

Azure IoT Cloud Workshop

Eric van Uum

Microsoft GBB WE – IoT Technology Professional
Eric.vanuum@Microsoft.com



Agenda (topics guideline)

Title	Content
Part 1 - Overview	Welcome and introductions Microsoft Cloud Vision Internet of Things Overview
Part 2 – Azure IoT solutions	Azure SaaS and PaaS solutions
Part 3 – 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)
Part 4 – Azure for IoT	Azure Services used in IoT
Part 5 – Advanced Analytics and Visualization	Getting from device data to insights with Azure Data and Machine Learning solutions
Part 6 - Hands-on	Hands-on Lab
Closing	Wrap up, next steps and Q&A

Introduction

Microsoft IoT is already delivering great results across industries

FINNING

"Fining's IoT solution has enabled customers to quickly solve business problems from a dashboard, transport more than 1 million additional tons of cargo via machine learning, reduce fuel consumption by reducing ideling by 17%, and increase ROI and competitiveness for the long term."



Rolls-Royce

Cutting fuel usage by 1 percent could save **\$250,000 per plane per year**



Tetra Pak's IoT business results show down-time cut down by up to 48 hours for each packaging line saving up to 30,000 Euros for customers."

HERSHEY'S

Ensure the licorice extruders on Twizzler's production line are **performing at peak optimization, saving over \$500K/year** on licorice alone.

Johnson Controls

Chillers restart **9x faster** than unconnected equipment, avoiding more than **\$300,000** in hourly downtime costs

RAC

By analyzing driving trends on its own patrol fleet, RAC has **reduced its accident rate by 25%, and reduced fuel usage by 20% - reporting annual savings of \$1.8 million**

DUNAV NET

By telling farmers such things as when to irrigate, how to control diseases and where to fight pests, agroNET provides an action plan to maximize efficiency. This solution has seen yield increases of 30% due to data & machine learning informed irrigation decisions and reductions in water use by 20%."

Rockwell Automation

Improves **access** to production and supply chain **data** worldwide, reducing downtime costs by as much as **\$300,000 per day**



thyssenkrupp

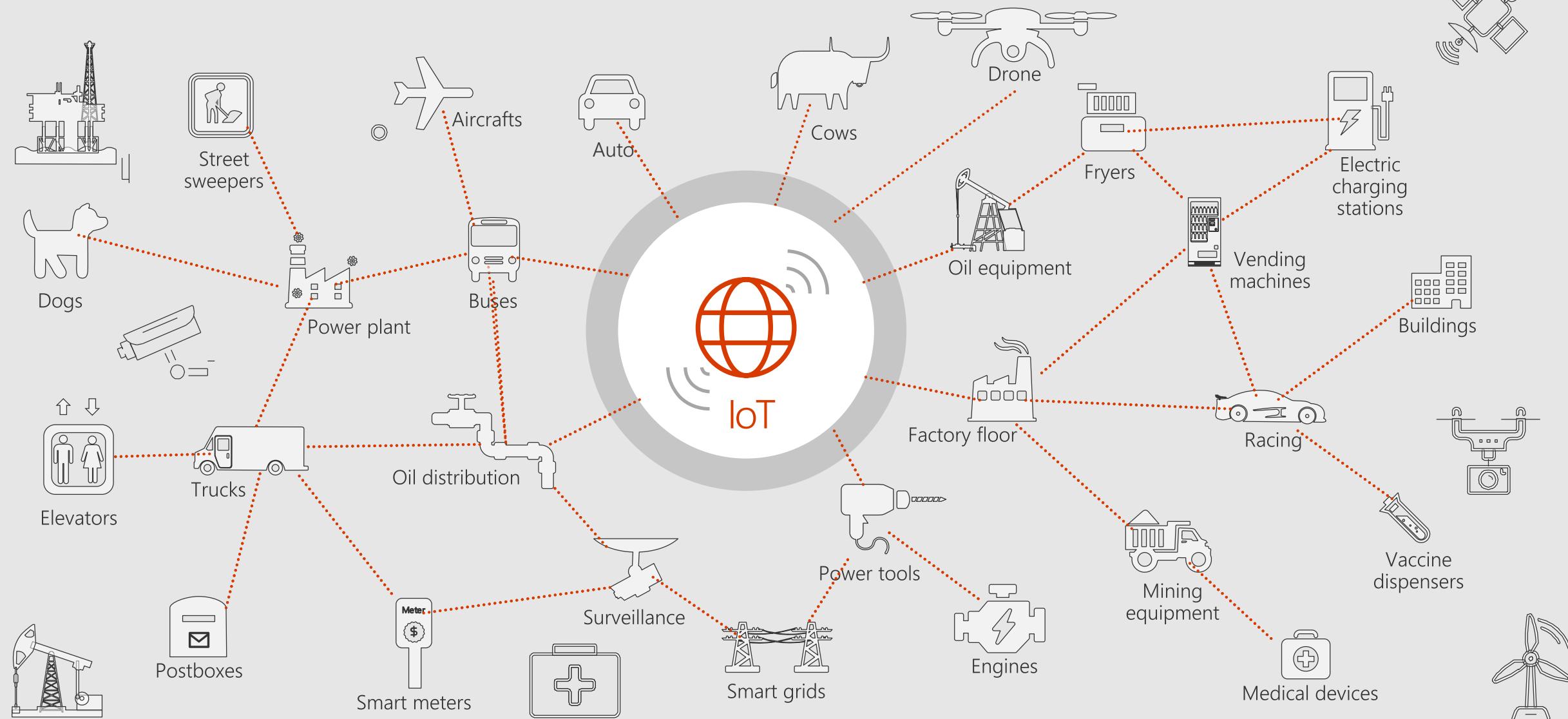
Gathers data from sensors and systems to create valuable business intelligence and **reduce downtime by 50%**



IoT Solutions Have a Common Pattern



The IoT Pattern is Broadly Applicable Across Industries



IoT projects can be complex



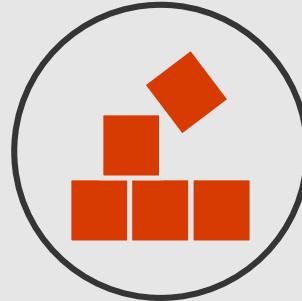
Difficult to maintain cohesive **security**



Time-consuming to get started

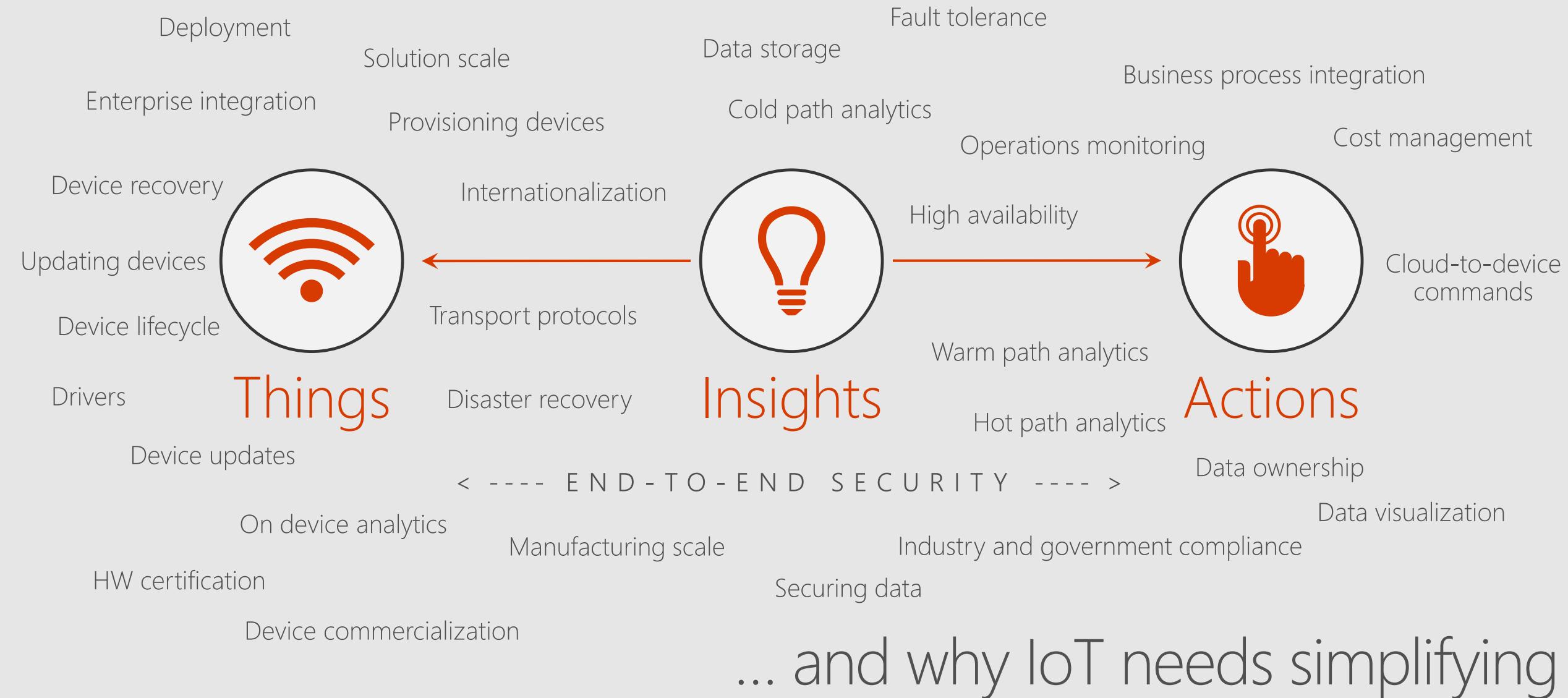


Incompatible with existing infrastructure



Challenging to scale over time

A more realistic view...



Microsoft is simplifying IoT

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

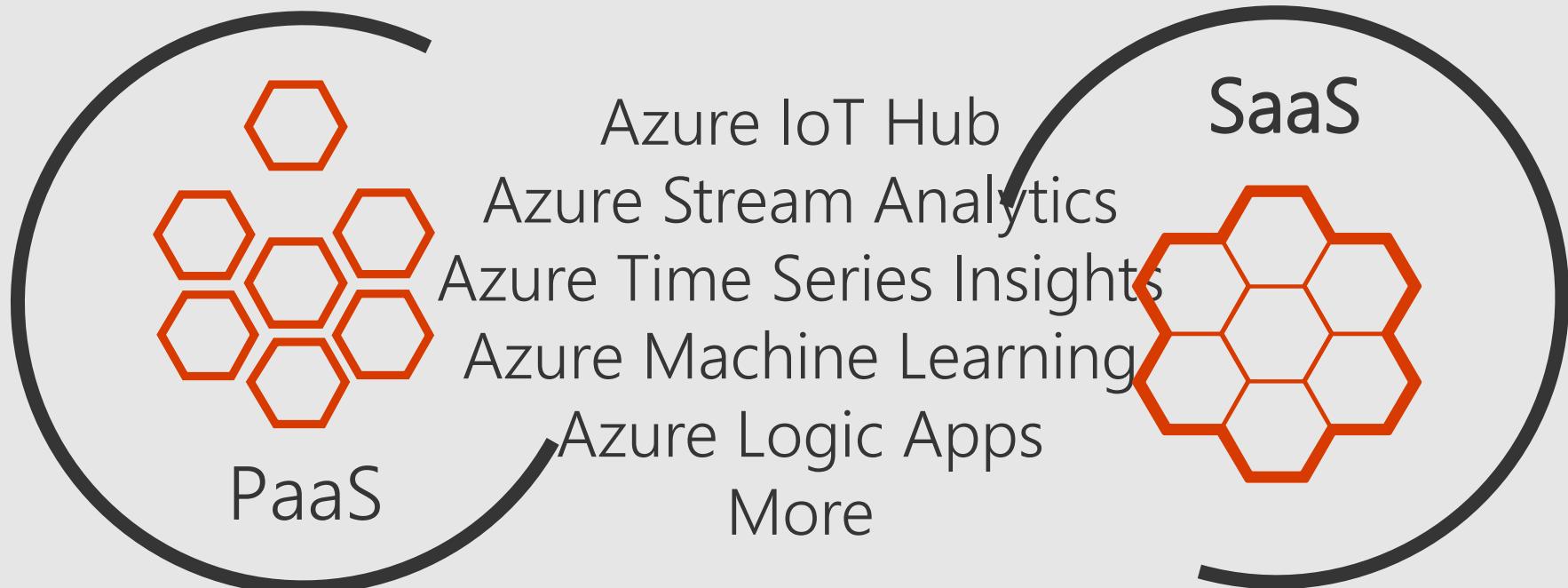
Preconfigured solutions for common IoT scenarios



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 – Remote Monitoring Update



Device Connectivity & Management



Data Ingestion and Command & Control



Stream Processing & Predictive Analytics



Workflow Automation and Integration



Dashboards and Visualization



Preconfigured Solutions

Remote monitoring

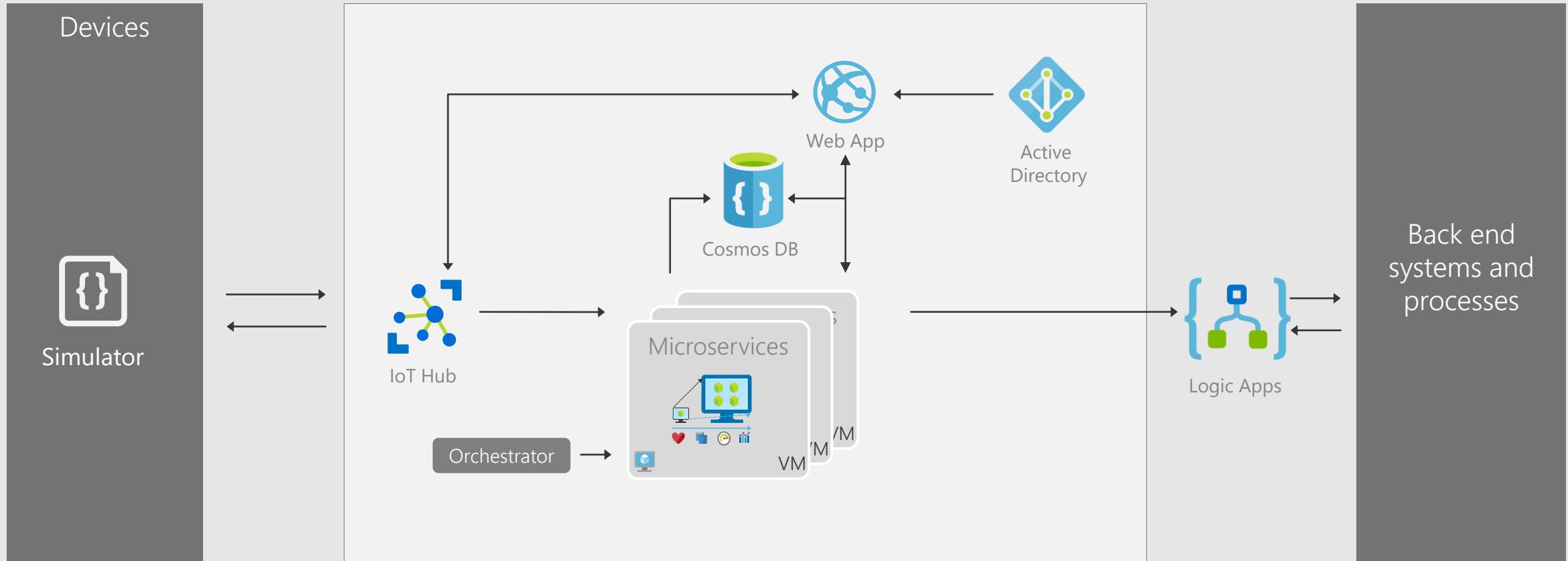
Predictive maintenance

Connected factory



New Microservice Based Architecture

Available in both .NET and Java, both Open Source



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 IoT Central

Builders



 Product Modeler

 Digital-twin Management

 Template Management

 Rules & Workflows

Administrators



 App Manager

 User Management

 Identity Management

Operators



 Intuitive discoverability

 Asset Visualizations

 Time-series Insights

 Device Management

Microsoft is simplifying IoT

Azure IoT Suite

Preconfigured solutions for common IoT scenarios



Remote Monitoring | Predictive Maintenance | Connected Factory

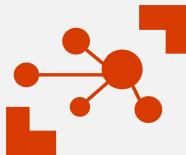
Microsoft IoT Central

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

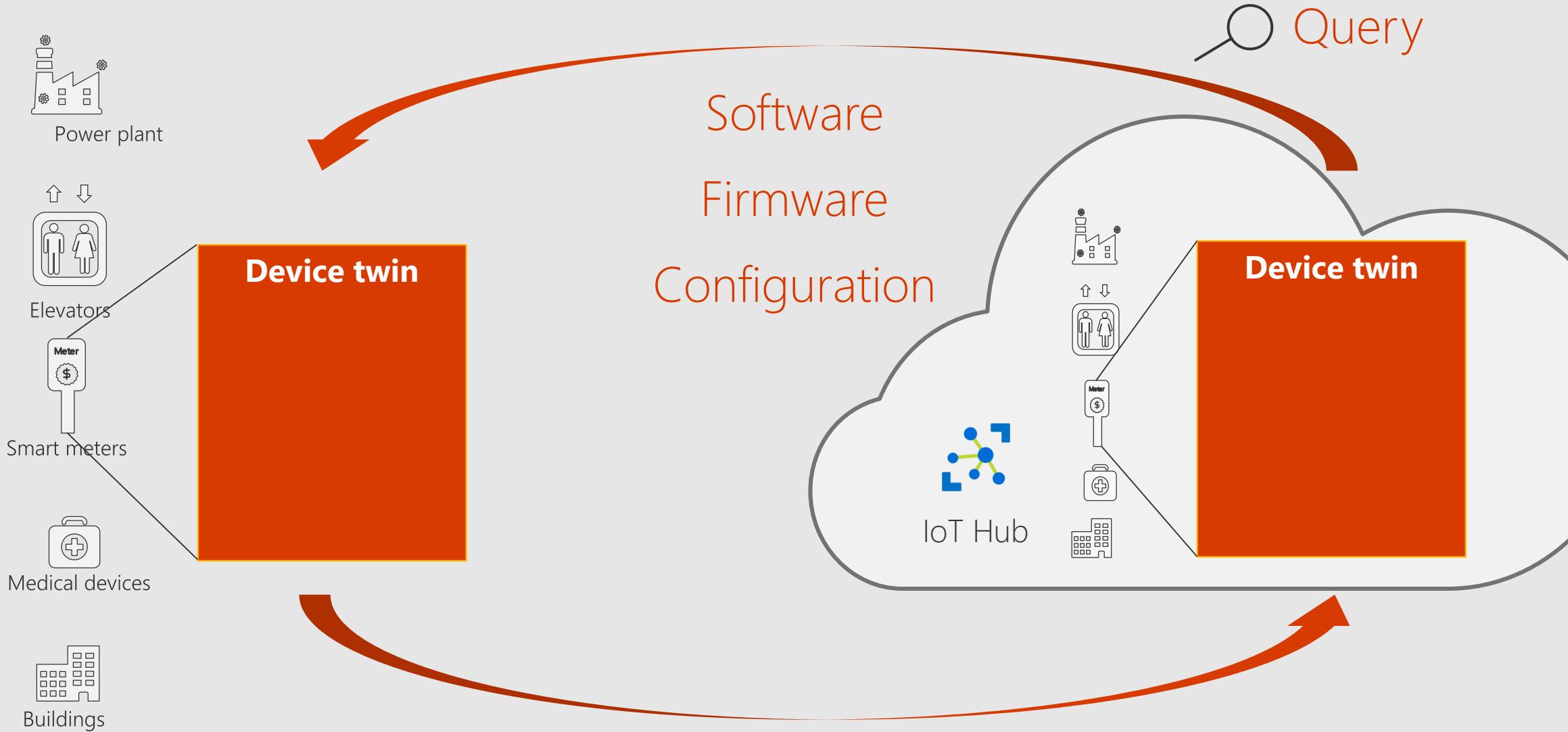
X.509 Support

IP Whitelisting/Blacklisting

Shared Access Policies

Firmware/Software Updates

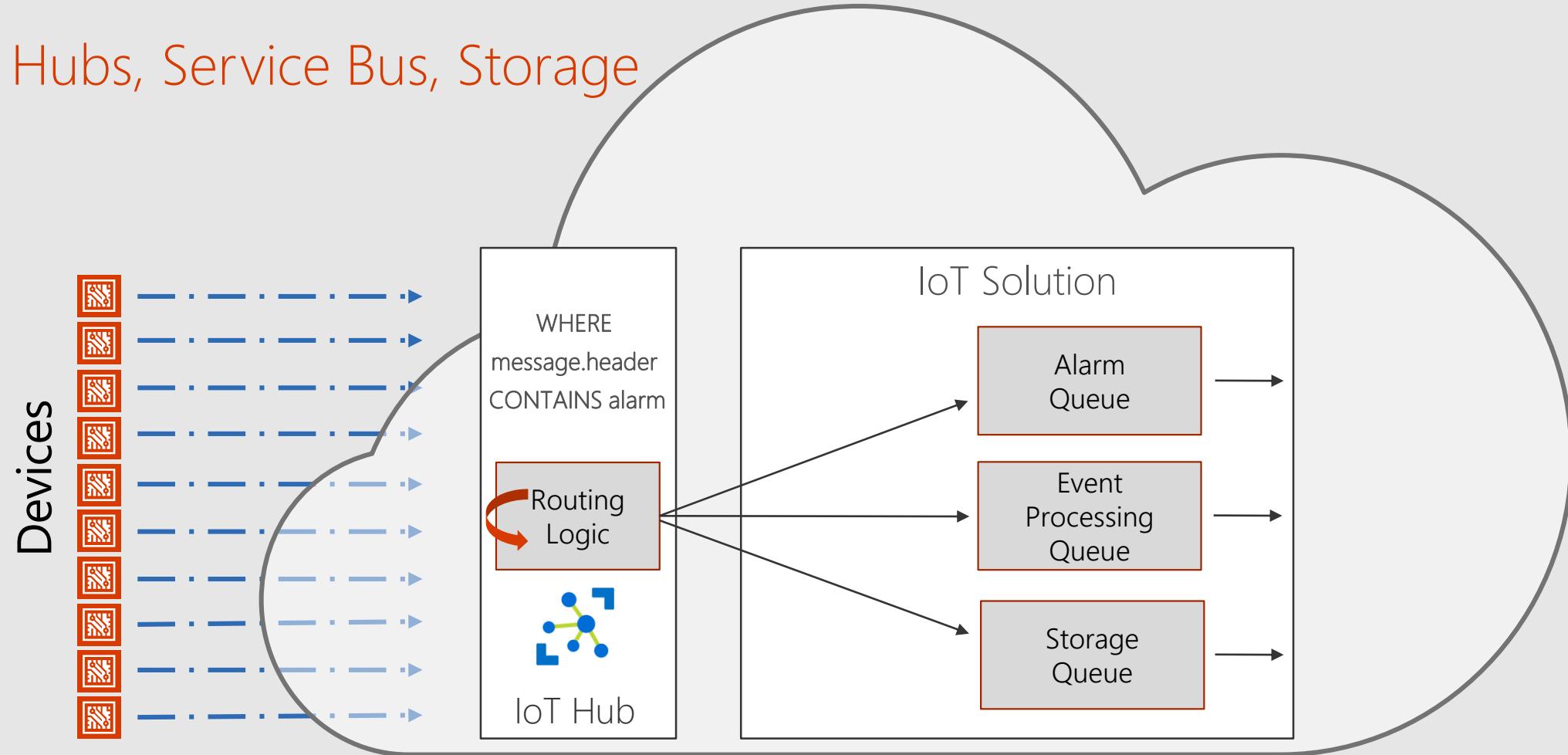
Azure IoT Hub Device Management



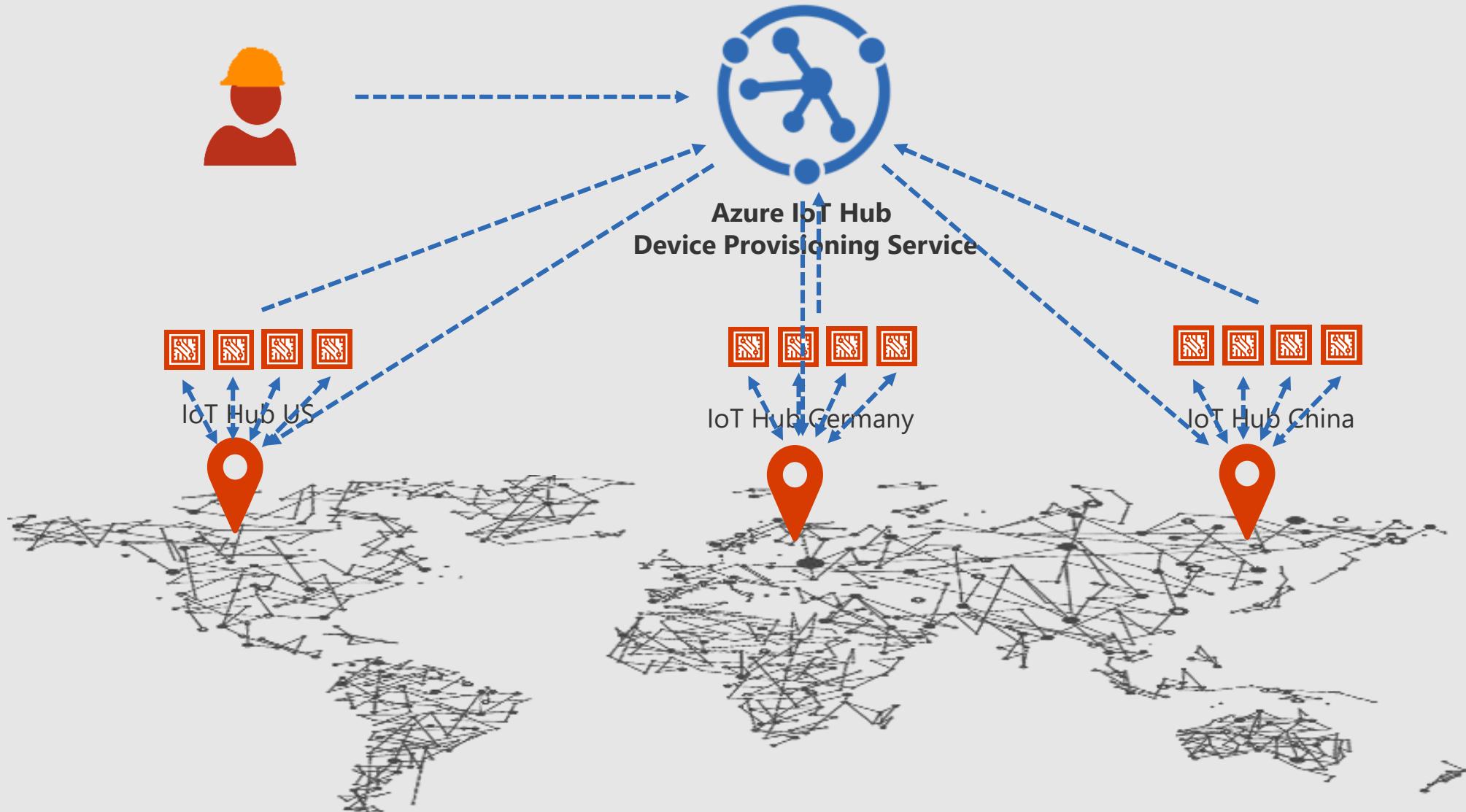
Azure IoT Hub Message Routes

Automatically route messages based on headers or JSON message body

Event Hubs, Service Bus, Storage



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

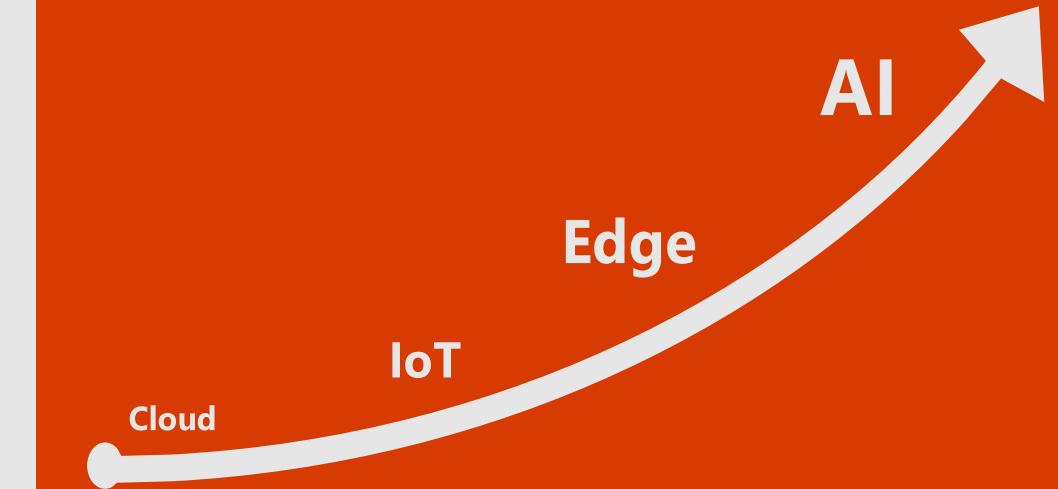
Harnessing signals from sensors and devices, managed centrally by the cloud

Edge

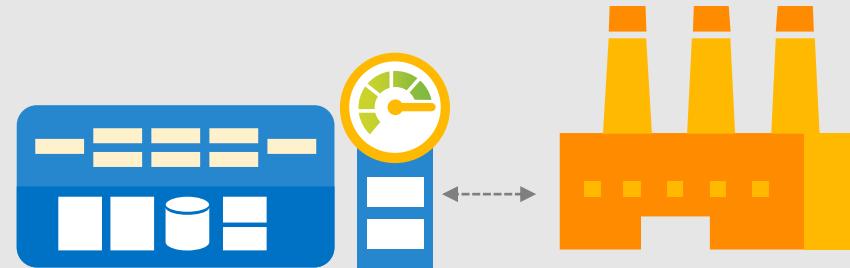
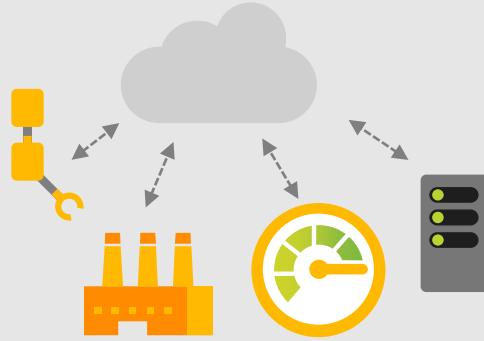
Intelligence offloaded from the cloud to IoT devices

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

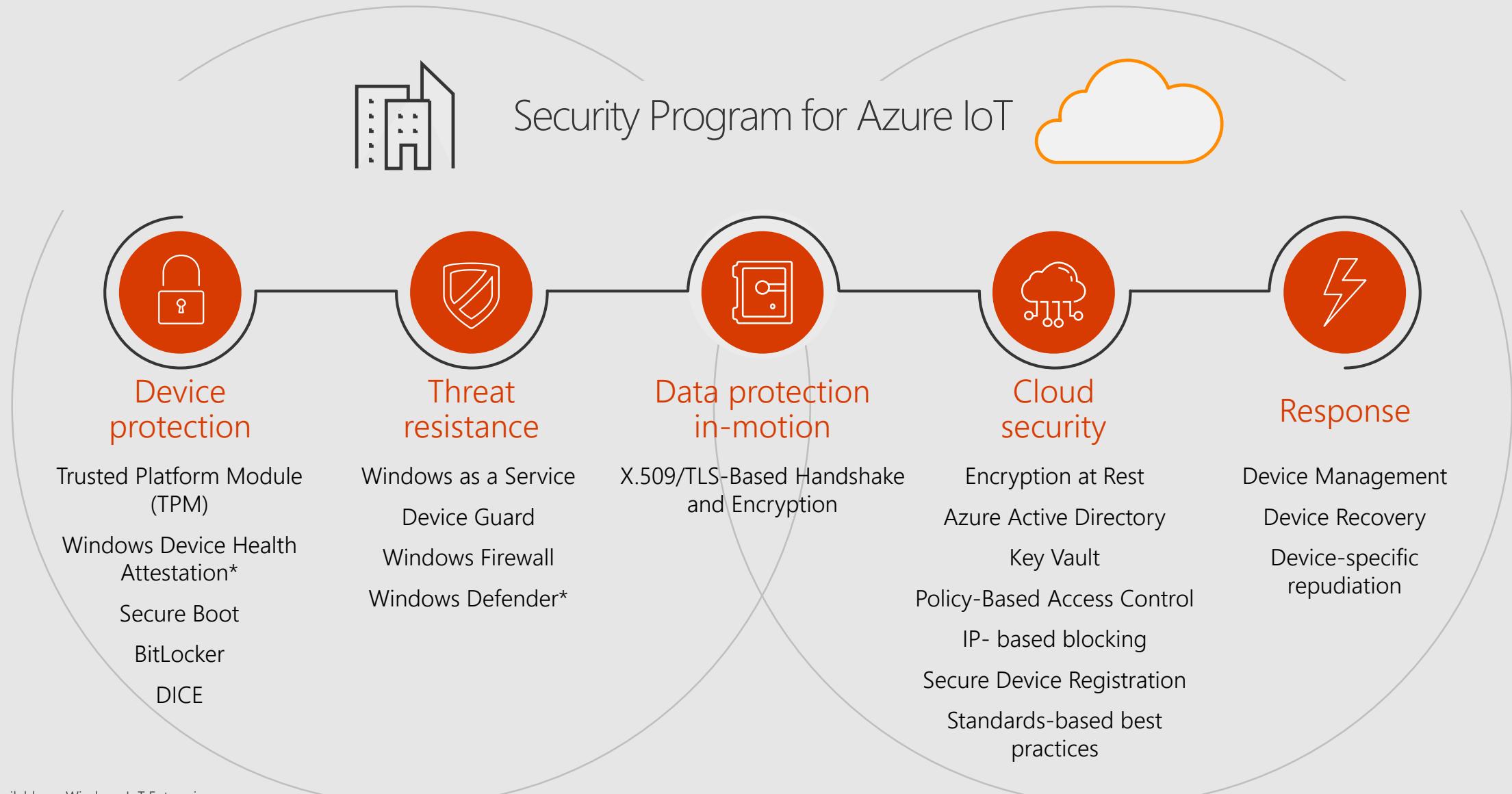
Low latency tight control loops require near real-time response

Protocol translation & data normalization

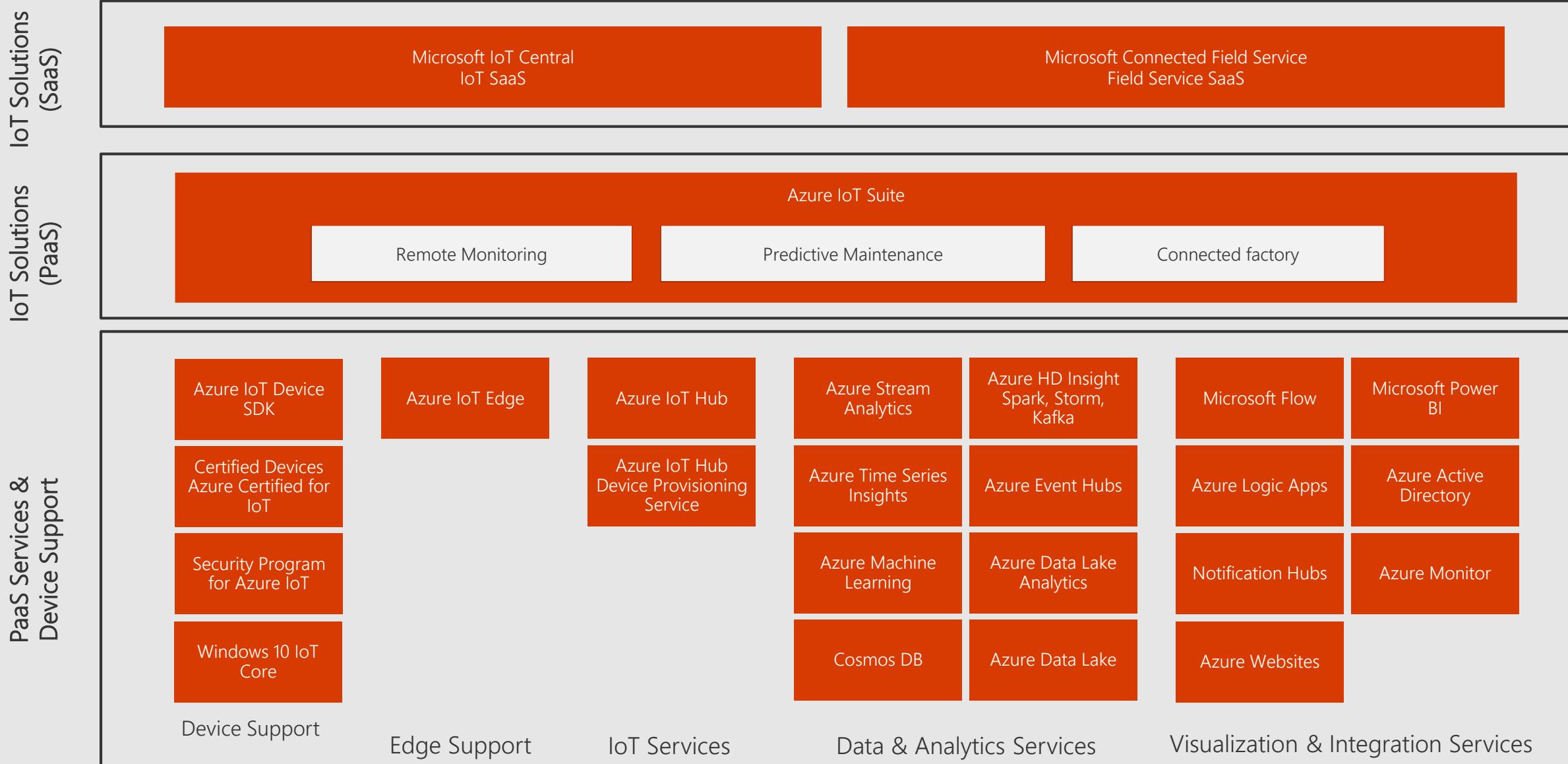
Privacy of data and protection of IP

Symmetry

Device to Cloud Security



A Comprehensive set of offerings for IoT



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

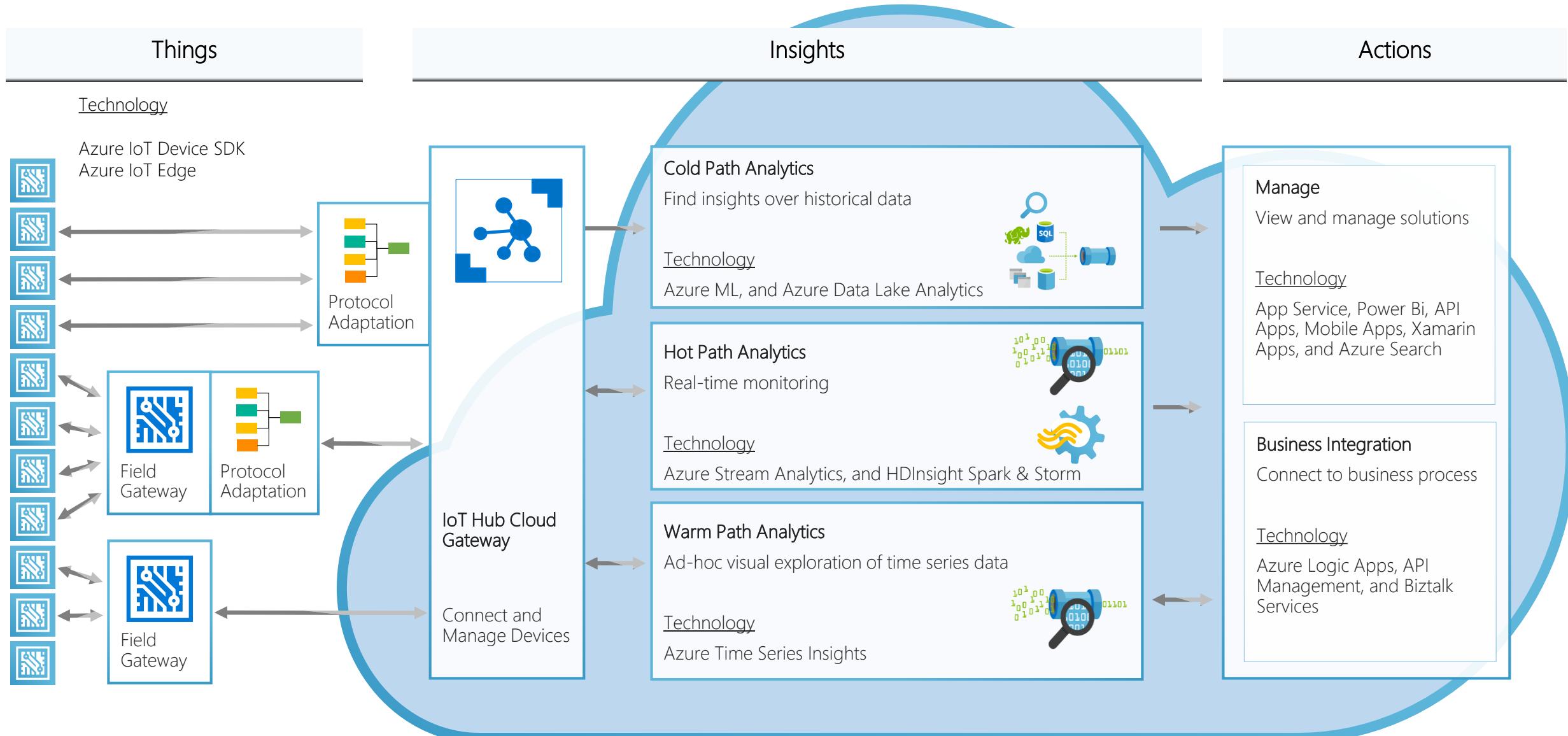


Predictive Maintenance



Connected Factory (new)

Azure IoT Suite: PaaS approach to IoT solutions



Preconfigured solutions

Solution types



Remote monitoring

Connect and monitor your devices to analyze untapped data and improve business outcomes by automating processes.



Connected factory

Accelerate your journey to Industrie 4.0 – connect, monitor and control industrial devices for insights using OPC UA to drive operational productivity and profitability.



Predictive maintenance

Anticipate maintenance needs and avoid unscheduled downtime by connecting and monitoring your devices for predictive maintenance.

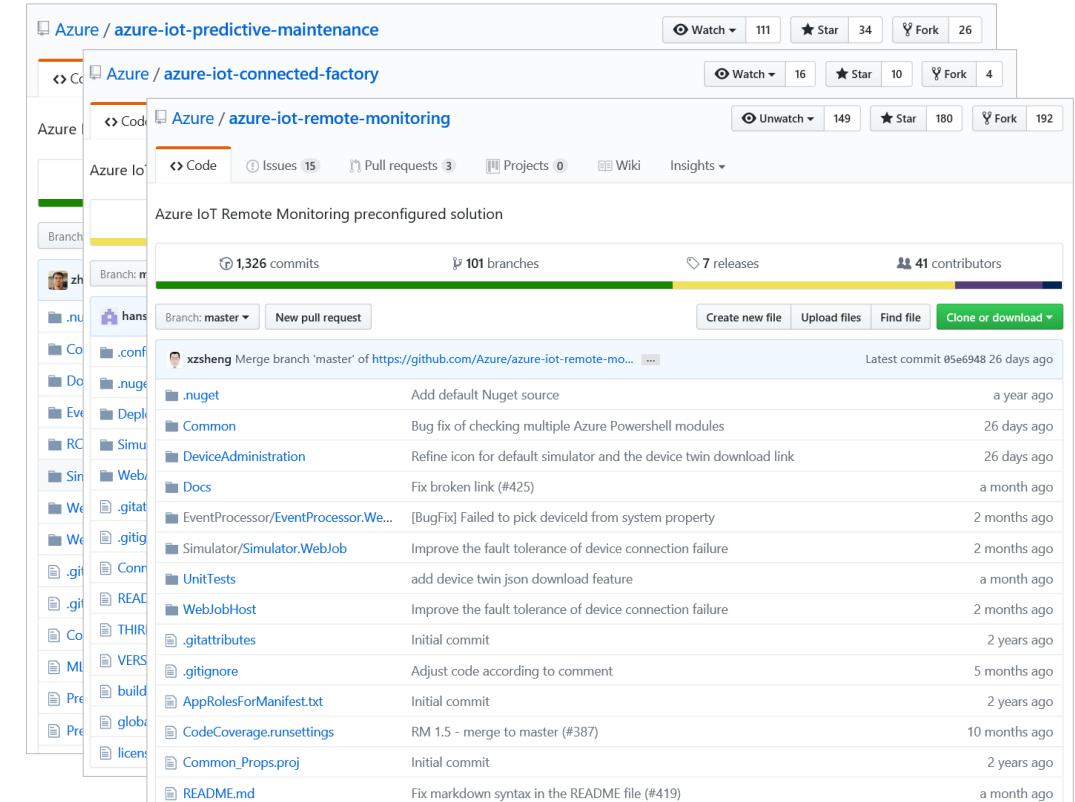
Select

Select

Select

www.azureiotsuite.com

Open sourced code



Azure / [azure-iot-predictive-maintenance](https://github.com/Azure/azure-iot-predictive-maintenance) Watch ▾ 111 ★ Star 34 Fork 26

Azure / [azure-iot-connected-factory](https://github.com/Azure/azure-iot-connected-factory) Watch ▾ 16 ★ Star 10 Fork 4

Azure / [azure-iot-remote-monitoring](https://github.com/Azure/azure-iot-remote-monitoring) Unwatch ▾ 149 ★ Star 180 Fork 192

Azure / [Azure / azure-iot-remote-monitoring](https://github.com/Azure/azure-iot-remote-monitoring) Issues 15 Pull requests 3 Projects 0 Wiki Insights ▾

Azure IoT Remote Monitoring preconfigured solution

1,326 commits 101 branches 7 releases 41 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download ▾

xzsheng Merge branch 'master' of https://github.com/Azure/azure-iot-remote-mo... Latest commit 05e6948 26 days ago

.nuget Add default NuGet source a year ago

Common Bug fix of checking multiple Azure Powershell modules 26 days ago

DeviceAdministration Refine icon for default simulator and the device twin download link 26 days ago

Docs Fix broken link (#425) a month ago

EventProcessor/EventProcessor.We... [BugFix] Failed to pick deviceld from system property 2 months ago

Simulator/Simulator.WebJob Improve the fault tolerance of device connection failure 2 months ago

UnitTests add device twin json download feature a month ago

WebJobHost Improve the fault tolerance of device connection failure 2 months ago

.gitattributes Initial commit 2 years ago

.gitignore Adjust code according to comment 5 months ago

AppRolesForManifest.txt Initial commit 2 years ago

CodeCoverage.runsettings RM 1.5 - merge to master (#387) 10 months ago

Common_Props.proj Initial commit 2 years ago

README.md Fix markdown syntax in the README file (#419) a month ago

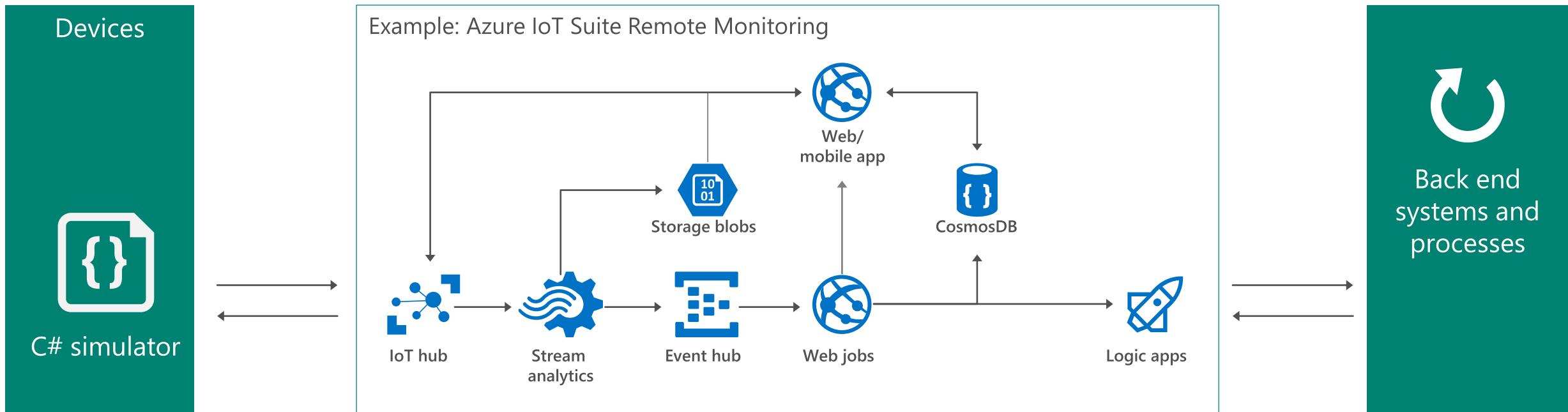
<https://github.com/Azure/azure-iot-predictive-maintenance>

<https://github.com/Azure/azure-iot-connected-factory>

<https://github.com/Azure/azure-iot-remote-monitoring>

Azure IoT Suite preconfigured solution

Remote monitoring | Predictive maintenance | Connected factory



Azure IoT Suite

Remote Monitoring V2

Preview released October 2017

GOALS



Beautiful & Functional



Agile & Scalable

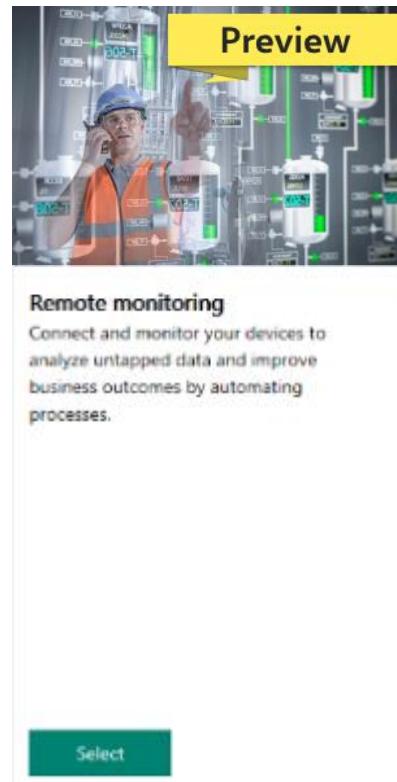


Customizable

Provisioning a Solution

www.azureiotsuite.com

Preview
October 2017



The screenshot shows a preview of the Azure IoT Suite. At the top right, there's a yellow "Preview" button. Below it, a section titled "Remote monitoring" describes connecting and monitoring devices to analyze untapped data and improve business outcomes by automating processes. A "Select" button is at the bottom.

Create Remote monitoring solution

Solution details

When to use this deployment:

Basic deployment is geared toward showcasing the solution. To reduce the cost of this demonstration, all of the microservices are deployed in a single virtual machine; this is not considered a production-ready architecture.

Our Standard deployment option should be used when you are ready to customize a production-ready architecture, built for scale and extensibility.

Creating a Basic solution will result in the following Azure services being provisioned into your Azure subscription at cost:

- 1 Azure Active Directory application
- 1 Virtual Machine (Standard D1 V2 (1 core, 3.5 GB memory))
- 1 IoT Hub (\$1 - Basic tier)
- 1 Cosmos DB Account (Standard)
- 1 Storage account (Standard-GRS)
- 1 Web Application

To learn more about our deployment options visit our [GitHub repository](#)

Approximate Cost:

The cost of the solution is an aggregate of the cost of the underlying Azure services. Pricing information for these services can be found [here](#). Usage amounts and billing details for your subscription can be found in the [Azure Portal](#).

In addition to the above Azure services, creating a solution will result in your being signed up for a subscription to the following Azure Marketplace offering(s), which are subject to the following terms:

[Bing Maps API for Enterprise \(Internal Website Transactions Level 1\)](#): terms of use and privacy statement.

Deployment option

Basic

Standard

Language

.NET

Java

Solution name

IgniteDemo0926

Subscription

Azure IoT Devices Product Team

Region

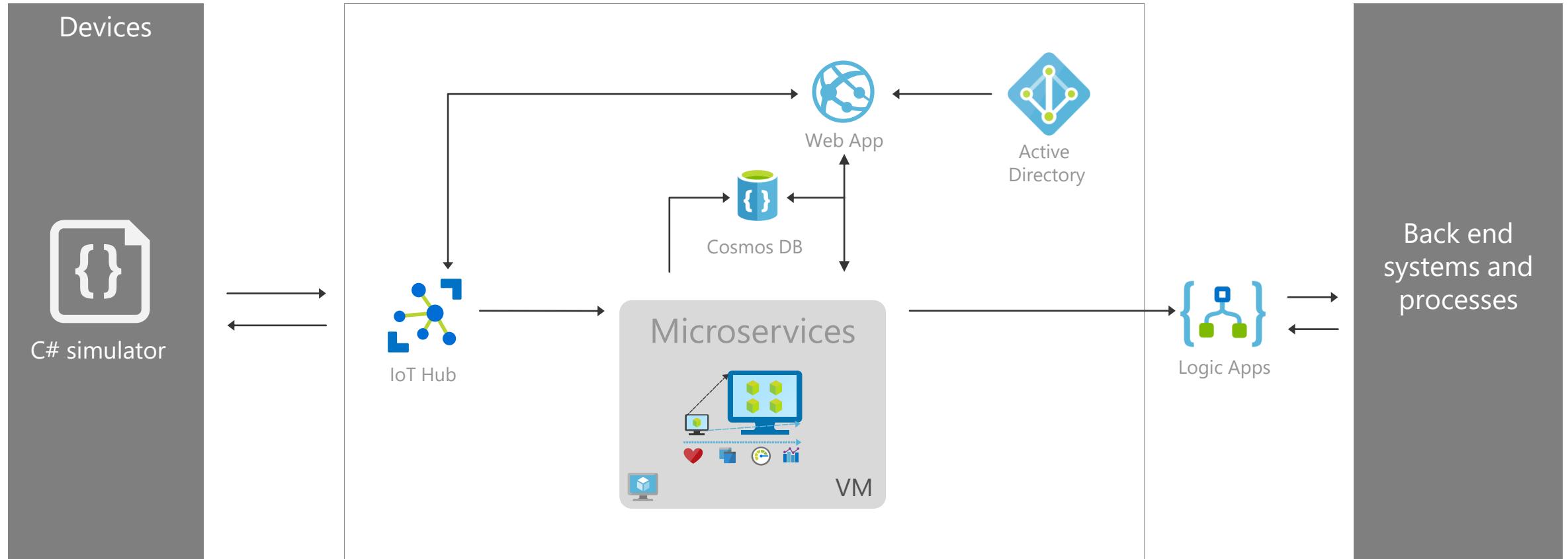
East US

Create solution

Cancel



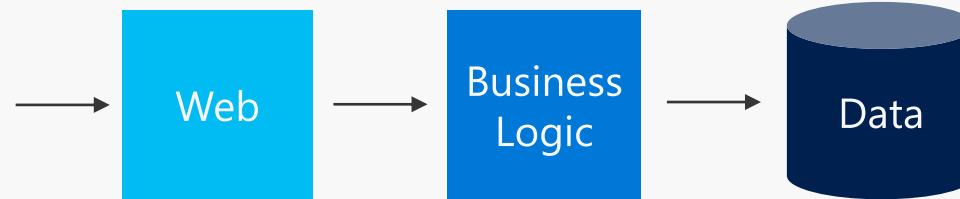
Azure IoT Suite Remote Monitoring – Basic Deployment



Application Models

Monolithic

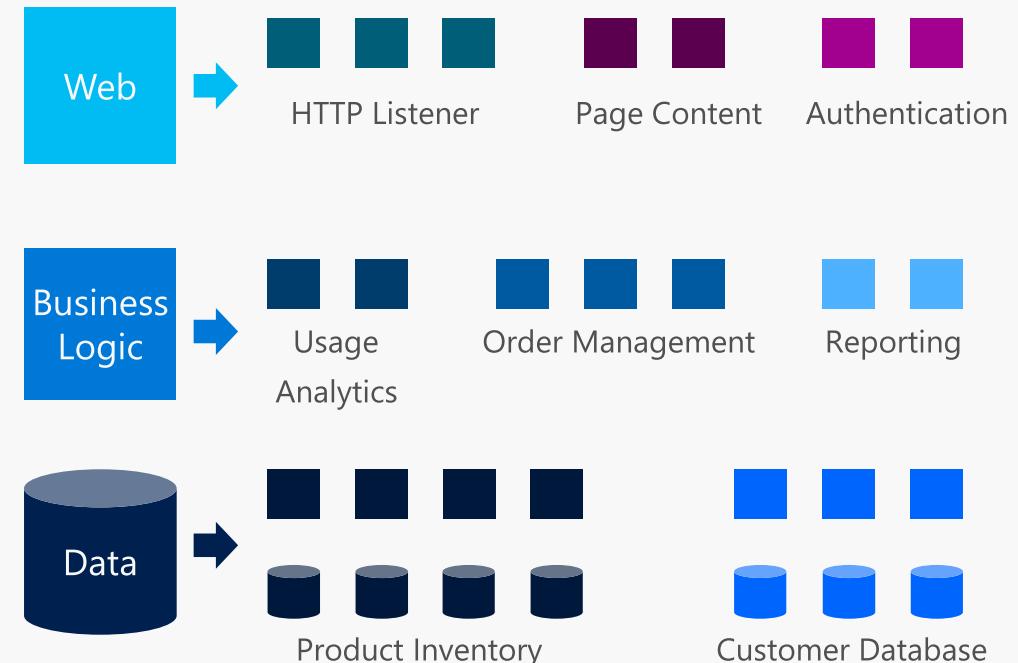
HW Influenced – scale, performance (caching)
Static – little incentive to decompose



3-Tier Monolithic Application

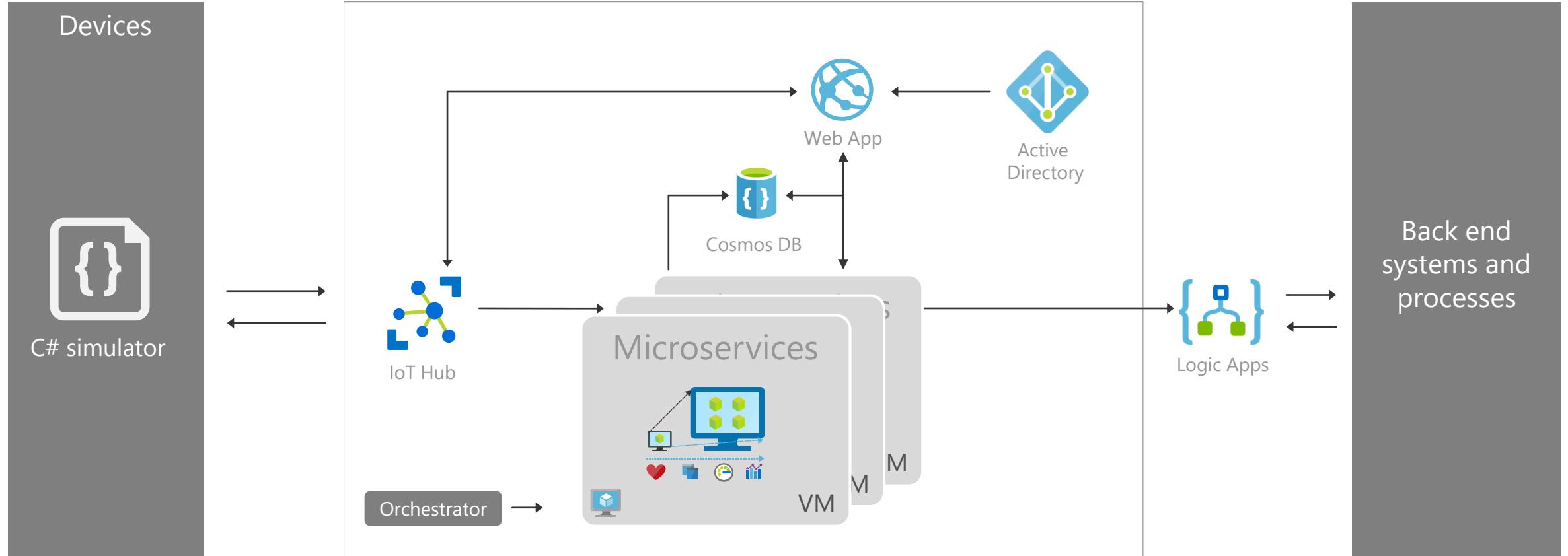
Microservices

Agility, Scale, Reliability
Deliver Isolated Value
Well defined contracts (RESTful interfaces)
Versioned Independently
Loose Coupling → Rapid Evolution

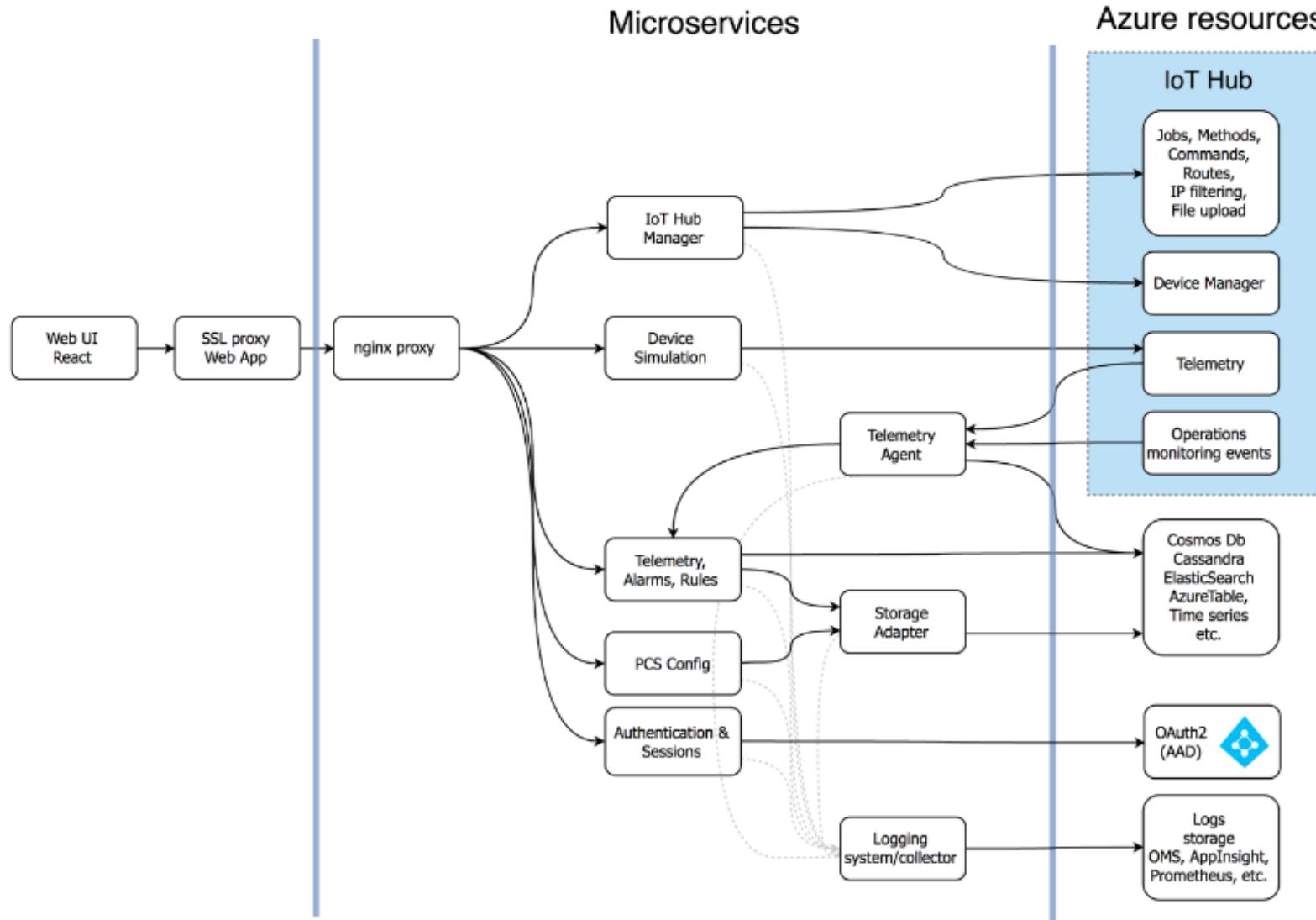


Breaking the Monolith into Microservices

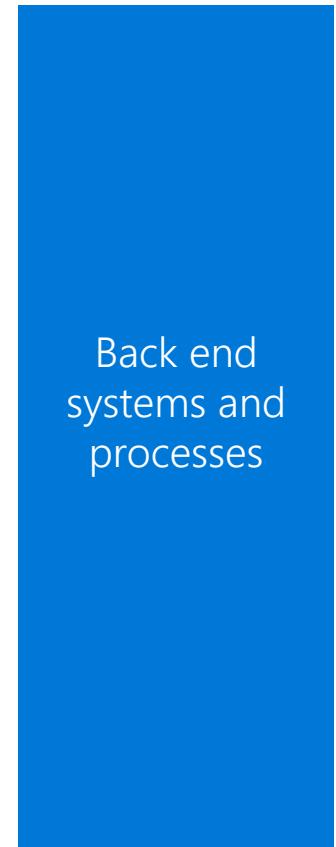
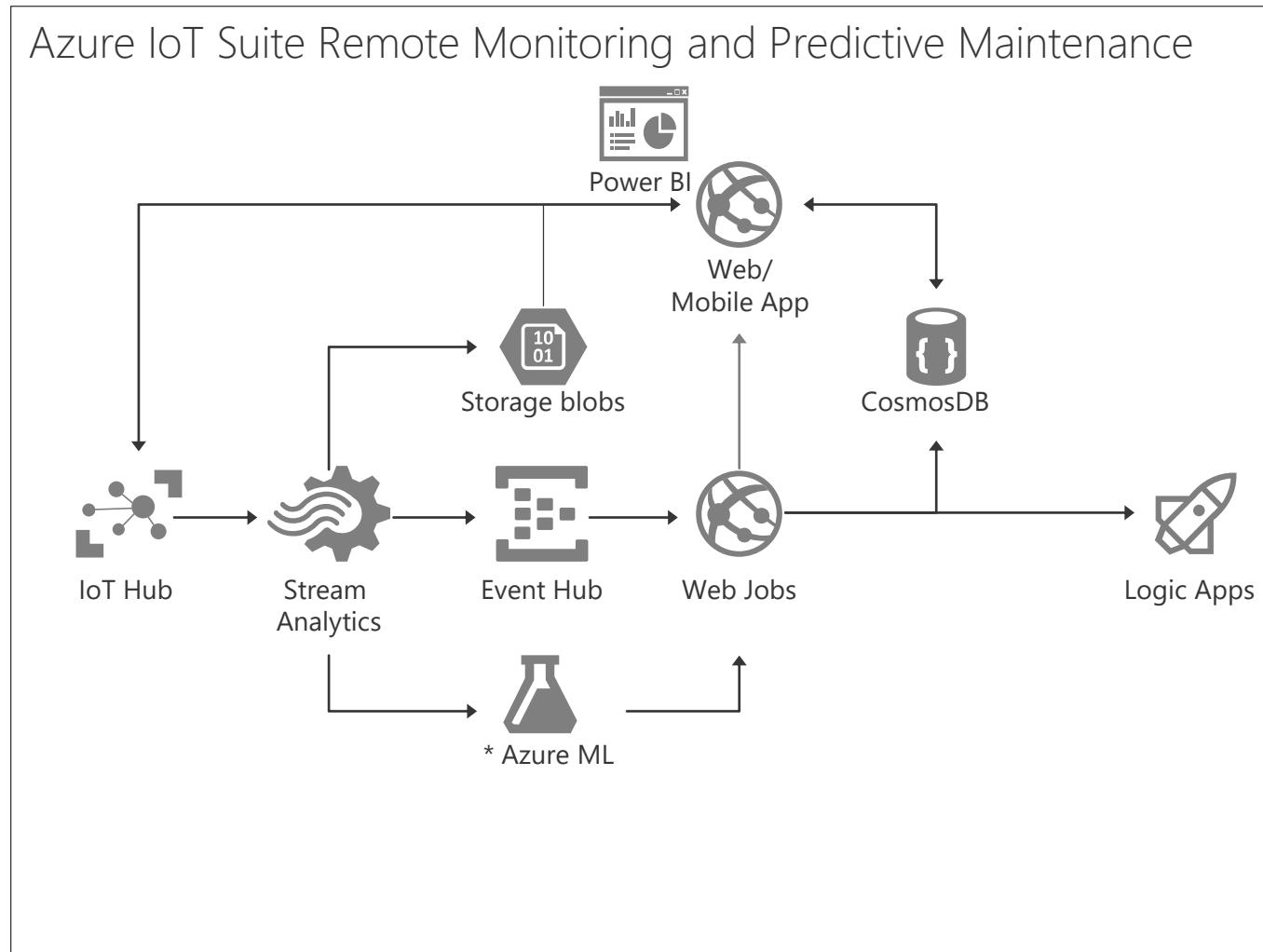
Azure IoT Suite Remote Monitoring – Standard Deployment



Remote Monitoring Microservices Architecture

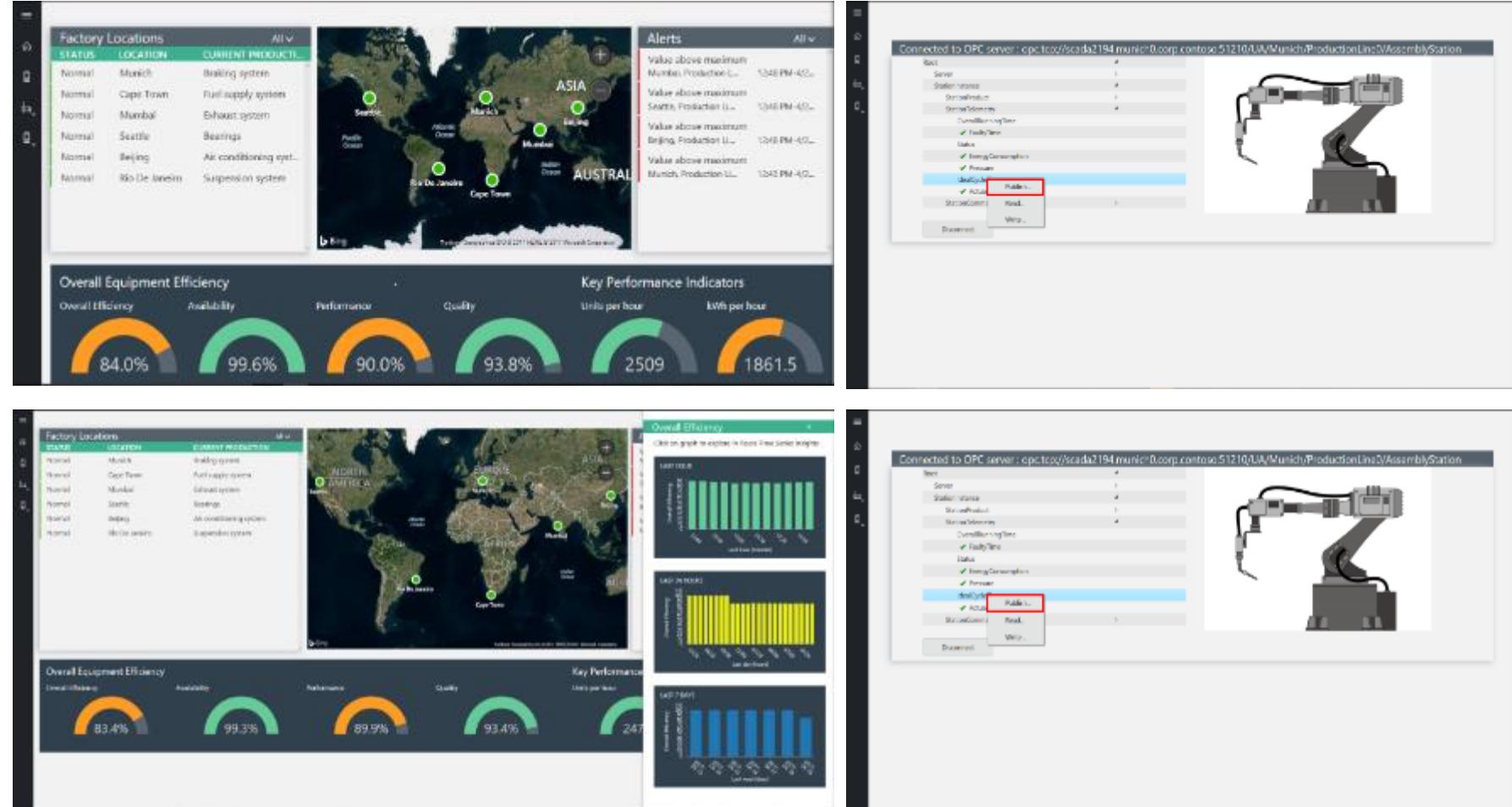


What is behind these IoT Suite preconfigured solutions



PCS: Connected Factory azureiotsuite.com

- Connect to OPC Devices via Azure IoT Gateway
- Get started in minutes, simple to configure
- Decide which data to ingest from OPC devices
- Find insights using included Azure Time Series Insights
- Optionally call methods on OPC devices to control them remotely



Customize your IoT Architecture

Elements of Azure IoT Suite

1. Connect and Manage Devices & Gateways



Preconfigured solutions



Gateway & Devices



Connect and control

2. Analyze streaming data



Real time analytics



Data visualization



Predictive analytics*

3. Integrate into business systems



Workflow integration



Push and broadcast notifications



ID and access management

4. Secure IoT Infrastructure



5. Customize IoT Architecture

* Only applies to Predictive Maintenance

Elements of Azure IoT Suite

1. Connect and Manage Devices & Gateways



Preconfigured solutions



Gateway & Devices



Connect and control

2. Analyze streaming data



Real time analytics



Data visualization



Predictive analytics*

3. Integrate into business systems



Workflow integration



Push and broadcast notifications



ID and access management

4. Secure IoT Infrastructure



5. Customize IoT Architecture

* Only applies to Predictive Maintenance

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

Which service to choose?

Azure Platform Services

Security & Management



Compute



App Services



Developer Services



Hybrid Operations



Integration



Analytics & IoT



Data



Media & CDN



Virtual Machines



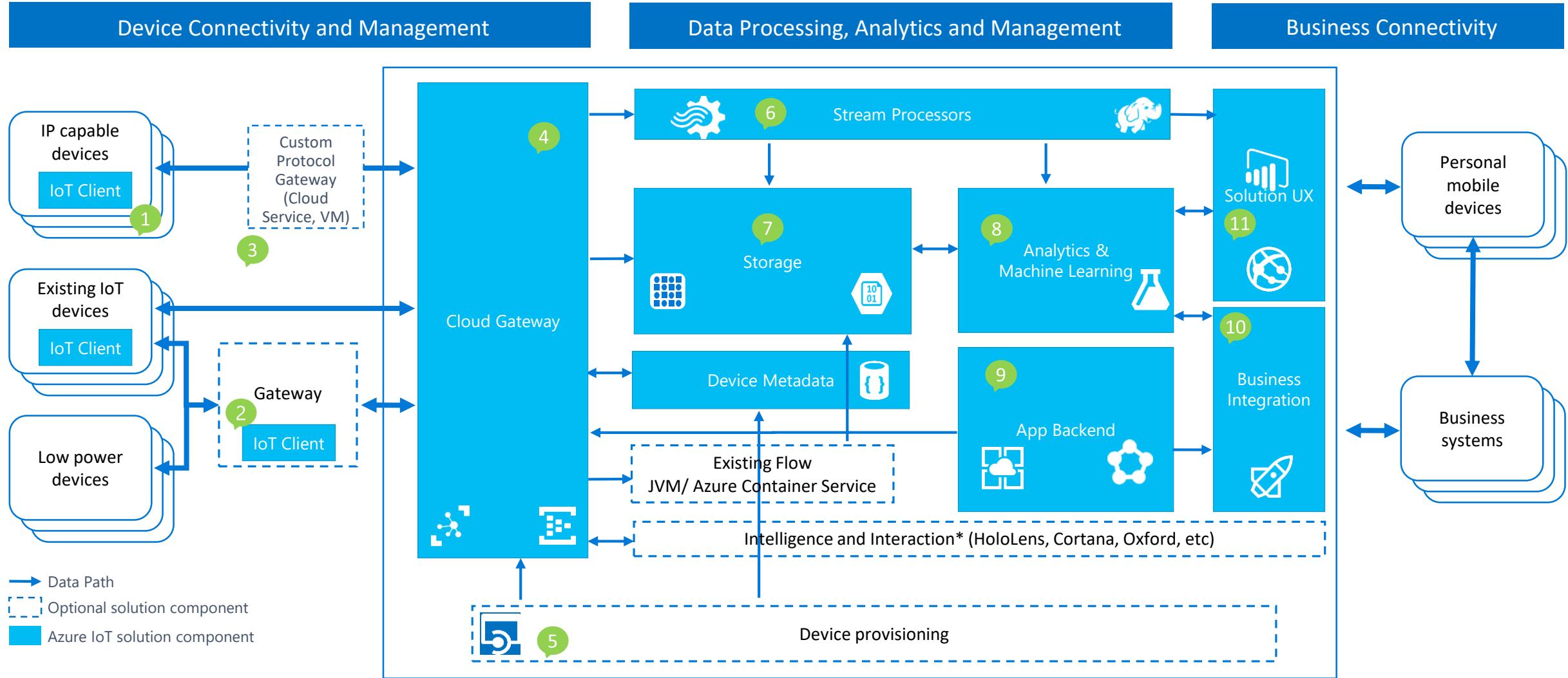
Storage



Networking



Azure IoT reference architecture

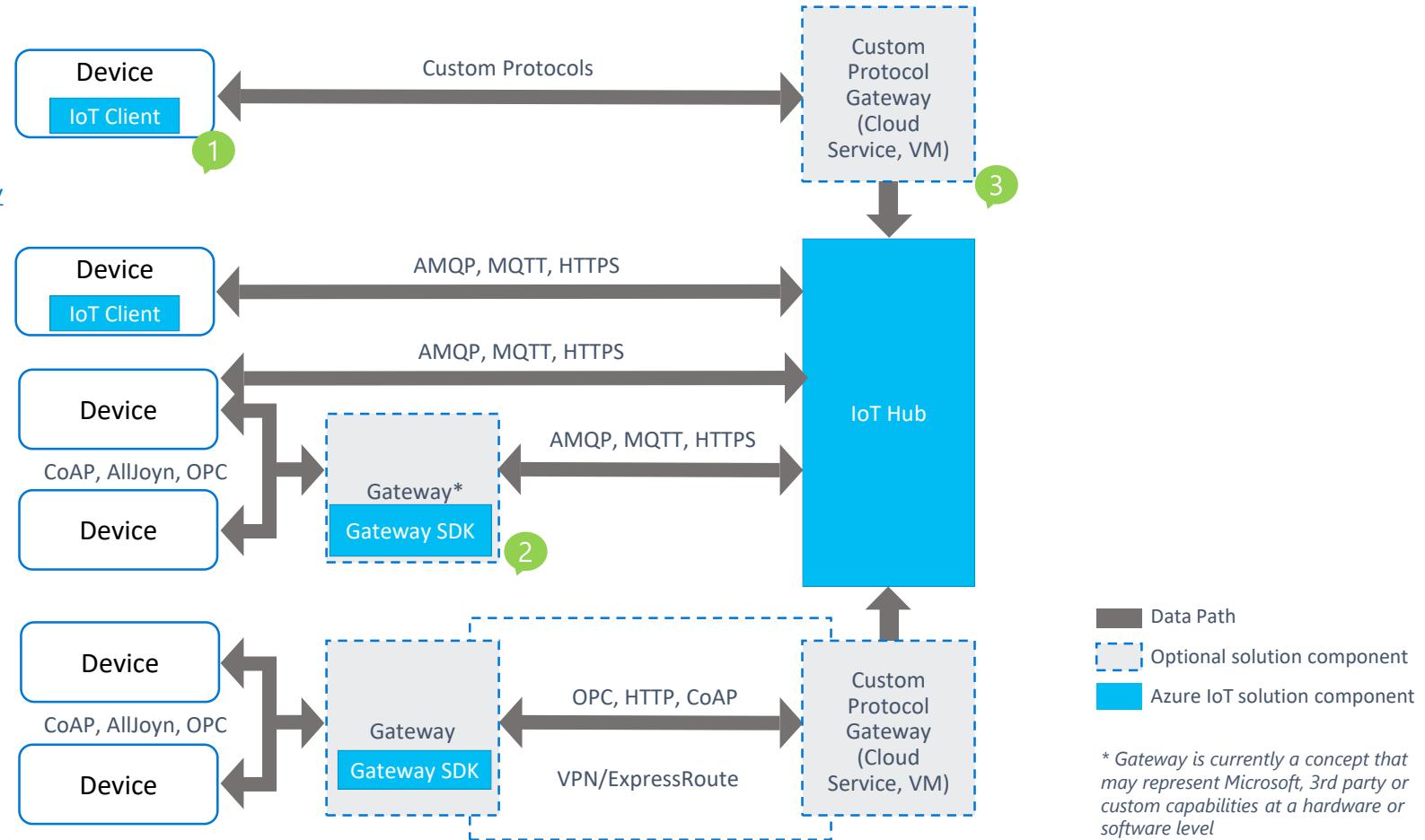


Device connectivity

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>



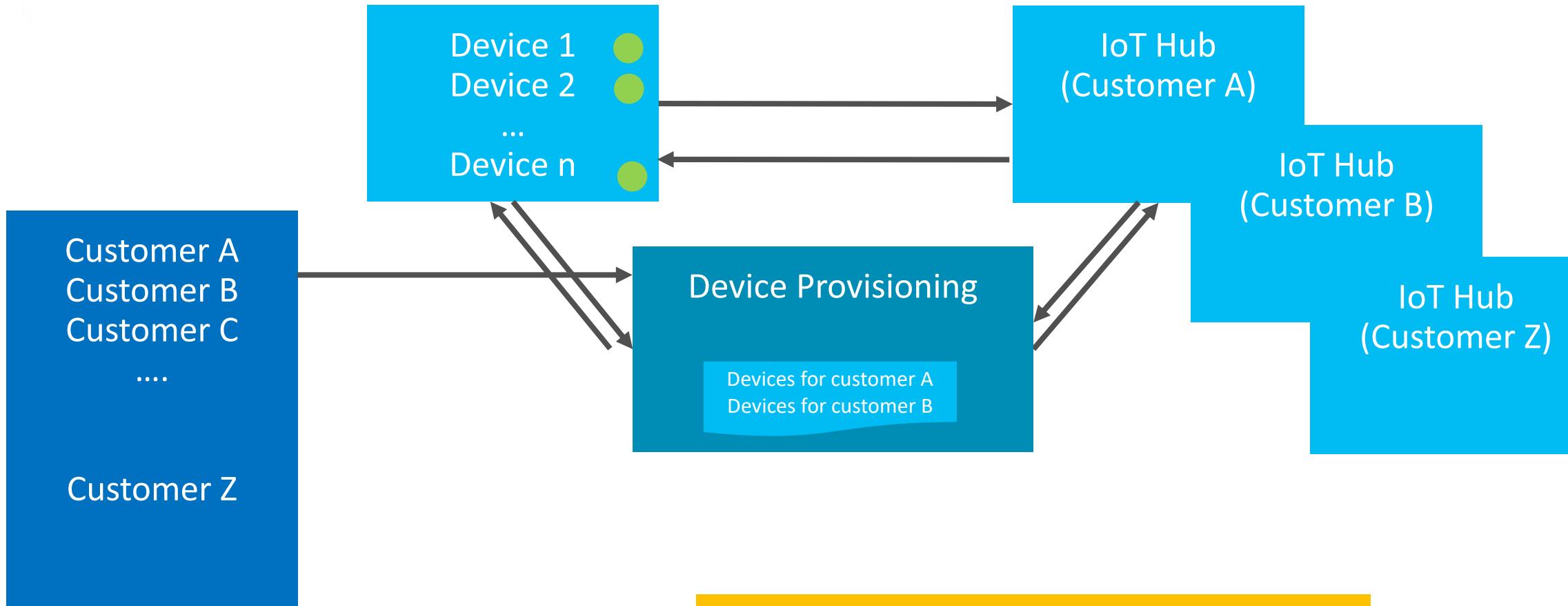
Scenarios addressed:

- Heterogeneity
- Network & Transport protocol

Cloud gateway

	Area	IoT Hub	Event Hub
Device Monitoring	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	Device twins 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.
Event Hub	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	Data Lake: 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.	SQL DW: 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	Unstructured data for text or binary data at massive scale (single container can be 500TB) Highly scalable Supports streaming scenarios and random access Data accessible via the REST API
Storage Queue	Durable message queueing for large workloads Rich client libraries for .NET, Java, Android, C++, Node.js, PHP, Ruby, and Python Data accessible via the REST API

Scenario addressed: Cold Storage

Queues considerations

	Storage Queue	Service Bus Queue
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

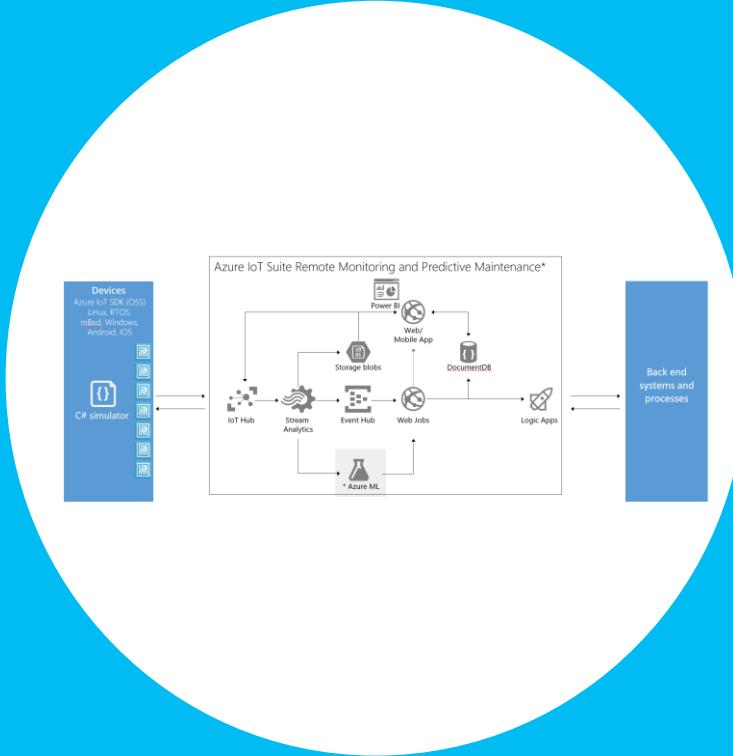
Solution UX considerations

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

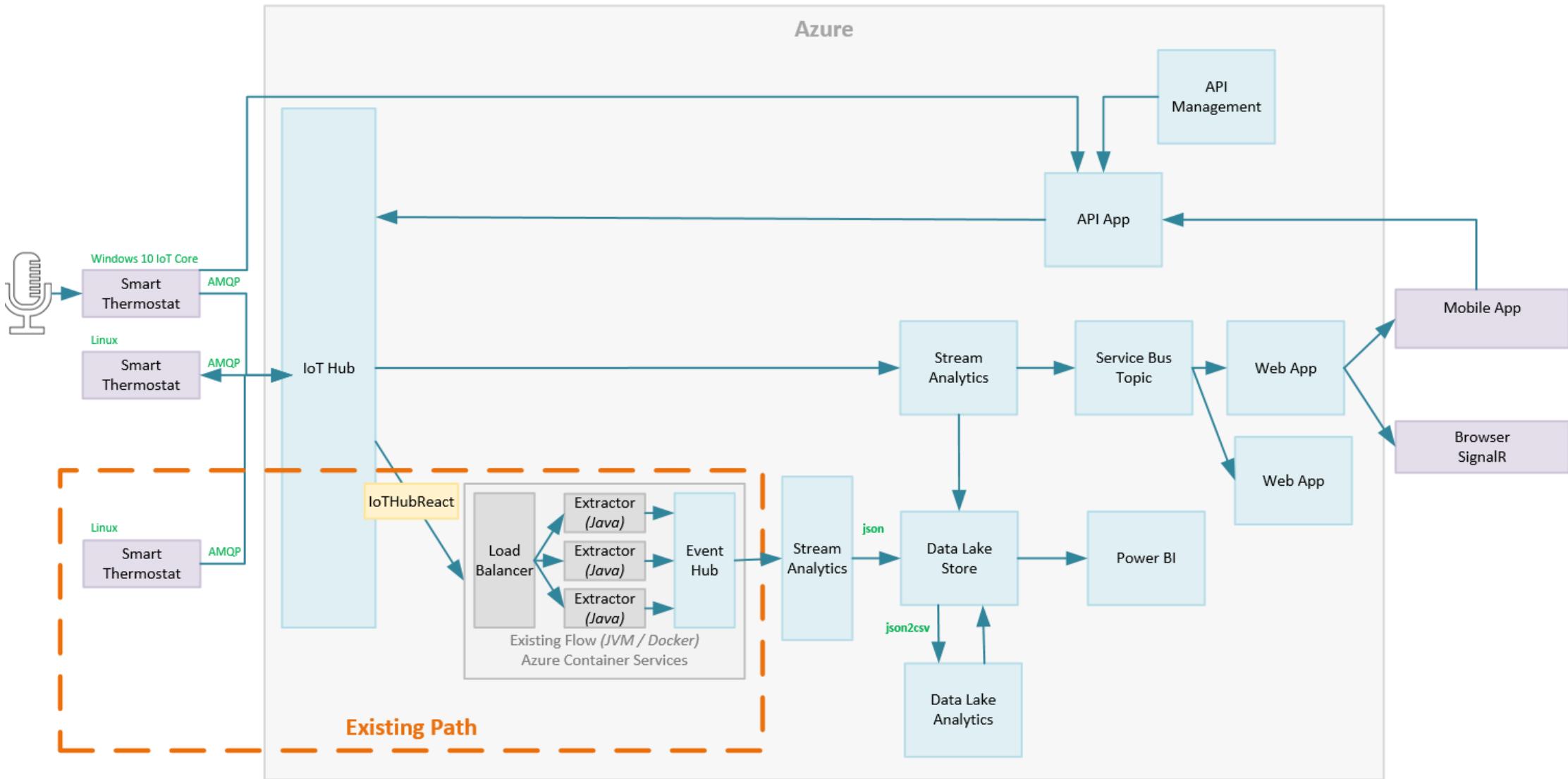
Scenario addressed:

- Visualization dashboards for operations

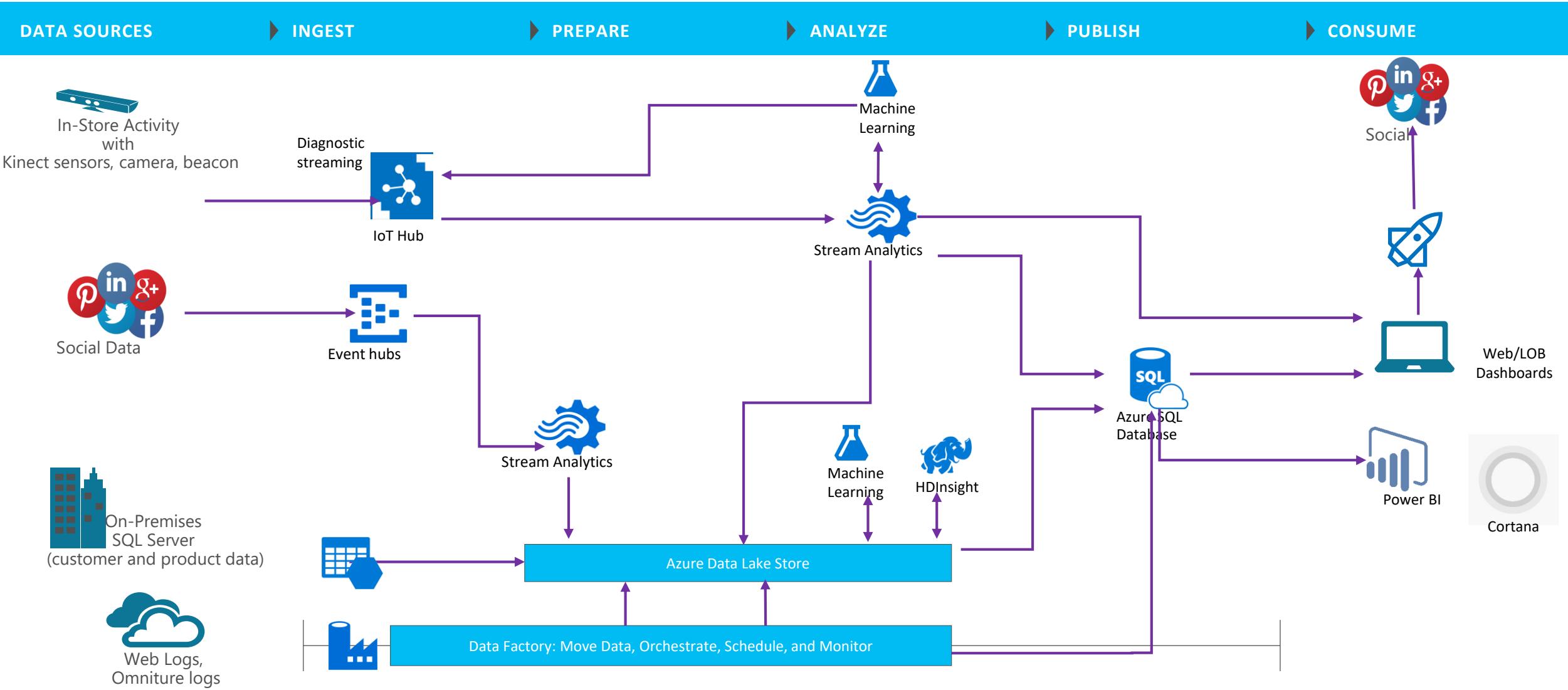
Custom architecture samples



Connected smart thermostat example

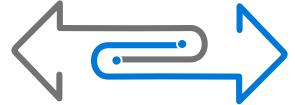


Retail architecture example



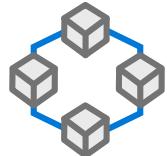
Azure IoT Hub

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

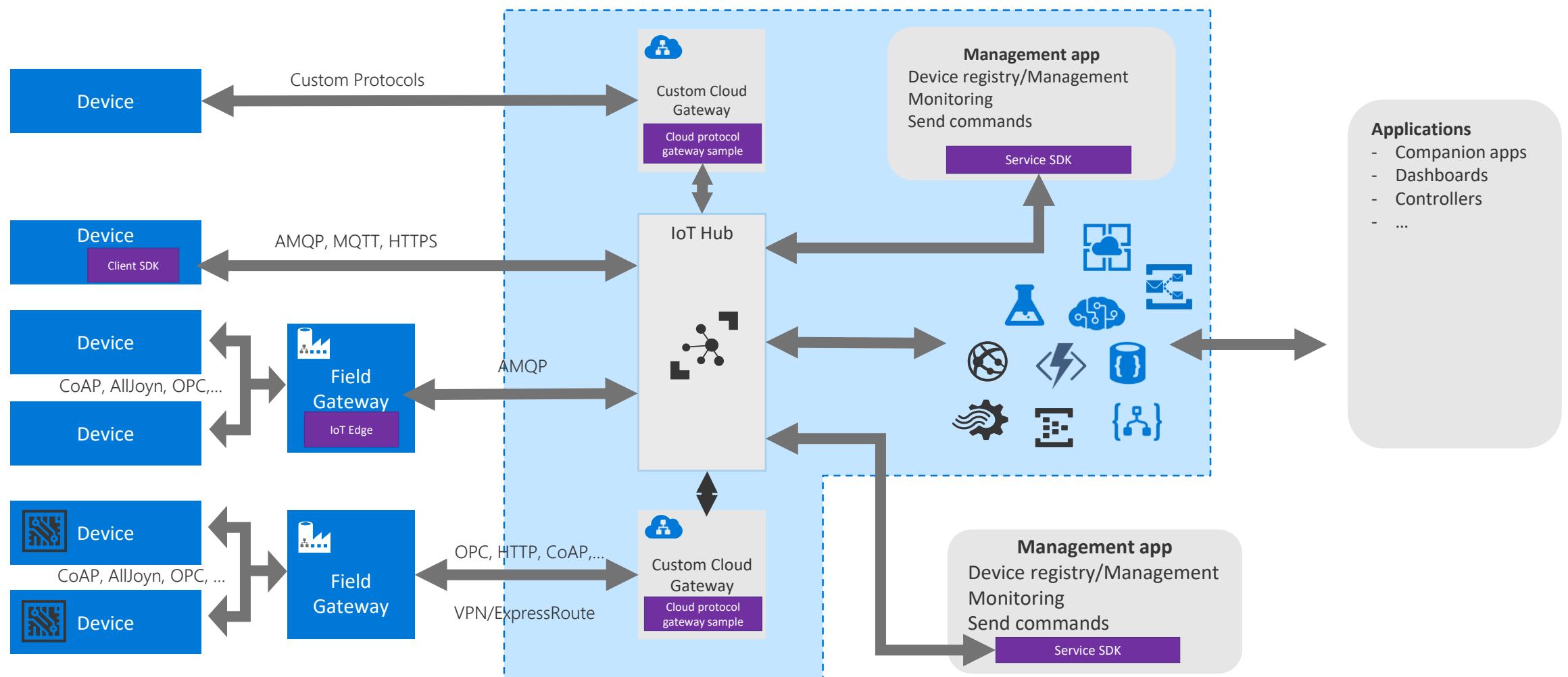
Billions of messages
Scale up and down
Declarative Message Routes
File Upload
WebSockets & Multiplexing
Azure Monitor
Azure Resource Health
Configuration Management



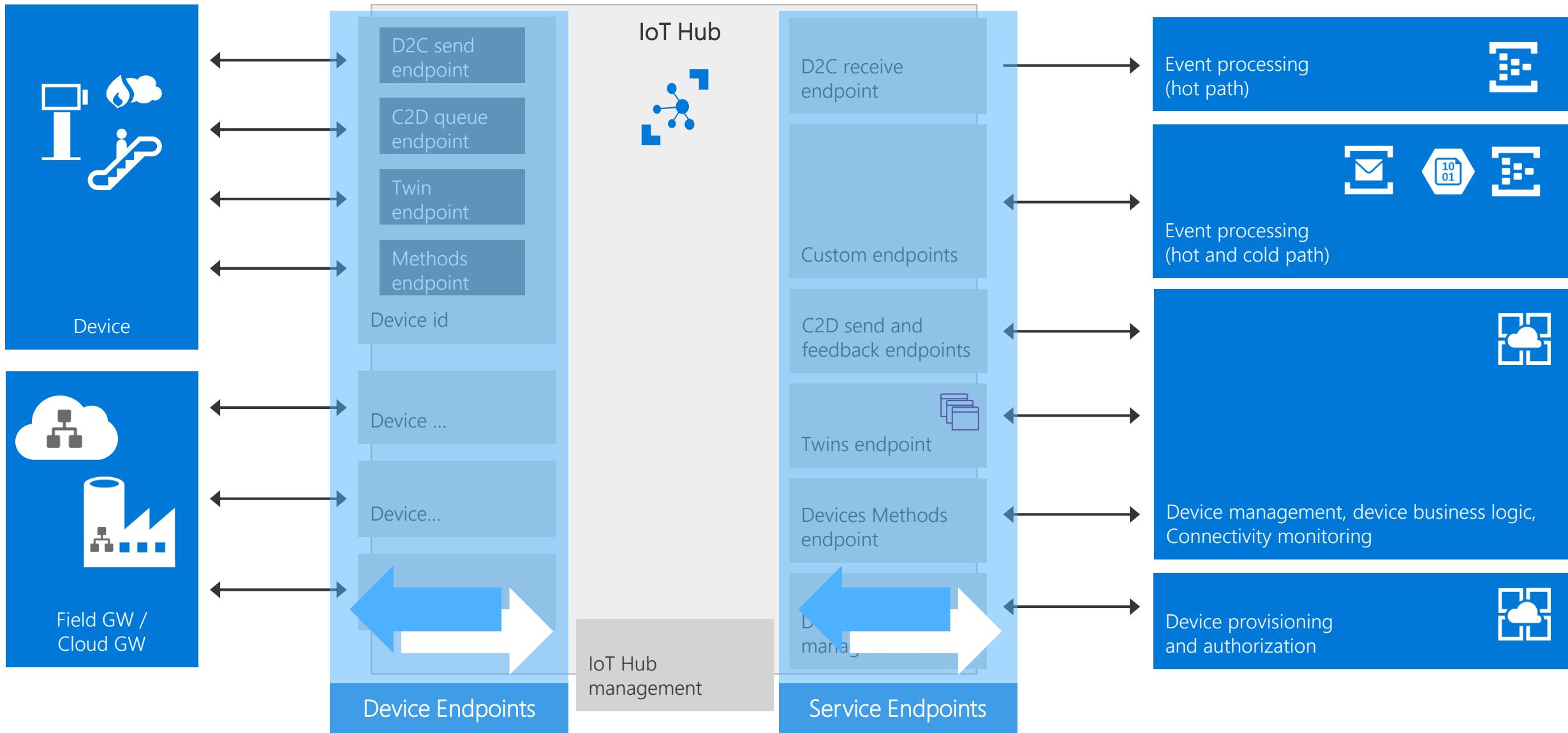
End-to-End Security

Per Device Certificates
Per Device Enable/Disable
TLS Security
X.509 Support
IP Whitelisting/Blacklisting
Shared Access Policies
Firmware/Software Updates

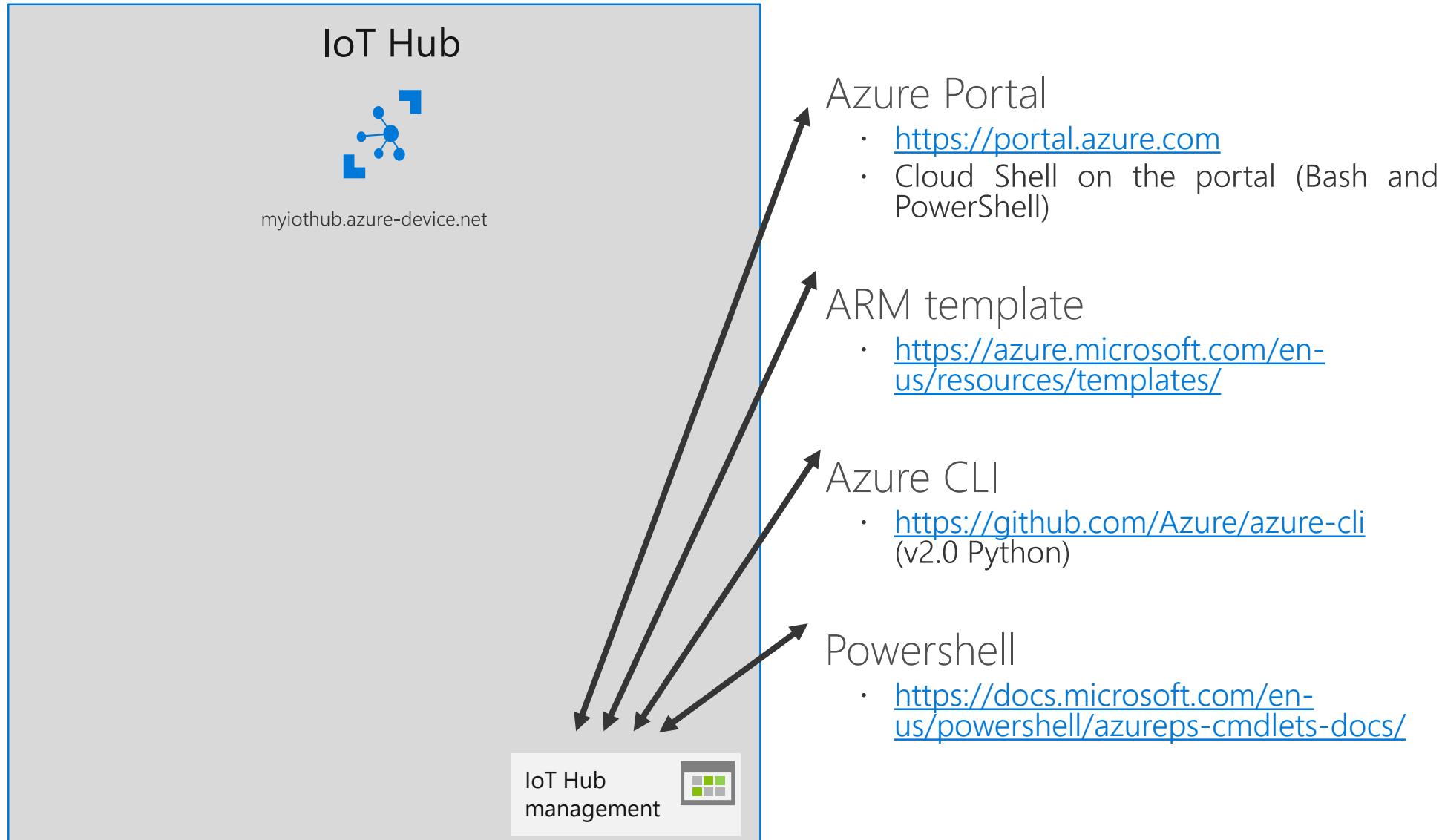
Azure IoT Hub



IoT Hub Key Functions



Creating your IoT Hub



Demo IoT Hub 101

Creation, tools, register device

Azure IoT Hub Tools

Device explorer (windows)

IoT Hub Explorer (NPM)

IoT Hub Diagnostics (NPM)

SDKs

Device SDK Libraries

SDK, agent libraries

Easily accessible libraries in GitHub

Cross platform/language support

RTOS, Linux, Windows, Android, or iOS
C, C#, JS, Java, Node.js, and Python

IoT protocols

MQTT, AMQP, and HTTP

Connect IP, and non-IP devices

Support for direct connection devices and resource constrained or non-IP devices via gateway and field protocols

Package & Libraries support

Nuget, Apt-Get, Npm, Maven, PyPi, Mbed lib, etc

Open source framework

Develop custom agents for your devices

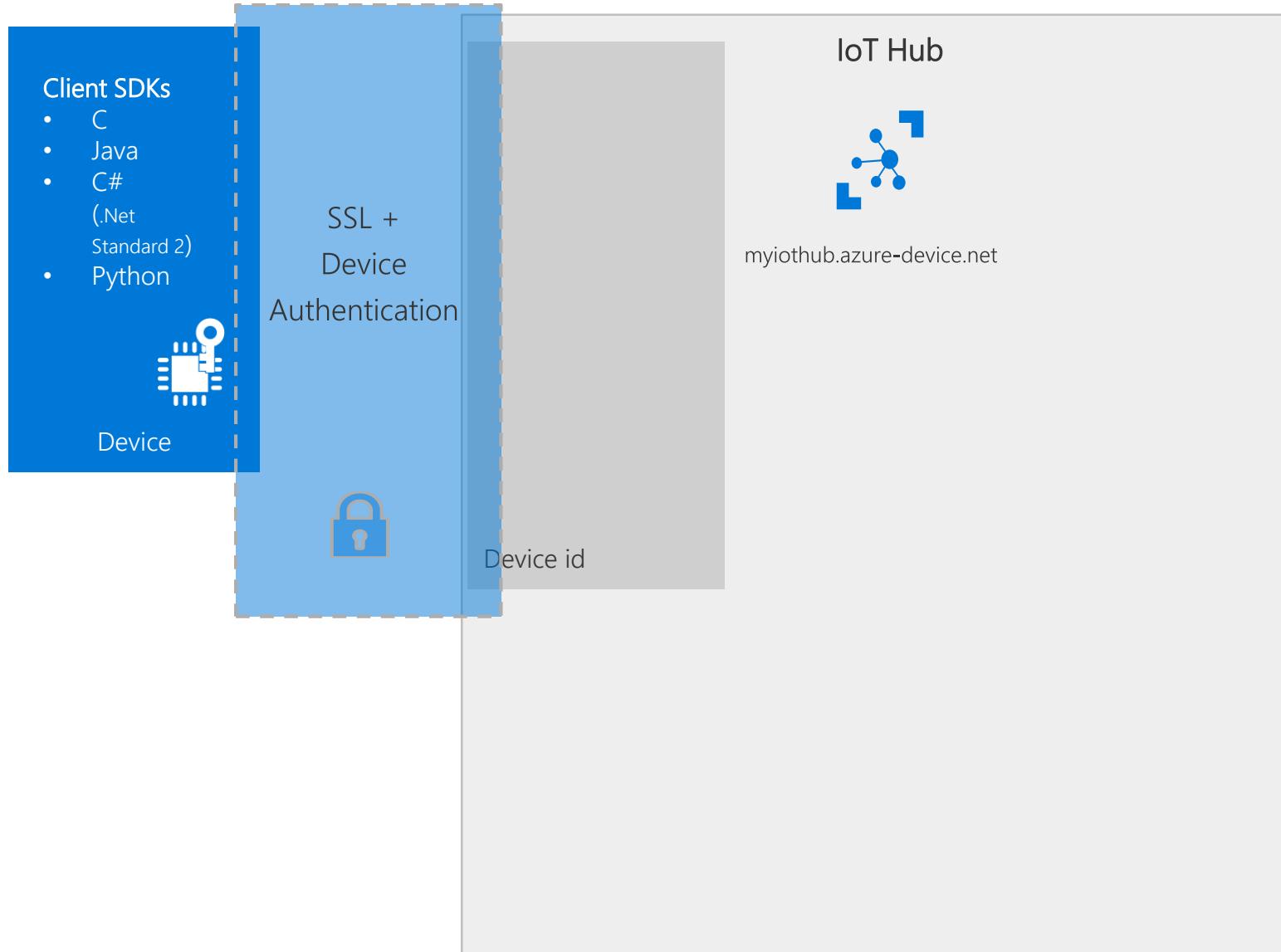
Secure communication

Secure D2C and C2D connectivity for messaging, device management and command and control

Dev tool support & samples

IoTHub-explorer, Device Explorer, iothub-diagnostics

Device Communication



Private Key -> SAS Token

x.509 Certificate

- Self Signed
- CA

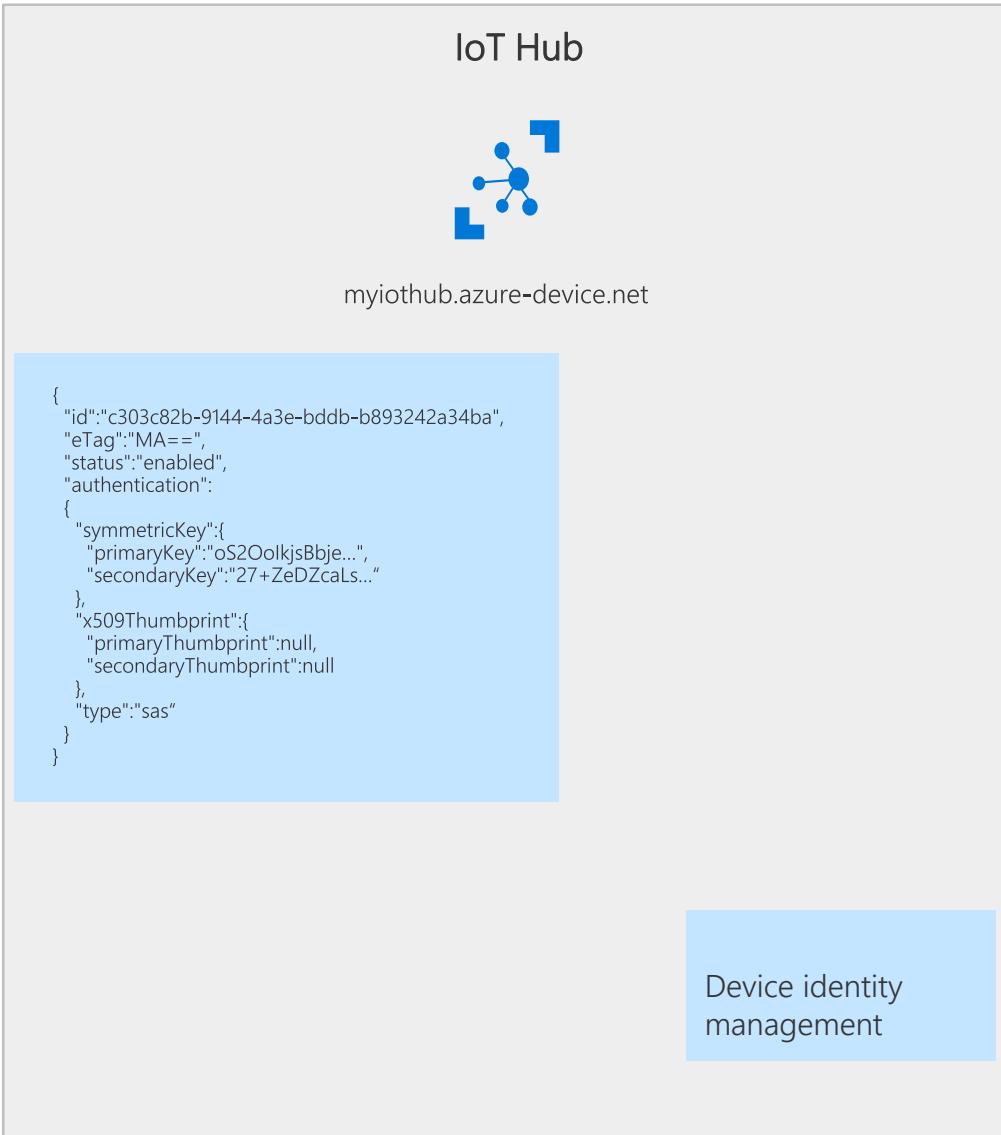
Device Registry

Unique id for each device

Unique credentials for authentication

- Private Key/SAS Token
- X.509 Certificates

Device Twin



Azure Portal
Development tools

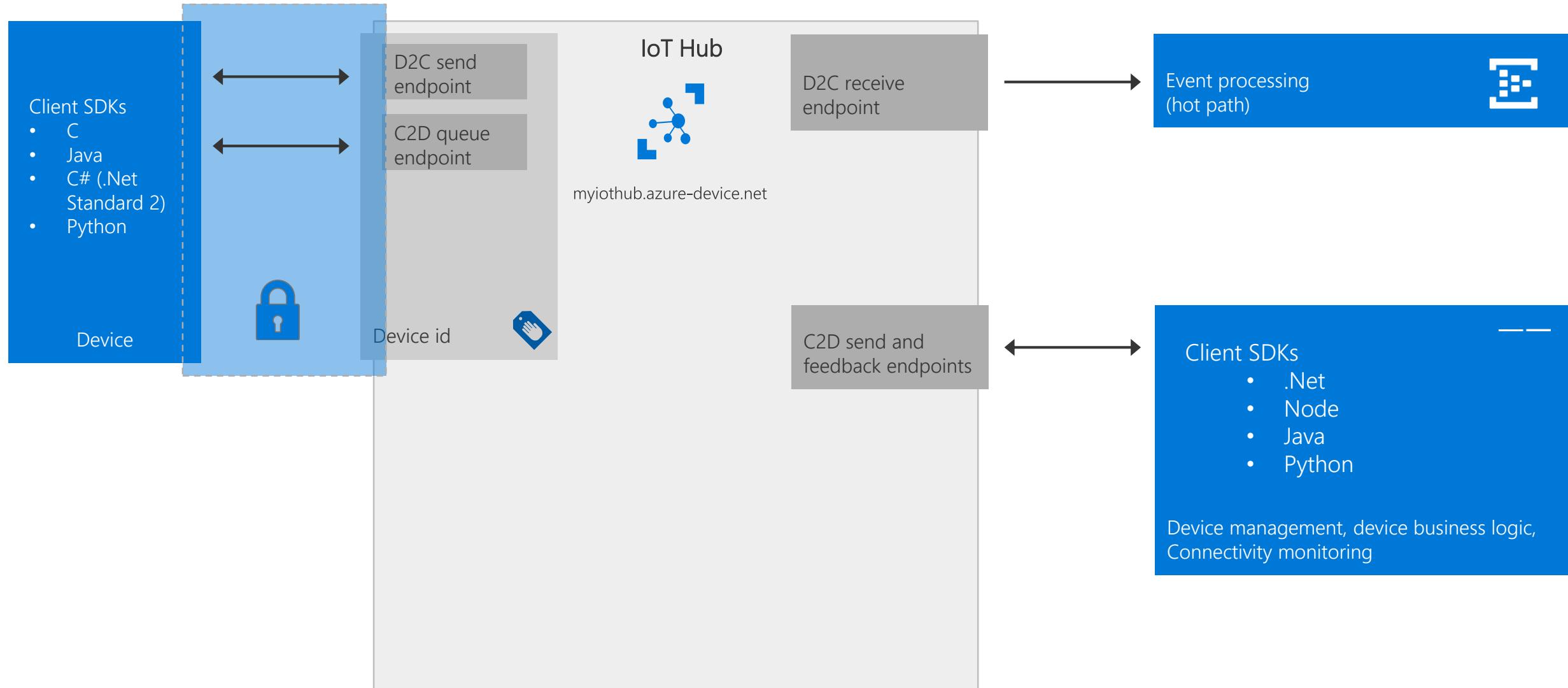
- Azure CLI
- VSCode extension

Using a service SDK

- .Net
- Node
- Java
- Python

Device provisioning
and authorization

Send/receive data



Structured data

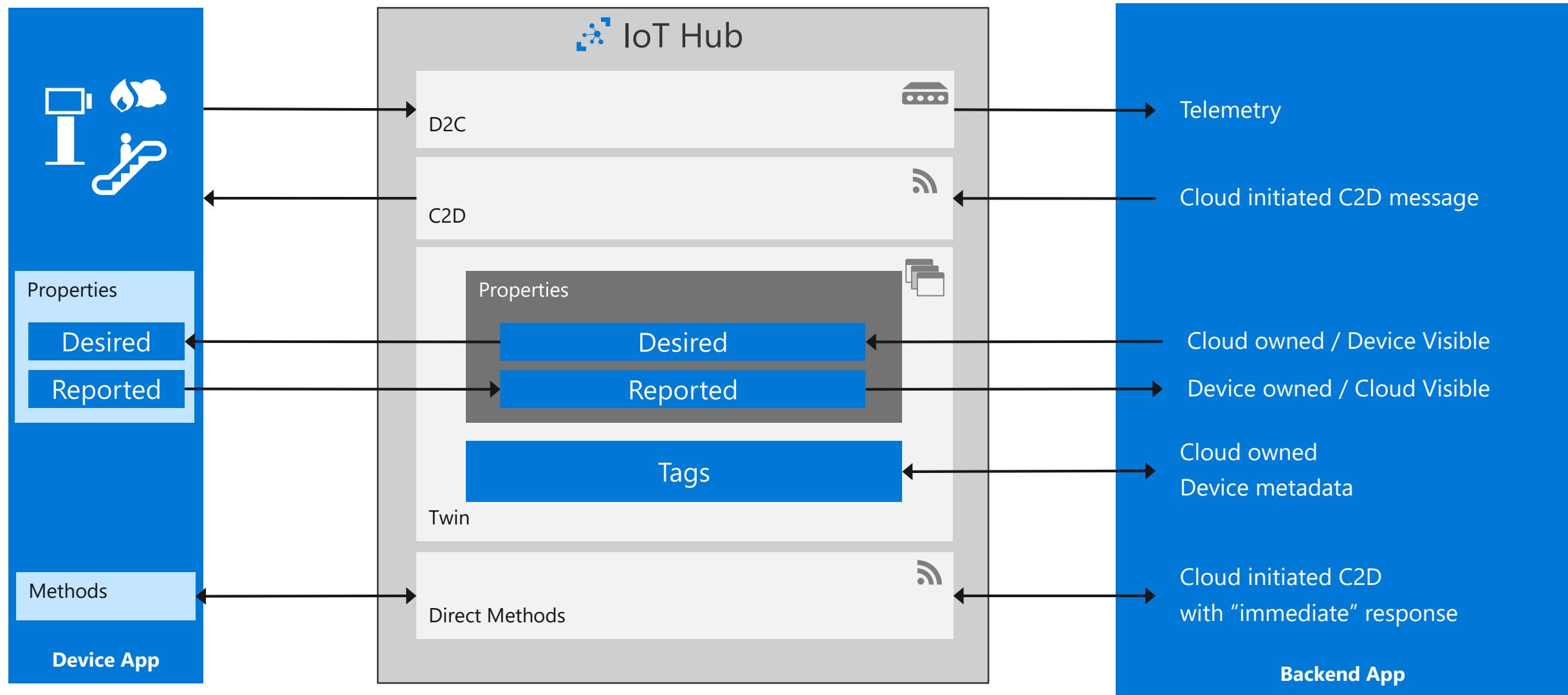
Device Twins

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

Device Direct Methods

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

Manage through Device Twin and Methods

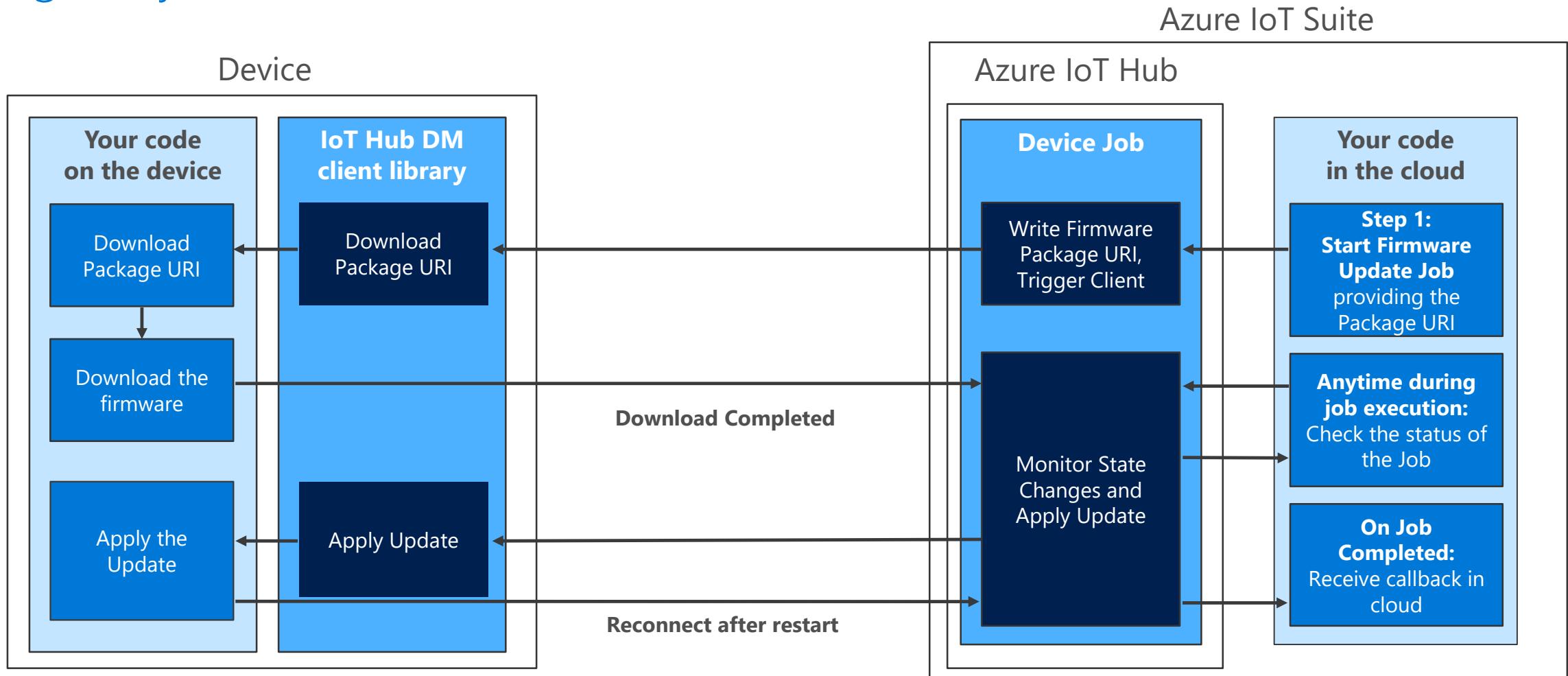


Demo

Device Twin

Device Job - Firmware Update Example

A Device Job is a multi-step device orchestration on a set of devices managed by Azure IoT Hub



IoT Hub Direct Method

Http Post Request:

Uri: {iot hub}/twins/{device id}/methods/

Header: authorization, request id, content type, content encoding

Body:

```
"methodName": "reboot",
"timeoutInSeconds": 60,
"payload": { "input1": "someInput"}
```

2

Device Request:

\$iothub/methods/POST/{method name}/?\$rid={request id}

2

Device Response:

\$iothub/methods/res/{status}/?\$rid={request id}

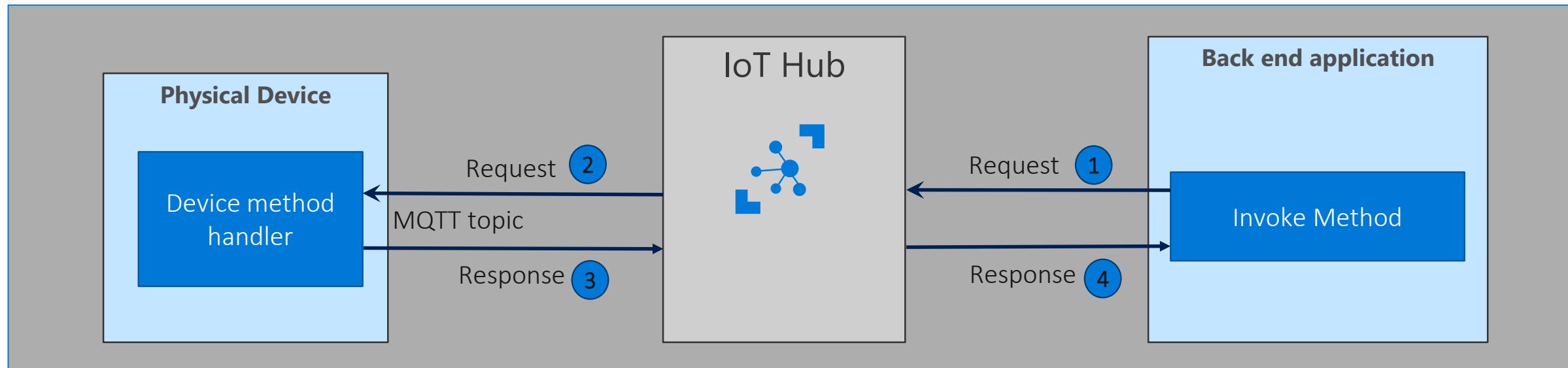
3

4. Http Response:

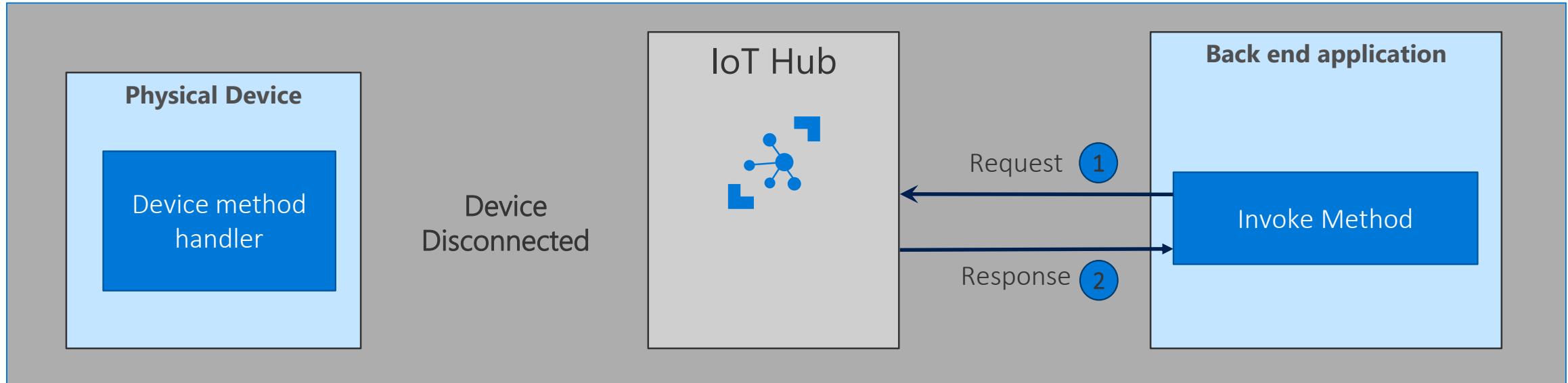
HttpStatusCode: 200, 404, 500

Body: "status" : 200, "payload" : {...}

4



IoT Hub Direct Method (disconnected device)



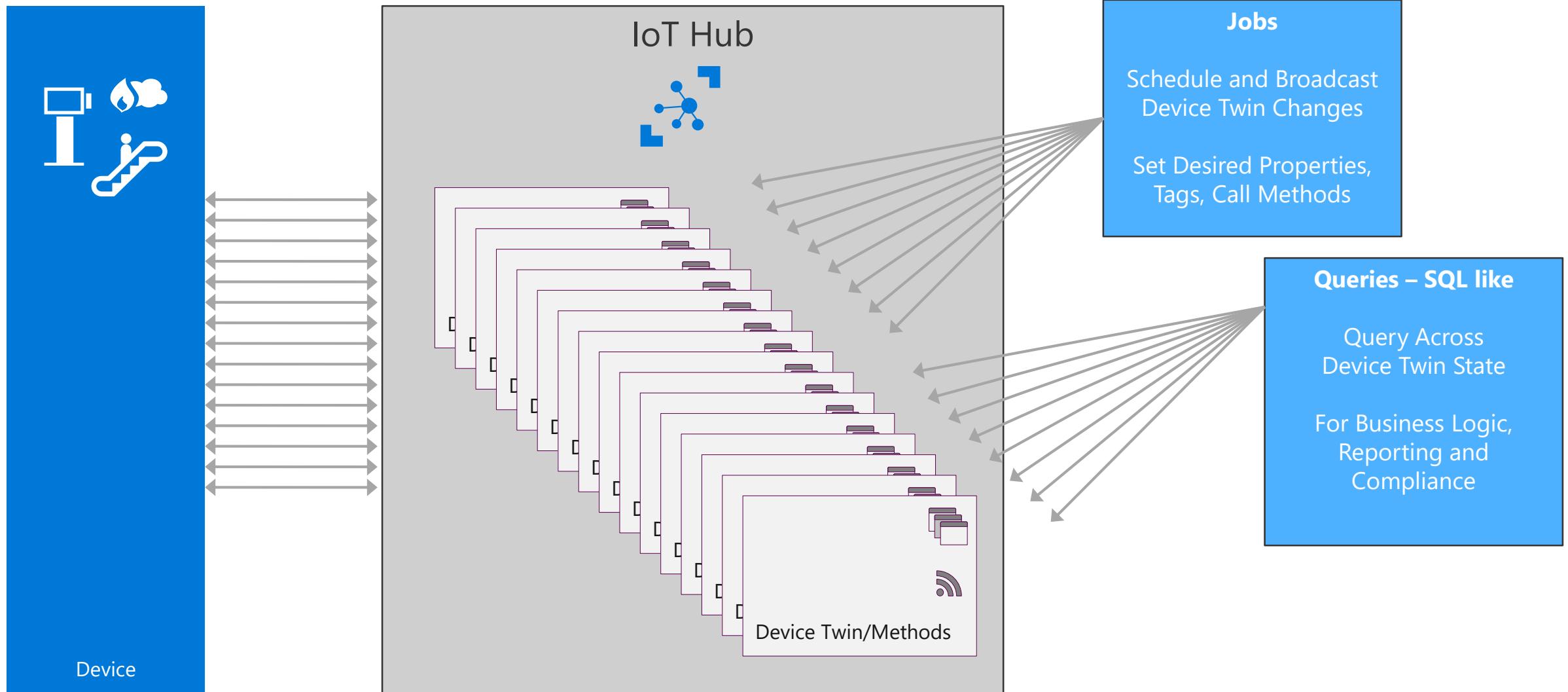
Http Response:
HttpStatuscode: 404
Body:

2

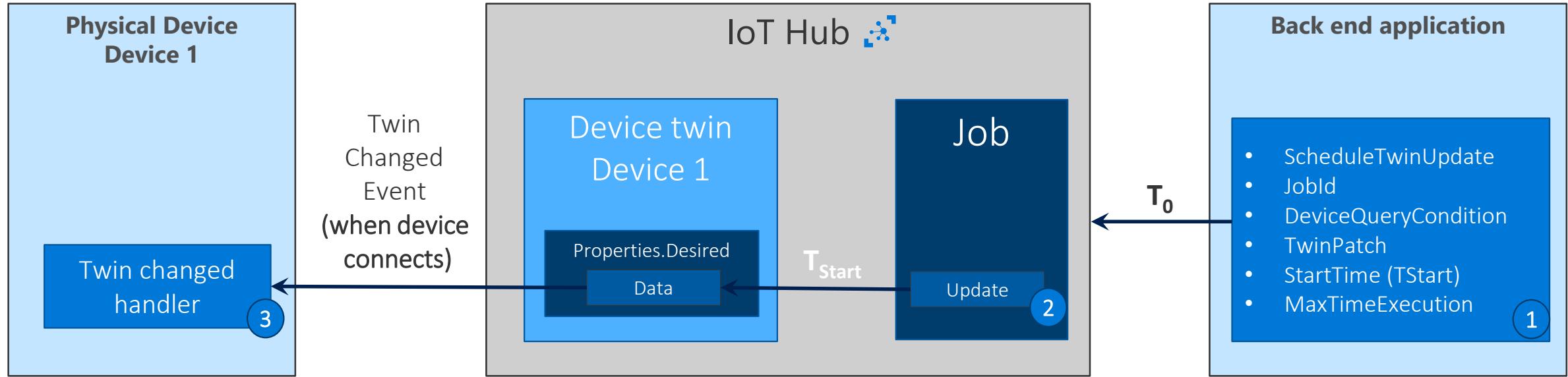
Demo

Direct Methods

Manage through Jobs and Queries



Job for scheduling device twin updates



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

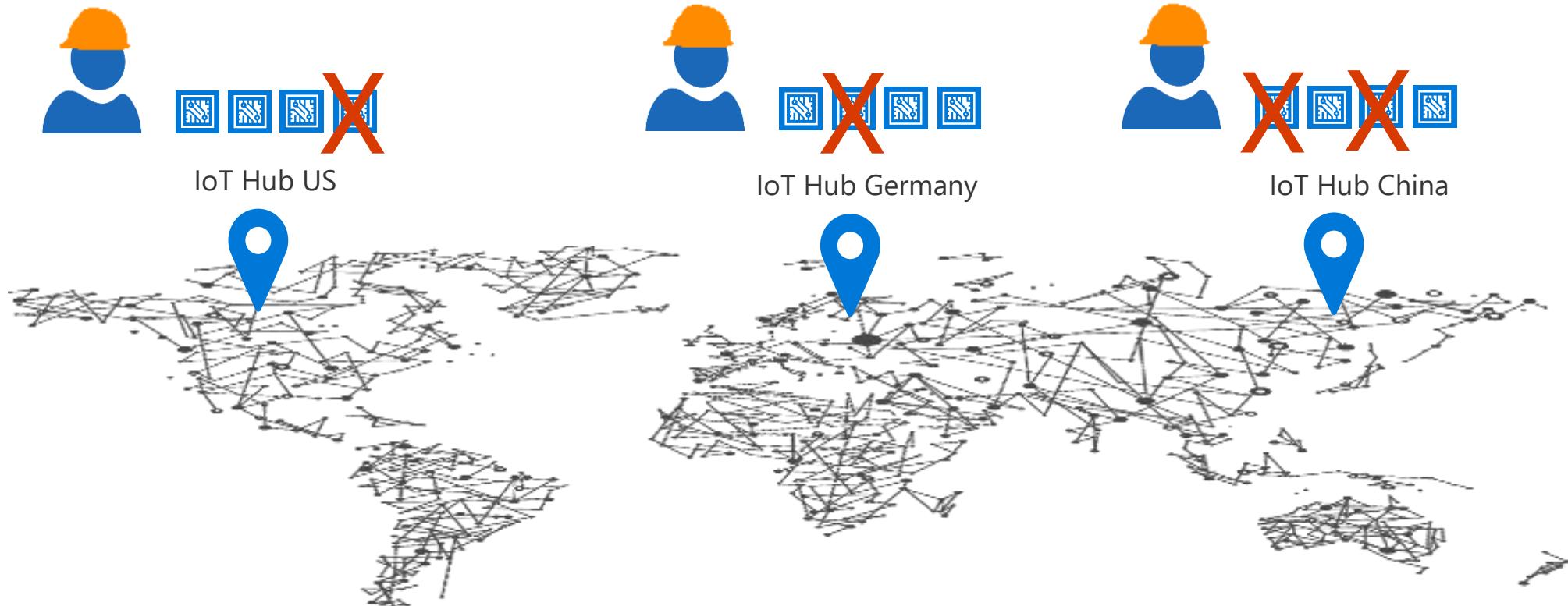
	Direct methods	Twin's Desired Properties	C2D messages
Scenario	Commands that require immediate confirmation	Long-running commands	One-way notifications to the device
Size	8KB requests / 8KB responses.	Maximum 8KB	64KB
Frequency	High	Medium	Low

Device Provisioning Service (Preview*)

* As of 6 Sept 2017

IoT Device Provisioning Challenges

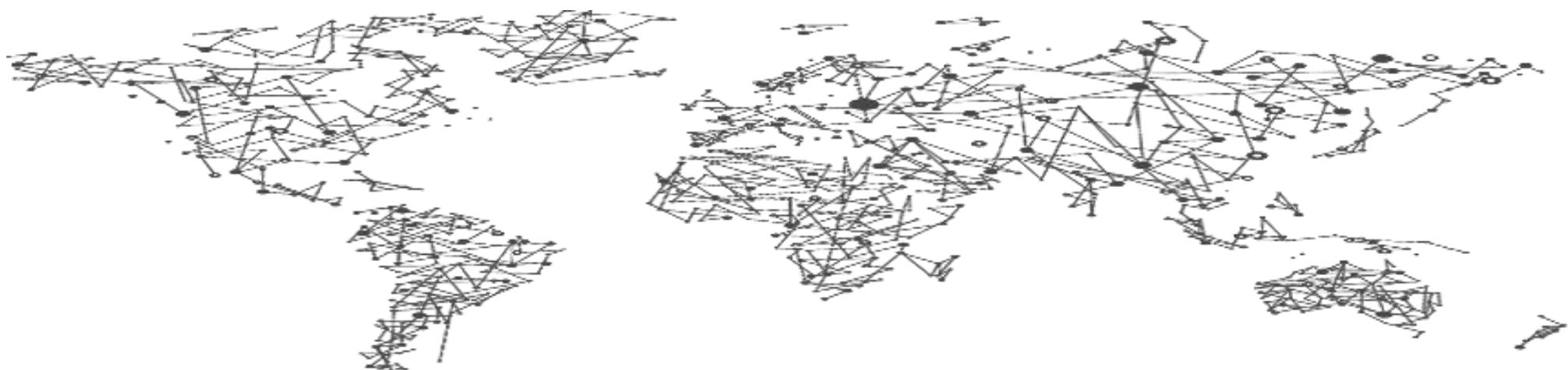
Adding devices to an IoT solution has historically been very manual
And error prone



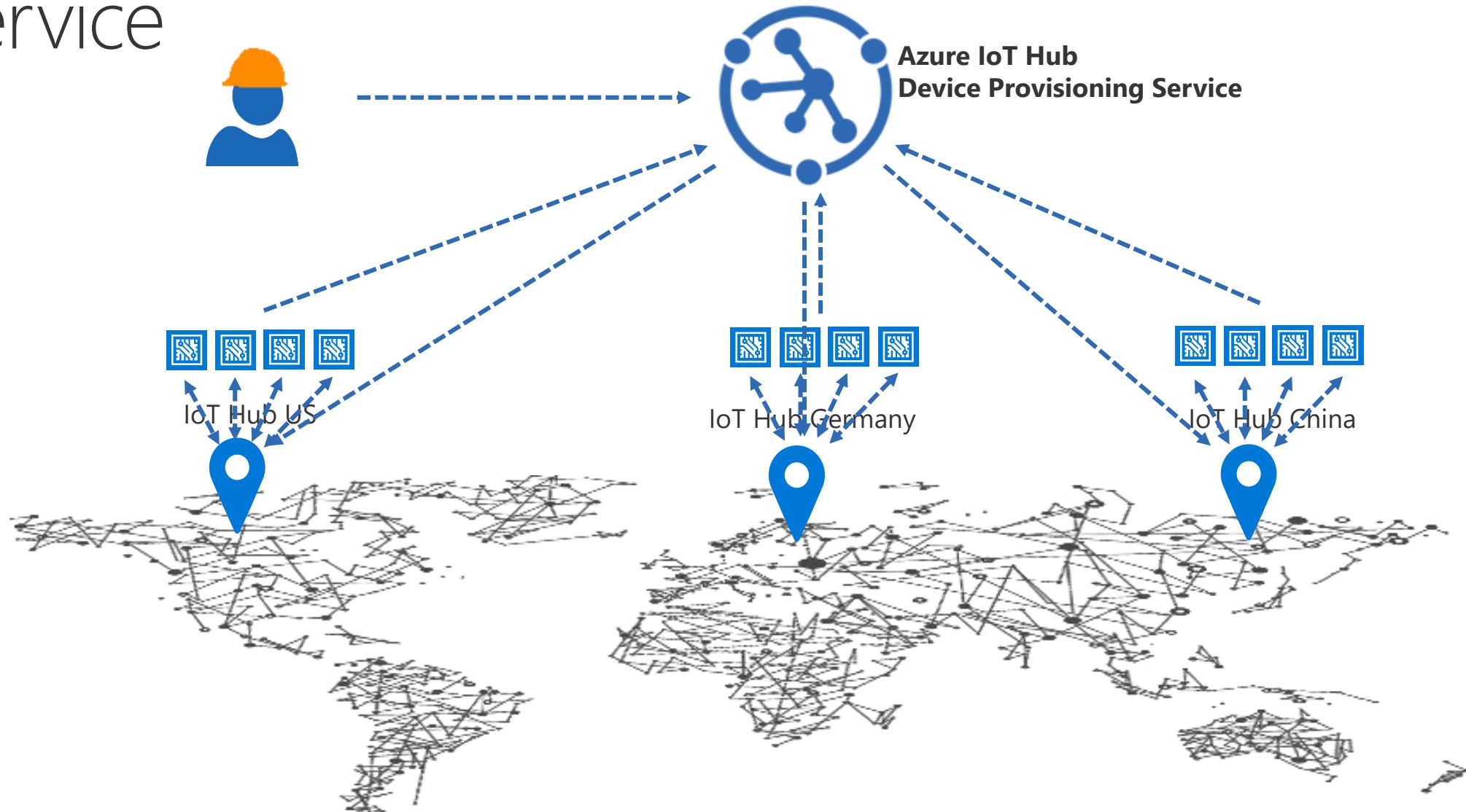
IoT Device Provisioning Challenges

What IoT needs is “plug and play” device provisioning

Where the automation and repeatability controlled in a central location



Available: Azure IoT Hub Device Provisioning Service



Available: Azure IoT Hub Device Provisioning Service

Global availability

Simplify with “plug and play” provisioning

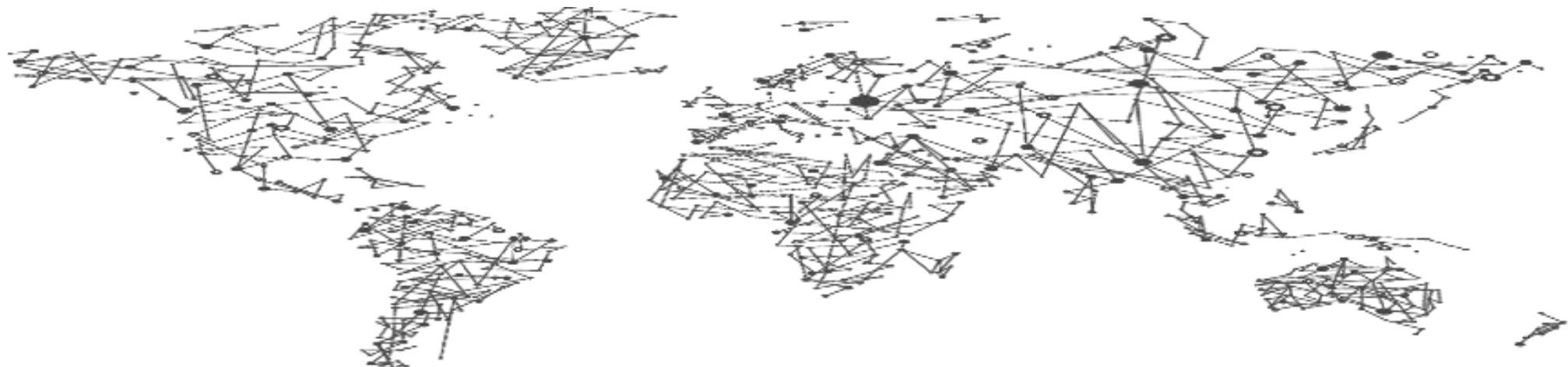
Minimize manual connection requirements

Enhanced security through HSM

Works in harmony with Azure IoT Hub



Azure IoT Hub
Device Provisioning Service



Azure IoT Hub Device Provisioning Service

Simplify with zero touch provisioning

Supports multiple locations

Easiest way to mass-provision devices

URL stability

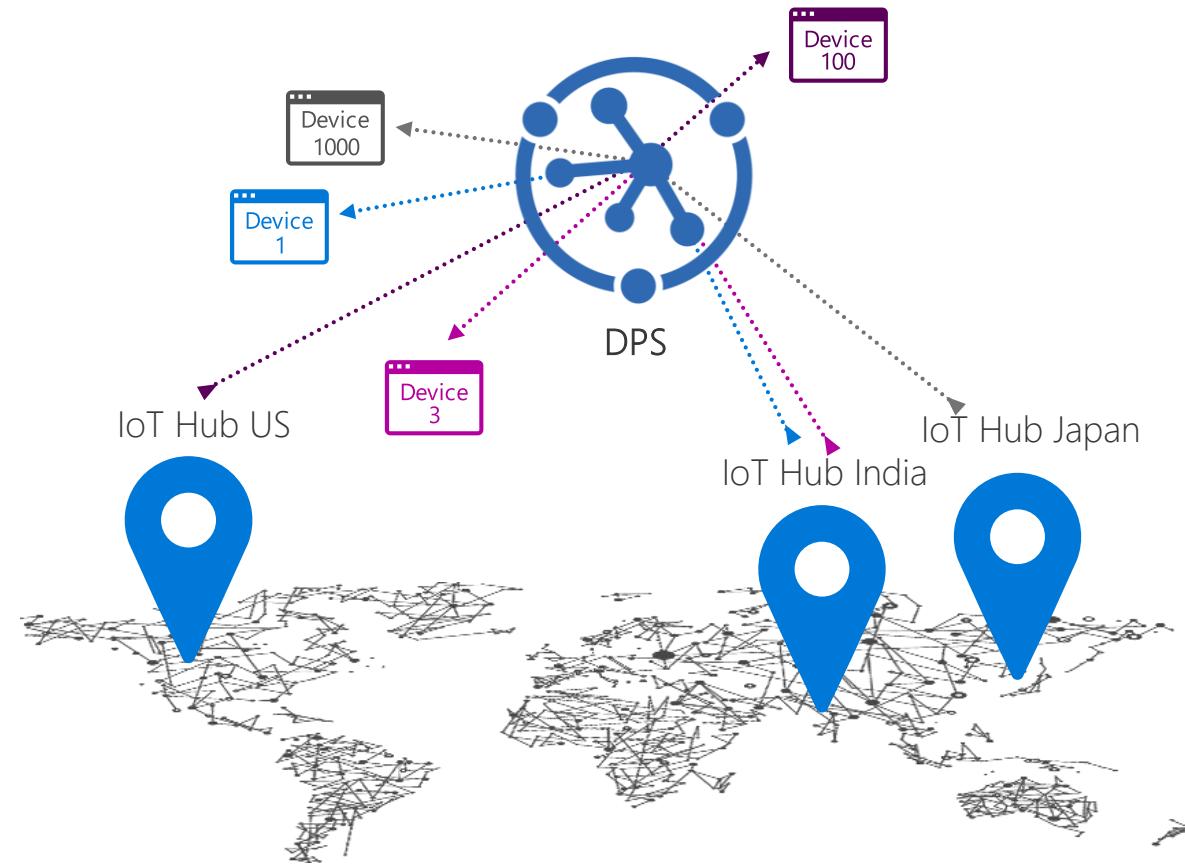
Enhanced security through HSM

For any device compatible with IoT Hub

Remove human error

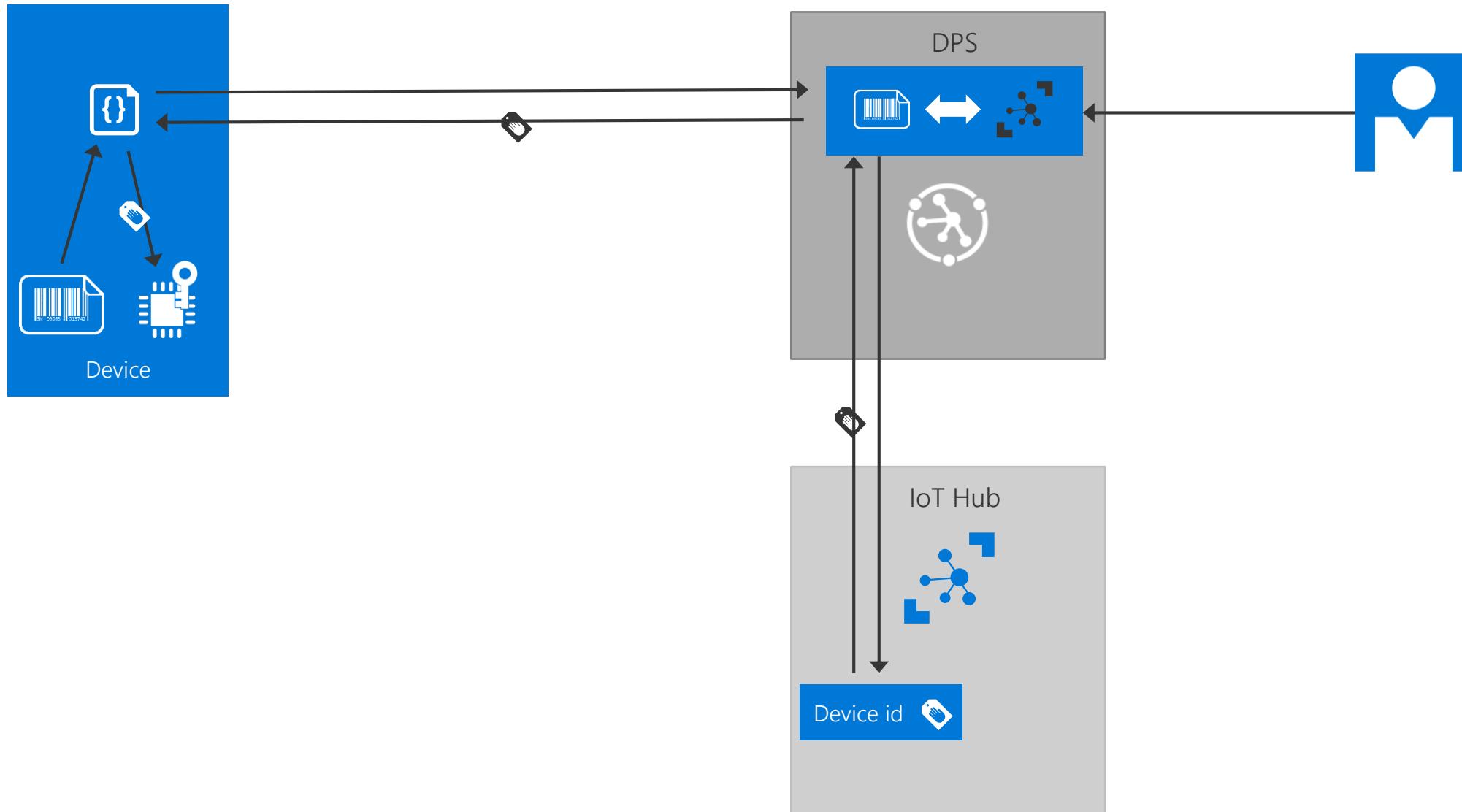
Minimize manual connection requirements

Multitenancy support

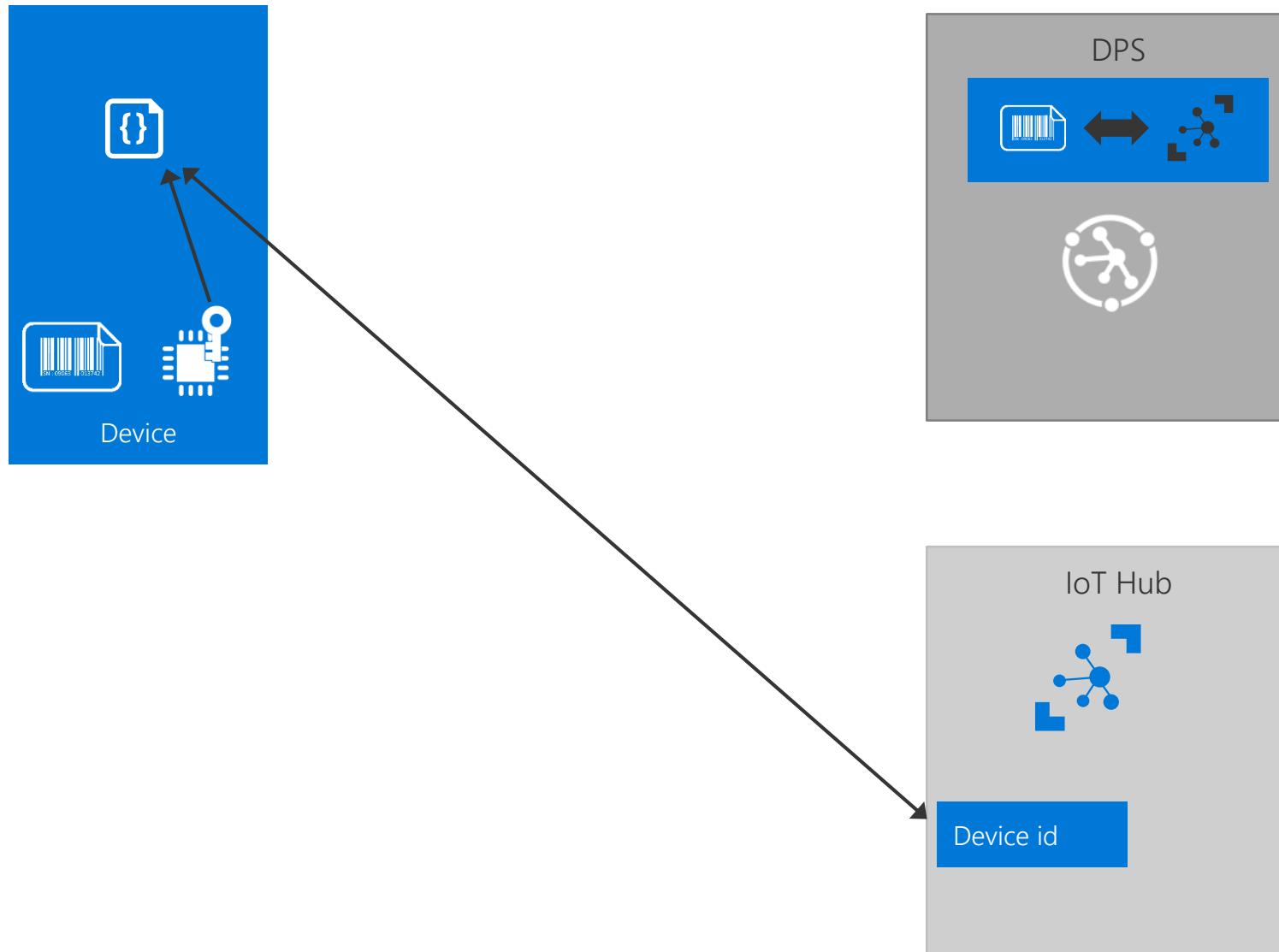


DPS knows exactly which IoT Hub to connect and provision

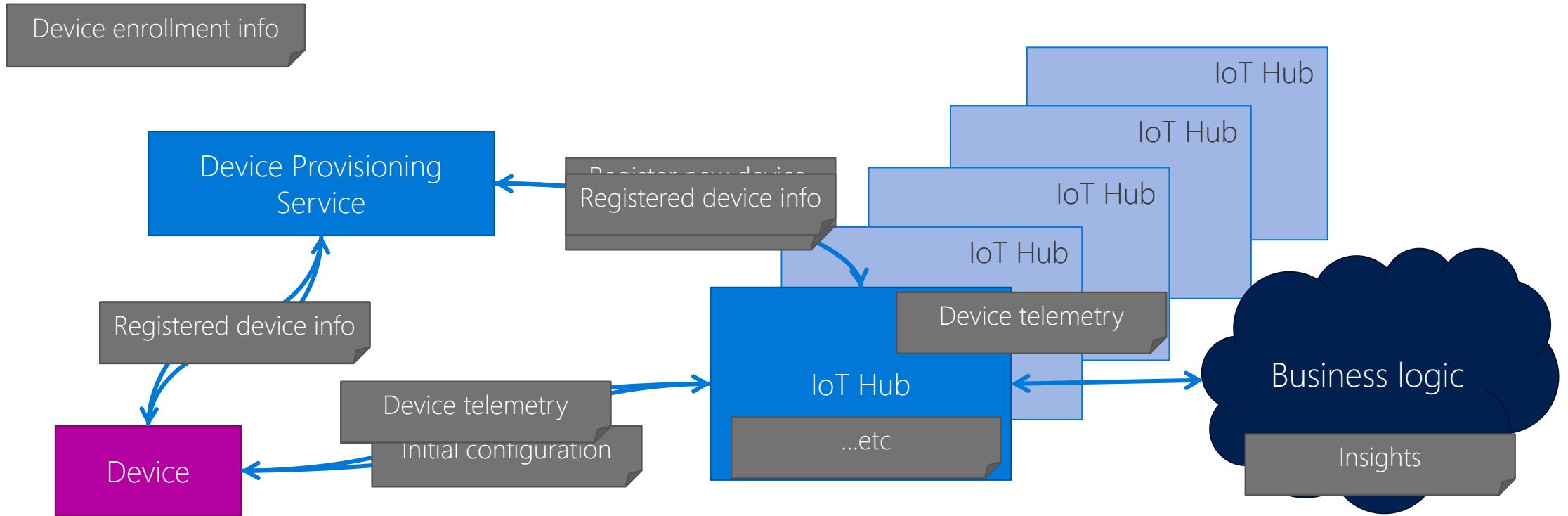
Device Provisioning Service



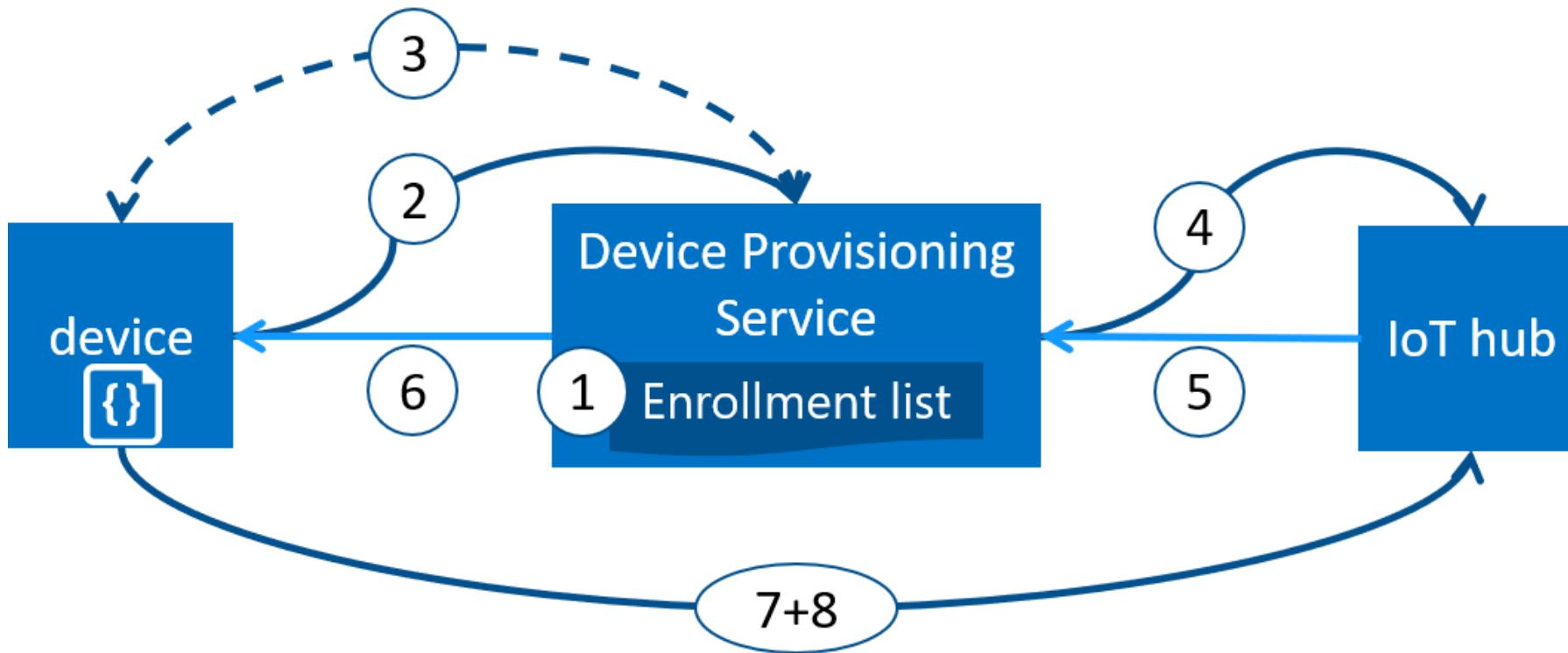
Device Provisioning Service



Provisioning with DPS



Flow



1. Device manufacturer adds the device registration information to the enrollment list in the Azure portal.
2. Device contacts the provisioning service endpoint set at the factory. The device passes the provisioning service its identifying information to prove its identity.
3. The provisioning service validates the identity of the device by validating the registration ID and key against the enrollment list entry using either a nonce challenge ([Trusted Platform Module](#)) or standard X.509 verification (X.509).
4. The provisioning service registers the device with an IoT hub and populates the device's [desired twin state](#).
5. The IoT hub returns device ID information to the provisioning service.
6. The provisioning service returns the IoT hub connection information to the device. The device can now start sending data directly to the IoT hub.
7. The device connects to IoT hub.
8. The device gets the desired state from its device twin in IoT hub.

Verifying a device's identity

Two types of device attestation supported

X.509: following the standard X.509 authentication flow

Trusted Platform Module: following the TPM standard for verifying possession of the TPM's private endorsement key

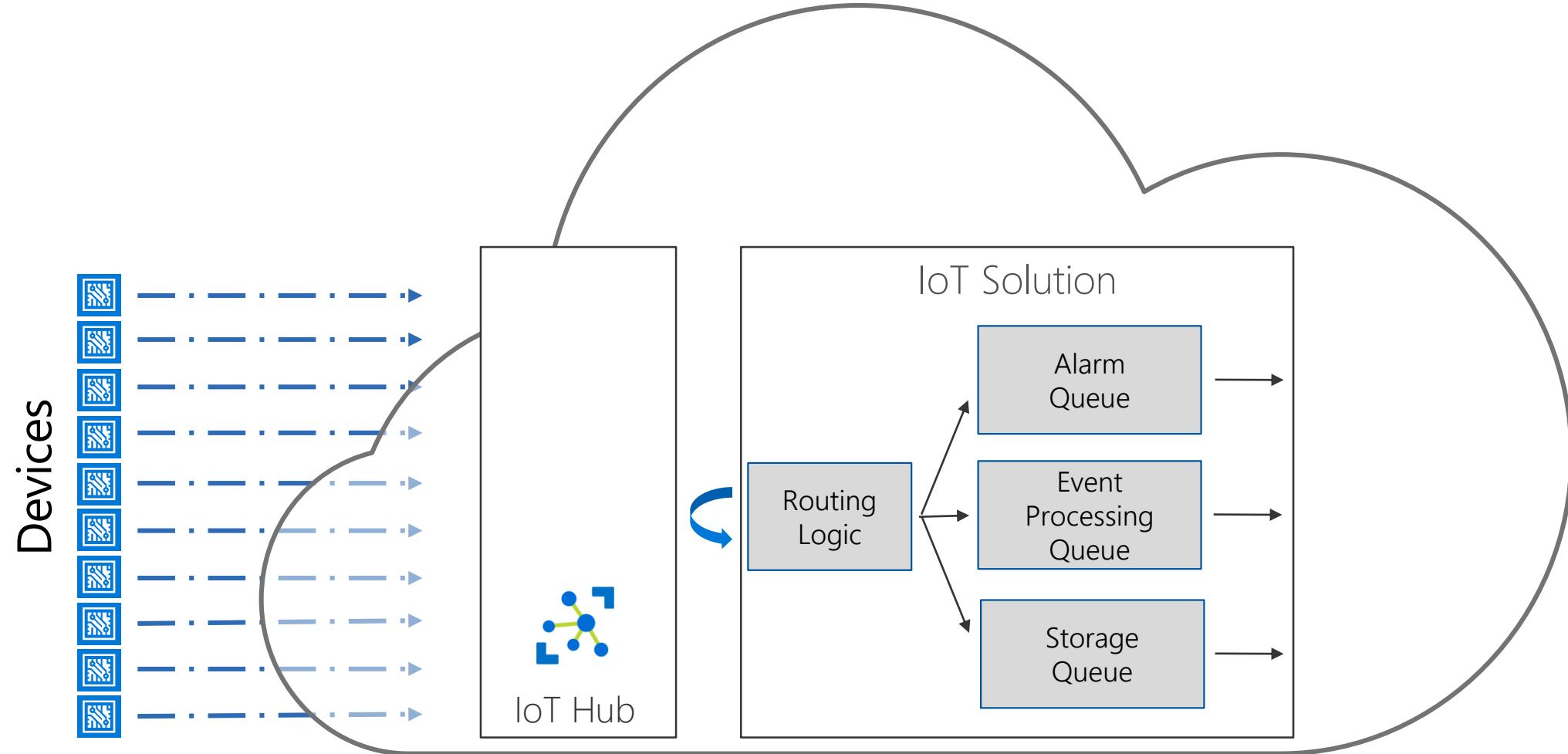
Four ways to store keys

HSM using X.509 certificates
Trusted Platform Module (a type of HSM)
Emulated X.509 certificates
Emulated TPM

IoT Hub Routing

Azure IoT Hub Message Routes

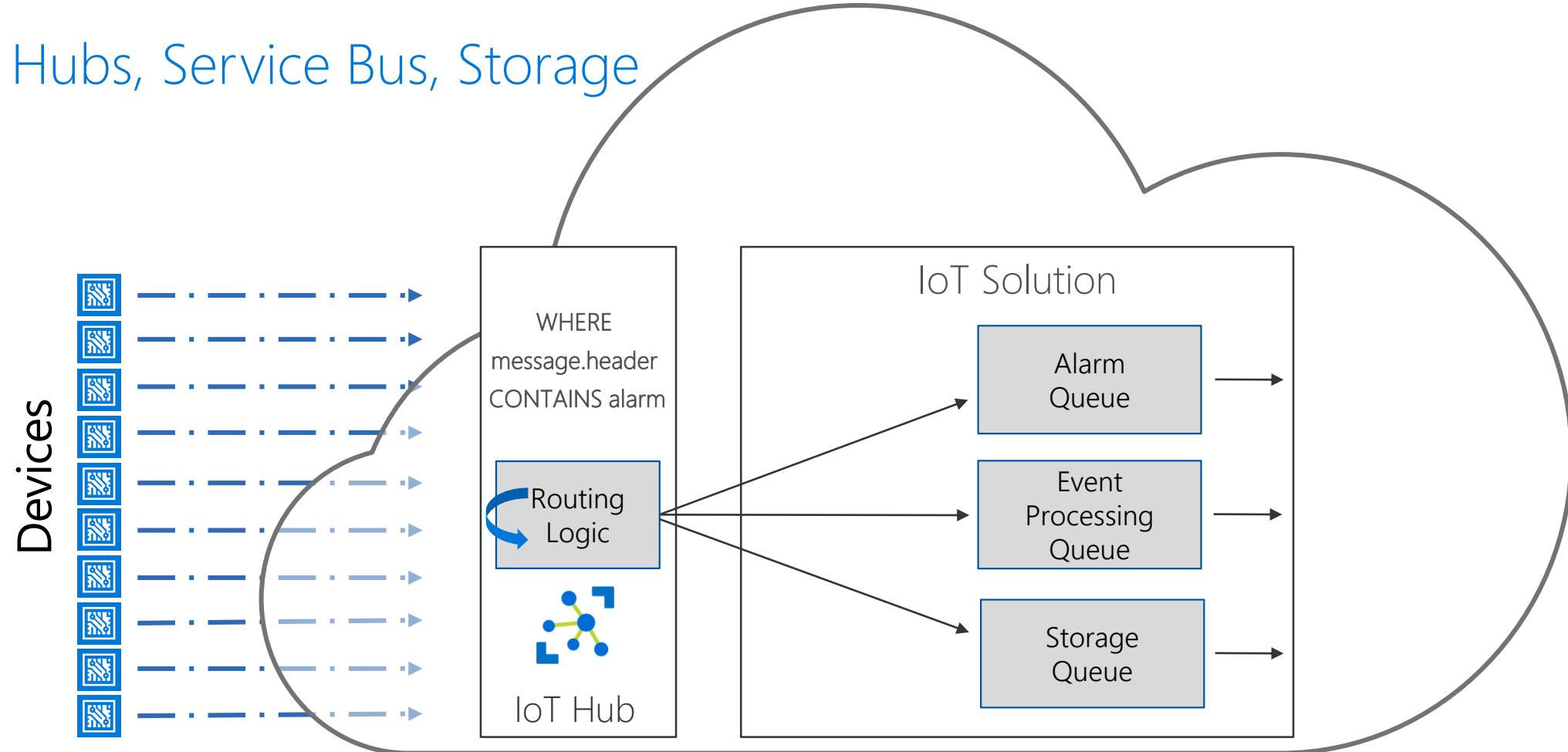
Remember email before inbox rules?



Azure IoT Hub Message Routes

Automatically route messages based on headers or JSON message body

Event Hubs, Service Bus, Storage



Routing messages

1. Endpoints

- Event Hub
- Service Bus Queue
- Service Bus Topic
- Storage Container (new)

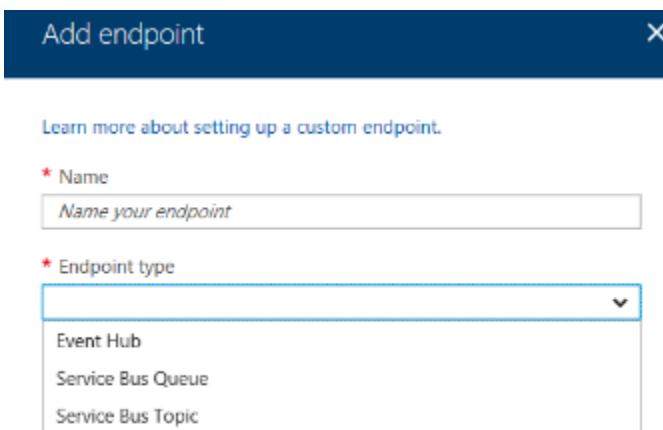
Add endpoint ×

Learn more about setting up a custom endpoint.

* Name

* Endpoint type

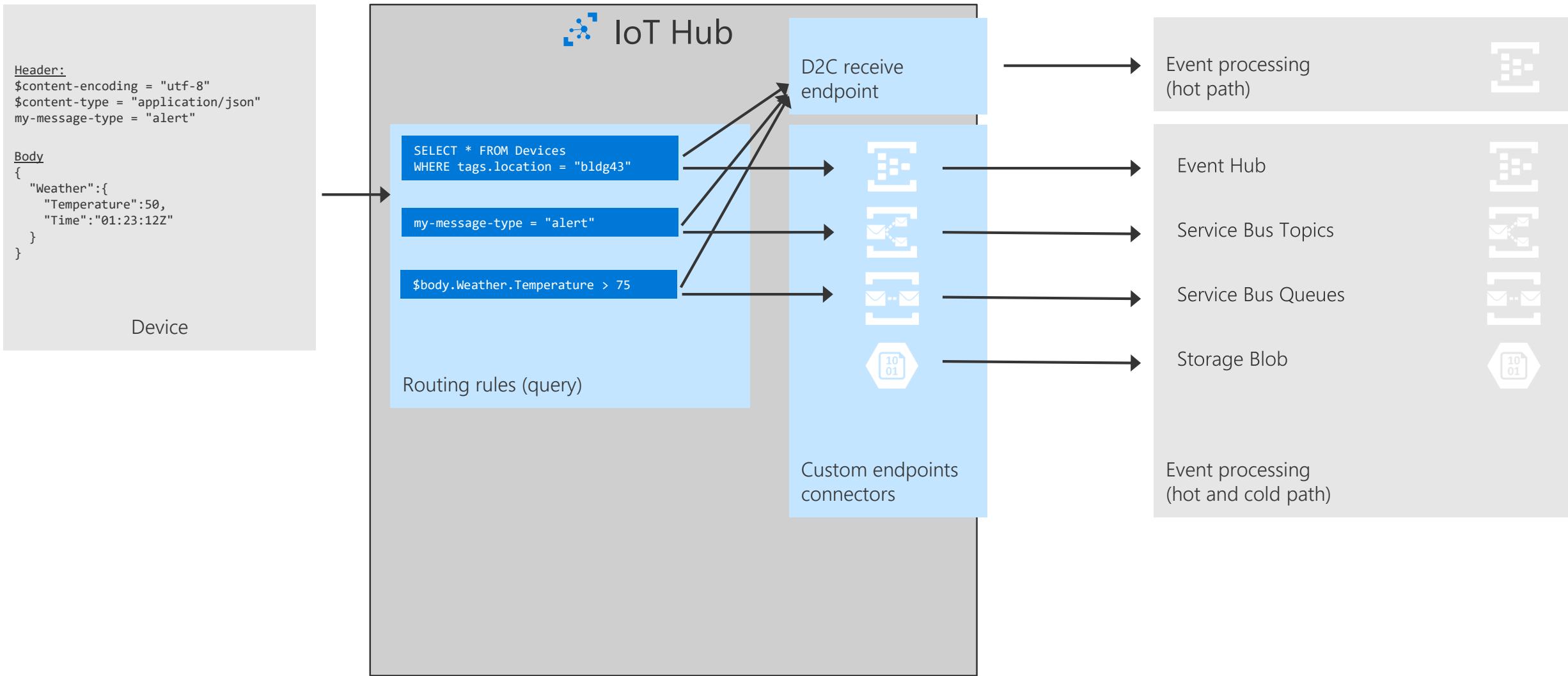
Event Hub
Service Bus Queue
Service Bus Topic



2. Routes

- Endpoint destination (point 1)
- Data source
(Device Messages, Twin Change Events, Device Lifecycle events)
- Query (on message header or body)
- Enable/disable routes

Routing telemetry data

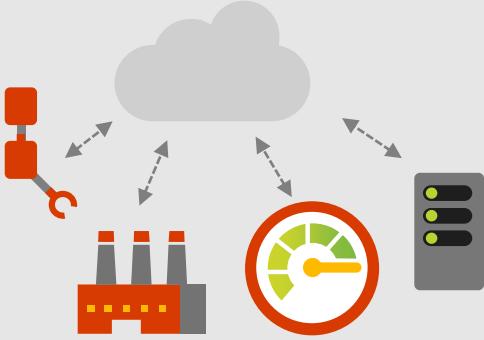


Demo

Endpoints & Routing

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

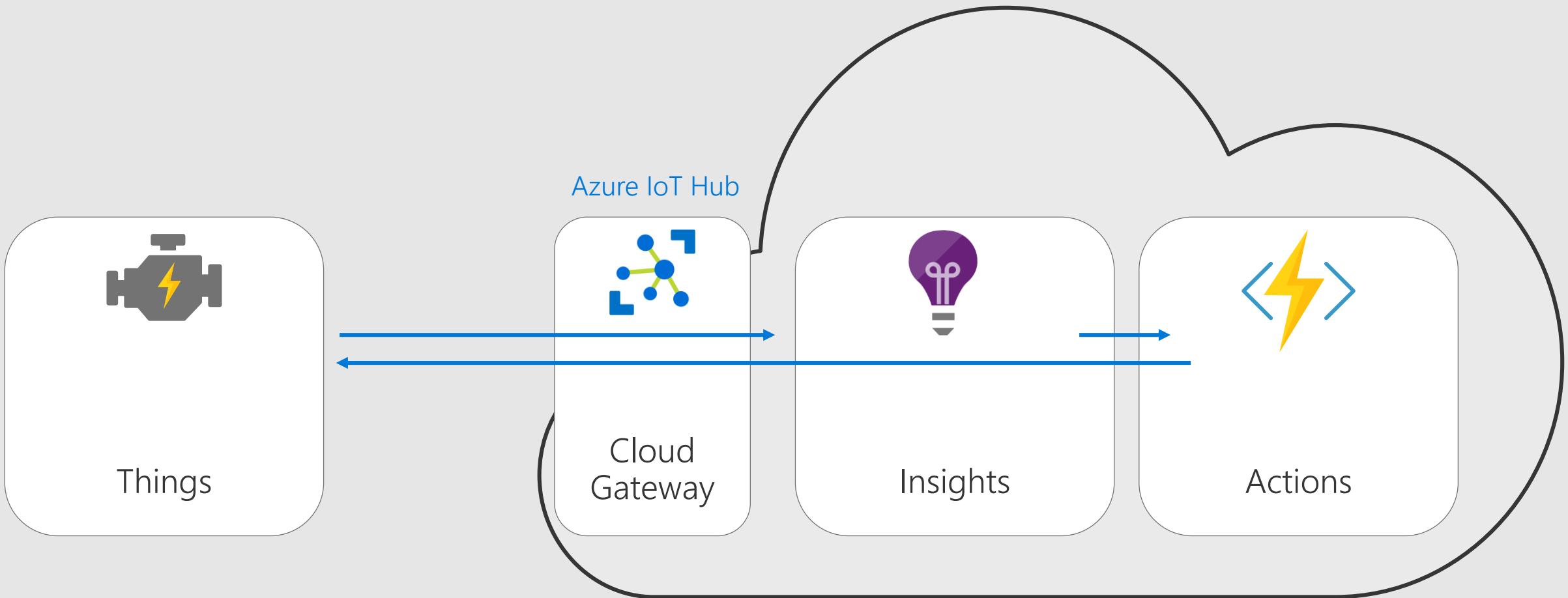
Low latency tight control loops require near real-time response

Protocol translation & data normalization

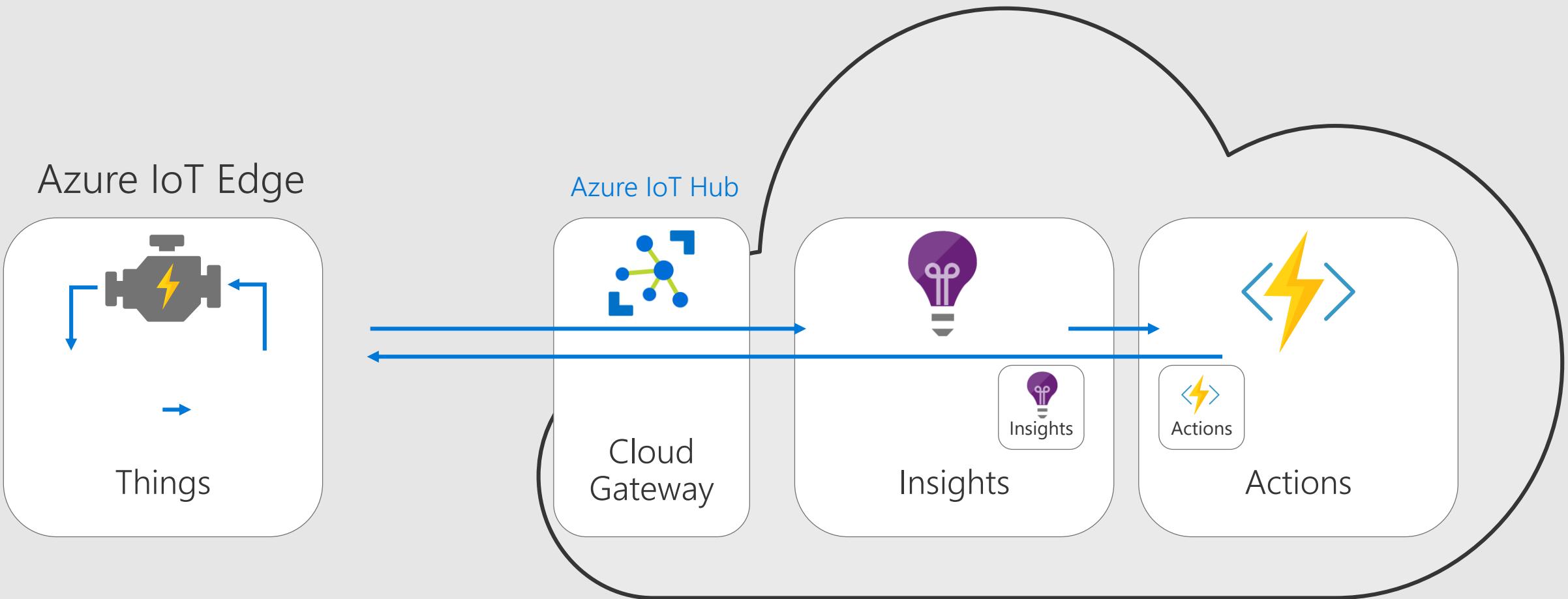
Privacy of data and protection of IP

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

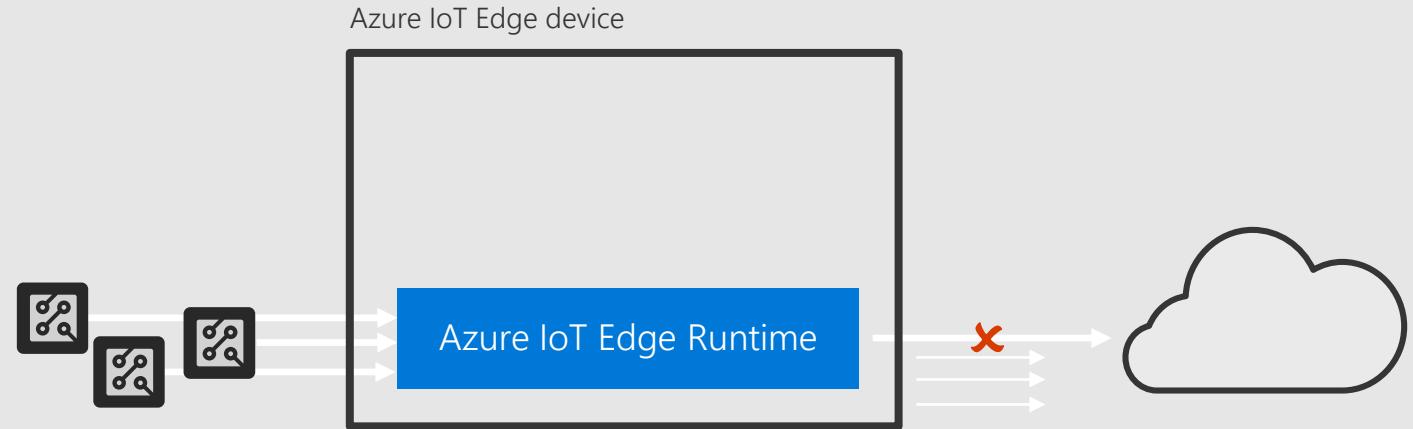
Edge Runtime provides fundamental services

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

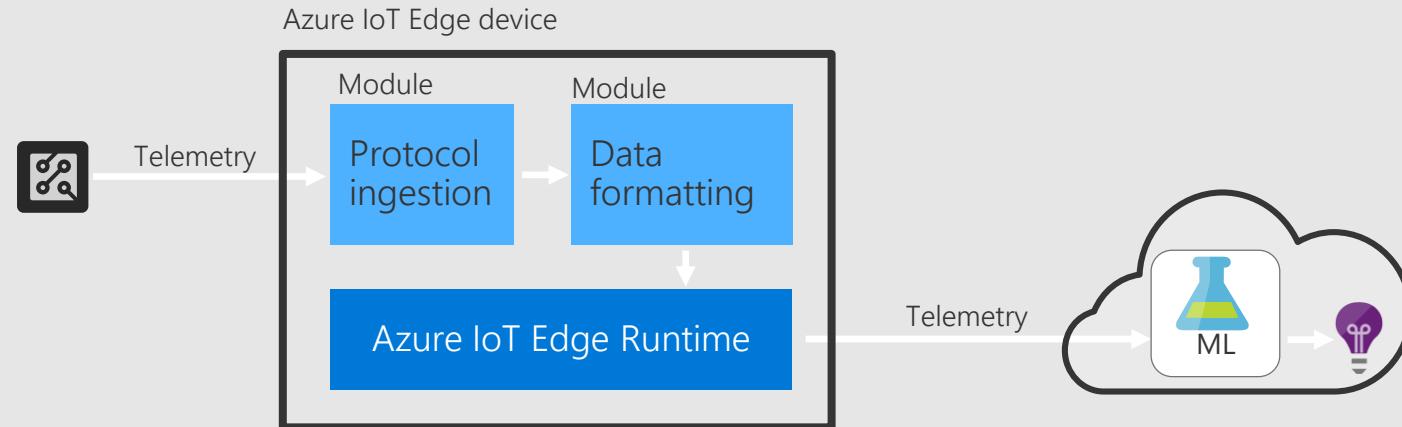
Each module performs an action

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.



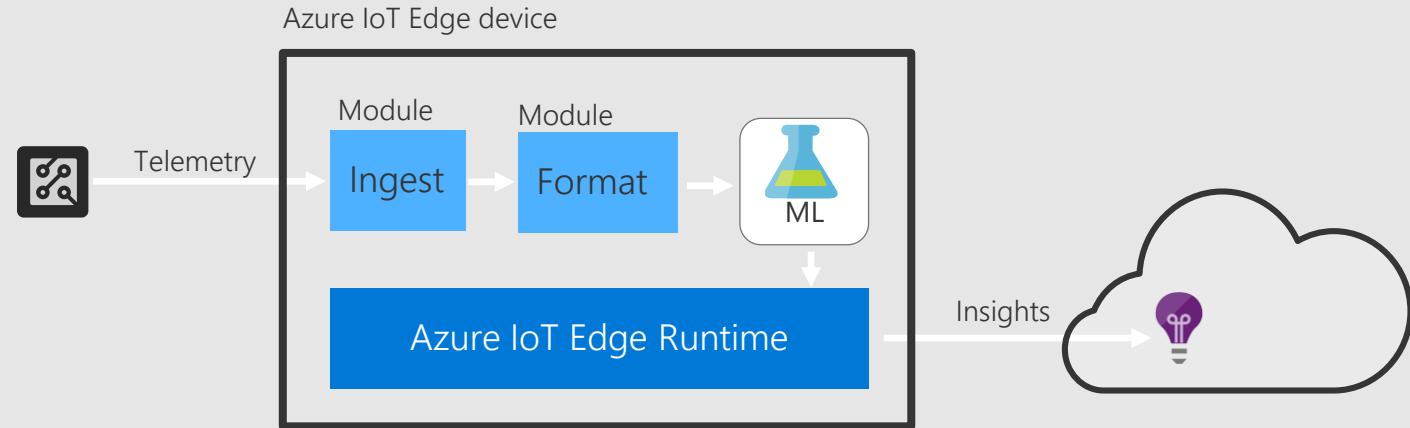
Concepts – Cloud off load

Modular architecture for cloud offload

Azure services provide AI modules provide true edge analytics

Ecosystem for 3rd 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

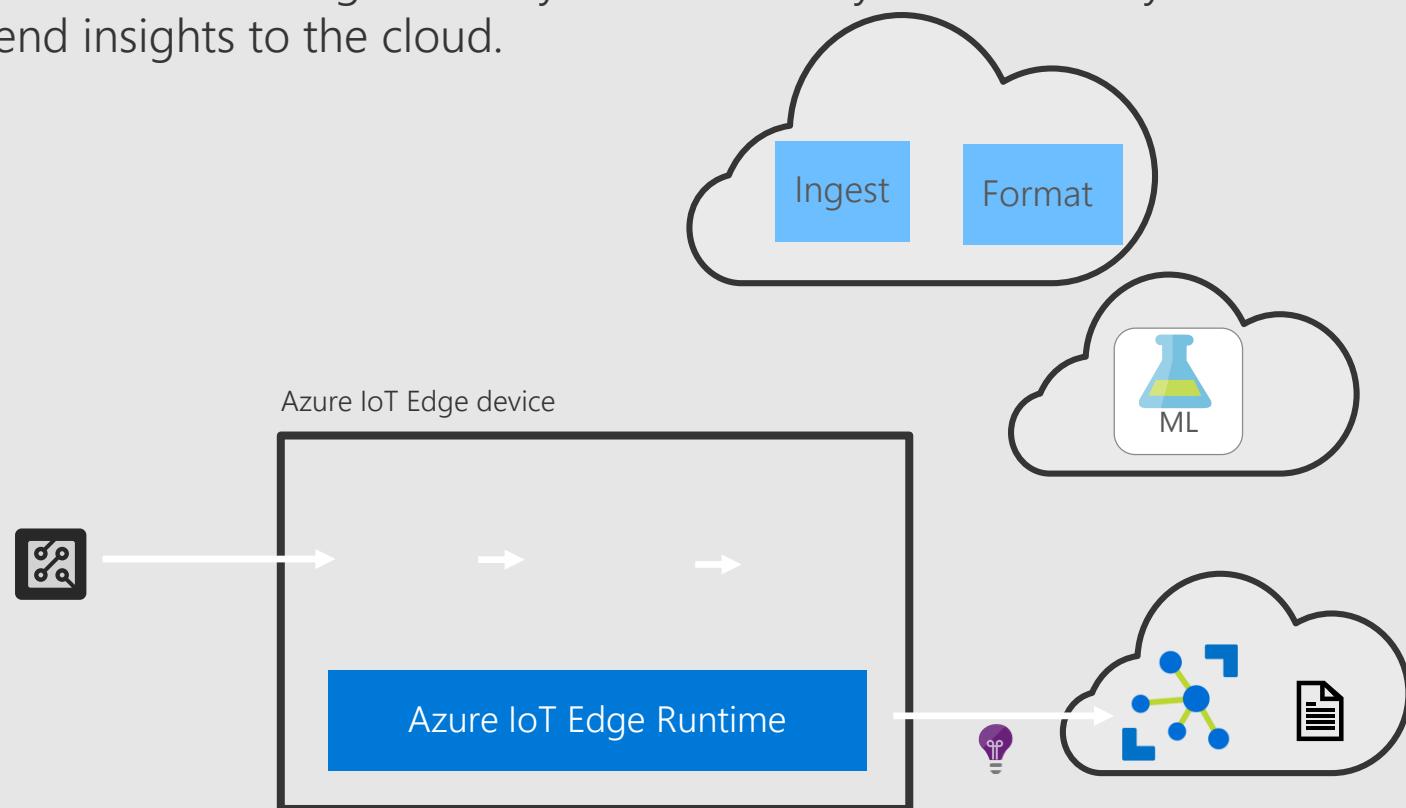
Configure a workflow

Target a device

Deploy

Monitor

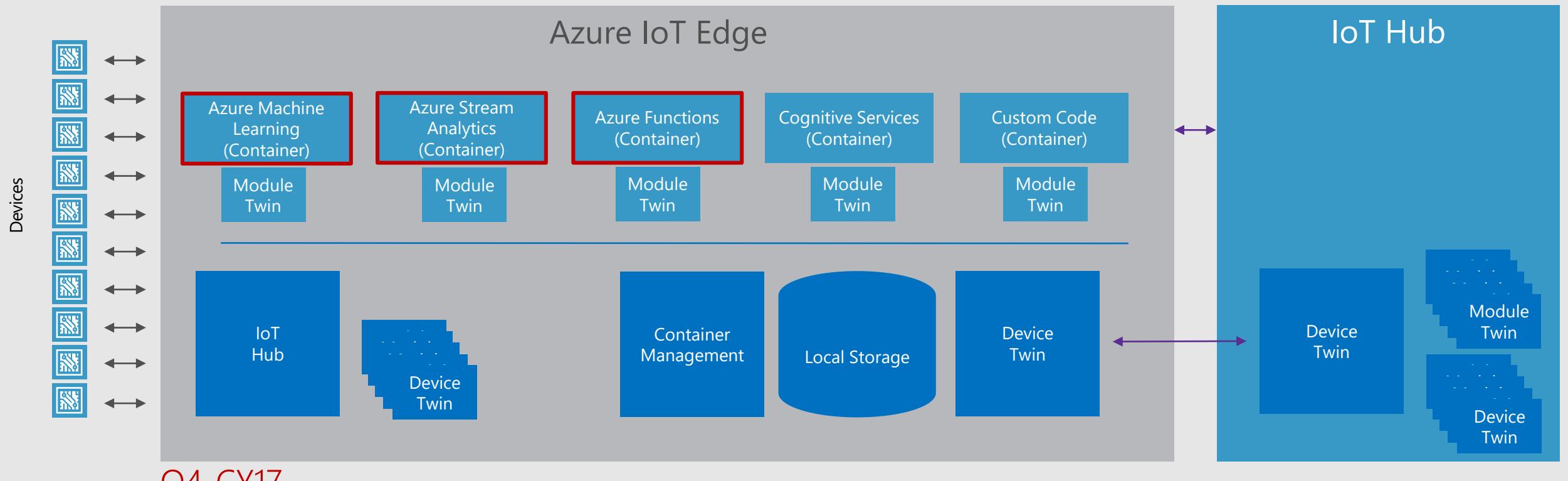
Scenario: Find insights locally from telemetry data and only send insights to the cloud.



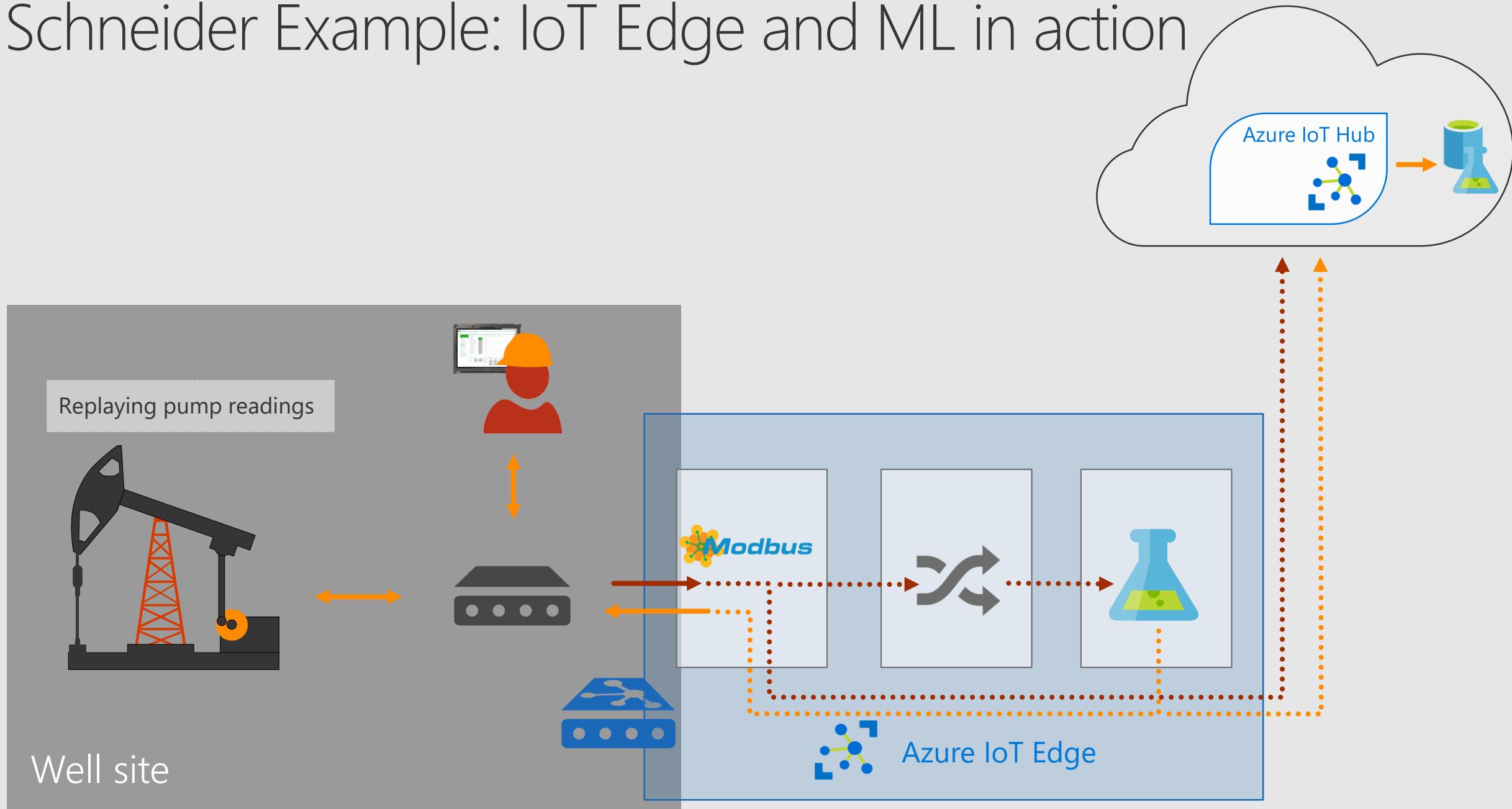
Azure IoT Edge

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



Schneider Example: IoT Edge and ML in action



Devices & Gateway SDK



SDK Library

SDK, agent libraries

Easily accessible libraries in GitHub

Cross platform support

Choose real time OS, platform and language

Device support

IP and access control capabilities

Connect IP, and non-IP devices

Support for direct connection devices and resource strained or non-IP devices via gateway and field protocols

Open source framework

Develop custom agents for your devices

Secure communication

Simple and secure D2C and C2D connectivity for messaging, device management and command and control

OS support

RTOS, Linux, Windows, Android, iOS etc.

Dev tool support & samples

IoTHub-explorer, Device Explorer, iothub-diagnostics

Service SDKs support

.NET

C# libraries supported:

- Windows Desktop (7,8,10)
- Universal Windows Platform



Node.js library:

- Node.js (v 0.10+)



Java library:

- Java (v 1.7+)



Python library:

- v 2.7.x
- v 3.5.x

Device SDKs support

C library:

- Microcontrollers
- RTOS
- Linux
- Windows



Node.js library:

- Node.js (v 0.10+)
- Node-RED



Java library:

- Java (v 1.7+)
- Android



C# libraries supported:

- Windows Desktop (7,8,10)
- Universal Windows Platform
- Windows 10 IoT Core
- Xamarin (iOS, Android)

.NET

Python library:



- v 2.7.x
- v 3.5.x

Device SDKs platform/OS support

Android (Java or Xamarin)	STM32
Arduino	TI RTOS
Debian Linux (v 7.5)	Ubilinux (v3.0)
ESP8266	Ubuntu Linux (v 14.04)
Fedora Linux (v 20)	Windows Desktop (7, 8, 10)
FreeRTOS	Windows IoT Core (v 10)
iOS (Xamarin)	Windows Server (v 2012 R2)
mbed OS (v 2.0)	Yocto Linux (v 2.1)
OpenWRT	
Raspbian Linux (v 3.18)	... more @ Azure Certified for IoT device catalog.

Packages and libraries

NuGet

C on Windows

.Net

Apt-get

C on Linux (Ubuntu/Debian)

Npm

Node.js

Maven

Java

PyPI

Python

Arduino lib

Mbed lib

IoT Hub Developers tools

IoTHub-explorer: node based CLI

npm install -g iothub-explorer

Device Explorer: Windows centennial app

Installer in releases of github.com/azure/azure-iot-sdks

New X-Plat UI tool coming soon

Coming soon!

Device discovery CLI

npm install -g device-discovery-cli

IoTHub Diagnostics tool

github.com/azure/iothub-diagnostics

IoT Developer Center Samples

Azure.com/iotdev

Azure IoT Edge scenario and benefits

End user IoT Edge scenarios

Security and isolation of devices

Bridge networks and isolate public Internet

Ensure cloud security and encryption

Integration with any device

Low resource devices

Device agents can be simplified

Connect new and legacy devices

Cloud offload

Run logic on premises to reduce network & cloud consumption

Low-latency / critical messages

Segregate message types

Edge processing

Enable time-sensitive decisions

Run edge analytics

Auditing - Central choke point to apply data policies

Batching

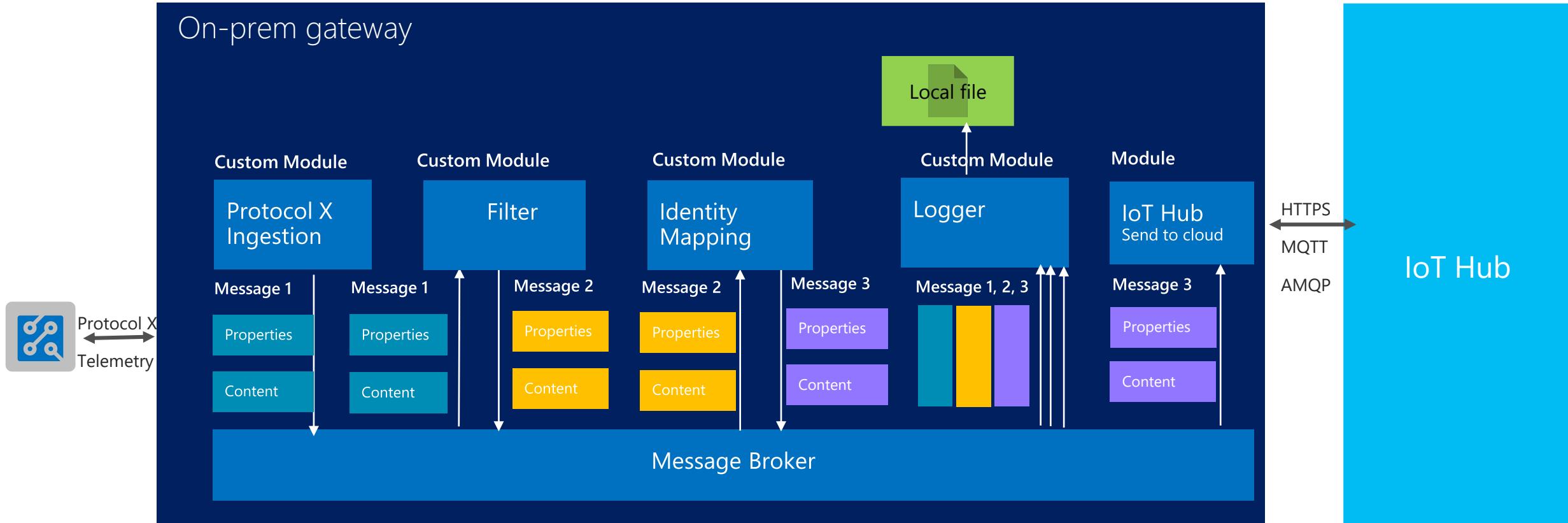
Optimize bandwidth and messages

Benefits

1. **SI & end user** – Reduced time to market and maintenance costs for gateway solutions
2. **OEM** - Increased margins on gateway hardware compared to IoT Devices
3. **ISV** - Monetize modules built on top of Azure IoT Gateway SDK

Azure IoT Edge SDK architecture (current)

<https://github.com/Azure/iot-edge>



BLE (Bluetooth Low Energy) Module

Logger Module

Identity Map Module

IoT Hub Module

[OPC-UA](#)

[Modbus](#)

Simulated Device Module

Hello World Module

...and many more to come

Division of work

SDK facilitates gateway software creation. It does not provide an out-of-the-box solution for every customer.

Provided by SDK

- Pluggable module architecture (SDK infrastructure)
- Simplified gateway creation (SDK infrastructure)
- Efficient D2C and C2D connectivity (modules)
- Protocol translation example (modules)

Partner/Customer work

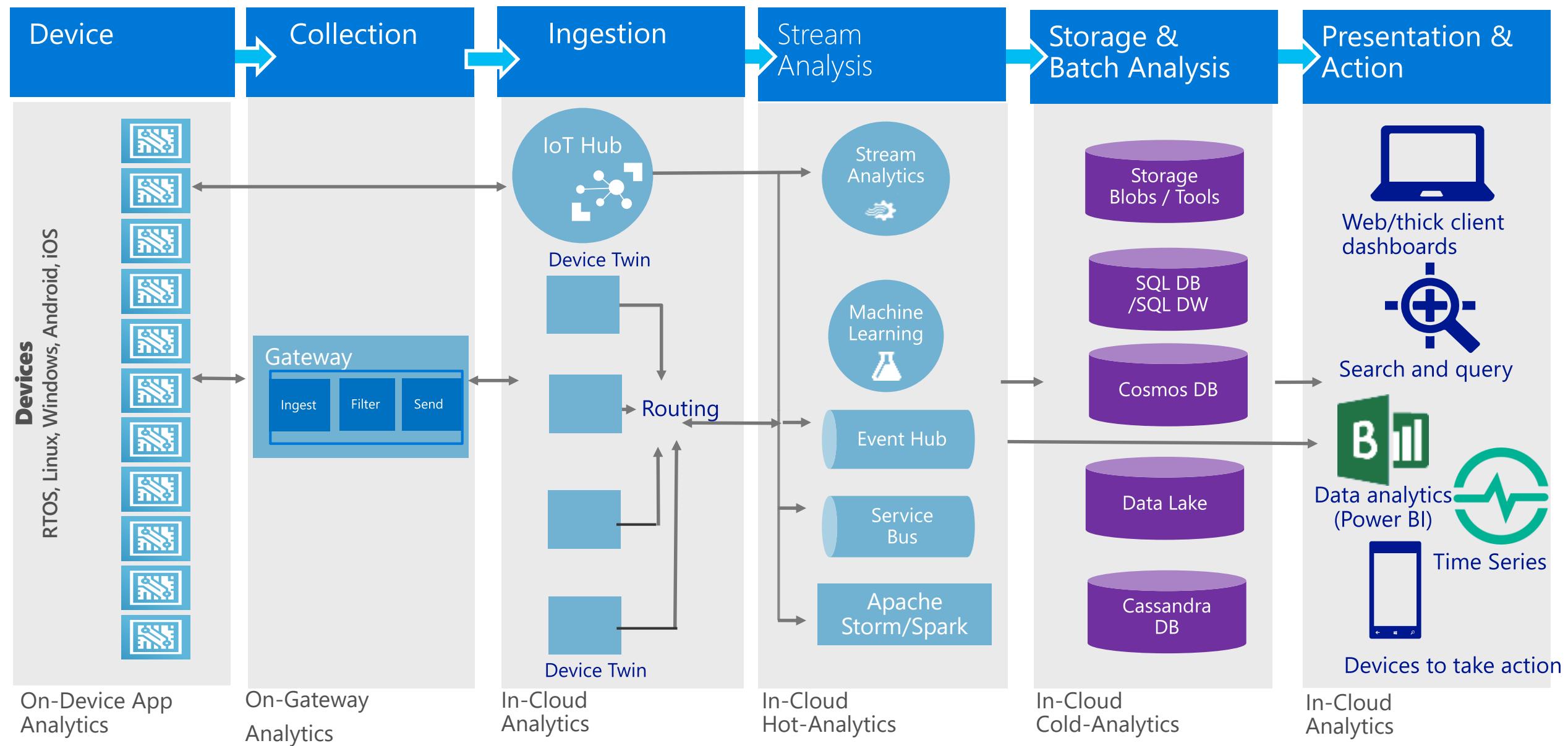
- Configuration of module pipeline
- Writing modules for any logic not provided by SDK
- Deployment and maintenance of gateway solution to hardware

Analytics & Visualization
(and other IoT services)

Other IoT – Azure Services

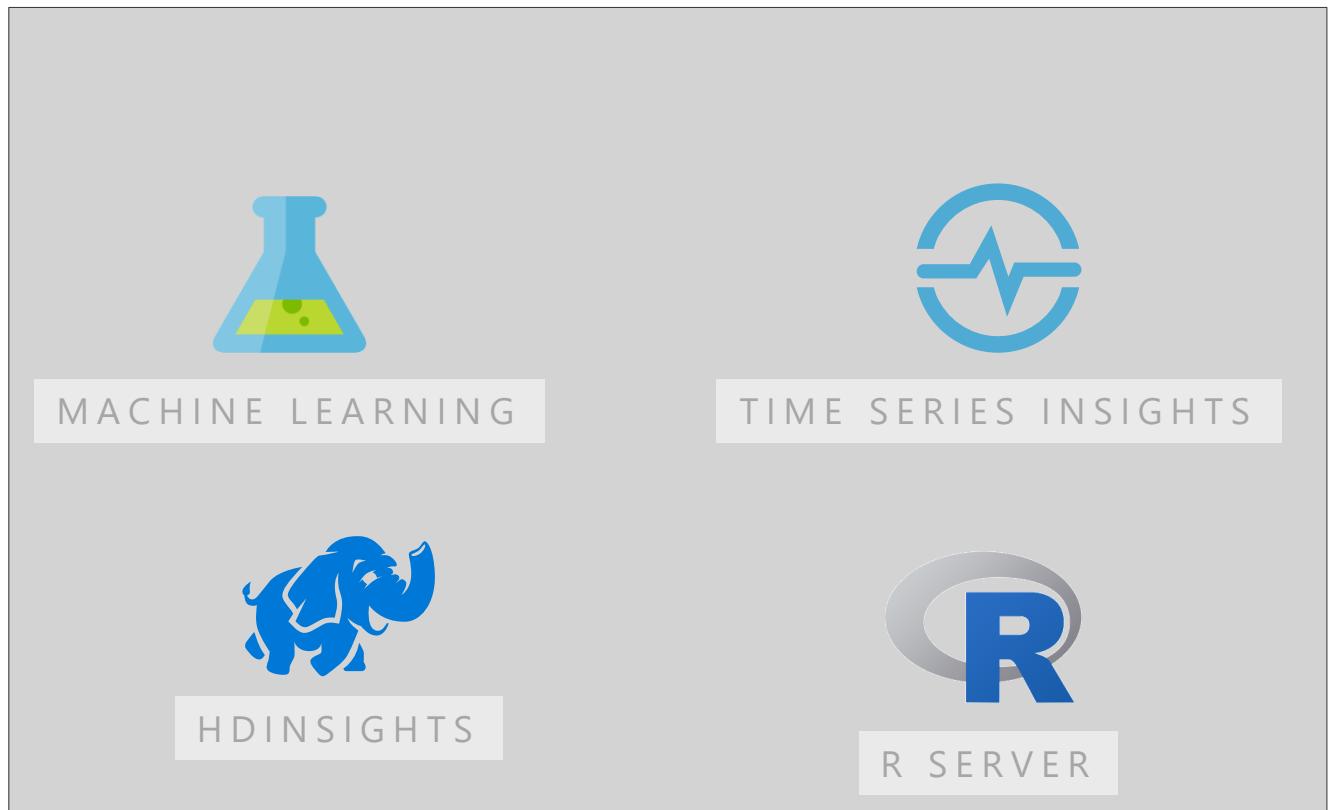
- Enterprise Integration
- Messaging
- Serverless
- Analytics & Visualization
 - Time Series Insights
 - Azure Stream Analytics
 - Azure ML
 - Cosmos DB

Azure IoT Analytics Patterns



Discover

- Mine historical data
- Discover trends and patterns
- Find key insights



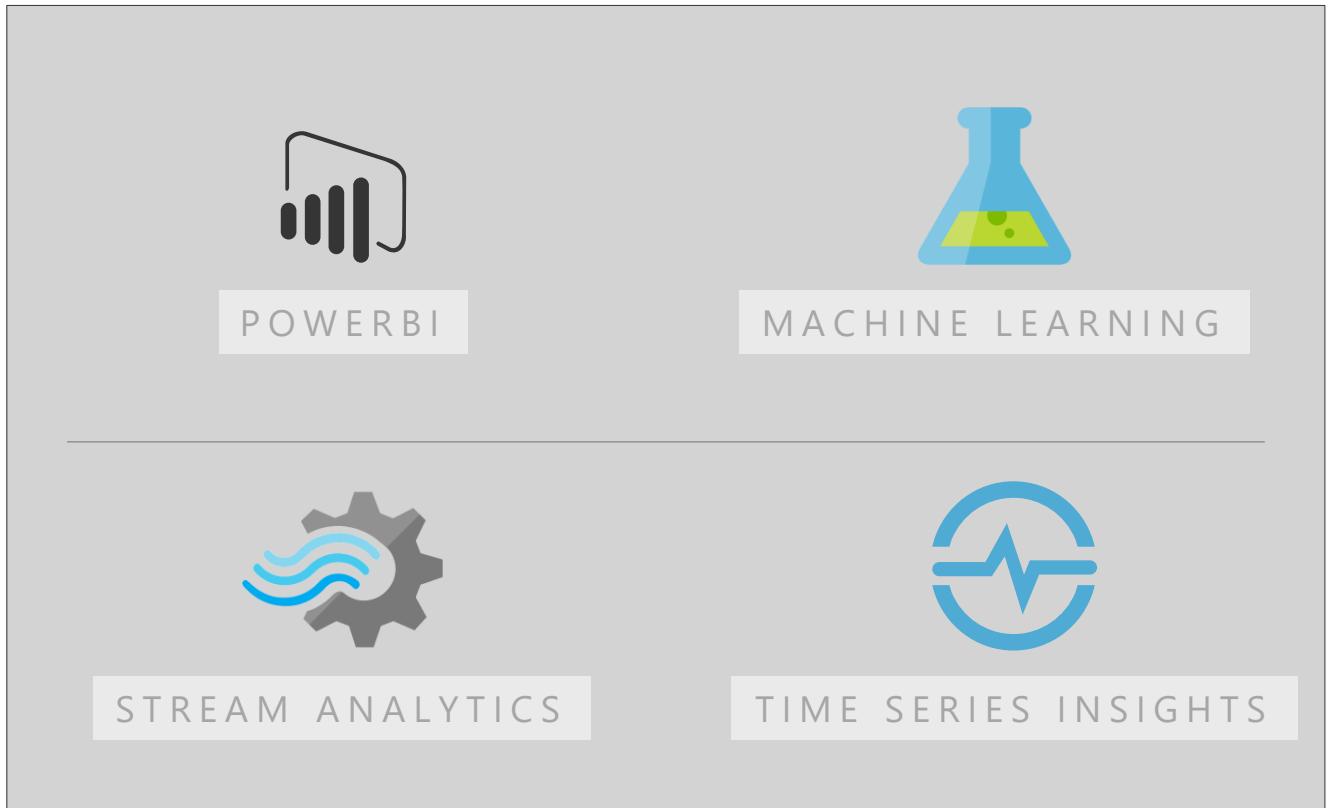
Operationalize

- Implement rules/thresholds
- Process real time data
- Mission critical reporting



Refine

- Root cause analysis
- Iterate rules and thresholds
- Continuous improvement

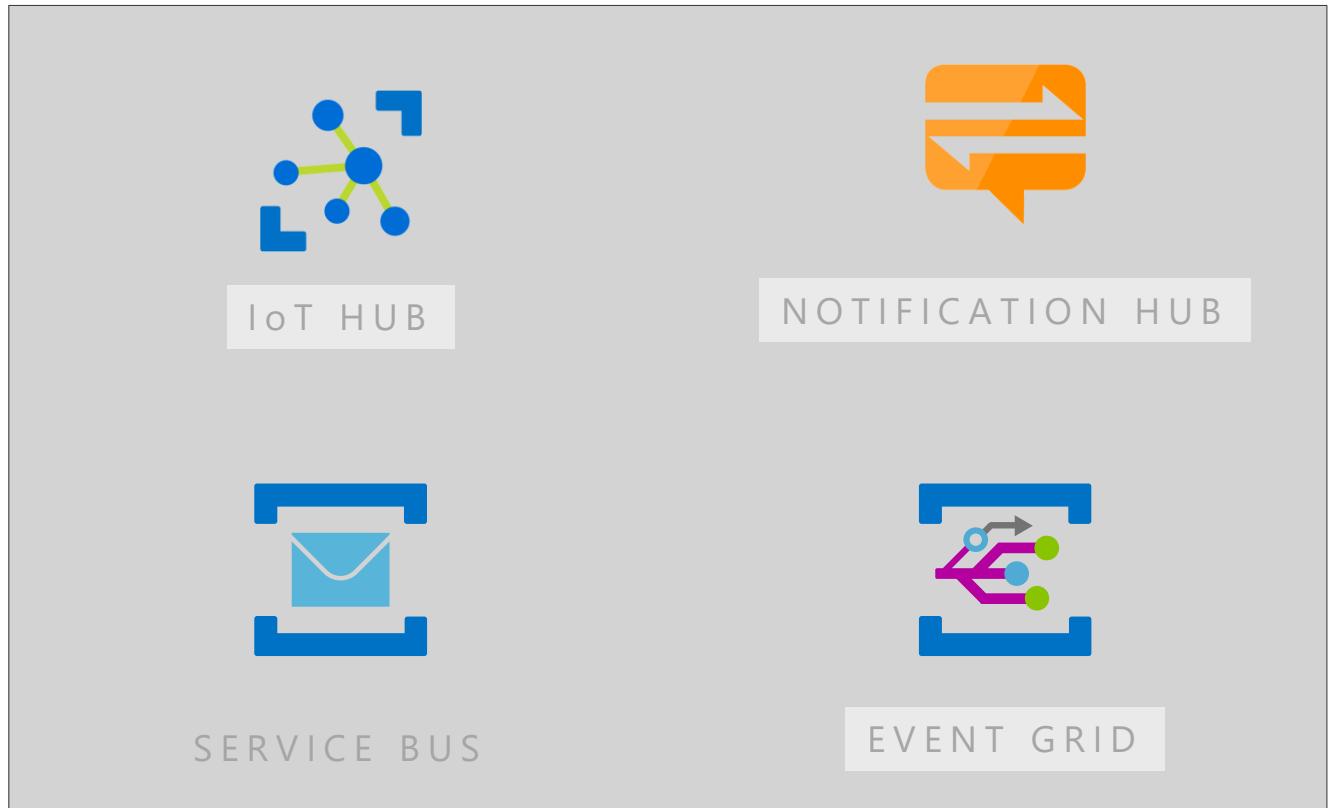


Respond

Manage devices

Send notifications and alerts

Execute workflows

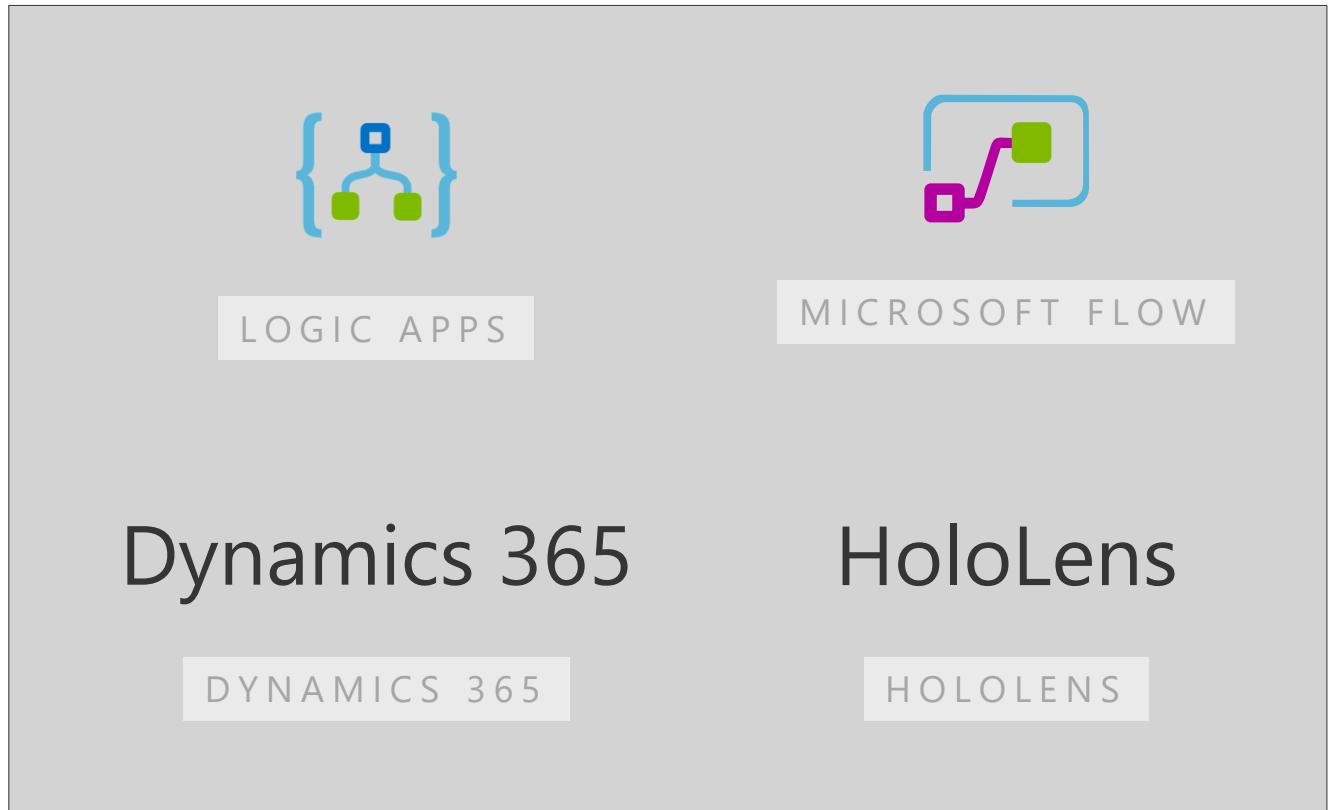


Integrate

Update and order inventory

Schedule maintenance

Empower employees (AR/VR)

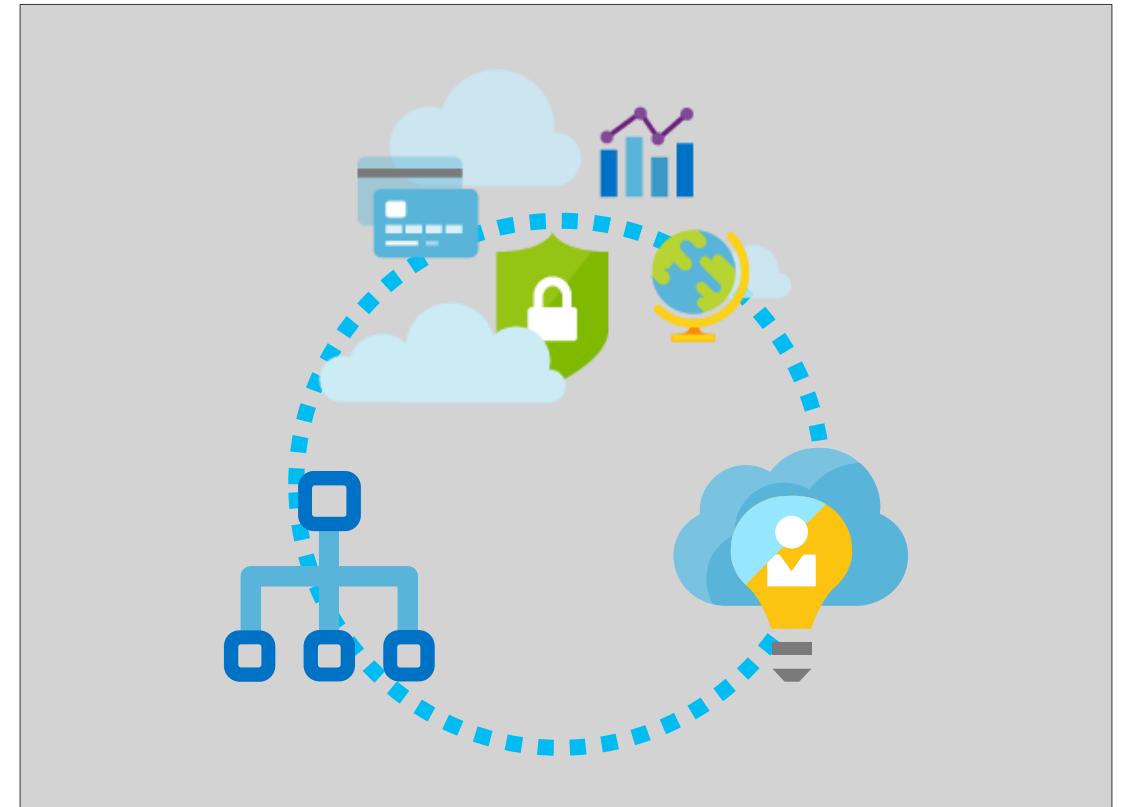


Improve

Improve products

Strengthen customer relationship

Streamline workflow efficiency



IoT Devices and “Time-Series” Data

- IoT sensor data typically consists of time base measurements
- Storing, indexing, querying, analyzing, visualizing can be challenging
- Especially considering “IoT scale”



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

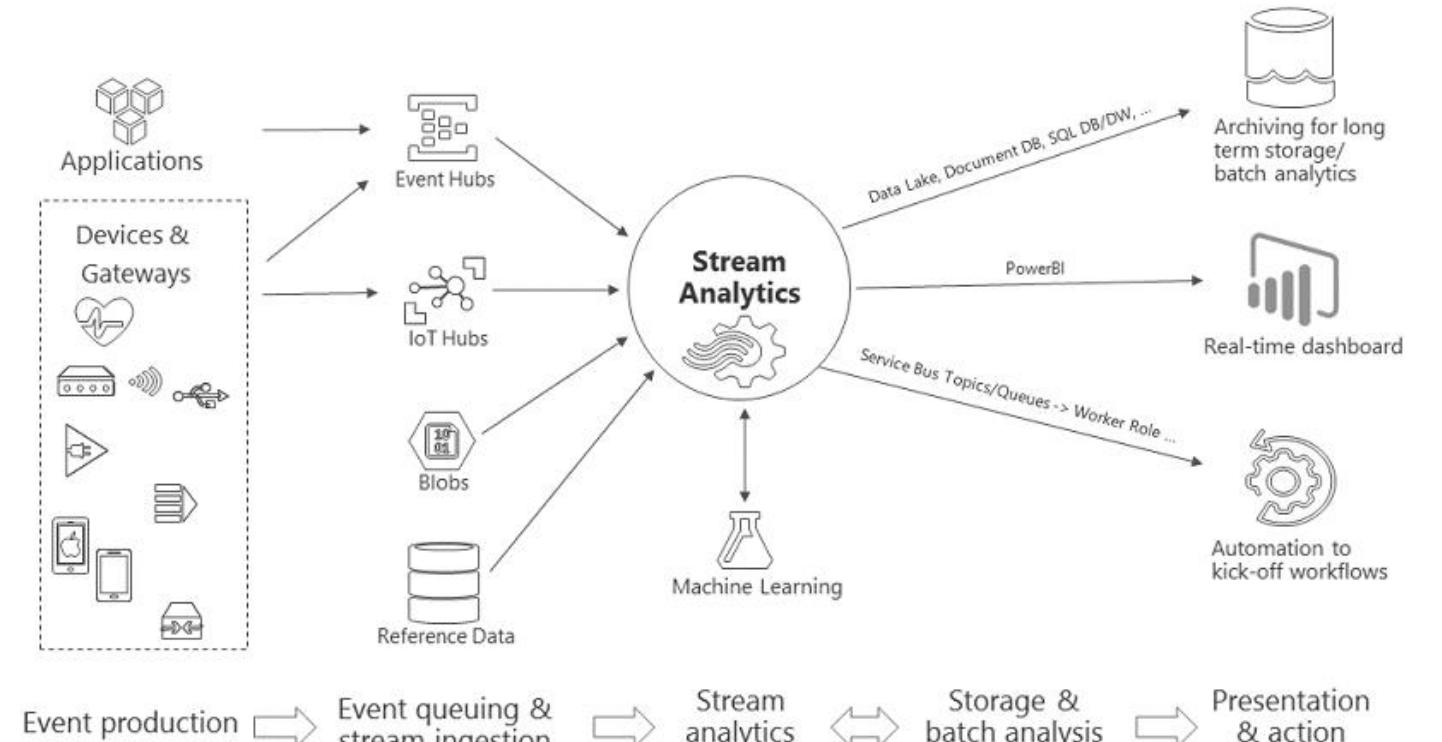


Demo

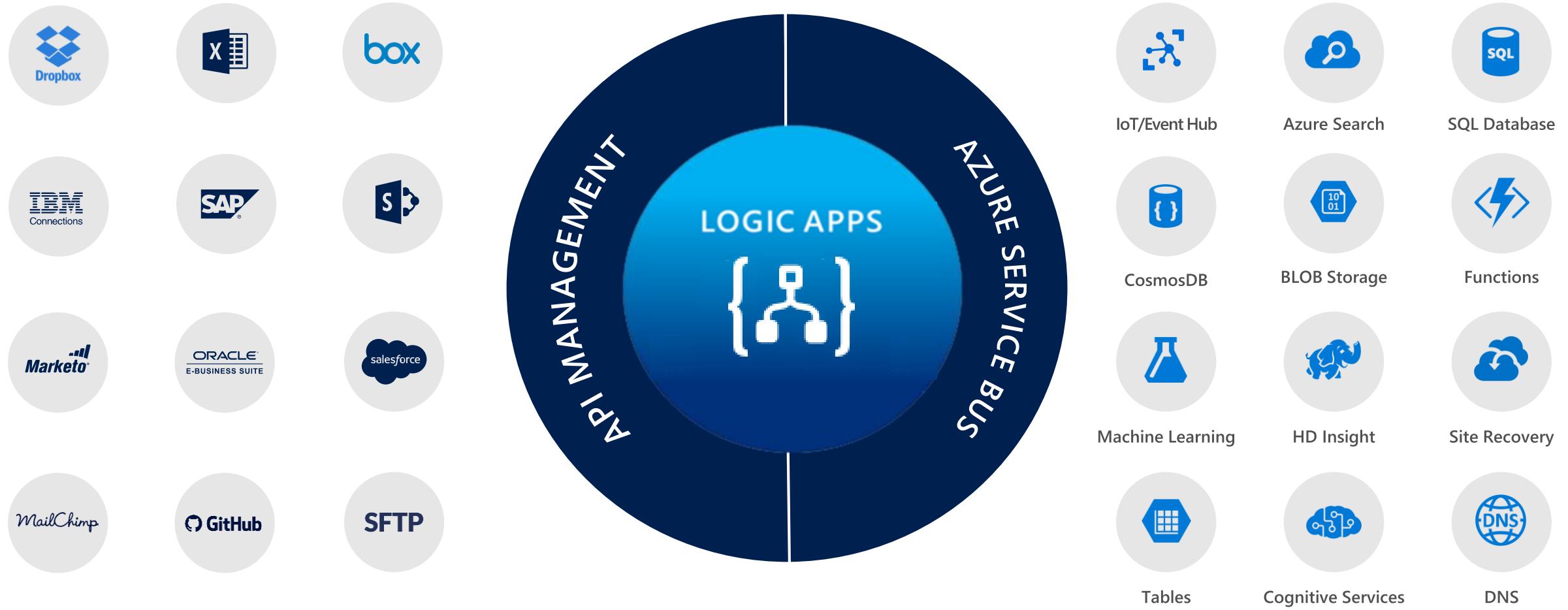
Time series insight

Azure Stream Analytics

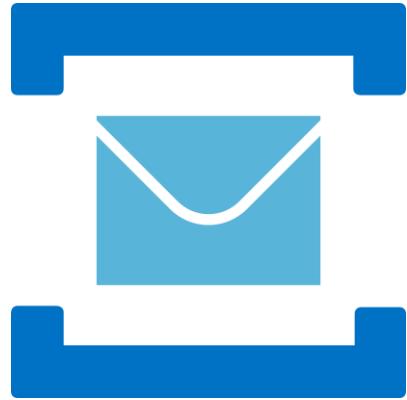
- Fully managed service
 - Pay as you go
 - Enterprise grade SLA
- Productivity
 - SQL Like query language
 - Aggregate data
 - Source/sink integrations
 - IoT Hub, Event Hubs
 - Power BI, Storage, Event Hubs, Data Lake, etc
 - Call custom functions and ML



Microsoft's Cloud Integration Platform



Messaging



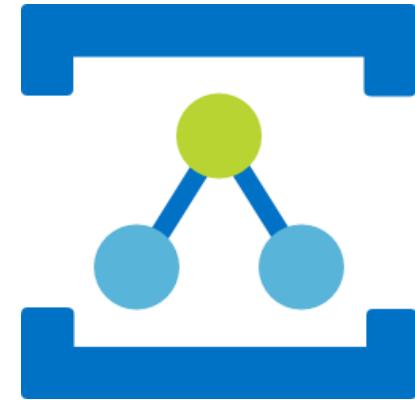
Service Bus

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



MongoDB API



Key-value



Column-family



Document



Graph

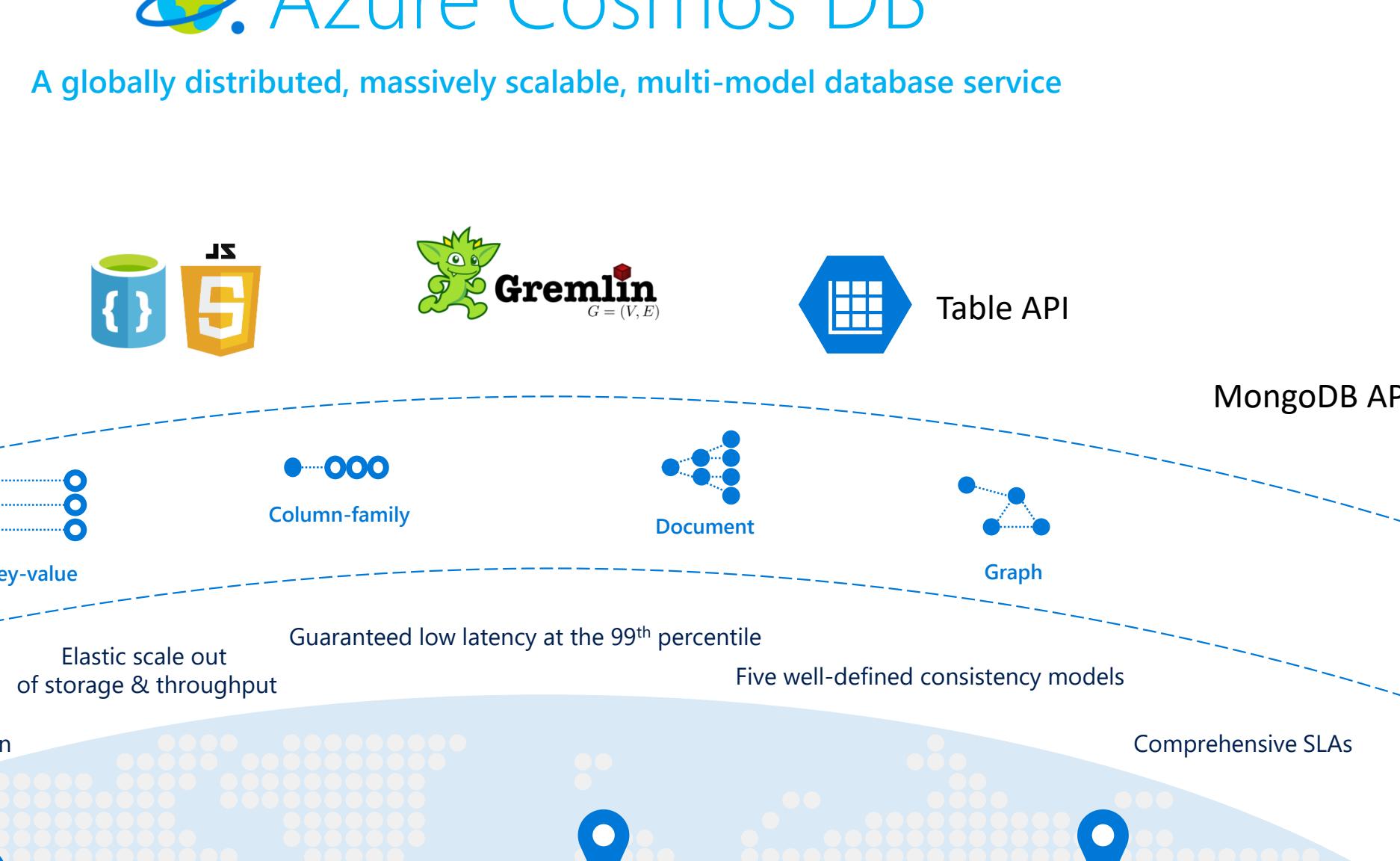
Turnkey global distribution

Elastic scale out
of storage & throughput

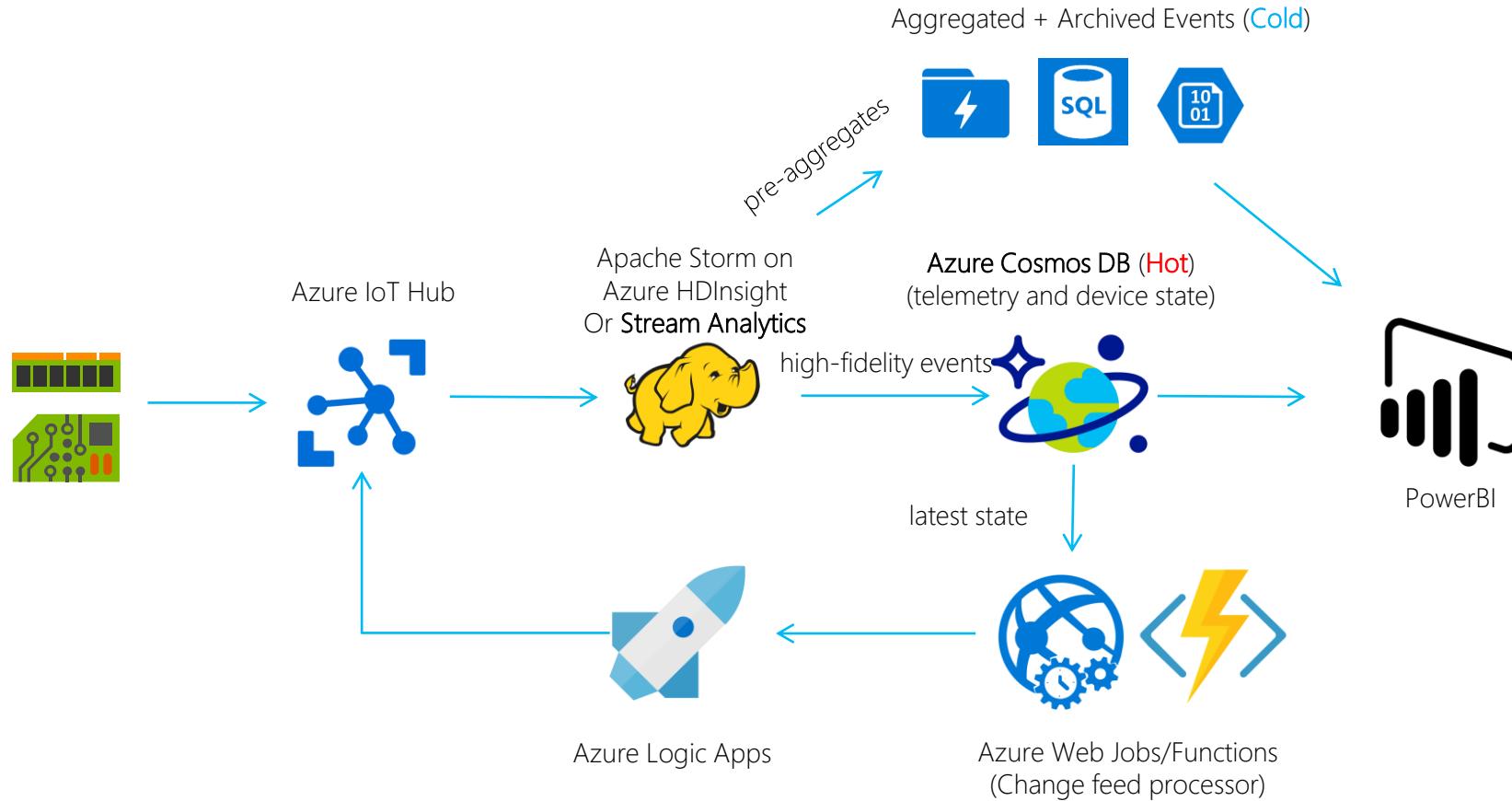
Guaranteed low latency at the 99th percentile

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)

Demo

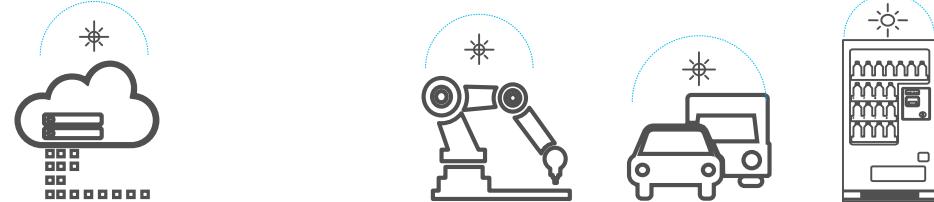
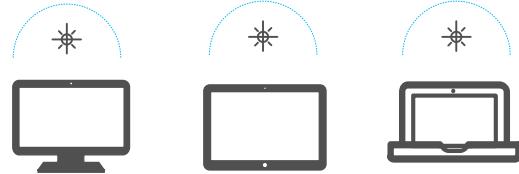
Functions

IoT Security

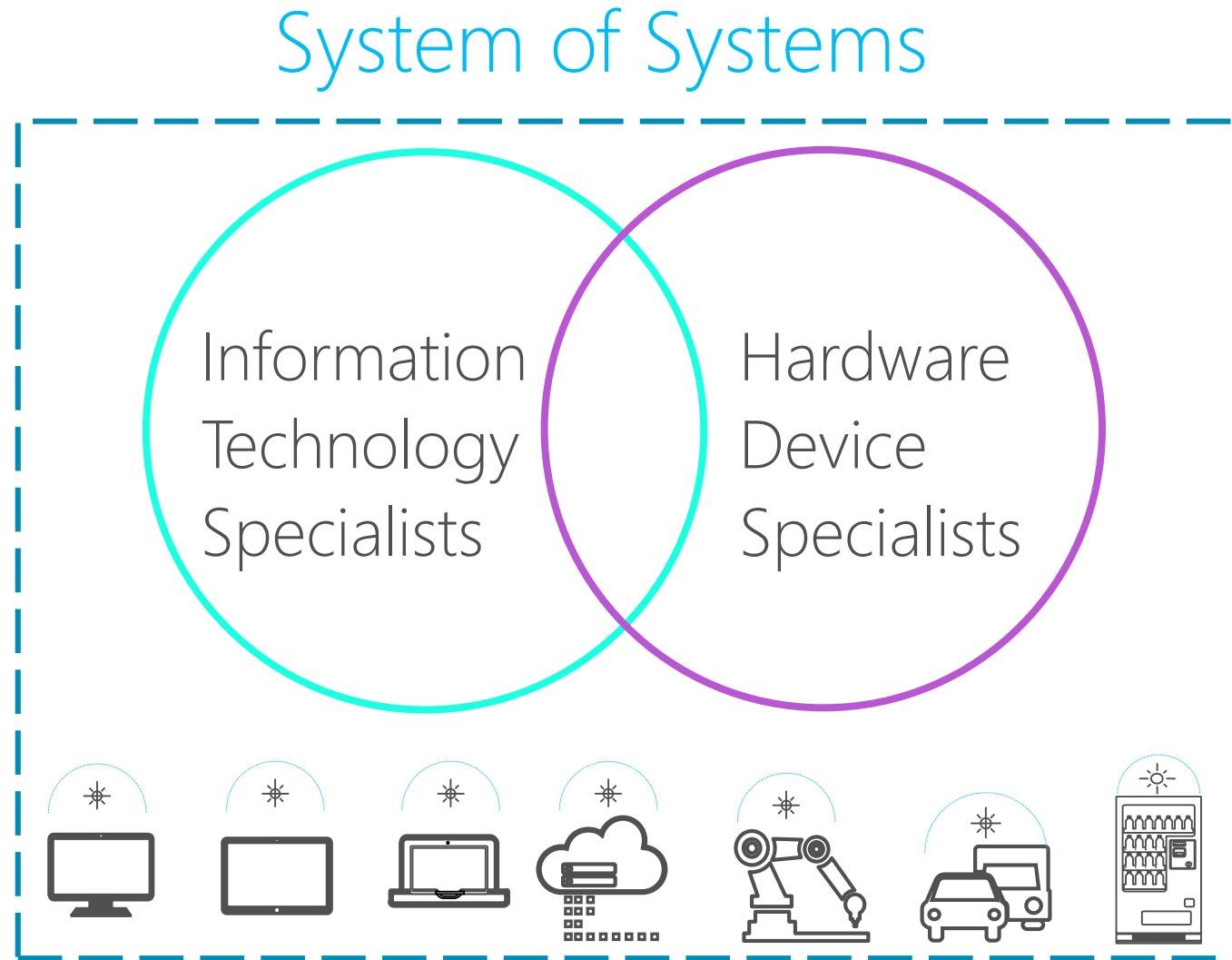


Why securing Internet of Things is hard

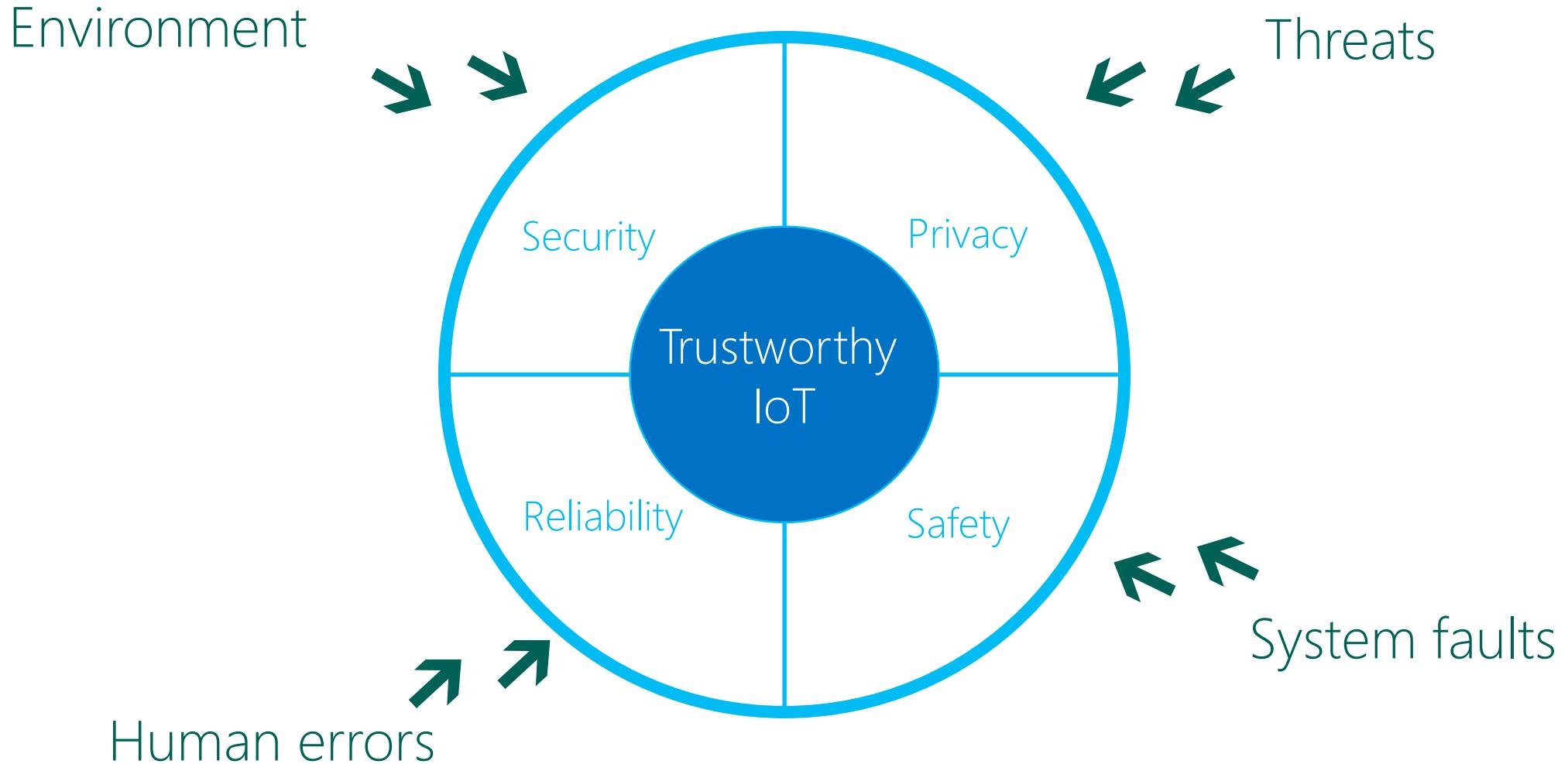
System of Systems



Why securing Internet of Things is hard



Trustworthy Internet of Things

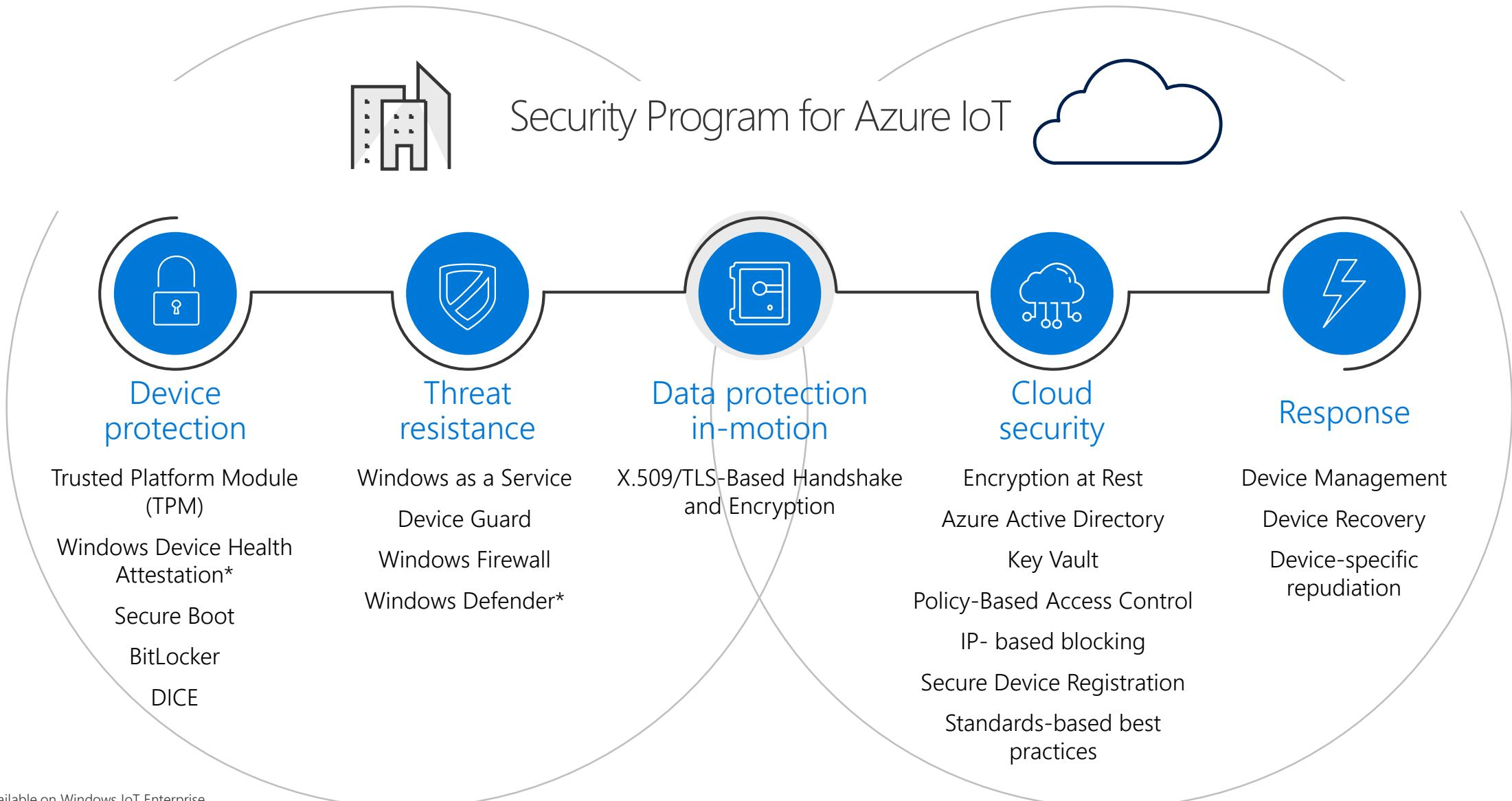


Driving Security Innovation: 7 Properties of Device Security

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

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



Security Program for Azure IoT

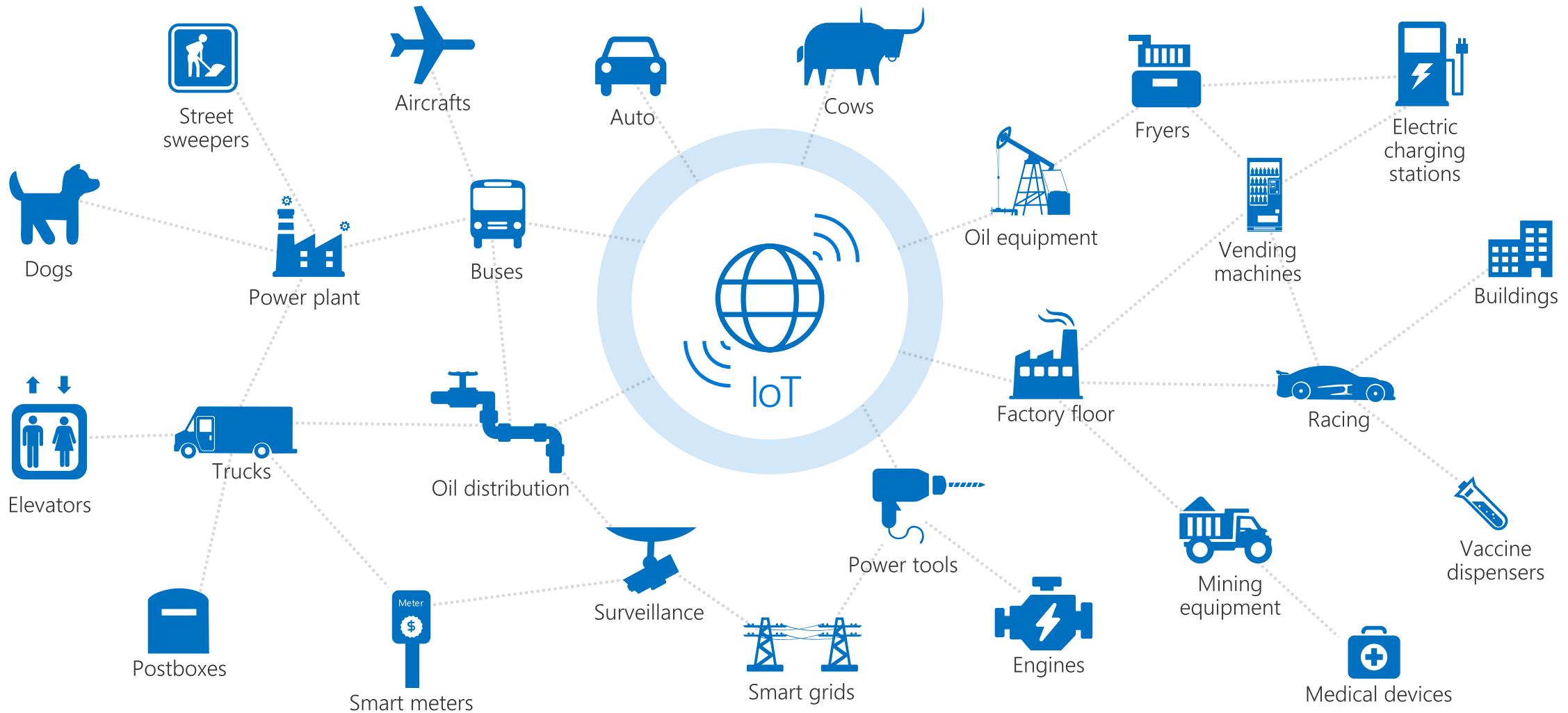
- Microsoft's Security Program
 - Trusted security auditors trained on Azure IoT
 - Discover issues, get recommended remediations
 - Keep your IoT Solution secure



Not all partners may be listed; check internetofyourthings.com for latest status

Customer cases

Innovation at work – real IoT use cases



IoT is already delivering results across industries



Connected chillers are back online 9x faster than unconnected equipment, avoiding more than \$300,000 in hourly downtime costs



"Power by the hour" model maximizes aircraft availability, while cutting fuel usage by 1%, saving \$250,000 per plane, per year



Data from sensors and systems create valuable business intelligence and reduce downtime by 50%



Access to production and supply chain data worldwide can reduce downtime costs by as much as \$300,000 per day



"The Microsoft Azure platform makes it a lot easier for us to deliver on our vision without getting stuck on the individual IT components. We can focus on our end solution and delivering real value to customers rather than on managing the infrastructure."

Richard Beesley

Senior Enterprise Architect Data Services
Rolls Royce





Rolls-Royce has over 13,000 engines for commercial aircraft in service around the world, and for the past 20 years, it has offered customers comprehensive engine maintenance services that help keep aircraft available and efficient. The company's TotalCare® Services employ a "*power by the hour*" model in which customers pay based on engine flying hours.

The responsibility for engine reliability and maintenance rests with Rolls-Royce, which analyzes engine data to manage customers' engine maintenance and maximize aircraft availability. Rolls-Royce is using the Microsoft Azure platform to fundamentally transform how it uses data to better serve its customers.

Rolls-Royce and Microsoft collaborate to create new digital capabilities

Rolls-Royce has over 13,000 engines for commercial aircraft in service around the world, and for the past 20 years, it has offered customers comprehensive engine maintenance services that help keep aircraft available and efficient. The company's TotalCare® Services employ a "power by the hour" model in which customers pay based on engine flying hours. The responsibility for engine reliability and maintenance rests with Rolls-Royce, which analyzes engine data to manage customers' engine maintenance and maximize aircraft availability. Rolls-Royce is using the Microsoft Azure platform to fundamentally transform how it uses data to better serve its customers.

"The Microsoft Azure platform makes it a lot easier for us to deliver on our vision without getting stuck on the individual IT components. We can focus on our end solution and delivering real value to customers rather than on managing the infrastructure."



Microsoft

Rolls-Royce



<https://aka.ms/rollsroyce>

Hands-on Labs

Hands-on lab – Remote monitoring

LAB 1: Deploy PCS Remote Monitoring (Preview) and review the code.

LAB 2: Add a “physical” device to the solution

LAB 3: Add Time Series Insights to the solution

LAB 4: Working with device state using device twins

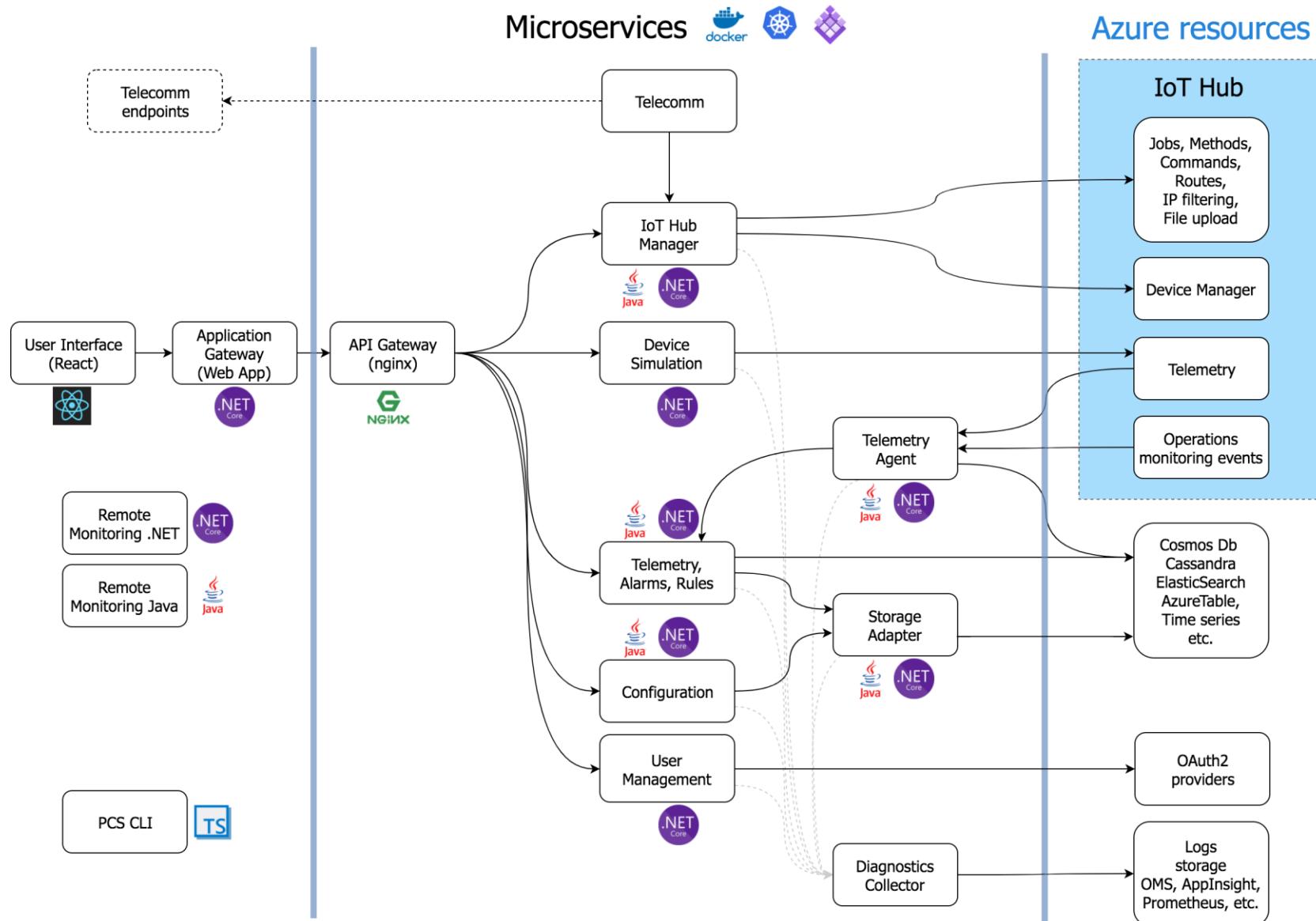
 1) Set up desired properties

 2) Change “physical” device to act upon change

LAB 5: Create a IoT Hub Job

LAB 6: Add a logic app for Enterprise Integration

PCS Remote Monitoring (preview)



LAB 1: Deploy Remote Monitoring (Basic)

Deploy the preconfigured solution

Before you deploy the preconfigured solution to your Azure subscription, you must choose some configuration options:+

1. Log on to azureiotsuite.com using your Azure account credentials, and click + to create a solution.
2. Click Select on the Remote monitoring tile.
3. On the Create Remote Monitoring solution page, enter a Solution name for your remote monitoring preconfigured solution.
4. Select a **Basic** or **Standard** deployment. If you are deploying the solution to learn how it works or to run a demonstration, choose the **Basic** option to minimize costs.
5. Choose either **Java** or **.NET** as the language. All the microservices are available as either Java or .NET implementations.
6. Review the **Solution details** panel for more information about your configuration choices.
7. Select the **Subscription** and **Region** you want to use to provision the solution.
8. Click **Create Solution** to begin the provisioning process. This process typically takes several minutes to run.
9. Now that you have deployed the remote monitoring solution, explore the capabilities of the solution dashboard



Solution types



Remote monitoring

Connect and monitor your devices to analyze untapped data and improve business outcomes by automating processes.

Select

Remote monitoring

Connect and monitor your devices to analyze untapped data and improve business outcomes by automating processes.

Preview highlights include:

- Redesigned user interface
- Microservices-based architecture
- Availability in both .NET and Java
- View an [interactive demo](#)

Select

Connected factory

Accelerate your journey to Industrie 4.0 – connect, monitor and control industrial devices for insights using OPC UA to drive operational productivity and profitability.

Select

Predictive maintenance

Anticipate maintenance needs and avoid unscheduled downtime by connecting and monitoring your devices for predictive maintenance.

Select

LAB 1: Get the code

Setup

1. Install git for Windows: <https://git-scm.com/download/win>
2. Open a terminal window or command shell
3. Create a directory to hold the code
4. Check out the repository and submodules

```
git clone --recursive https://github.com/Azure/azure-iot-pcs-remote-monitoring-dotnet.git
```

LAB 1: Code walk-through

Name	Date modified	Type	Size
📁 .git	5-11-2017 14:57	File folder	
📁 .github	2-11-2017 19:56	File folder	
📁 auth	2-11-2017 19:57	File folder	
📁 cli	2-11-2017 19:57	File folder	
📁 config	2-11-2017 21:07	File folder	
📁 device-simulation	2-11-2017 21:18	File folder	
📁 docs	2-11-2017 19:56	File folder	
📁 iothub-manager	2-11-2017 21:06	File folder	
📁 reverse-proxy	2-11-2017 19:56	File folder	
📁 scripts	2-11-2017 19:56	File folder	
📁 storage-adapter	2-11-2017 20:59	File folder	
📁 telemetry	2-11-2017 21:03	File folder	
📁 telemetry-agent	2-11-2017 21:01	File folder	
📁 webui	2-11-2017 19:57	File folder	
📄 .gitattributes	2-11-2017 19:56	Text Document	1 KB
📄 .gitignore	2-11-2017 19:56	Text Document	11 KB
📄 .gitmodules	2-11-2017 19:56	Text Document	1 KB
📄 .travis.yml	2-11-2017 19:56	YML File	1 KB
📄 CONTRIBUTING.md	2-11-2017 19:56	MD File	3 KB
📄 DEVELOPMENT.md	2-11-2017 19:56	MD File	1 KB
📄 LICENSE	2-11-2017 19:56	File	2 KB
📄 README.md	2-11-2017 19:56	MD File	17 KB

Each of the modules has a Visual Studio solution. Open a solution and review the code.

Understand how to customize the solution:

<https://docs.microsoft.com/en-us/azure/iot-suite/iot-suite-remote-monitoring-customize>

LAB 2: Adding a “physical” device to the solution

In this LAB, you implement a two devices that sends telemetry to the remote monitoring preconfigured solution using node.js. One chiller and one custom device

1. Create a new device in the Remote Monitoring solution.
2. Open bash on Windows 10 (activate if needed).
3. Create a Chiller Node.js solution using the online tutorial
<https://docs.microsoft.com/en-us/azure/iot-suite/iot-suite-connecting-devices-node#create-a-nodejs-solution>.
4. Start the node.js solution and see the outcome on the Remote Monitoring Solution.

LAB 2: Adding a “physical” device to the solution

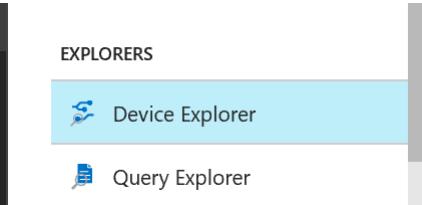
1. Create a new device on the Remote Monitoring solution.
2. Copy the *remote_monitoring.js* file to *my_custom_device.js*
3. Define completely new telemetry data for the device and adjust the .js file to represent them. Don’t forget to adjust the *connectionString* to the new device.

LAB 3: Add Time Series Insights to the solution

1. 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>
2. 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>
3. Grant data access to a Time Series Insights environment using Azure portal
<https://docs.microsoft.com/en-us/azure/time-series-insights/time-series-insights-data-access>
4. Access your Time Series Insight environment
<https://insights.timeseries.azure.com/>
5. Have a look at the Temperature by Device Id. Add pressure to the time series.

LAB 4: Working with device state using device twins

1. Open the Azure Portal and browse to your IoT Hub
2. Select your “physical” 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: Working with device state using device twins

1. Open the node.js file of your device in bash
2. 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');  
    });  
});
```

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

LAB 5: Create a IoT Hub job

1. Adjust your “physical” device and add a direct method “ReactOnJob”
 - a) Add “ReactOnJob” to SupportedMethods
 - b) Add the client.onDeviceMethod for “ReactOnJob”
 - c) Implement the defined method to update a property.
2. Create a nodejs job “jobService”
 - a) 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.

Use <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-node-node-schedule-jobs> as your guideline

LAB 6: Add a logic app for Enterprise Integration

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);  
}
```

Training

Training for Azure IoT

Training provider	Class	Price
	<u>Developing IoT Solutions with Azure IoT</u>	Free (auditing) or \$99 (certificate of completion)
 Microsoft Virtual Academy	<u>Introduction to Azure IoT</u>	Free
 Linux Academy	<u>Azure IoT Essentials</u>	Free
 Linux Academy	<u>IoT for the Enterprise</u>	Paid (Free 7-day trial, \$19/month afterwards)
	<u>Internet of Things with Azure</u>	Paid (Free 7-day trial, \$29/month afterwards)
	<u>Building IoT Solutions with Azure</u>	Paid (Free 7-day trial, \$29.95/month afterwards)

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

Wrap up & Q&A

