



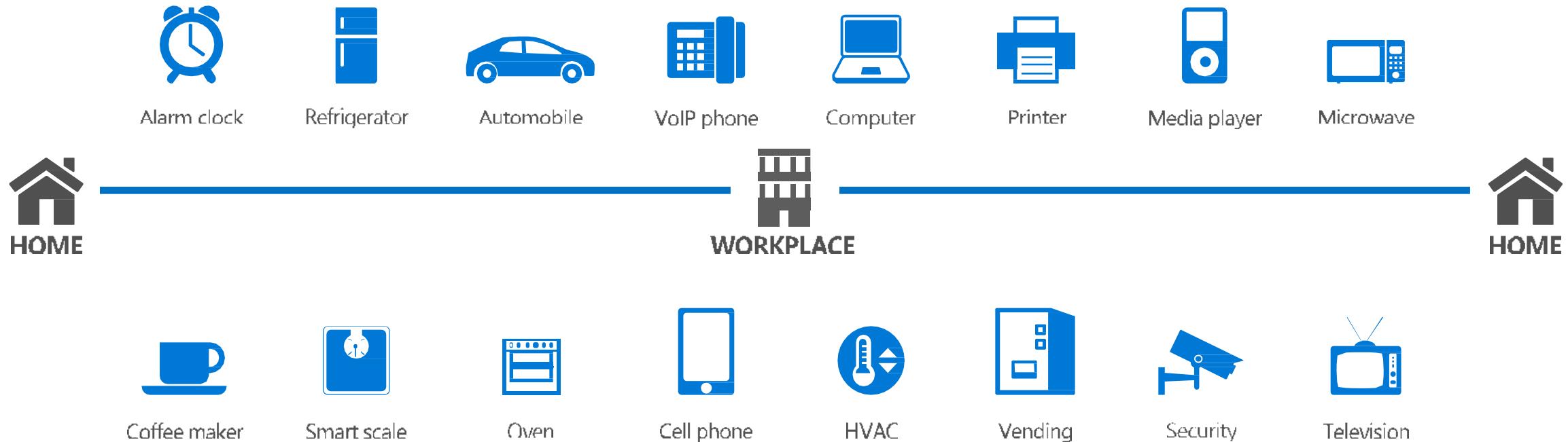
Internet of Things with Microsoft Azure

Fadil
Gravicode

About Me



IoT 2010



IoT 2016



Medication adherence



Health monitoring



Pet tracking



Behavior modification



Object tracking



Child and elder monitoring



Sports and fitness



Smart lighting



Indoor navigation



Beacons and proximity



Trip tracking and car health



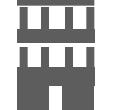
HOME



Smart appliances



Food and nutrition tracking



WORKPLACE



Identity



Office equipment



Smart vending machines



HOME



Bike ride stats and protection



Control



Home security



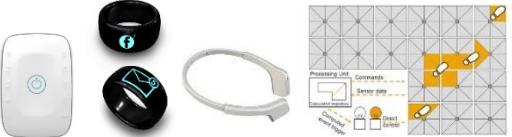
Home automation



Leak detection



Garden, lawn and plant care



New devices and sensors



Entertainment systems

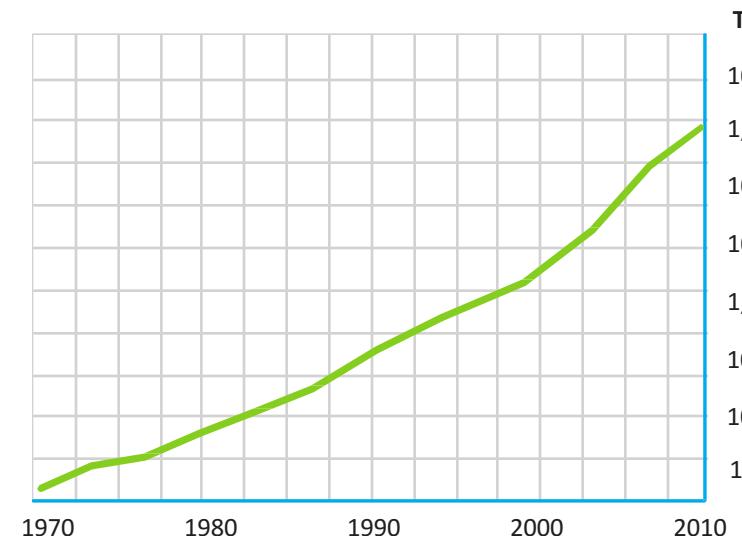
\$7.2 TRILLION

worldwide market for IoT solutions
by 2020

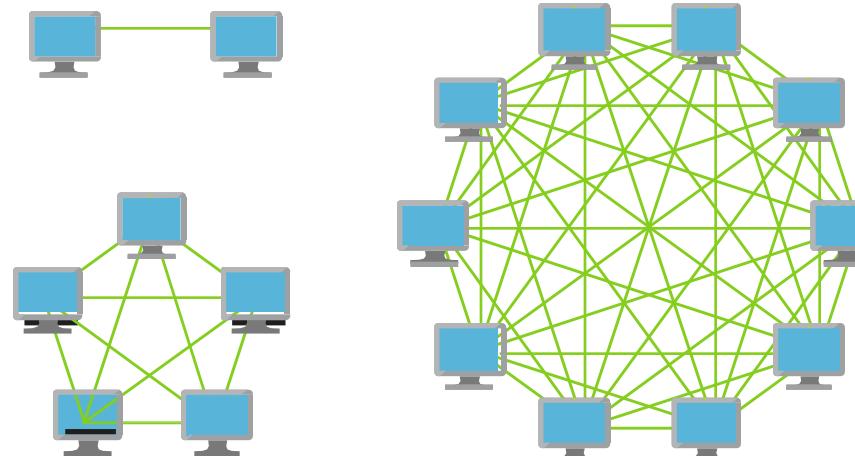
IDC: Worldwide and Regional Internet of Things (IoT) 2014–2020 Forecast

Disruptive Forces

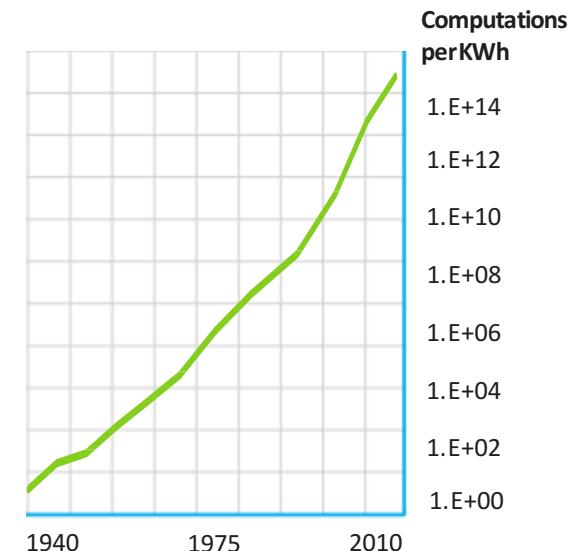
Moore's Law



Metcalf's Law



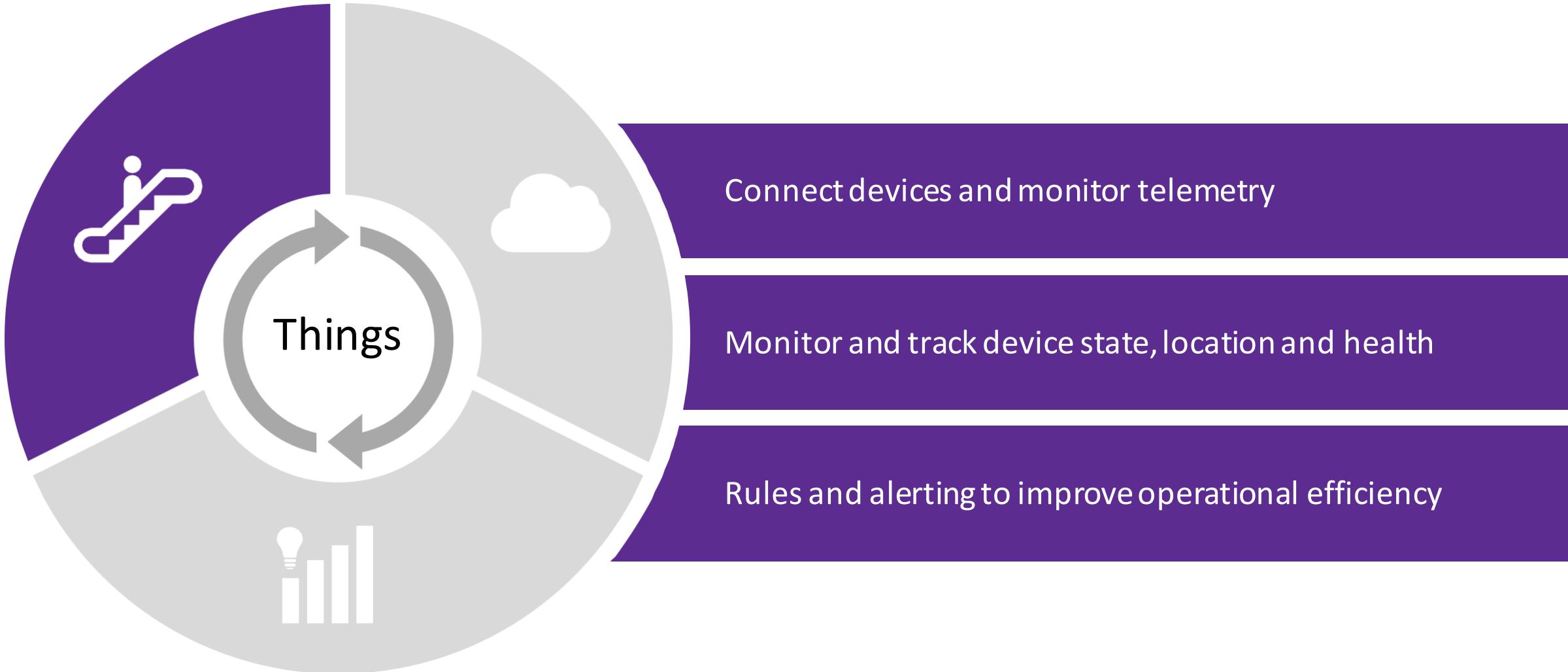
Koomey's Law



And more importantly:
what can you do by combining and analyzing signals from all of these IoT devices?

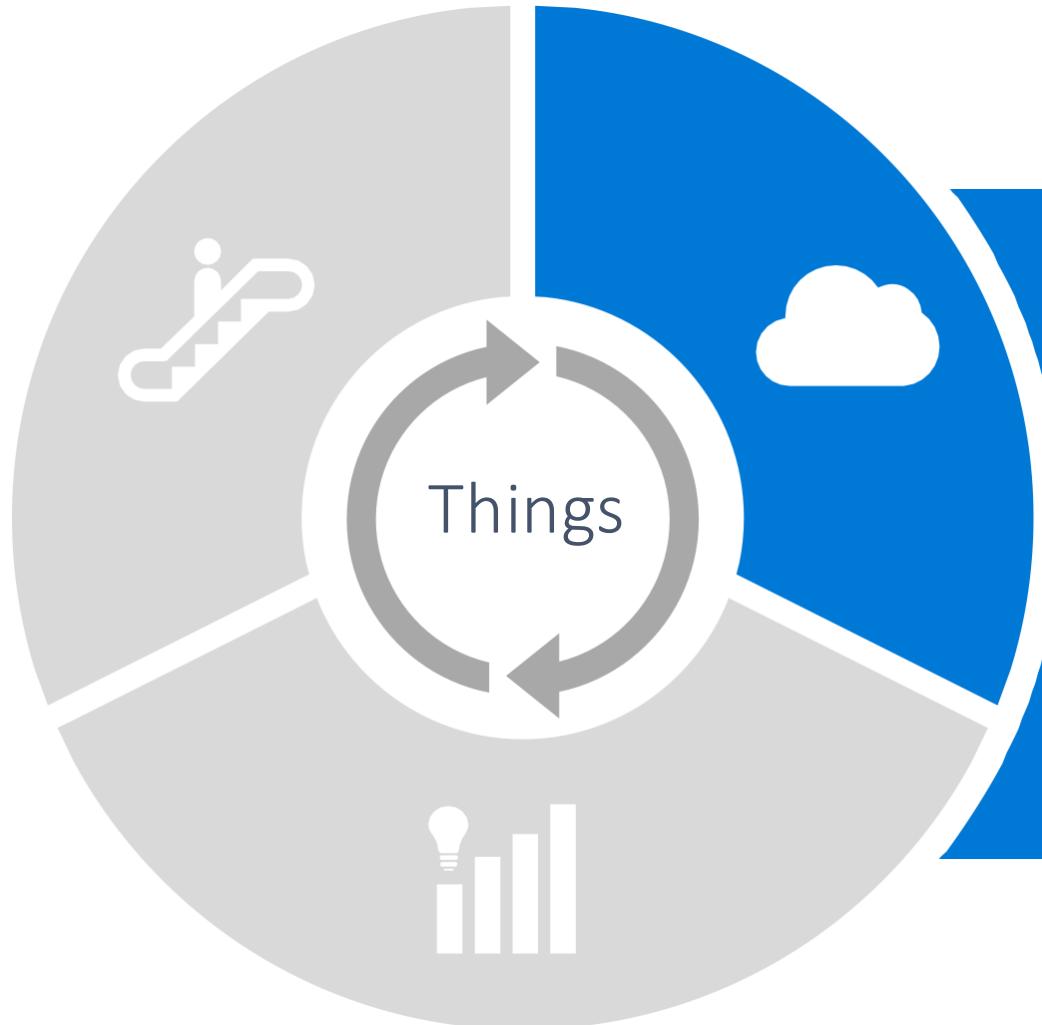
IoT Business Maturity Model

1. Operational efficiency



IoT Business Maturity Model

2. Business Intelligence



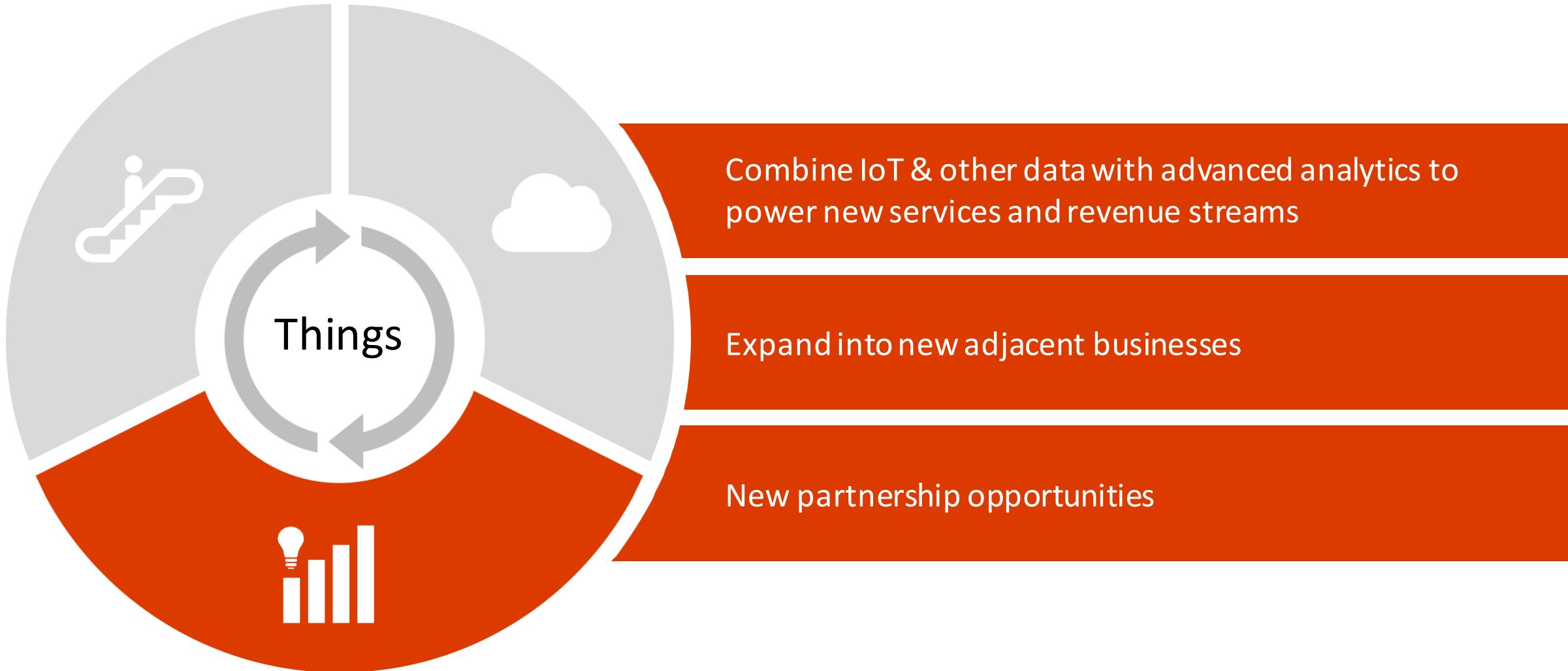
Analyze and visualize data from devices

Discover patterns in device data using predictive analytics

Operationalize insights to improve things and processes in realtime

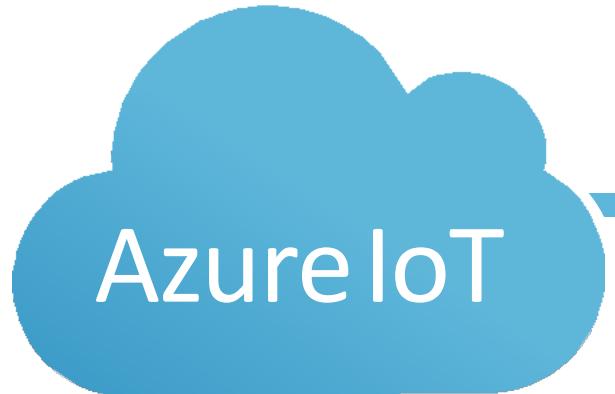
IoT Business Maturity Model

3. Business Transformation



IoT Device and Cloud Patterns

Cloud Based IoT Solutions



Easy to provision, use and manage

Pay as you go, scale as you need

Global reach, hyper scale

End to end security & privacy

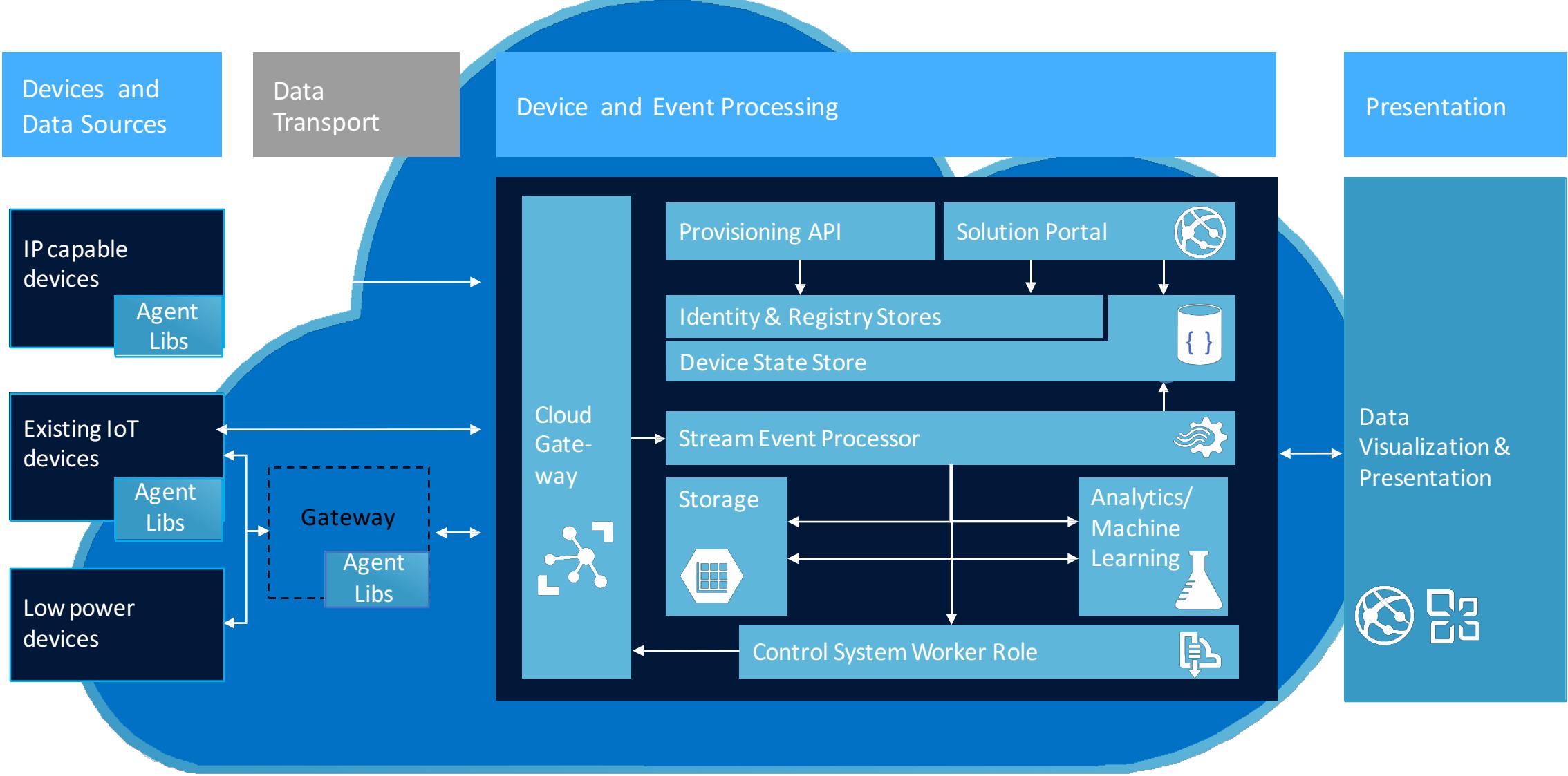
Three parts of an IoT solution

1 Device connectivity & management

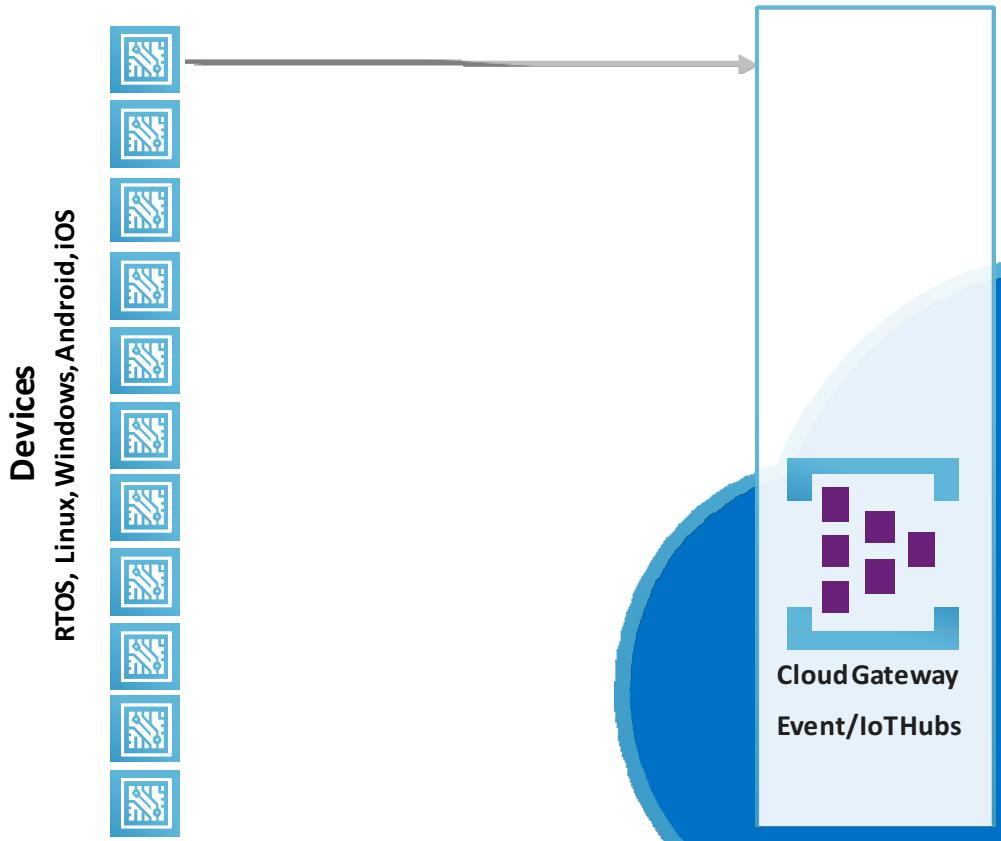
2 Analytics & operationalized insights

3 Presentation & business connectivity

Azure IoT Reference Architecture



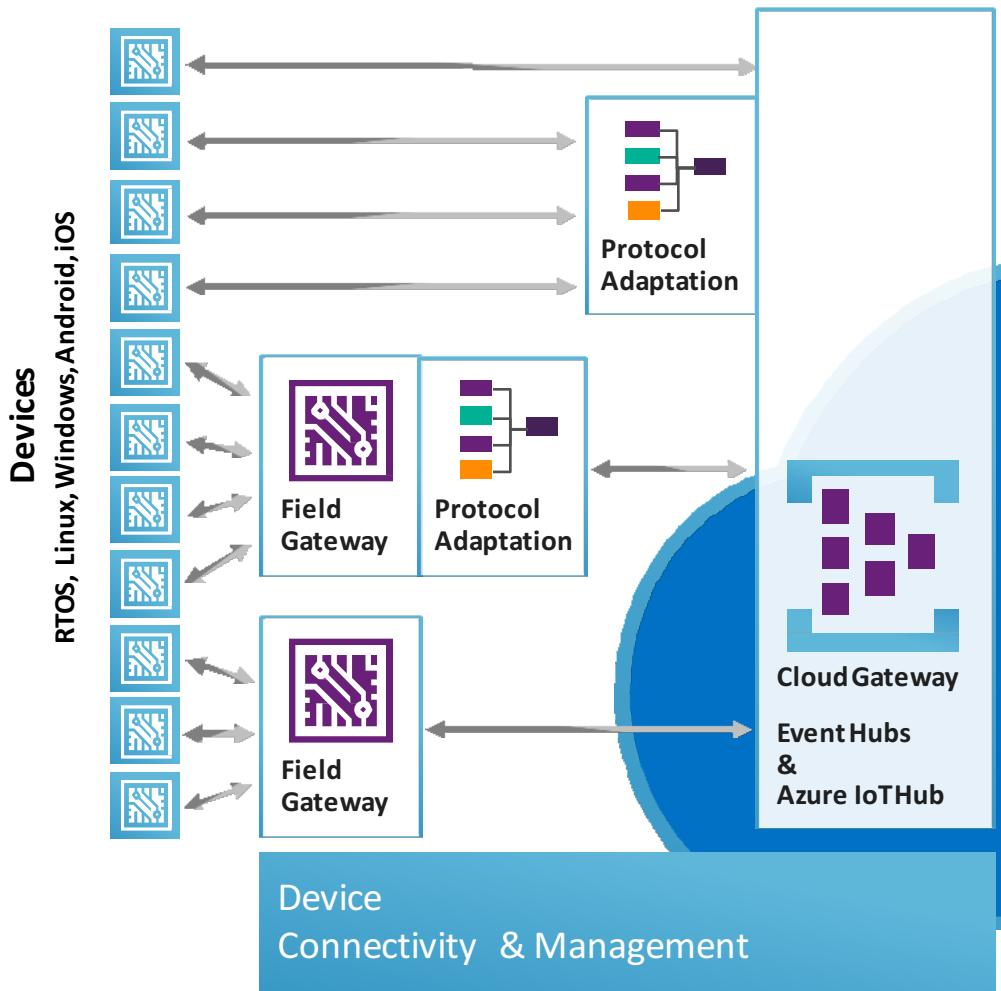
IoT Device & Cloud Patterns



Event Hubs

- High scale telemetry ingestion service
- HTTP/AMQP protocol support
- Each Event Hub supports
 - 1 million publishers
 - 1GB/s ingress
- Generally available worldwide
 - Tens of Billions of messages per day
 - Tens of TB ingested per day
 - And rising...

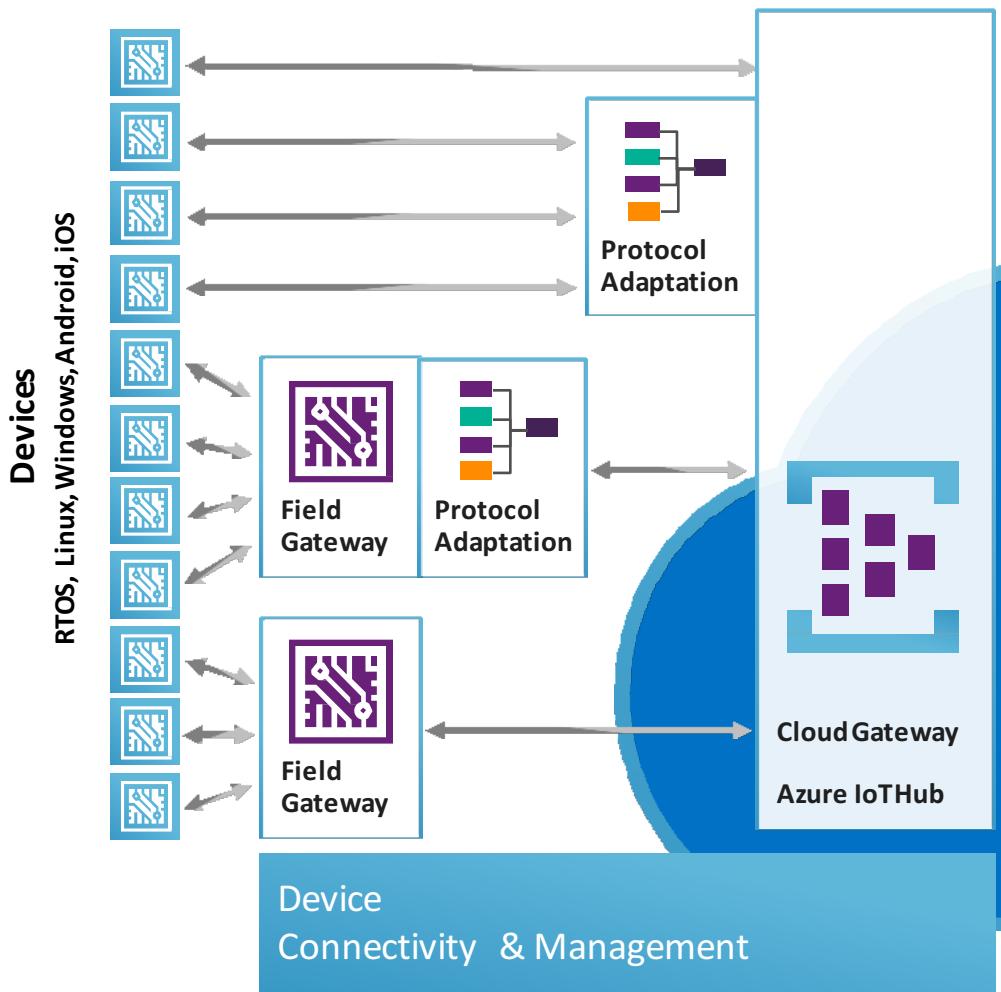
IoT Device & Cloud Patterns



Additional IoT Needs

- Command & control
- Device identity
- Device registry
- Device and State management
- Protocol translation and gateways

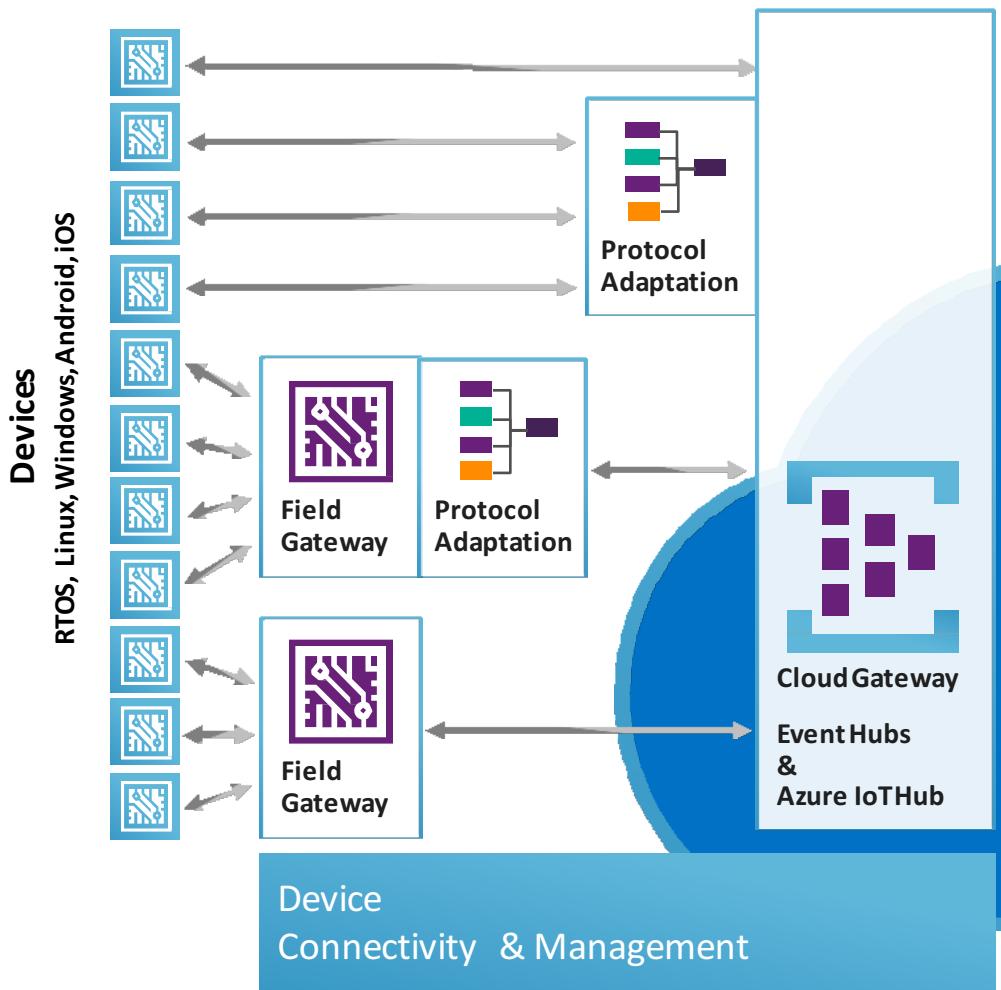
IoT Device & Cloud Patterns



Azure IoT Hub

- Bi-directional device <-> cloud
- Up to 10 million devices
- Telemetry ingestion
- Command & control
- Device registry & identity
- Bulk import/export of device identities
- Device Management
- HTTP/AMQP/AMQP-WS/MQTT
- Extensible protocol support
- Operations Monitoring

IoT Device & Cloud Patterns

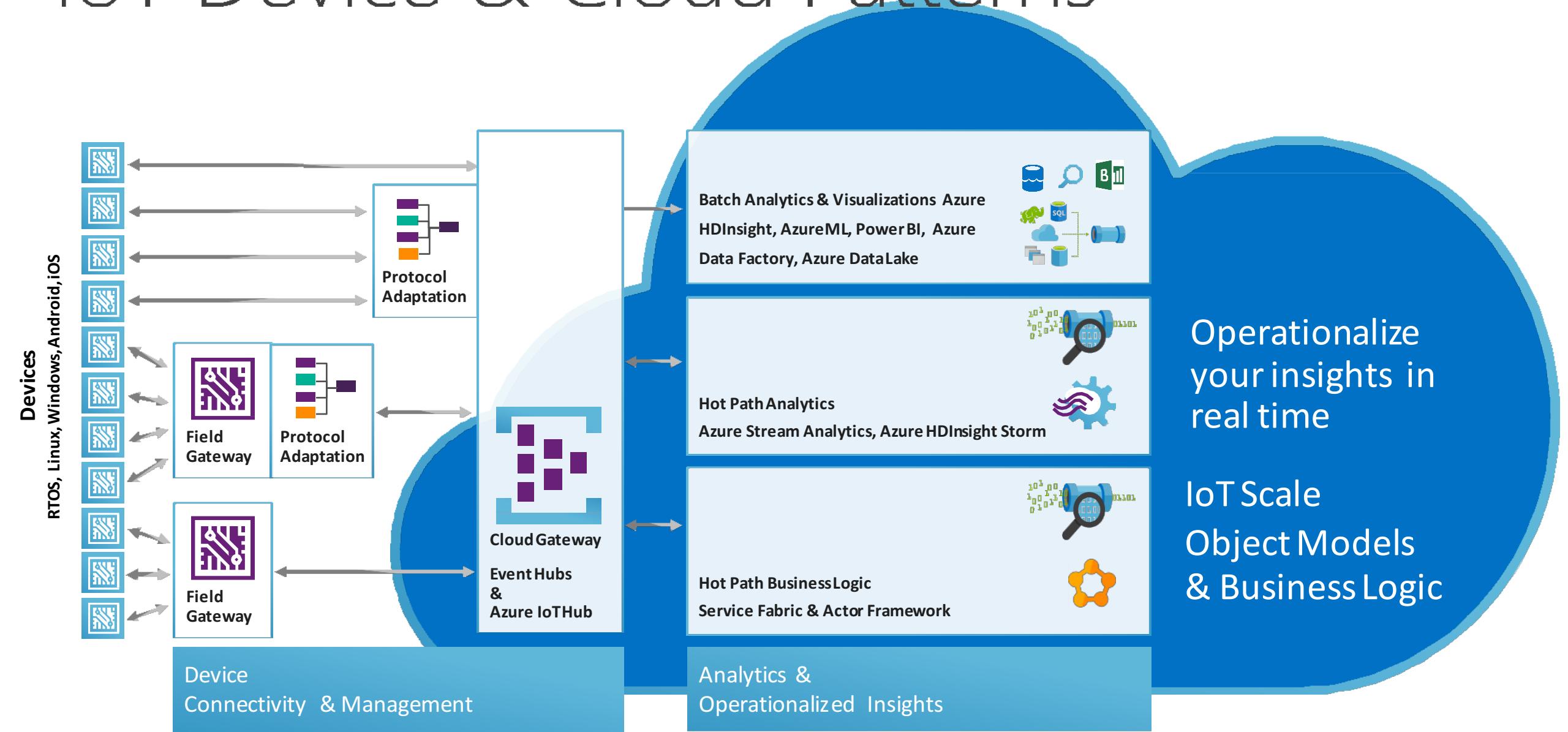


Cross-Platform Device Support

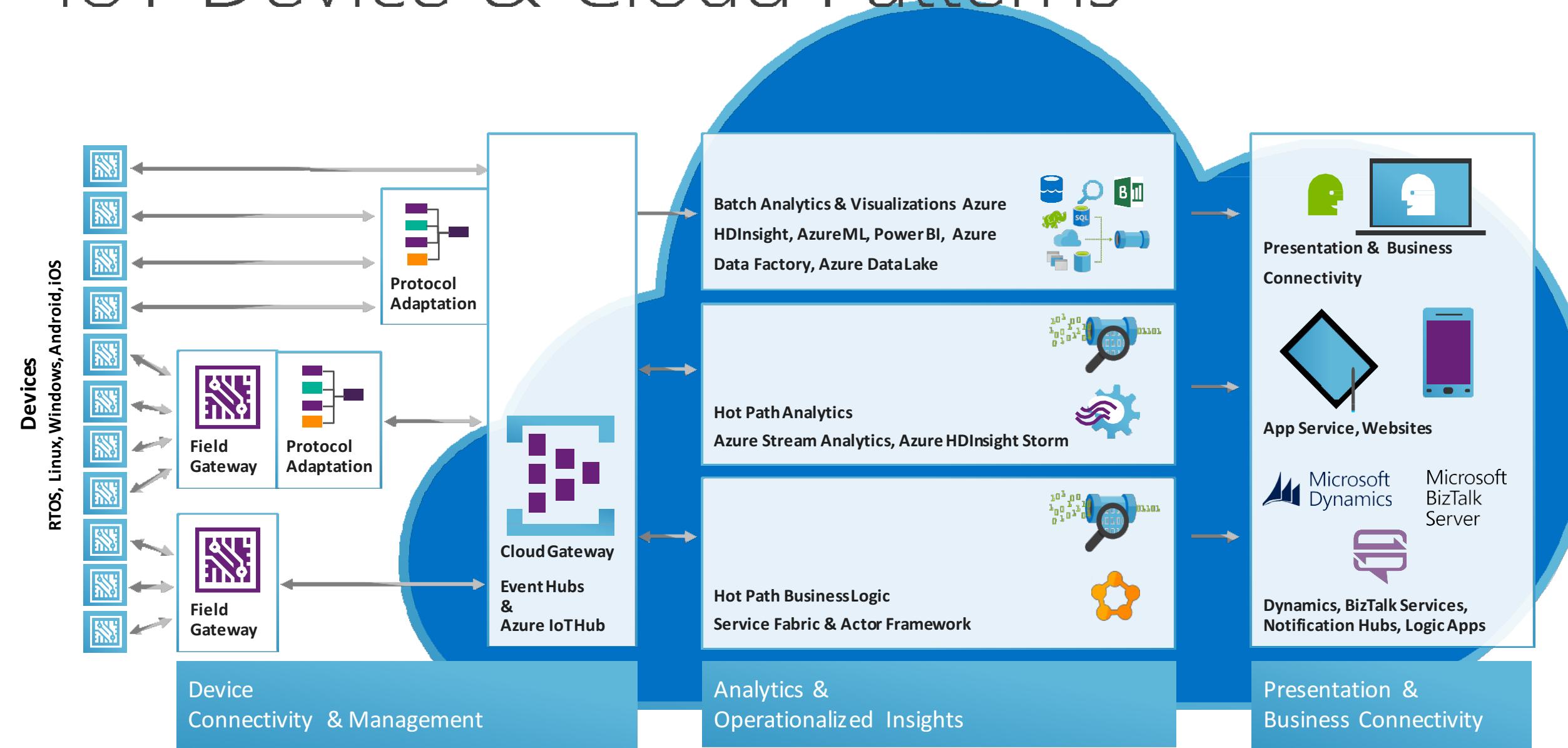
- Open source “agent” framework
- Simple, secure device <-> cloud connectivity & management
- RTOS, Linux, Windows, Android, iOS
- Easy to use, not required



IoT Device & Cloud Patterns



IoT Device & Cloud Patterns



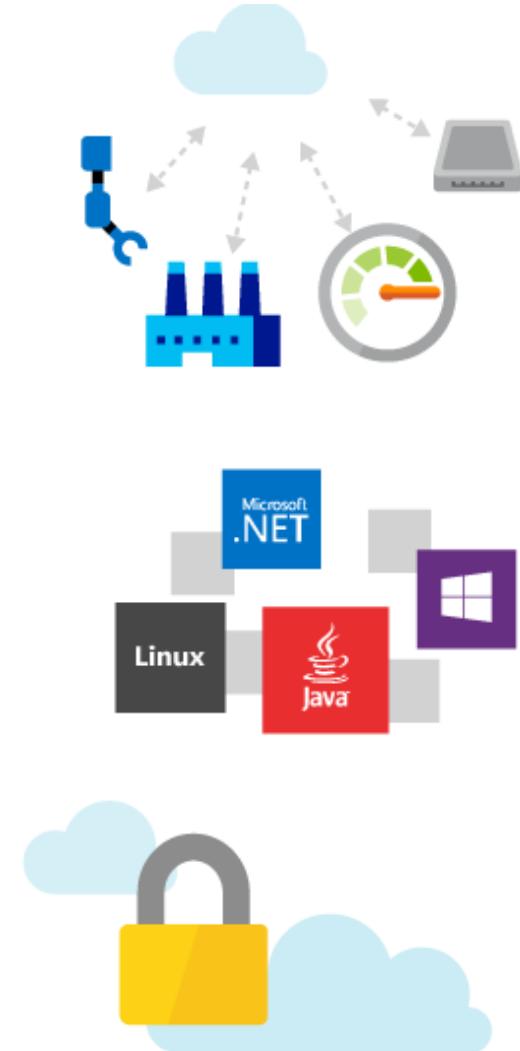
Microsoft Azure IoT services

Devices	Device Connectivity	Storage	Analytics	Presentation & Action
	Event Hubs	SQL Database	Machine Learning	App Service
	IoT Hubs	Table/Blob Storage	Stream Analytics	Power BI
	Service Bus	DocumentDB	HDInsight	Notification Hubs
	External Data Sources	External Data Sources	Data Factory	Mobile Services
				BizTalk Services

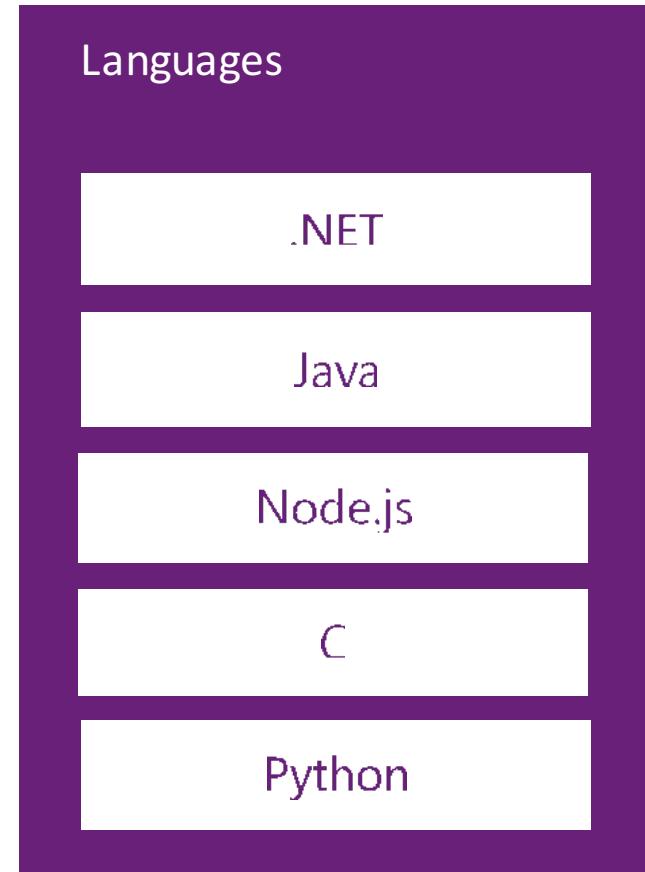
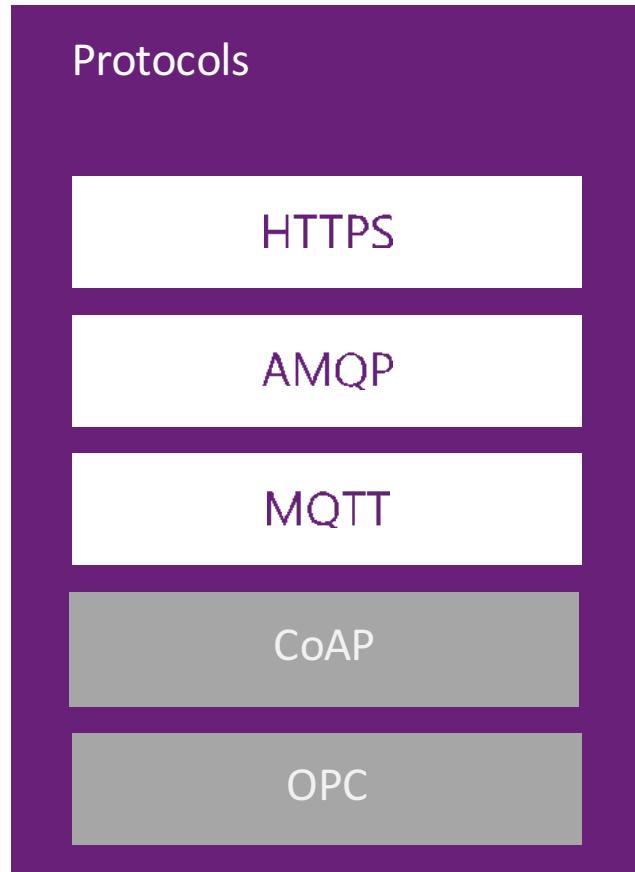
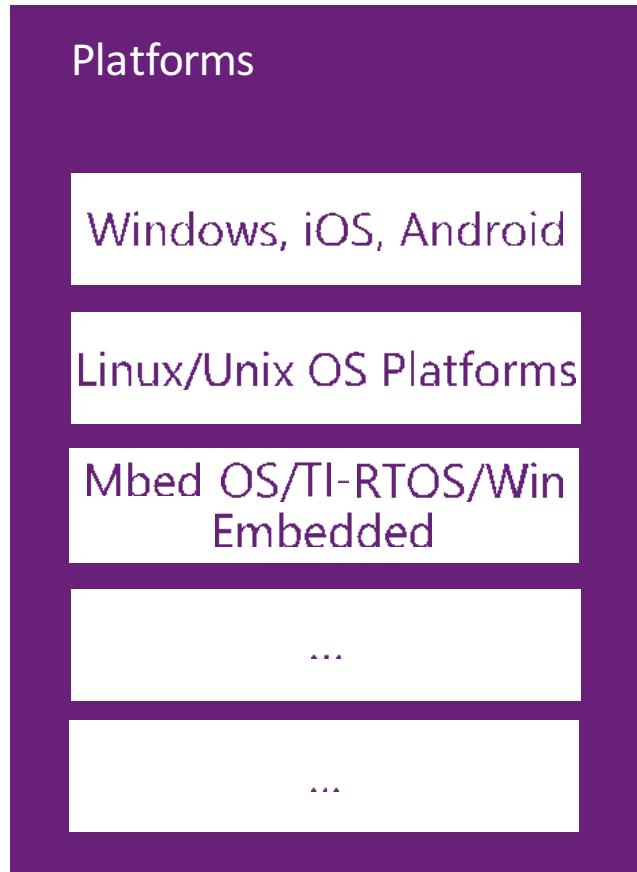
Azure IoT Hub

IoT Hub at a glance

- Establish bi-directional communication with millions of IoT devices
- Work with platforms and protocols that you know
- Authenticate per device for security-enhanced IoT solutions
- Based on Azure Event Hubs technology; delivering tens of billions of messages per day

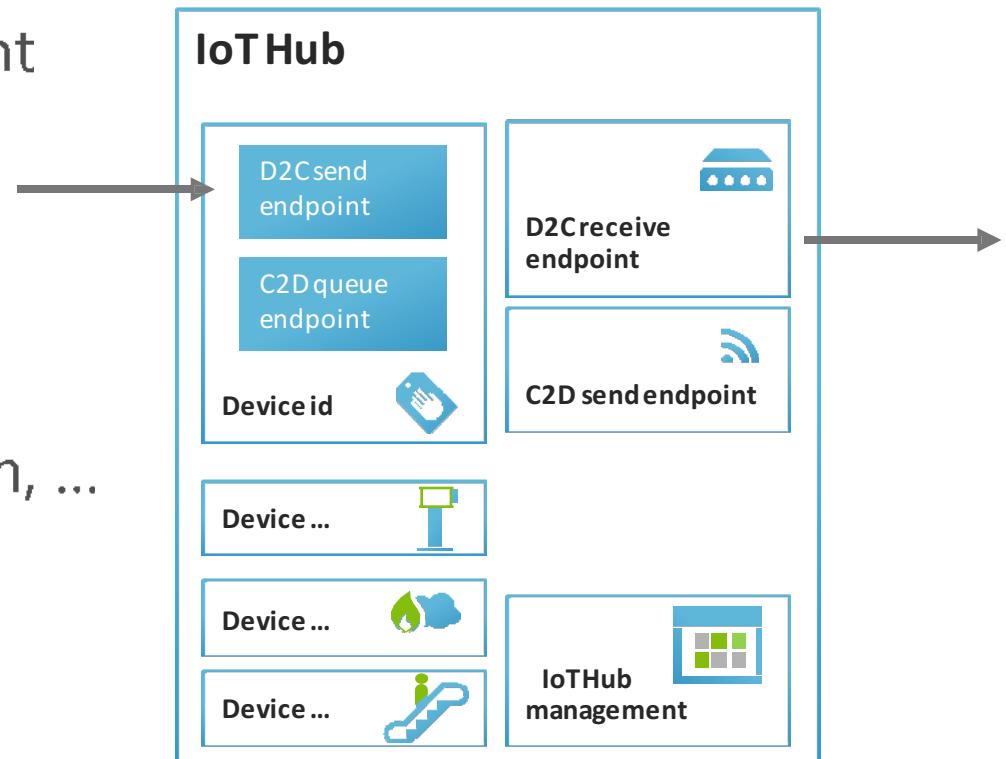


What's supported?



D2C – Messages

- **Interface**
 - AMQP / MQTT and HTTPS device-side endpoint
 - AMQP service-side endpoint
 - Device and service SDKs
- **Compatible with Event Hubs**
 - Partitioned receiver, client check-pointing
 - Integrations with Azure Stream Analytics, Storm, ...
 - 100% compatible with Event Hubs receivers
- **IoT Hub services for D2C**
 - Millions of simultaneously connected devices
 - Per-device authentication
 - Connection-multiplexing



D2C – Consuming Events

Code against SB msg API

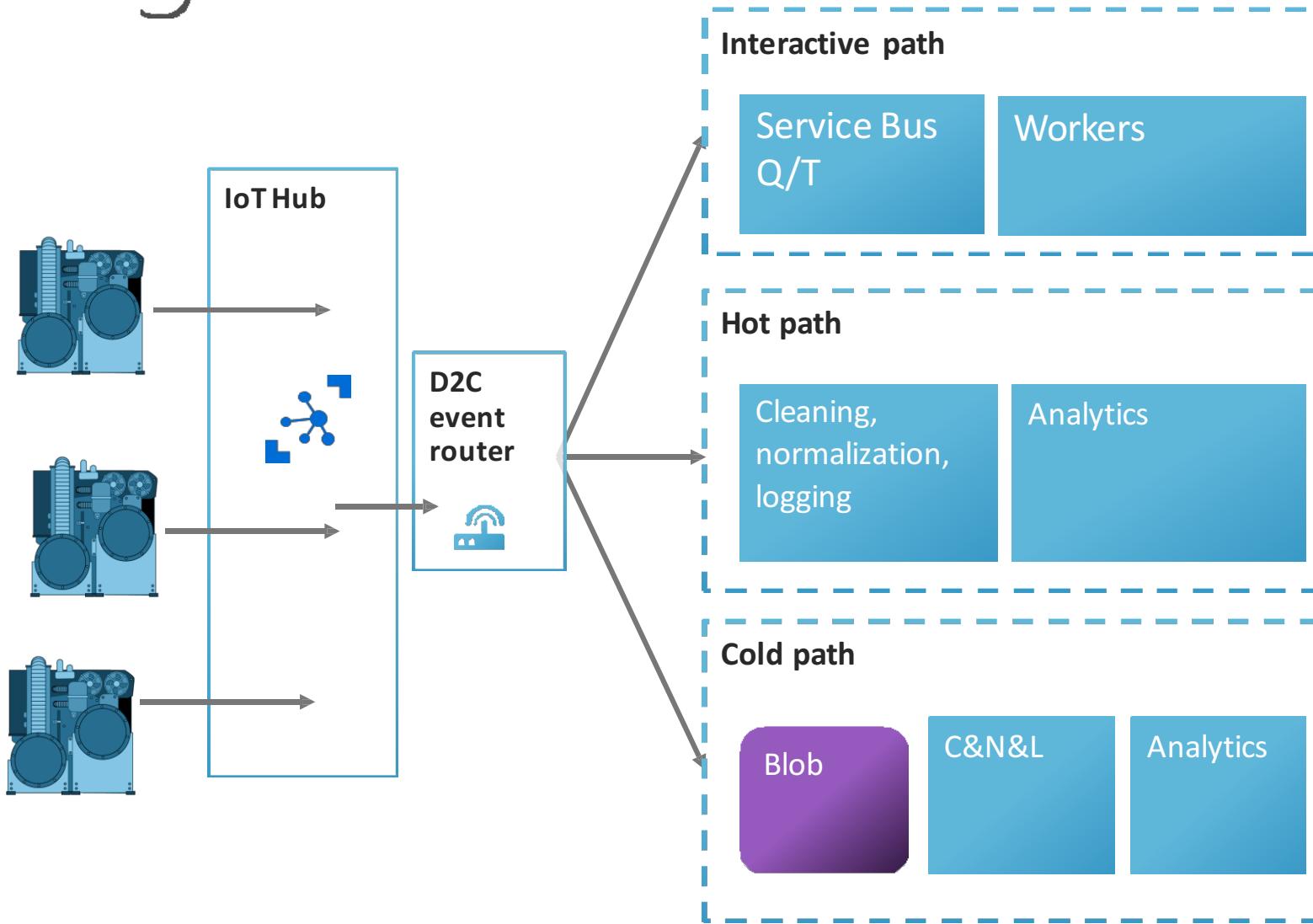
- Easy event processor implementation
- Enable deduplication on Q
- Easy routing to different subcomponents
- Create SB Q/T based on expected throughput

Use any event processor

- Easy to select only “real-time” data.
- Storm / Spark
- Custom code
- Azure Stream Analytics

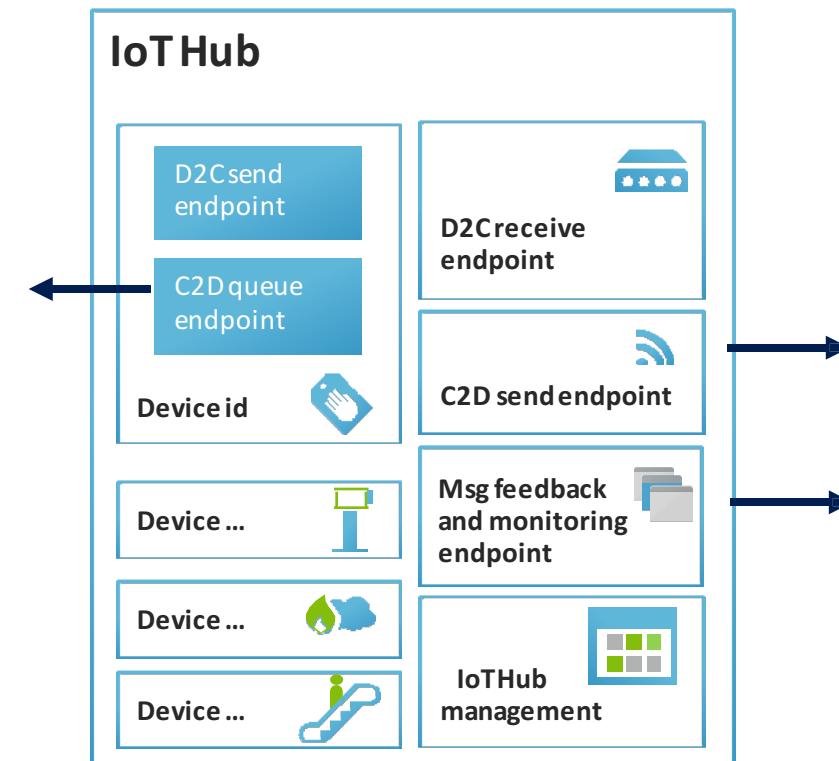
Run batches from blob

- Reuse existing batch code
- Easy with Azure Data Factory



C2D - Messages

- **Interface**
 - AMQP, MQTT, HTTPS device-side endpoint
 - AMQP service-side endpoint
- **At-least-once semantics**
 - Durable queued messages (TTL up to 2 days)
 - Device acknowledges receipt
(Send - Receive - Abandon OR Complete)
- **TTL and receipts**
 - Per-message TTL
 - Per-message positive and negative receipts
- **Command lifecycle pattern**
 - Use correlated D2C for responses
 - Use feedback information to retry
 - Store command state in command registry



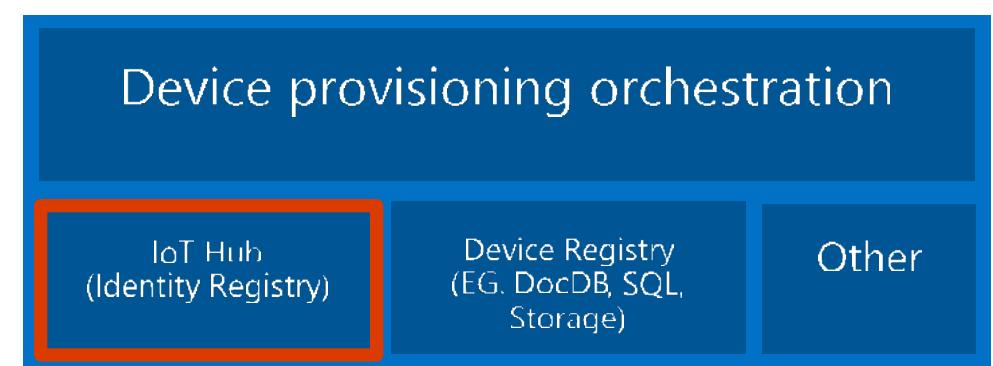
Management - Provisioning

- **Identity Registry**

- Each IoT hub has a device identity registry that you can use to create per-device resources in the service
- Does not contain any application metadata
- Can be accessed like a dictionary using the **deviceId** as the key
- Does not support expressive queries

- **Operations:**

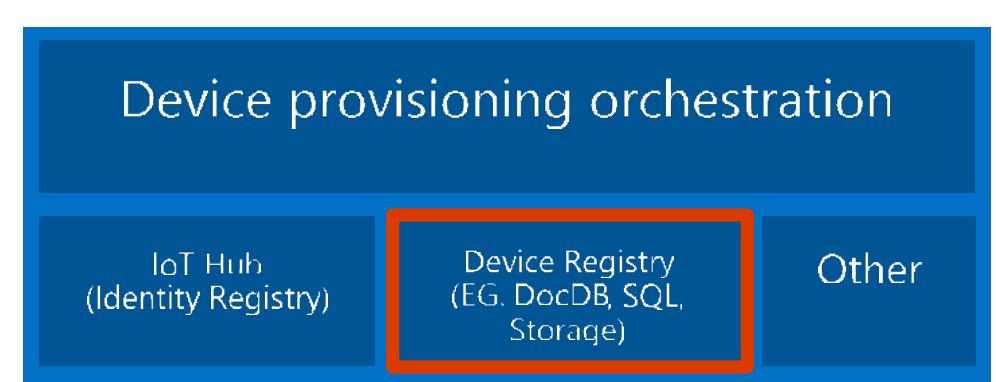
- *Create device identity*
- *Update device identity*
- *Retrieve device identity by ID*
- *Delete device identity*
- *List up to 1000 identities*



Management - Provisioning

- **Device Registry**

- Provides the ability to add application-specific metadata to your devices
- Use a per-device authentication model
- The device registry serves as a consistent view of device data
- Typically stored in DocDB, SQL DB, or similar



Management - Provisioning

- **Disabling devices**
 - You can disable devices by updating the **status** property of an identity in the registry
 - You would typically do this in two scenarios:
 - *For any reason, you consider a device is compromised or has become unauthorized*
 - *During a provisioning orchestration process or decommissioning process*
- **Batch Operations**
 - Imports / Exports at scale are long-running jobs that use a customer-supplied blob container to read and write device identity data
 - You are able to export device identities in bulk from the IoT hubs registry

Management - Security (Transport)

- **Transport Security & Authentication**
 - Various AuthN options including certificates and pre-shared keys
 - The connection established between devices and IoT Hub is TLS (Transport Layer Security) based
 - Communication is encrypted
 - Server is authenticated using its own X.509 certificate sent to the device during hand shaking
 - Security keys are never sent over the wire - always token/signature based
- **Custom Authentication Options**
 - Multi-factor Authentication
 - Active Directory

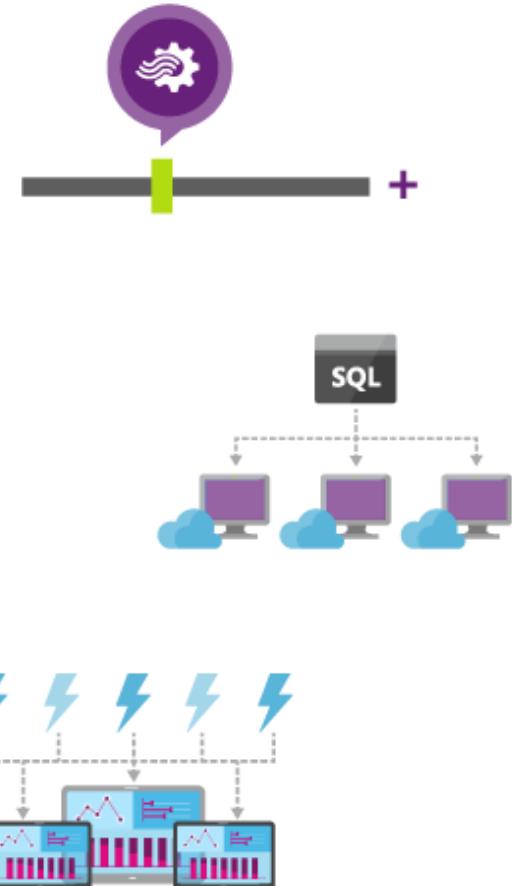
Management – (Access Policies)

- **Service-side Communication**
 - RegistryRead and RegistryReadWrite - Grant permissions to the Identity registry
 - Service connect - Grants access to service-facing communication and monitoring endpoints
- **Device-side communication**
 - Device Connect - Grants access to device-facing communication endpoints
 - Device Identity Registry - Each device identity has a user-specified deviceId with a set of security credentials and status flag to enable/disable connectivity.

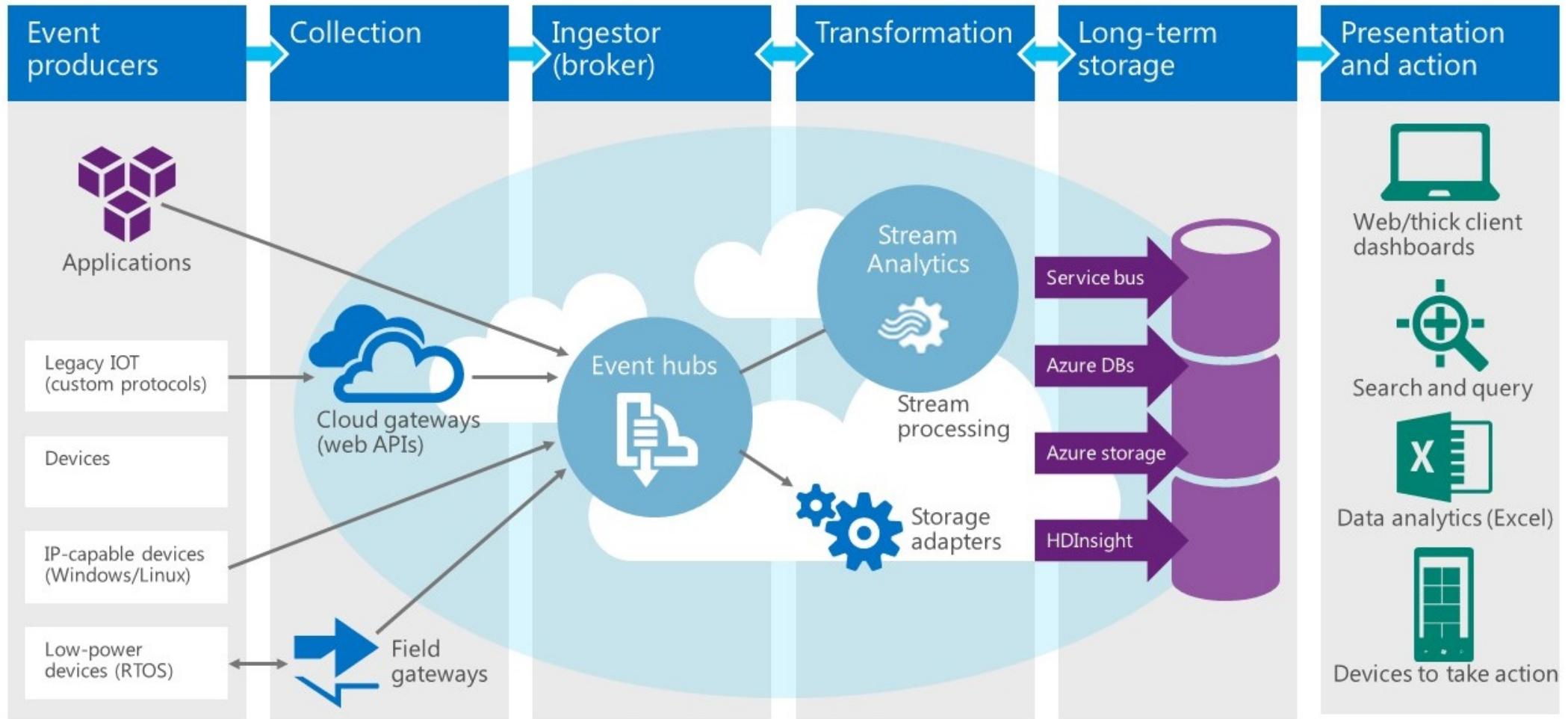
Azure Stream Analytics

ASA – Benefits

- Allows event processing of large volumes of event data at high velocity
- Analysis is done as data arrives
- Declarative ‘SQL like’ query processing support
- High throughput, low latency processing
- Real-time stream computation options
- Enable rapid development



Stream Analytics at Work



ASA - Inputs

Data stream inputs

- **Event Hub:** Feed events into Azure in real time, and Stream Analytics jobs can process them
- **IoT Hub:** Stream information collected from connected devices directly into a streaming analytics job
- **Blob Storage:** Can be used as an input source for ingesting bulk data

Reference Data

- Static or slow changing used for performing look-ups
- Azure Blob Storage is the only supported reference input
- Source blobs limited to 50MB in size

Input Data Types

- CSV, JSON and Avro Serialized

ASA - Queries

- Performs transformations and computations over streams of events
- Stream Analytics Query Language
 - *Subset of T-SQL syntax*
 - Multiple built-in functions
 - Supports multiple data types as well as complex
 - 3 types of windowing
 - *Tumbling*
 - *Hopping*
 - *Sliding*
 - Queries can be created via Azure Portal or via API calls

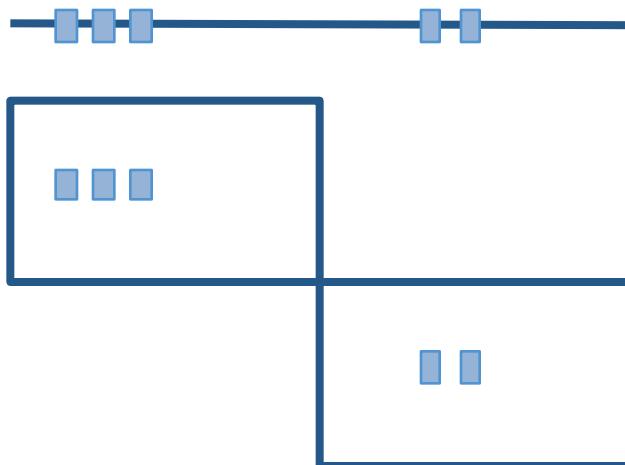


```
1 WITH [StreamData] AS (SELECT * FROM [IoTHubStream1])
2
3 SELECT *
4 INTO [Telemetry]
5 FROM [StreamData]
6
7 SELECT
8     DeviceId,
9     AVG(Humidity) AS [AverageHumidity],
10    MIN(Humidity) AS [MinimumHumidity],
11    MAX(Humidity) AS [MaxHumidity],
12    5.0 AS TimeframeMinutes
13 INTO
14     [TelemetrySummary]
15 FROM
16     [StreamData]
17 WHERE
18     [Humidity] IS NOT NULL
19 GROUP BY
20     DeviceId, SlidingWindow (mi, 5)
```

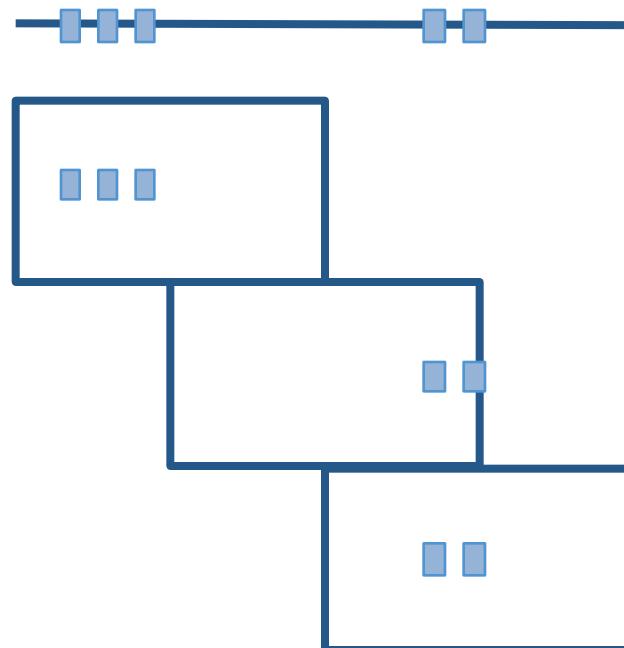
Windowing

Count or aggregate events over a specified time period

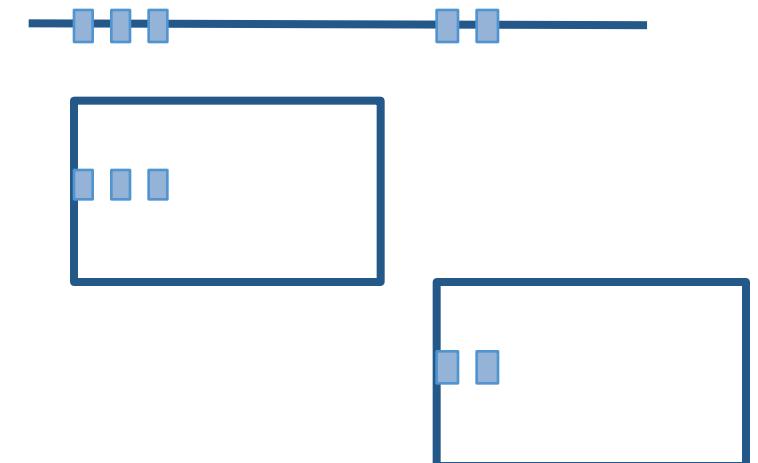
TumblingWindow



HoppingWindow



SlidingWindow



ASA - Outputs

- Multiple options for storing output and viewing analysis results
Route to one or more outputs
- Flexibility in the consumption and storage of the job output for data warehousing and other purposes

Available Outputs:

- SQL Database
- Blob Storage
- Event Hub
- Power BI
- Table Storage
- Service Bus Queues and Topics
- Document DB

Development SDKs

SDK

- SDK and agent libraries easily accessible in GitHub
- Cross platform support – choose OS, platform and language
- Device support with IP and access control capabilities
- Connect IP and non-IP devices via gateway and field protocols
- Open source framework to accommodate development of custom agents for devices
- Simple and secure D2C and C2D connectivity for messaging, device management and command and control
- Multiple OS support - RTOS, Linux, Windows, Android, iOS etc

SDKs - Device

- **Operating Systems Supported**

- Debian Linux (v 7.5)
- Fedora Linux (v 20)
- mbed OS (v 2.0)
- Raspbian Linux (v 3.18)
- Ubuntu Linux (v 14.04)
- Windows Desktop (7, 8, 10)
- Windows IoT Core (v 10)
- Windows Server (v 2012 R2)
- Yocto Linux (v 2.1)

- **C Libraries Supported**

- Debian Linux (v 7.5) HTTPS, AMQP, MQTT
- Fedora Linux (v 20) HTTPS, AMQP, MQTT
- mbed OS (v 2.0) HTTPS, AMQP
- Ubuntu Linux (v 14.04) HTTPS, AMQP, MQTT
- Windows Desktop (7,8,10) HTTPS, AMQP, MQTT
- Yocto Linux (v 2.1) HTTPS, AMQP

SDKs – Device (Registry Manager)

- **Name Space: Microsoft.Azure.Devices**
- **RegistryManager**
- RegistryManager.CreateFromConnectionString()
- RegistryManager.OpenAsync()
- RegistryManager.AddDeviceAsync()
- RegistryManager.GetDevicesAsync()
- RegistryManager.RemoveDeviceAsync()

SDKs – Device Client

- **Name Space:** `Microsoft.Azure.Devices.Client`
- **Key Object:** `DeviceClient`
- `DeviceClient.CreateFromConnectionString()`
- `DeviceClient.OpenAsync()`
- `DeviceClient.SendEventAsync()`
- `DeviceClient.ReceiveAsync()`
- `DeviceClient.UploadToBlob()`

SDKs – Service Client

- **Name Space:** `Microsoft.Azure.Devices`
- **Key Object:** `ServiceClient`
- `ServiceClient.CreateFromConnectionString()`
- `ServiceClient.OpenAsync()`
- `ServiceClient.SendCloudToDeviceMessageAsync()`
- `ServiceClient.ReceiveFeedbackAsync()`

Tools and Demo

DEMO:

Azure IoT Hub – Ingest

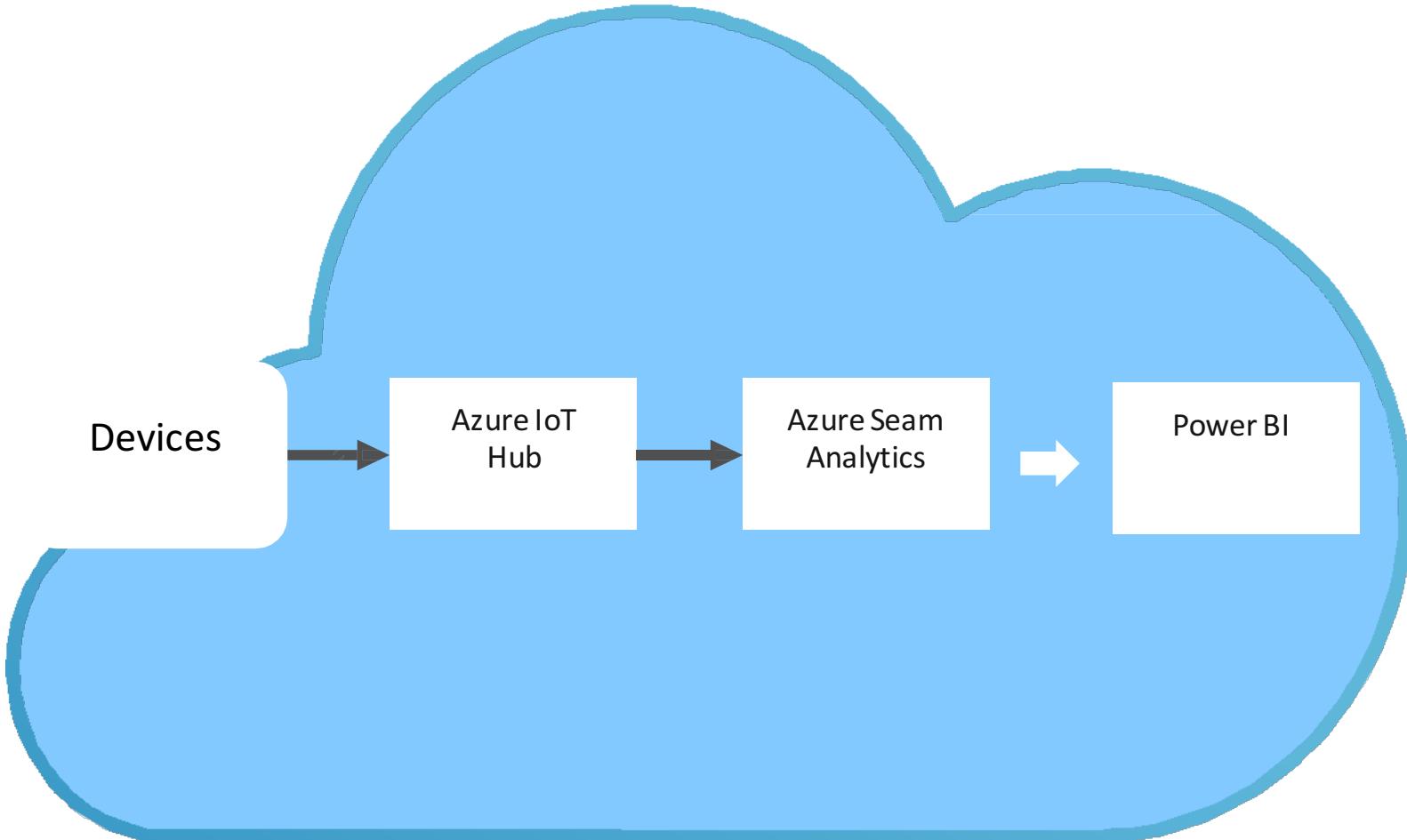
Setup Devices

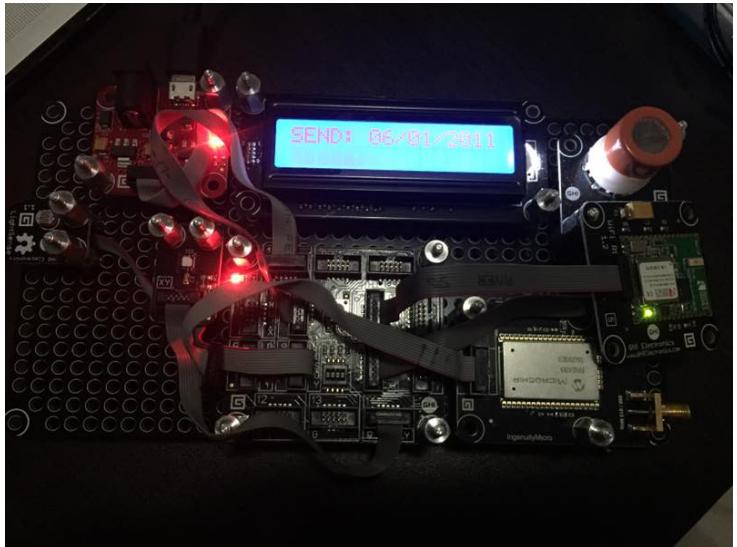
Stream Analytics - Query

Power BI - Visualize



Demo Architecture

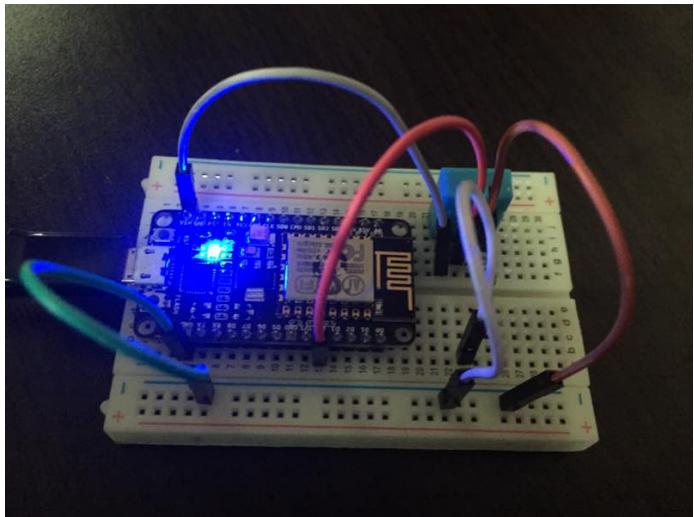




G120 - .NET Microframework / C#



Tessel- Open WRT / Node.js



NodeMcu – Arduino C



Raspi 3 – Win IoT Core / C#

THANK YOU
See you in another chapter...

Please visit our page:
<https://www.facebook.com/netgadgeteerindonesia>
<https://www.facebook.com/buitenzorgmakersclub>
And share your ideas and awesome ‘things’ to us