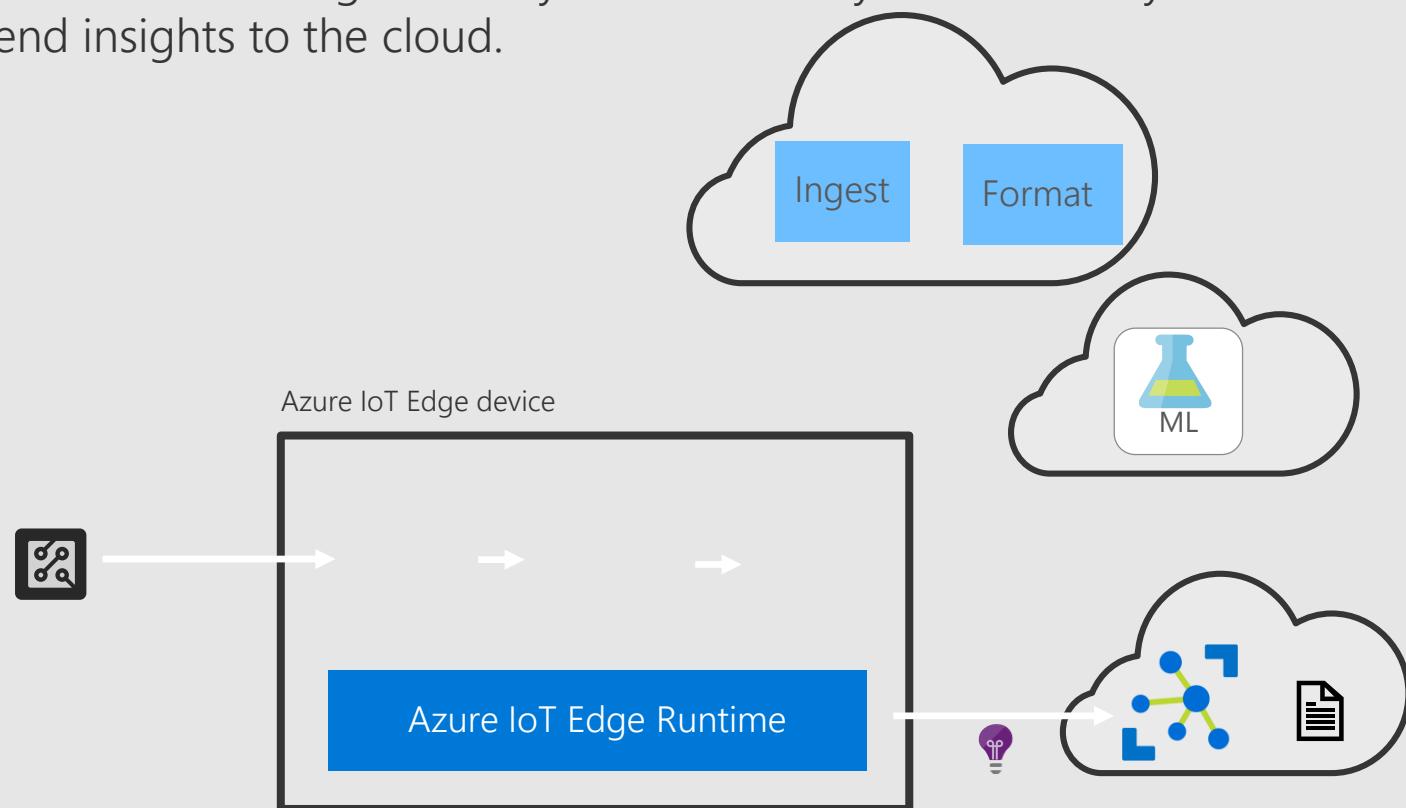
Configure a workflow

Target a device

Deploy

Monitor

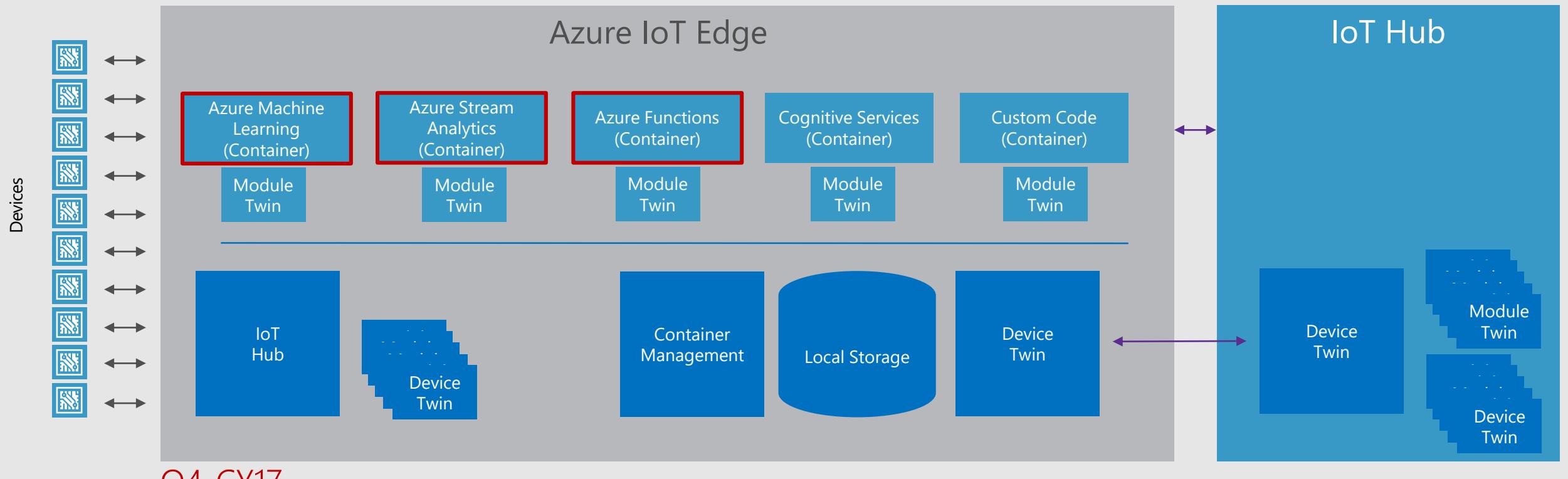
Scenario: Find insights locally from telemetry data and only send insights to the cloud.



# Azure IoT Edge

Container based workloads  
Azure Functions  
Azure Stream Analytics  
Azure Machine Learning  
Cognitive Services

Offline / Synchronized Device & Module Twins  
Local Storage  
Container Management  
Local IoT Hub  
HA/DR, Cloud Dev/Test Support



# Demo

Azure IoT Edge

# Lab 6:

*Deploy an IoT Edge device.*

# Lab 6 Steps

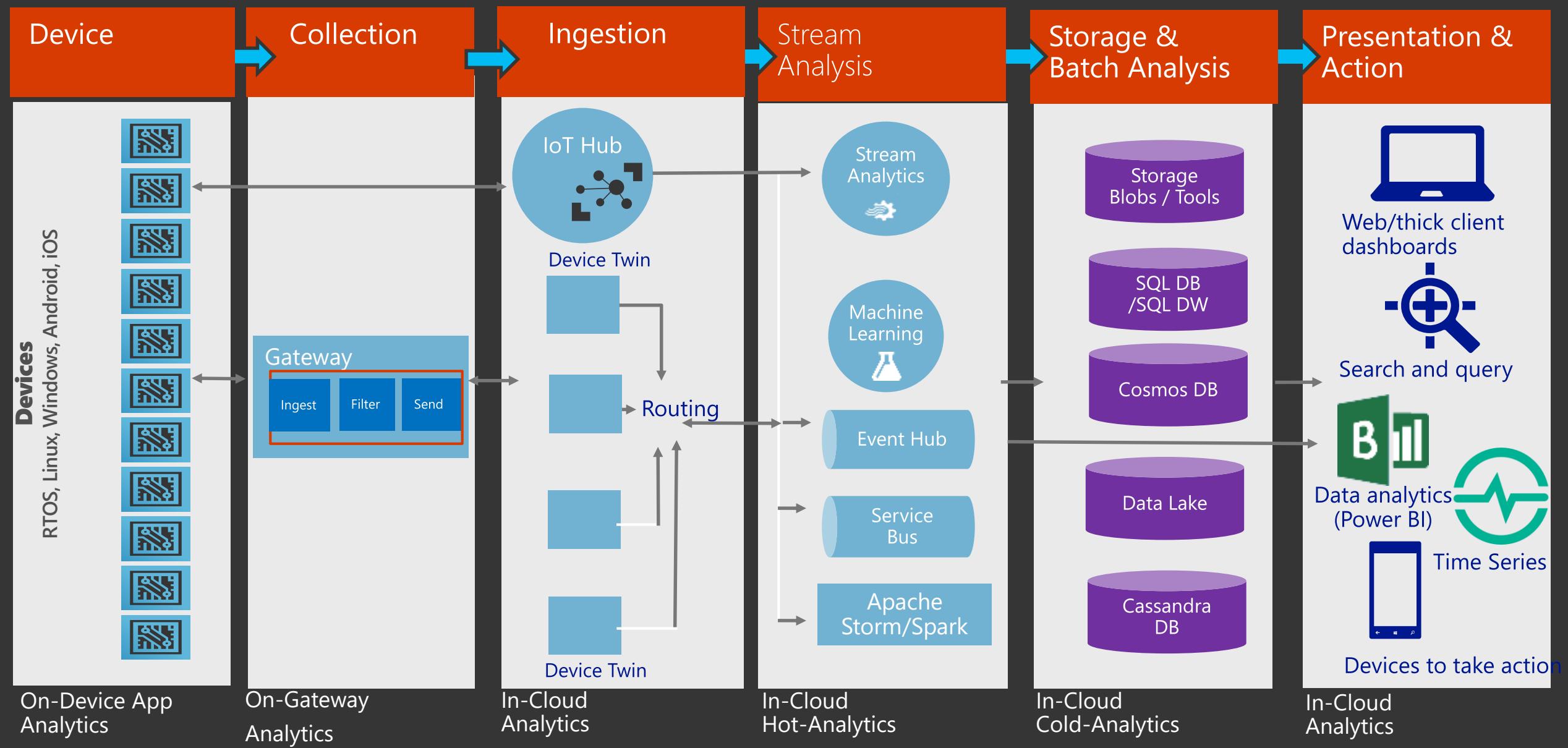
1. Create a Ubuntu VM in your resource group to act a an IoT Edge.  
<https://docs.microsoft.com/en-us/azure/iot-edge/quickstart-linux>.
2. Deploy and run the IoT Edge using the Linux quick start:  
<https://docs.microsoft.com/en-us/azure/iot-edge/quickstart-linux>.
3. Deploy Azure Function as an IoT Edge module:  
<https://docs.microsoft.com/en-us/azure/iot-edge/tutorial-deploy-function>.
4. Extra: Create a custom C# module:  
<https://docs.microsoft.com/en-us/azure/iot-edge/tutorial-csharp-module>.

# Analytics & Visualization (and other IoT services)

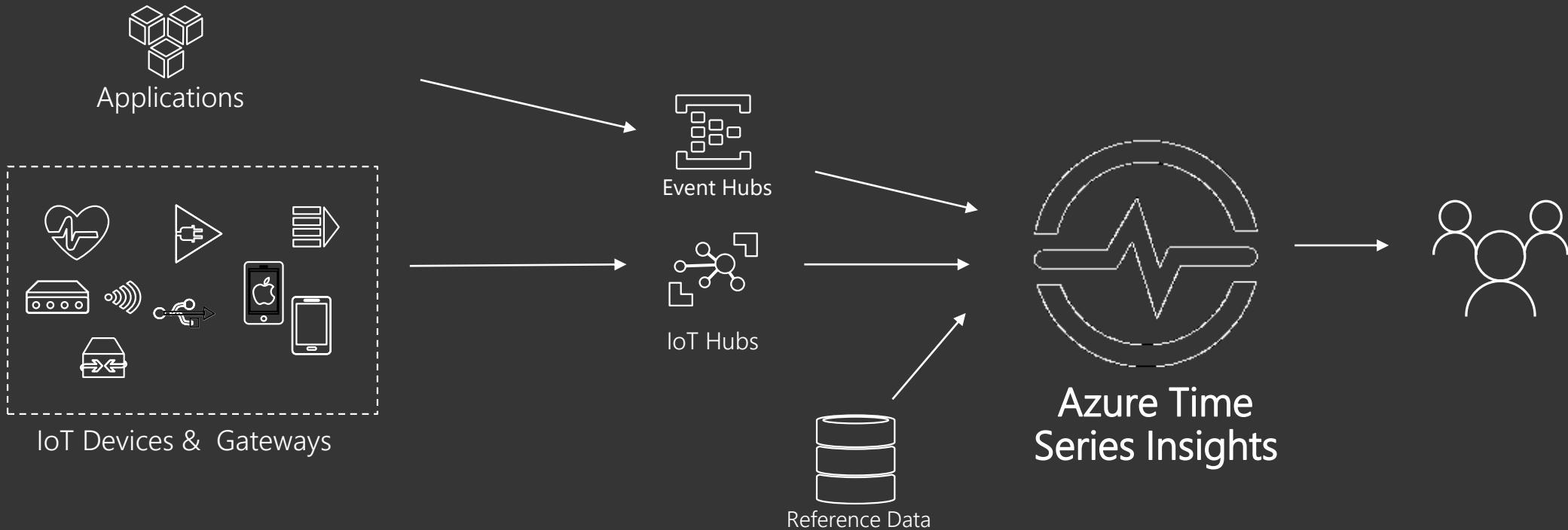
# Other IoT – Azure Services

- Analytics & Visualization
  - Time Series Insights
  - Azure Stream Analytics
  - Azure ML
  - Cosmos DB
- Enterprise Integration
- Messaging
- Serverless

# Azure IoT Analytics Patterns



# Time Series Insights



Event Sources



Event Ingestion



Storage + Analytics +  
Presentation & Action  
+ Query API

# Demo

Time series insight

# Lab 7:

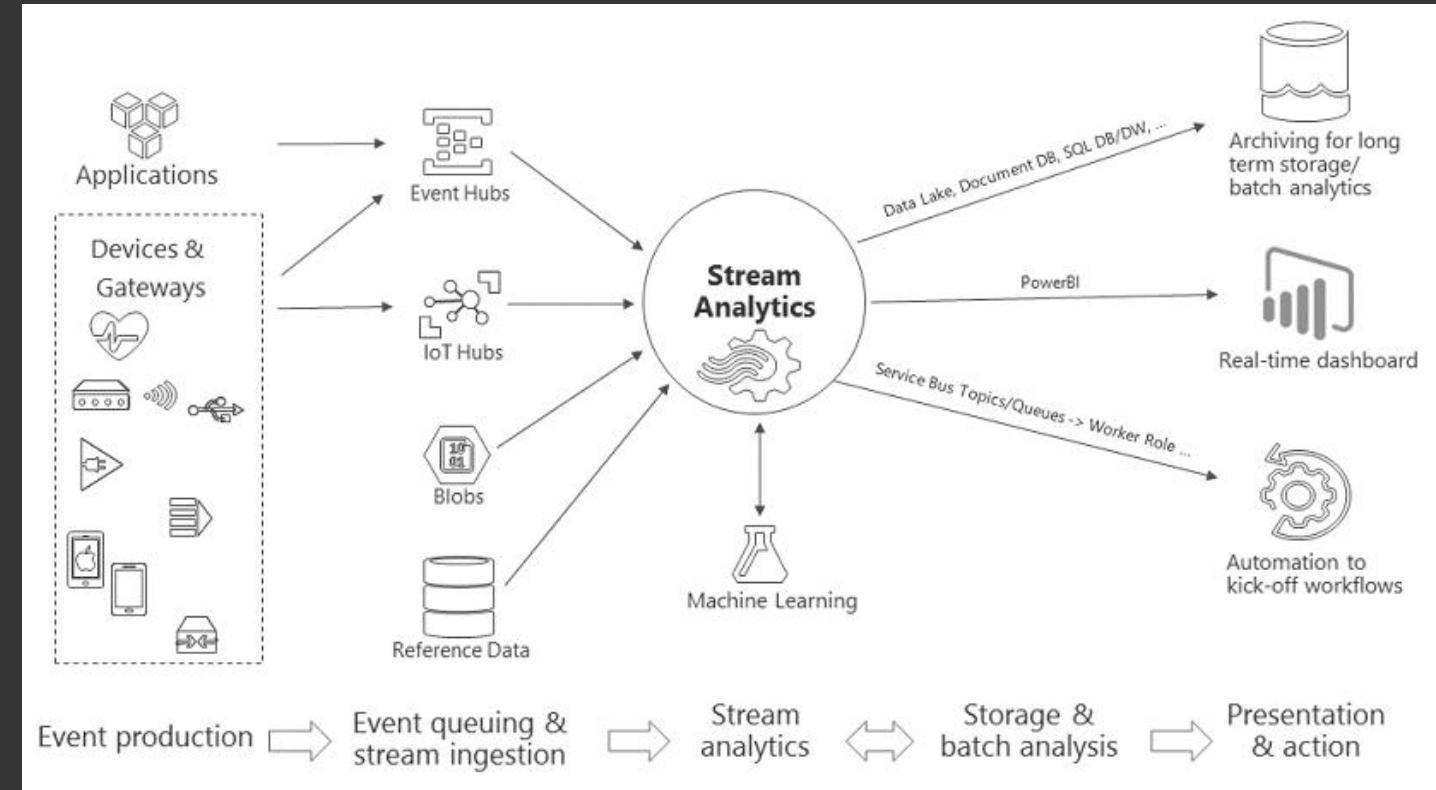
*Add Time Series Insights to the Remote Monitoring solution.*

# Lab 7 Steps

- Create a new Time Series Insights environment in the Azure portal  
<https://docs.microsoft.com/en-us/azure/time-series-insights/time-series-insights-get-started>
- Create the IoT Hub event source for your Time Series Insights environment using the Azure portal  
<https://docs.microsoft.com/en-us/azure/time-series-insights/time-series-insights-how-to-add-an-event-source-iothub>
- Access your Time Series Insight environment  
<https://insights.timeseries.azure.com/>
- Have a look at the Temperature by Device Id. Add pressure to the time series.

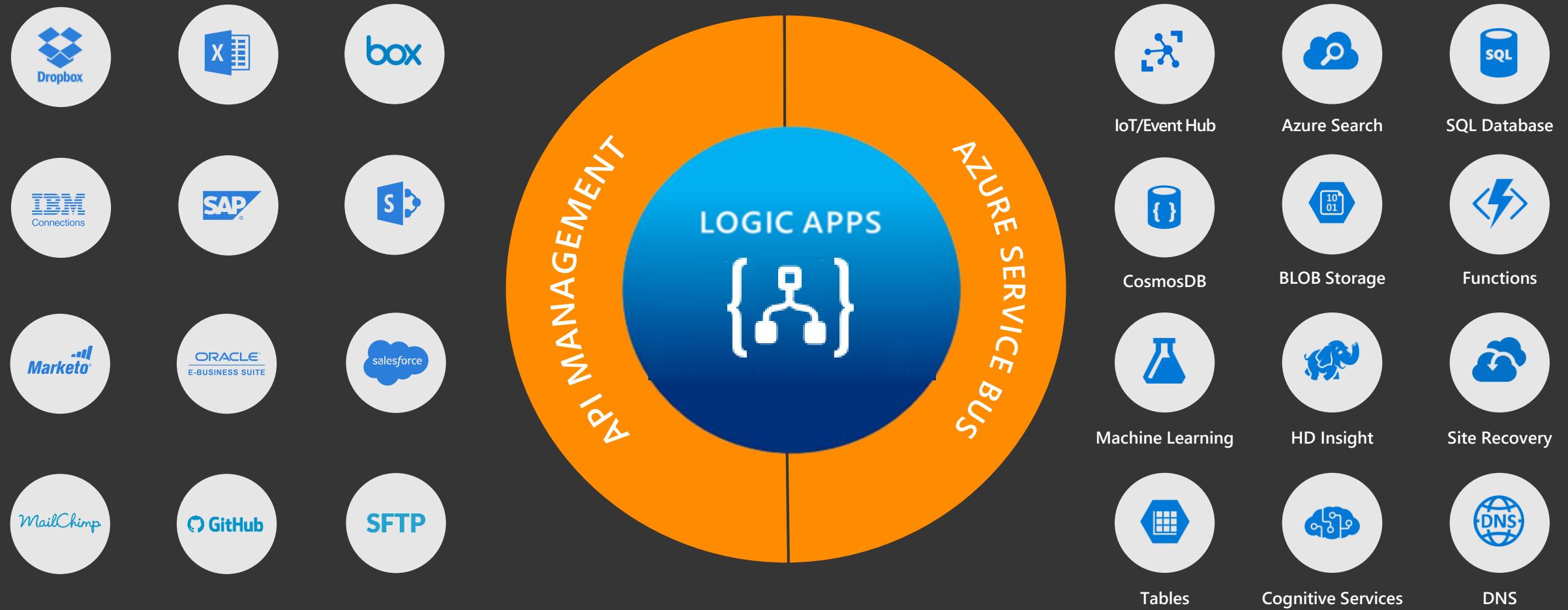
# Azure Stream Analytics

- Fully managed service
  - Pay as you go
  - Enterprise grade SLA
- Productivity
  - SQL Like query language
  - Aggregate data
  - Source/sink integrations
    - IoT Hub, Event Hubs
    - Power BI, Storage, Event Hubs, Data Lake, etc ....
  - Call custom functions and ML



# Microsoft's Cloud Integration Platform

About 39% of developers are focused on building cloud based Application integration



# Messaging



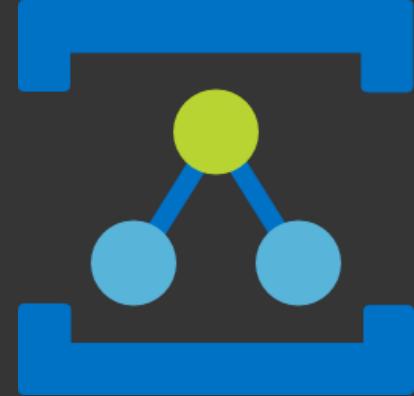
## Service Bus

Reliable asynchronous  
message delivery



## Event Hubs

Distributed data  
streaming



## Relay

Secure two way  
communication without  
changes to your  
network

# What is Azure Functions?

- Cloud-scale Event Handlers in no time

Azure Functions Composing cloud apps becomes simple

- Scales to demand & pay for what you use



- Develop in: C#, Node.js, Python, PHP, and more
- Schedule event-driven tasks across services
- Expose Functions as HTTP API endpoints
- Fully Open Source
- Running on Serverless Infrastructure

# Azure Cosmos DB

*A globally distributed, massively scalable, multi-model database service*



**SQL**



Table API

MongoDB API



Key-value



Column-family



Document



Graph

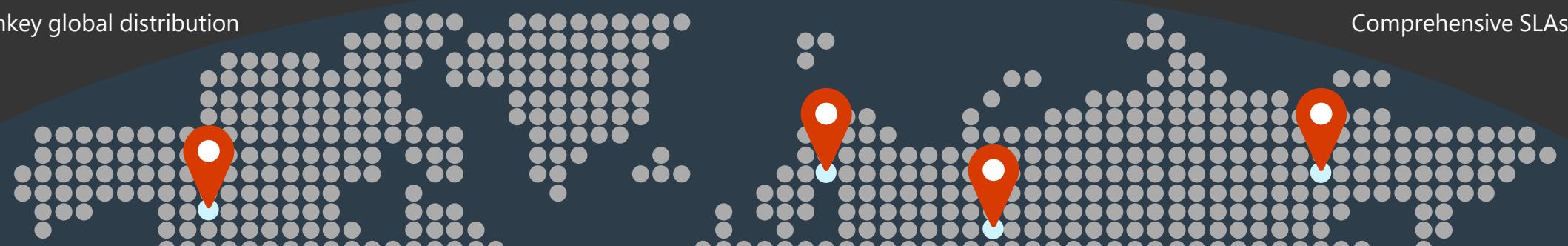
Turnkey global distribution

Elastic scale out  
of storage & throughput

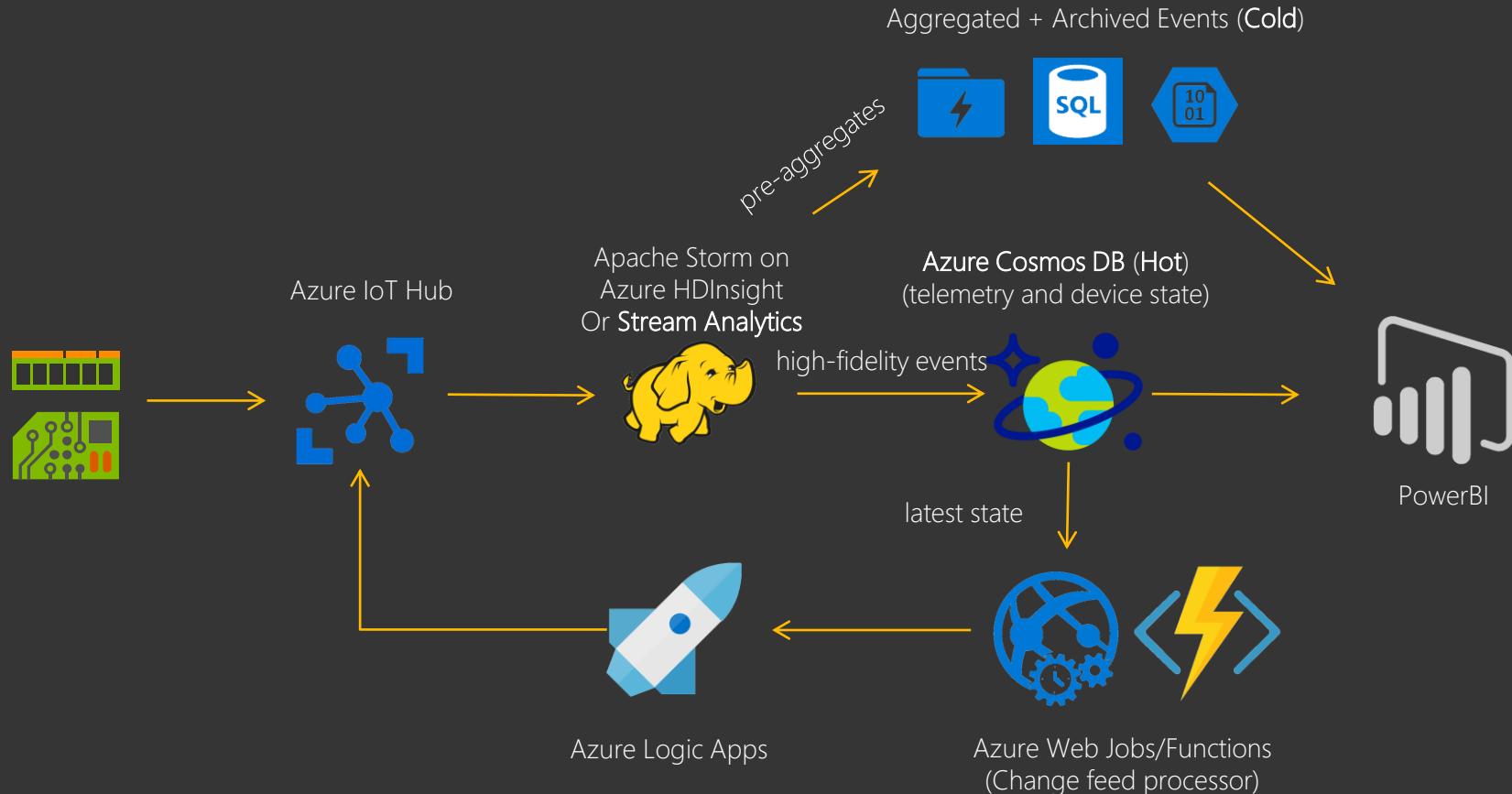
Guaranteed low latency at the 99<sup>th</sup> percentile

Five well-defined consistency models

Comprehensive SLAs



# Internet of Things – Telemetry & Sensor Data



# Power BI

- Business Analytics
  - Multiple sources
  - Enterprise security
  - Cross platform tools
- 
- Streaming Data Sets
    - Stream Analytics integration
    - APIs available (Azure Functions support)



# Lab 8:

*Add a logic app for Enterprise Integration.*

# Lab 8 Steps

1. Open the Azure Portal and add a Logic App using the following walk-through:

<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-monitoring-notifications-with-azure-logic-apps>

- a) Create a service bus
- b) Add an endpoint and routing rule  
Query string: temperature > ???. (whatever you want to use as trigger)
- c) Create and configure the Logic App
- d) Test it adjusting your “physical” device

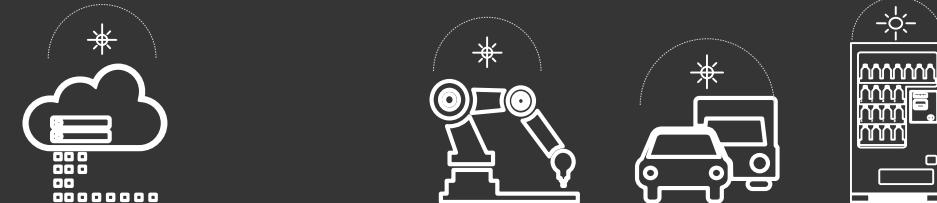
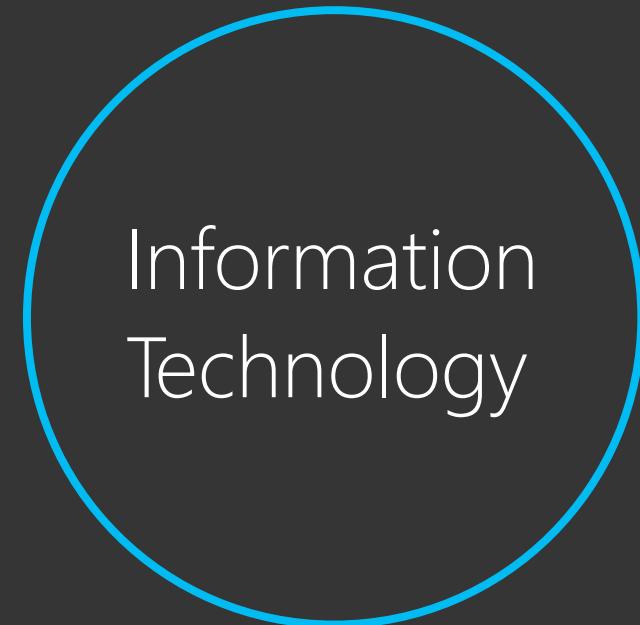
```
function generateRandomIncrement() {  
    return ((Math.random() * 5) - 1);  
}
```

# IoT Security

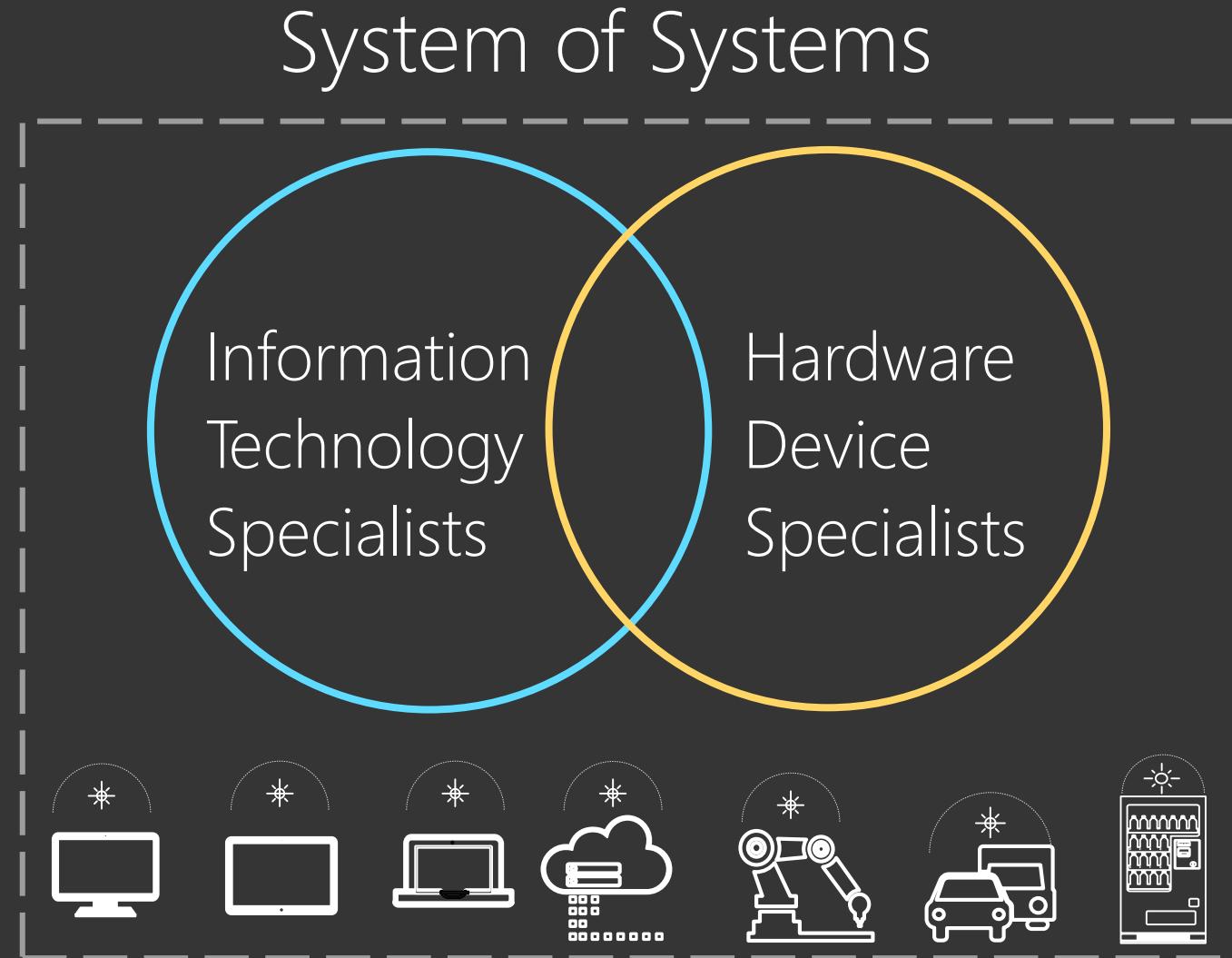


# Why securing Internet of Things is hard

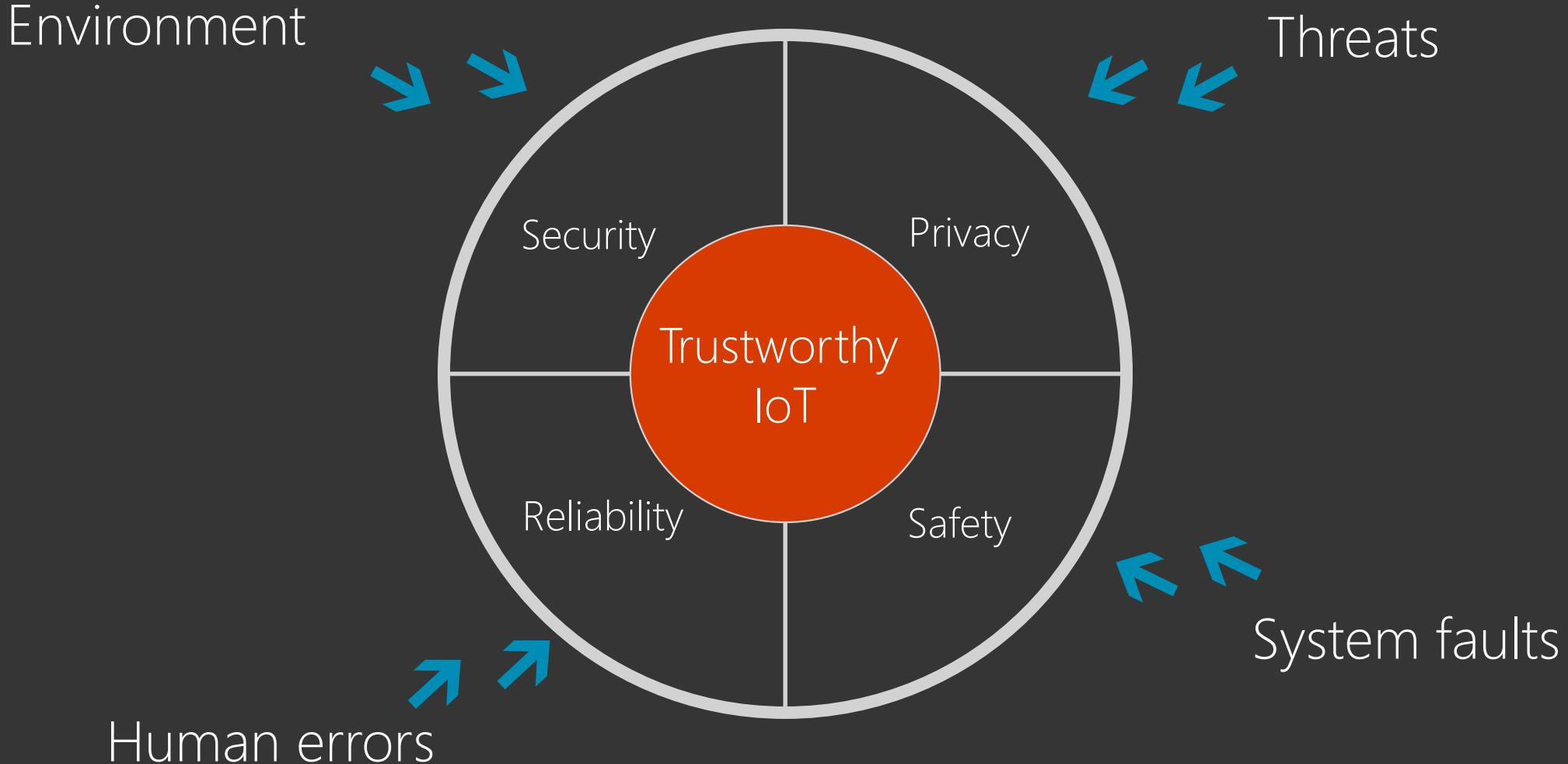
System of Systems



# Why securing Internet of Things is hard



# Trustworthy Internet of Things

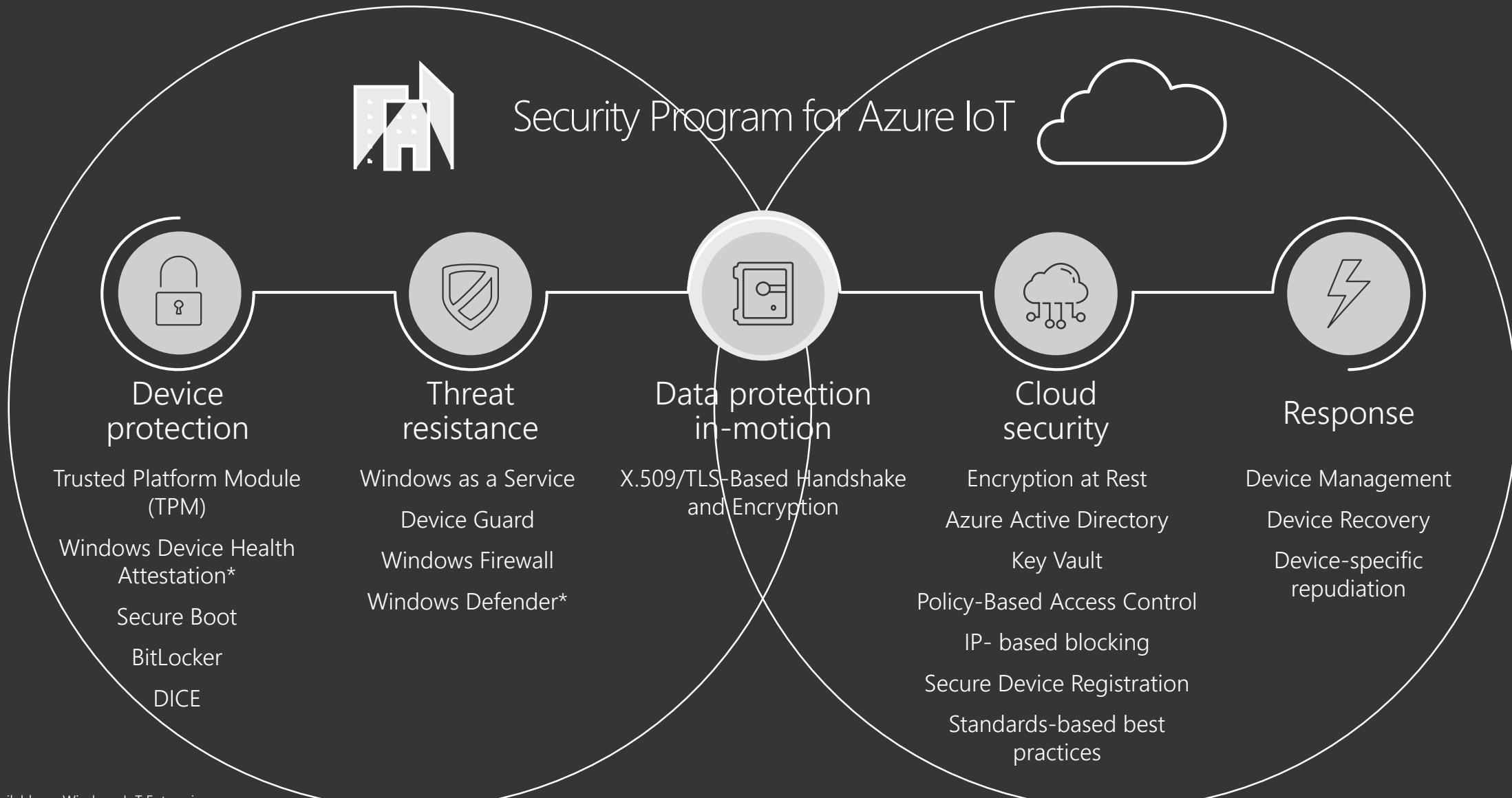


# Driving Security Innovation: 7 Properties of Device Security

- Well understood security principles and practices
- Device security rooted in hardware, but guarded with secure, evolving software
- [aka.ms/7properties](http://aka.ms/7properties)

Property	Hardware-based Root of Trust	Small Trusted Computing Base	Defense in Depth	Compartmentalization	Certificate-based Authentication	Renewable Security	Failure Reporting
Key Questions	Does the device have a unique, unforgeable identity that is inseparable from the hardware?	Is most of the device's software outside the device's trusted computing base?	Is the device still protected if the security of one layer of device software is breached?	Does a failure in one component of the device require a reboot of the entire device to return to operation?	Does the device use certificates instead of passwords for authentication?	Is the device's software updated automatically?	Does the device report failures to its manufacturer?

# Device to Cloud Security



# Customize your IoT Architecture

# Elements of Azure IoT Suite

Connect and Manage  
Devices & Gateways

Analyze streaming data

Integrate into business  
systems



Preconfigured solutions



Real time analytics



Workflow integration



Gateway & Devices



Data visualization



Push and broadcast  
notifications



Connect and control



Predictive analytics



ID and access  
management

4. Secure IoT Infrastructure



5. Customize IoT Architecture

# IoT architecture challenges

- Network and protocol
- Security, Privacy and Trust
- Heterogeneity
- Serialization
- Streaming Throughput
- Cost
- Data Volume
- Auto-discovery
- Legacy and fragmentation

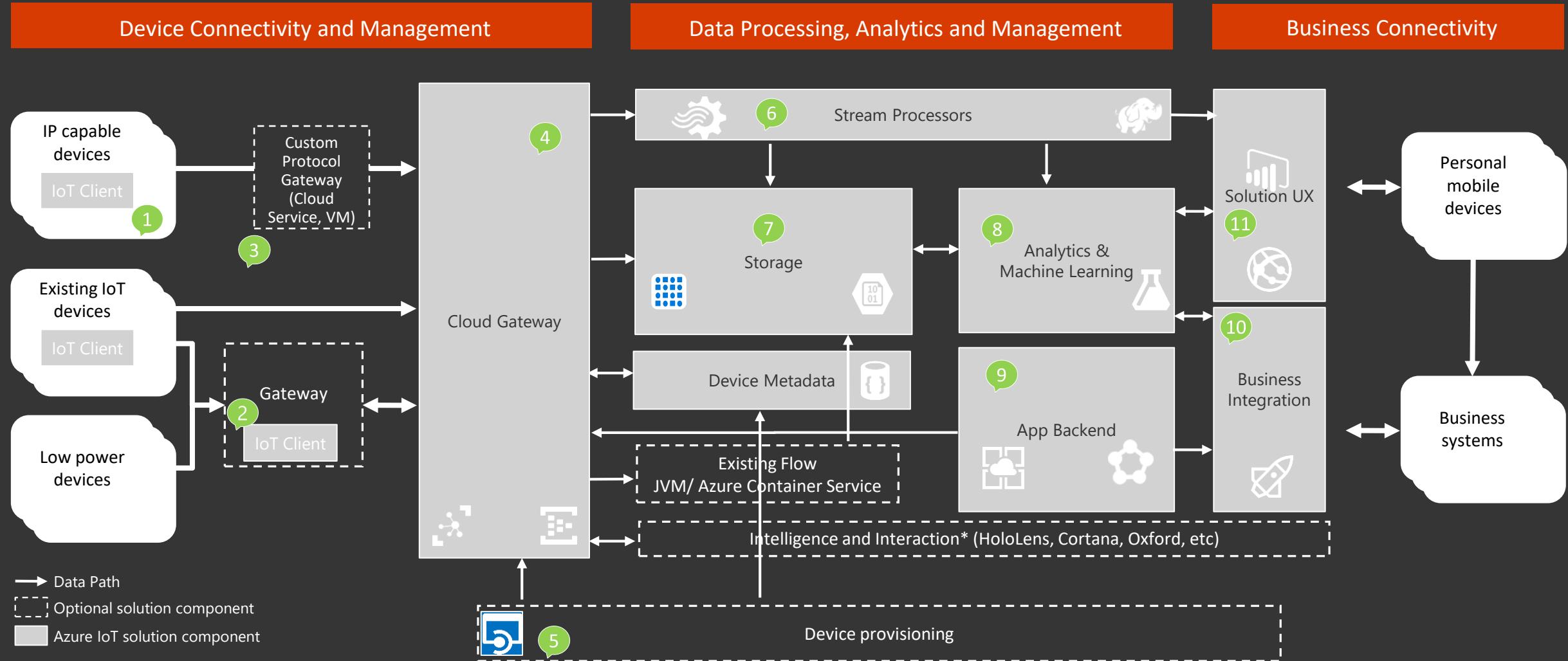
# IoT architecture characteristics/variables

- Message Type
- Message Speed
- Message Frequency
- Device Volume
- Message Volume
- Connectivity Level
- Data structure
- Communication direction

# IoT real-life use cases and scenarios

Industry	Use case	Scenarios patterns
Manufacturing	Connected Vehicle Food Traceability Production Asset Mgmt Manufacturing Operations Connected Field Service	Device Connectivity & Management Multi-protocol support Connect legacy devices without replacing existing infrastructure
Consumer	Home Security Smart Appliances	Remote Monitoring Command and Control Service customer equipment
Government	Automated Public Transit Environmental Monitoring Detection Public Infrastructure Asset Mgmt Public Safety and Emergency Response	Edge computing Security & Environmental Monitoring Real-time analytics
Healthcare	Remote Health Monitoring Clinical Care Personal Wellness Connected Medical Device	Data Transformation and Routing Predictive Maintenance Role Based Access Control
Retail	In-Store Consumer Digital Offer Personalized promotion	Alerts and Notification Data Visualization
Transportation	Air traffic monitoring Asset Fleet management Freight monitoring	Integration with other business services Process atomic and batch data stream Manage Intermittent-Connected Devices
Utility	Smart grid Smart Building	Data security from connectivity to storage Geo-availability

# Azure IoT reference architecture

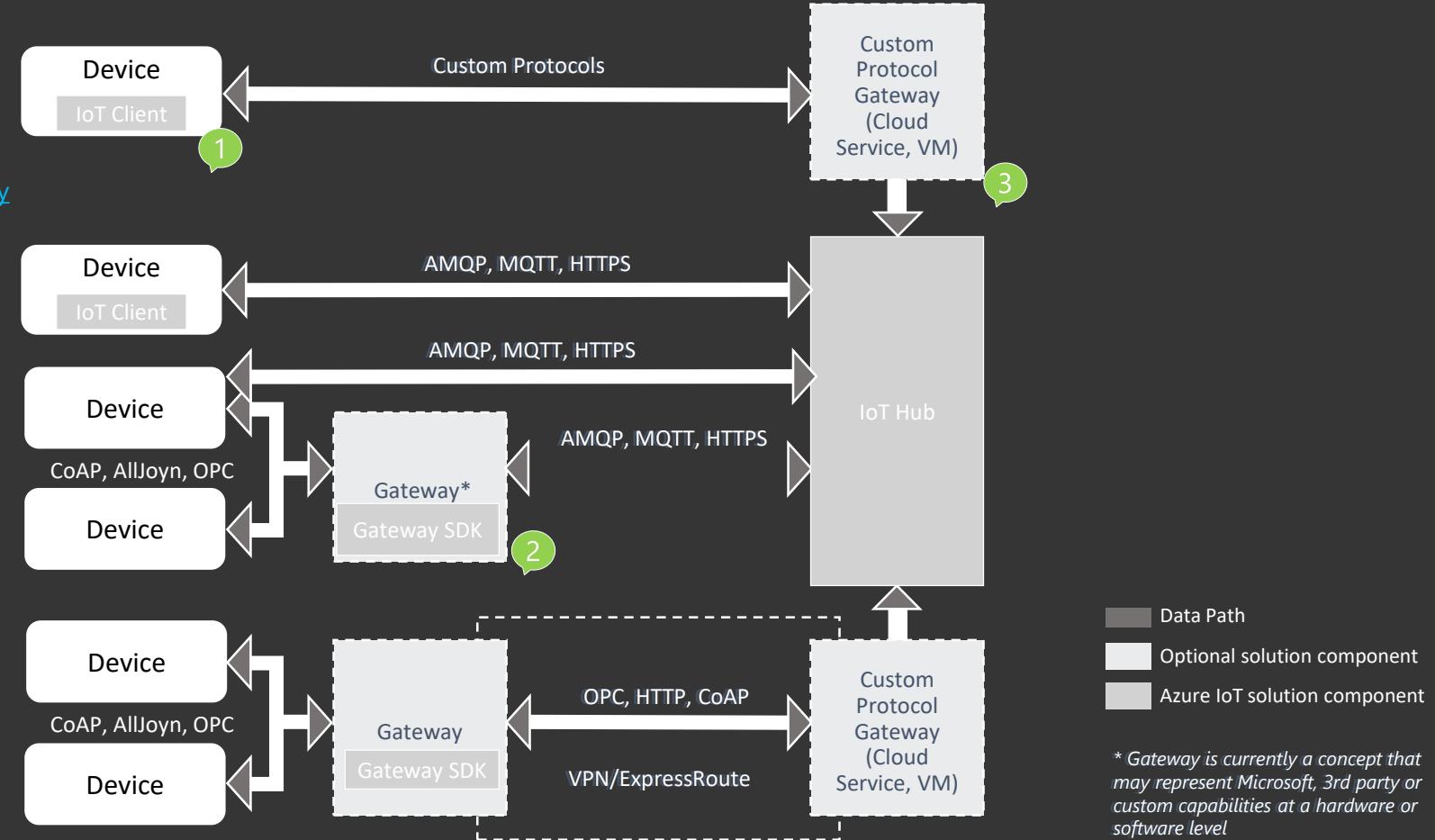


# Device connectivity

1 <https://github.com/Azure/azure-iot-sdks>

2 <https://github.com/Azure/azure-iot-gateway-sdk>

3 <https://github.com/Azure/azure-iot-protocol-gateway>



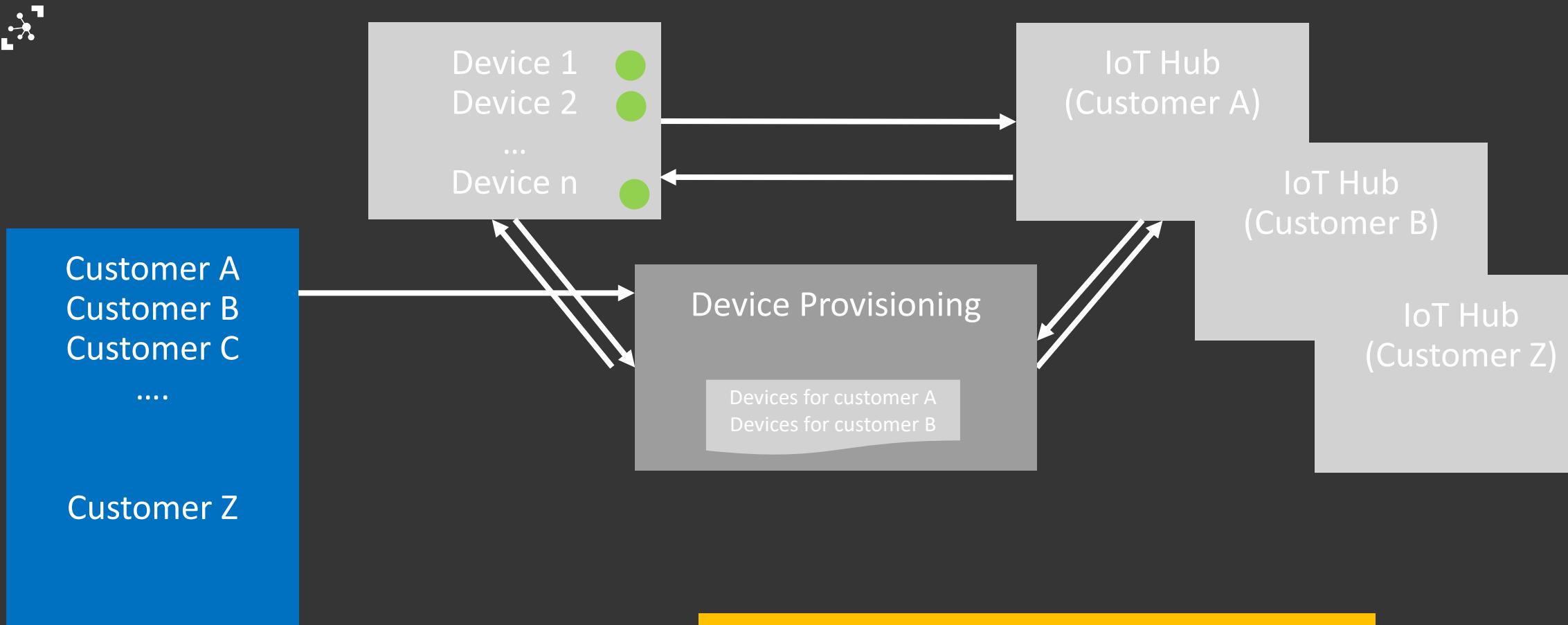
## Scenarios addressed:

- Heterogeneity
- Network & Transport protocol

# Cloud gateway

	Area	IoT Hub	Event Hub
IoT Hub 	Communication Pattern	Device-to-cloud event ingress and cloud-to-device messaging.	Only event ingress (device-to-cloud scenarios)
	Security	Per-device identity and revocable access control.	Event Hub-wide Shared Access Policies, with limited revocation support using publisher's policies.
	Scale	Millions of simultaneously connected devices.	Limited number of simultaneous connections: up to 5000 AMQP connection.
	Device state information	<a href="#">Device twins</a> can store and query device state information.	No device state information can be stored.
	Device direct RPC methods	Yes	No
	Operations	Individual device authentication errors, throttling, and bad format exceptions.	Exposes only aggregate metrics.
	File Upload	Yes	Not supported.
	Device SDK	Large variety of platforms and languages, OS.	Supported on .NET, C.
	Device protocol	MQTT, MQTT over WebSockets, AMQP, AMQP over WebSockets, HTTP and Protocol Adapter	MQTT, MQTT over WebSockets, AMQP, AMQP over WebSockets, and HTTP

# Device provisioning



## Scenario addressed:

- Auto-discovery and provisioning of devices

# Stream processors

Processor Name	UX	Monitor	IDE	Scale	Connectors	SQL	Data Format
Stream Analytics	Yes	Yes, via Azure Portal or REST API	Yes, VS, Azure Portal, Power Shell	Yes, 1MB/s	Out: Azure Event Hubs, Azure Blob Storage, Azure Tables, Azure SQL DB, and PowerBI.	Yes, an easy to use SQL language support is available	Json, Avro, CSV
Event Processor Host	No	No	VS, Nuget package	No, not massively	Yes, only Event Hub to Storage	No, custom code	Custom code
Apache Storm on HD Insight	Yes	Yes, Storm UI and REST API	Yes, Azure Portal, PowerShell, Visual Studio and REST APIs	Yes, top limit defined by quota	Out: PowerBI, Azure Event Hubs, Azure Blob Store, Azure DocumentDB, SQL and HBase.	No, users must write code in Java C# or use Trident APIs.	Any format via custom code

**Scenario addressed:** Real-time event processing

# Storage considerations

Relational Database  SQL DB, SQL Server Stretch DB	Two-dimensional rows/columns  Strongly typed column data  Declarative schema  Proven stability and security	Flexible query-ability, reporting  For when model is known; query unknown  Variable data structure  Scaling out
Big Data  Data Lake Store, SQL DW	<b>Data Lake:</b> All types of data could be stored(structure, unstructured, semi-structured, raw). Schema on read. Design for low cost storage. Highly agile, configure as needed.	<b>SQL DW:</b> provides more familiar and interactive query environment. Can use as a primary store or may move data there via batch from Data Lake/Blobs. Expensive for large data volume. Less agile, fixed configuration.
Key-Value Databases  Table Storage, Redis Cache	Simple Map (Key, value)  Semi-Structured  De-normalized  When you have & will query by the key	Extremely fast  Scalable  Interacting with indexes  Searching on properties within the Value  Complex queries
Document Databases  DocumentDB	No schema, no relationships  Collections of documents  Documents are JSON objects  No schema, no relationships	Collections of documents  Documents are JSON objects  Complex relational data  Elaborate joins

**Scenario addressed:**  
- Optimize for storage

# Storage considerations

Blob Storage	<p>Unstructured data for text or binary data at massive scale (single container can be 500TB)</p> <p>Highly scalable</p> <p>Supports streaming scenarios and random access</p> <p>Data accessible via the REST API</p>
Storage Queue	<p>Durable message queueing for large workloads</p> <p>Rich client libraries for .NET, Java, Android, C++, Node.js, PHP, Ruby, and Python</p> <p>Data accessible via the REST API</p>

**Scenario addressed:** Cold Storage

# Queues considerations

	<b>Storage Queue</b>	<b>Service Bus Queue</b>
Ordering Guarantee	No	Yes; First-In-First-Out (FIFO)
Delivery Guarantee	At-Least-Once	At-Least-Once At-Most-Once
Maximum message size	64 KB	256 KB
Maximum message Time to Live (TTL)	7 days	Unlimited
Maximum throughput	Up to 2,000 messages per second	Up to 2,000 messages per second
Average latency	10 ms	20-25 ms

# Analytics & Machine Learning considerations

Hot-Path Analytics	Azure IoT Routes, Stream Analytics, Azure HDInsight Storm, Service Bus Queues and Topics
Cold Path Analytics	Azure HDInsight, Azure ML, Azure Data Lake, Data Lake Analytics, Azure Data Factory
Hot Path Business Logic	Service Fabric & Actor Framework
Machine Learning	Pattern Detection, Behavior Prediction, Plausibility Analysis, Anomaly and Fraud Detection

**Scenario addressed:** Advanced analytics, Predictive Analytics

# App Backend - High-scale compute models

## Scale-appropriate compute models

### Actor Frameworks / Service Fabric Reliable Actors

Distributed compute fabric hosting device (or device-group) actors

Message-based activation of actors and dispatch of messages

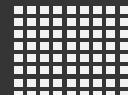
### Service Fabric Reliable Collections

Highly available with replicated and local state management

Simple programming experience in vastly scalable compute nodes

### Serverless code

### Azure functions



#### Scenario addressed:

- Hyper-scale compute processing
- Serverless architecture
- Data format transformation

# Business Integration considerations

Logic Apps	For graphical way to model your business processes as a series of steps or a workflow. For triggers on connectors(like email arrival), for actions.
API Management	For creation of API gateway and developer portal in minutes with security, analytics, monitoring and alert support
Data Factory	For moving and transformation of your data based on schedule; rich visualization, many data sources and sinks
Service Bus	For reliable message ordering queuing (FIFO) and durable publish/subscribe messaging
BizTalk Services	For integration with SAP, Oracle EBS, SQL Server, PeopleSoft

## Scenario addressed:

- Integration with business connectivity

# Solution UX considerations

App Service	Best choice for most web apps, easily create a new site using the framework and tools of your choice. Has Web Apps, Mobile Apps, API Apps, Logic Apps and Web Jobs
Notification Hub	Designed for massive scale notifications to devices and apps.
Power BI	Easy way to create rich, interactive dashboards for visualizations and analysis
Bing Maps	For interactive and static maps, geocoding, route and traffic data
Active Directory	For authentication and authorization control of your users

## Scenario addressed:

- Visualization dashboards for operations

# Training

# Training for Azure IoT

Training provider	Class	Price
	<a href="#">Developing IoT Solutions with Azure IoT</a>	Free (auditing) or \$99 (certificate of completion)
 Microsoft Virtual Academy	<a href="#">Introduction to Azure IoT</a>	Free
 Linux Academy	<a href="#">Azure IoT Essentials</a>	Free
 Linux Academy	<a href="#">IoT for the Enterprise</a>	Paid (Free 7-day trial, \$19/month afterwards)
 CloudAcademy	<a href="#">Internet of Things with Azure</a>	Paid (Free 7-day trial, \$29/month afterwards)
 <small>Real Code. Real Labs. Real Learning.</small>	<a href="#">Building IoT Solutions with Azure</a>	Paid (Free 7-day trial, \$29.95/month afterwards)

Learn more: <http://aka.ms/iottraining>

# Wrap up & Q&A



# Thank you!