2018 Global Azure BOOTCAMP

ARGOMENTO

Sviluppare un portale per gestire la tua soluzione IoT Hub

Marco Parenzan

Eccomi!

Marco Parenzan

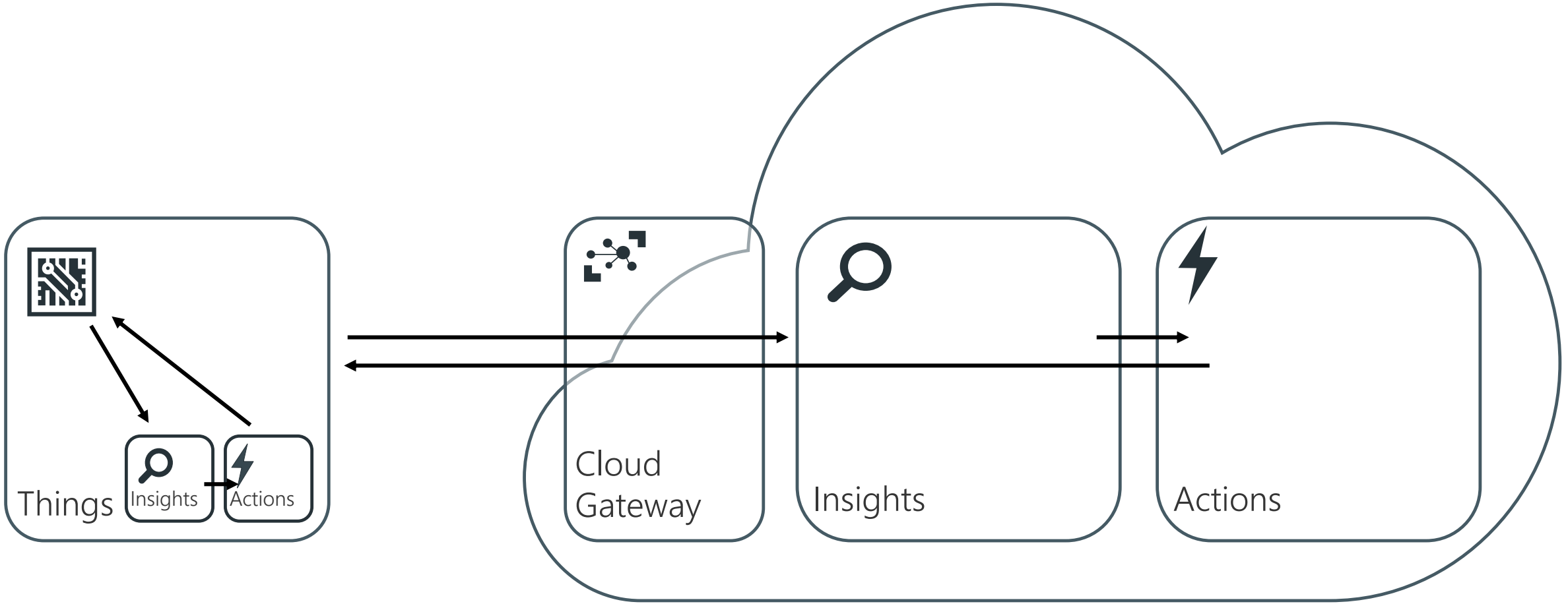
@marco_parenzan

marcoparenzan

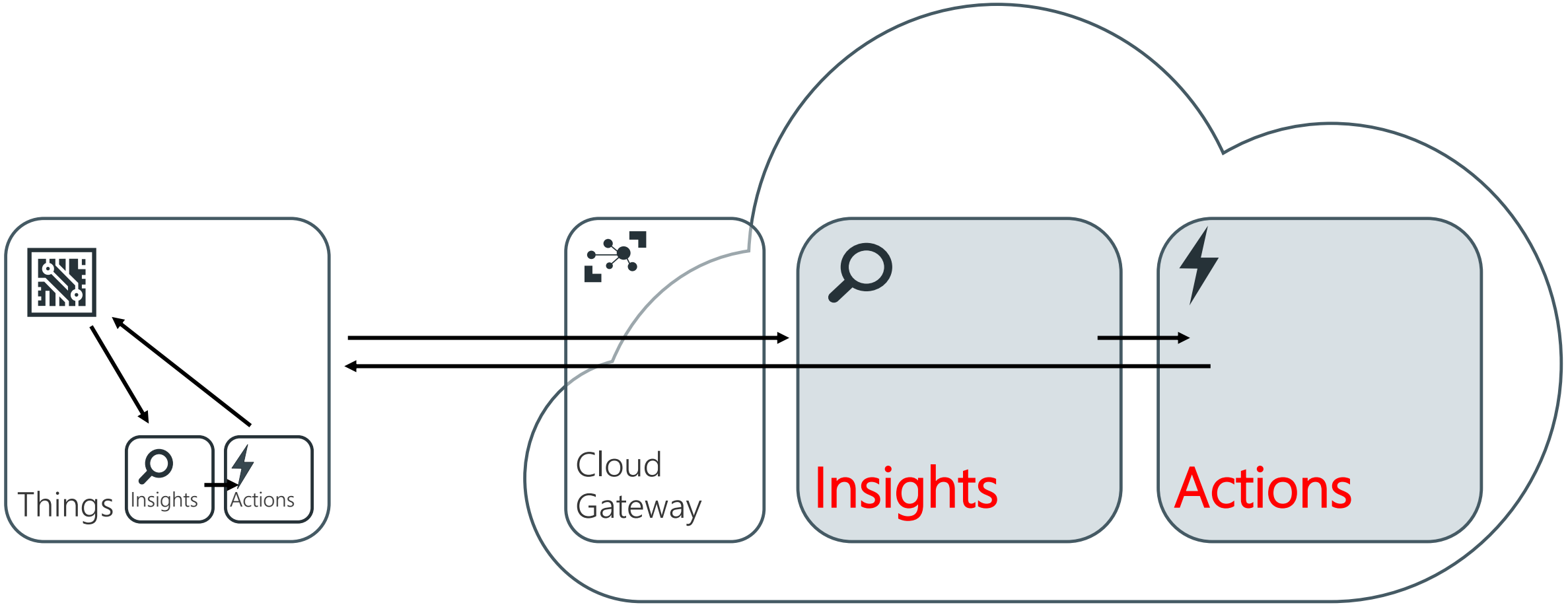
marcoparenzan

AZURE
FOR THE INTERNET OF THINGS

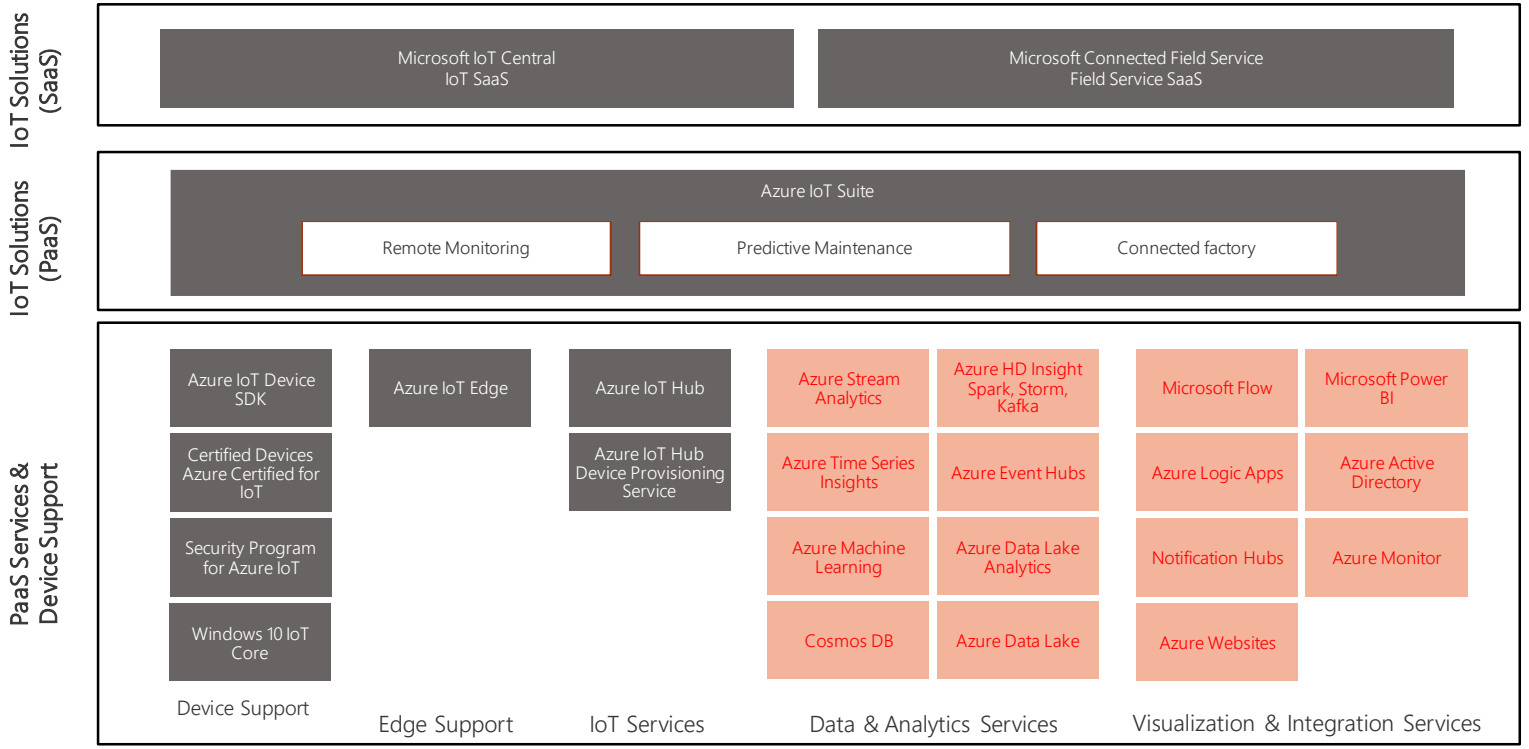
IoT Pattern + Edge



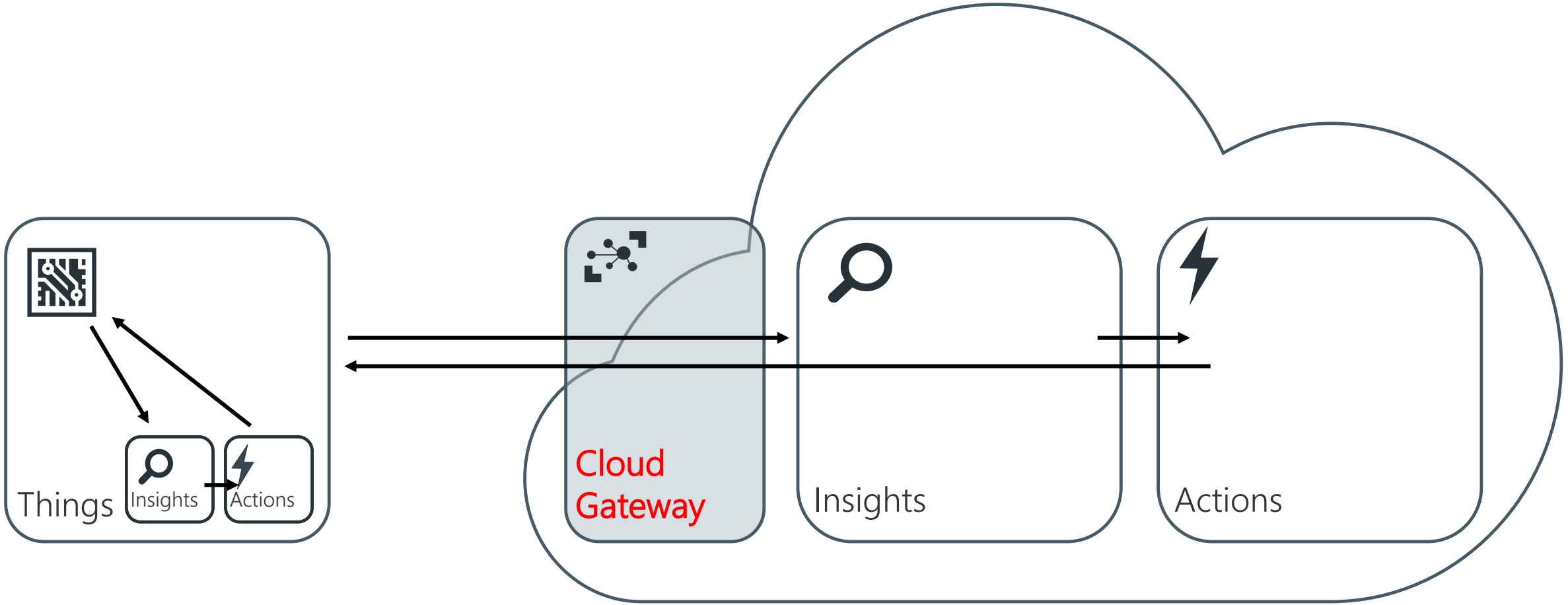
Azure “Insights & Actions”



Comprehensive set of offerings for IoT



The cloud gateway



Azure IoT Hub

Bi-directional communication

Bi-directional communication

- Millions of Devices
- Multi-language, open source SDKs
- HTTPS/AMQPS/MQTTS
- Send Telemetry
- Receive Commands
- Device Management
- Device Twins
- Queries & Jobs

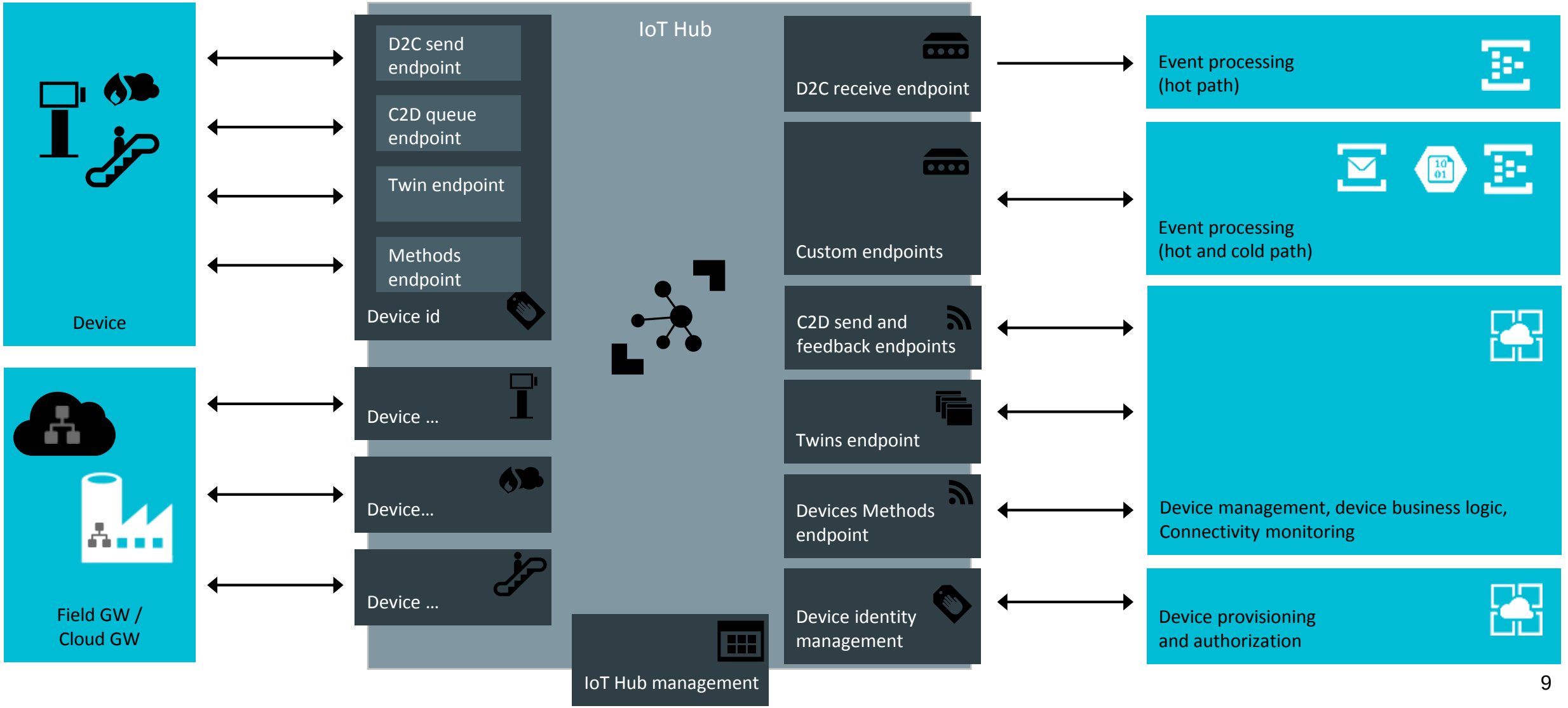
Enterprise scale & integration

- 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 Polices
- Firmware/Software Updates

IoT Hub Messaging



Message

Opaque body

Application Properties

System Properties

Receiving a message as a service

guarantees reliability and durability handling messages.

handles intermittent connectivity on the device side.

at least once

1day (default) to 7 days

Custom Processor, Stream Analytics, Azure Func

Limitations and Guidance

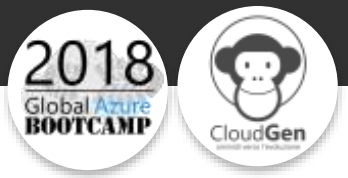
Transient storage

50 messages

48 hours

64Kb

IoT Hub messaging pricing and scaling



Service instance made of units

Device messages (limit 256Kb) are billed in 4Kb chunks (0,5Kb for Free tier)

Twins messages (limit 8Kb) are billed in 0,5Kb chunks

F1 Free	B1 Basic	B2 Basic	B3 Basic	S1 Standard	S2 Standard	S3 Standard
8k messages/unit/day	400k messages/unit/day	6M messages/unit/day	300M messages/unit/day	400k messages/unit/day	6M messages/unit/day	300M messages/unit/day
Device-to-cloud telemetry	Device-to-cloud telemetry	Device-to-cloud telemetry	Device-to-cloud telemetry	Device-to-cloud telemetry	Device-to-cloud telemetry	Device-to-cloud telemetry
Message routing	Message routing	Message routing	Message routing	Message routing	Message routing	Message routing
Cloud-to-device commands	Upgradable to standard	Upgradable to standard	Upgradable to standard	Cloud-to-device commands	Cloud-to-device commands	Cloud-to-device commands
IoT Edge				IoT Edge	IoT Edge	IoT Edge
Device management				Device management	Device management	Device management
Unable to display pricing	8,43 EUR PER IOT HUB UNIT	42,17 EUR PER IOT HUB UNIT	421,65 EUR PER IOT HUB UNIT	21,08 EUR PER IOT HUB UNIT	210,83 EUR PER IOT HUB UNIT	2.108,25 EUR PER IOT HUB UNIT

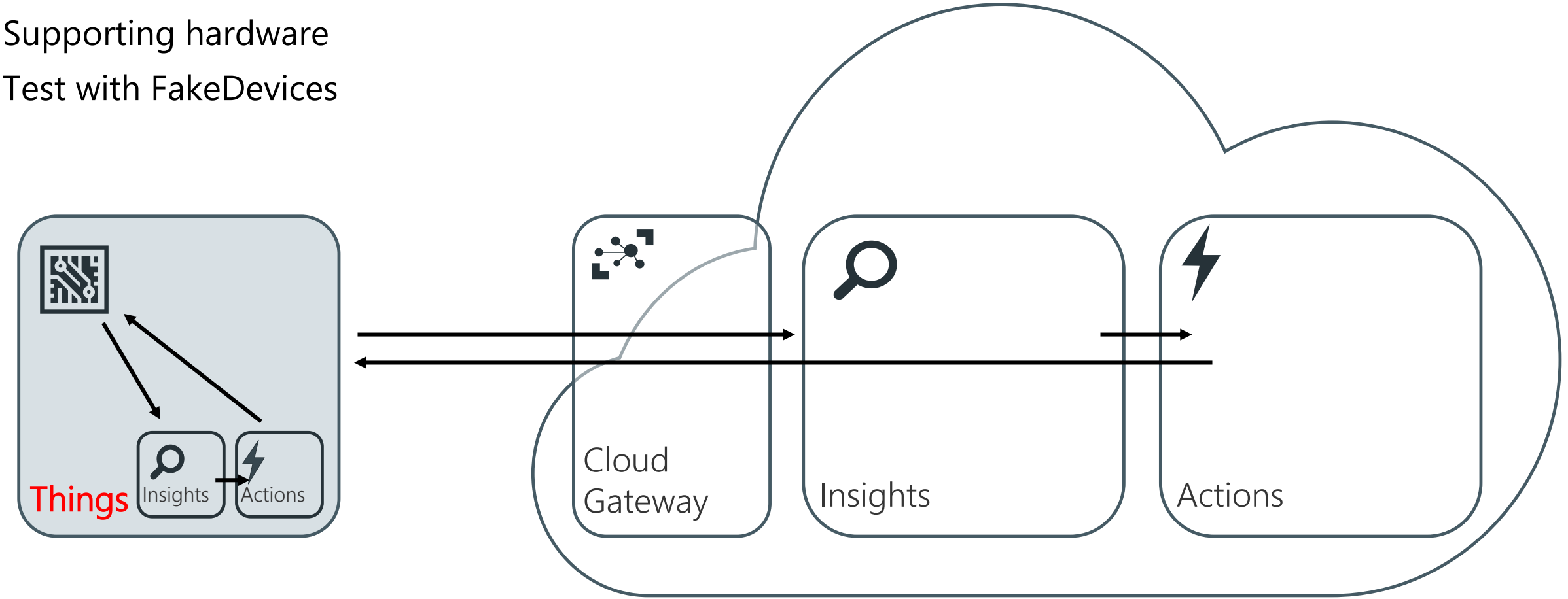
Developing things

Comprehensive set of SDK

Supporting languages

Supporting hardware

Test with FakeDevices

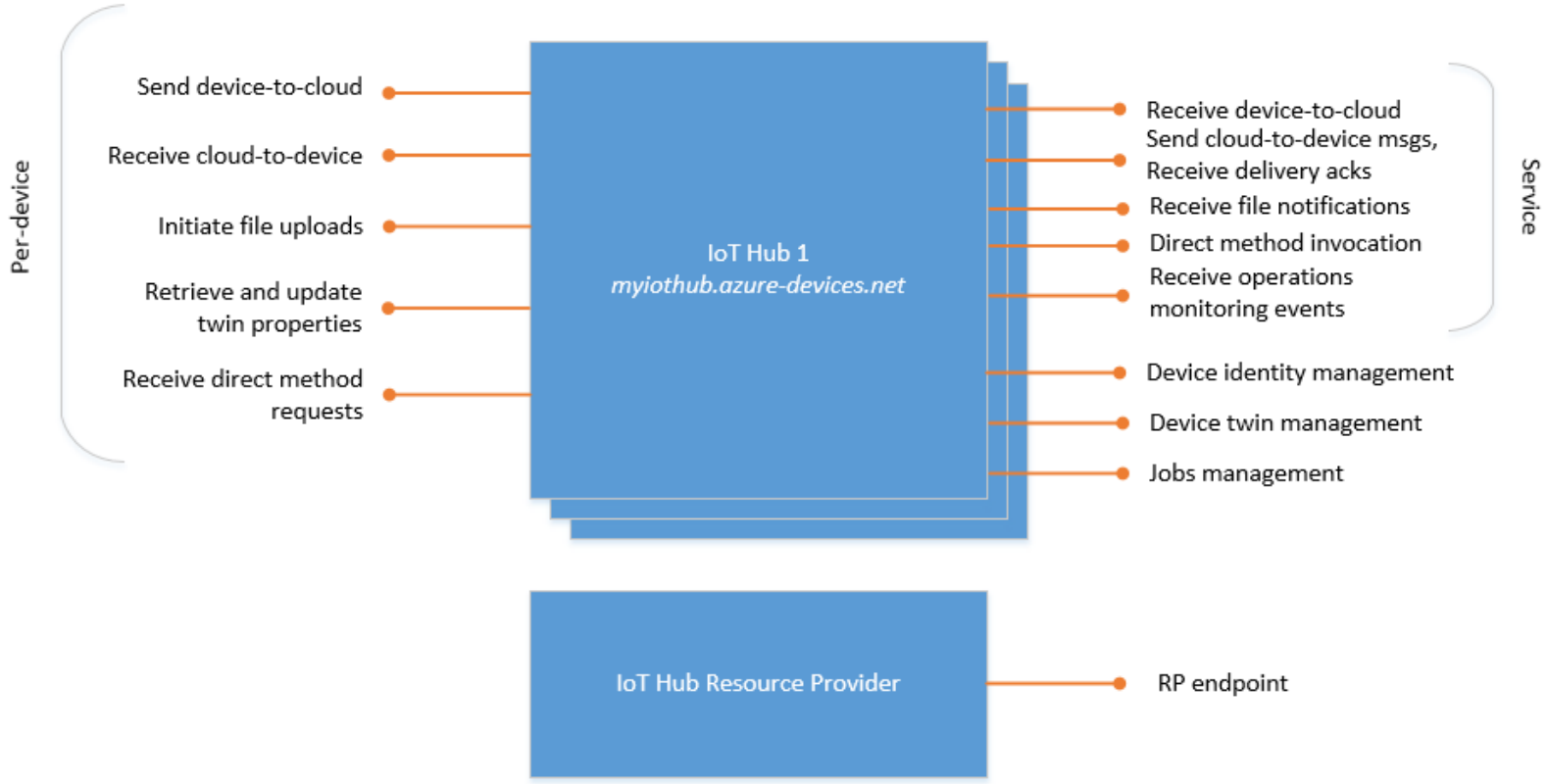


Device has a «functional» lifetime

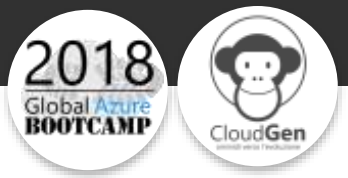
Device has a «non functional» lifetime

Managing Device Indentities with Azure IoT Hub

IoT Hub Endpoints



What is a Shared Access Policy?



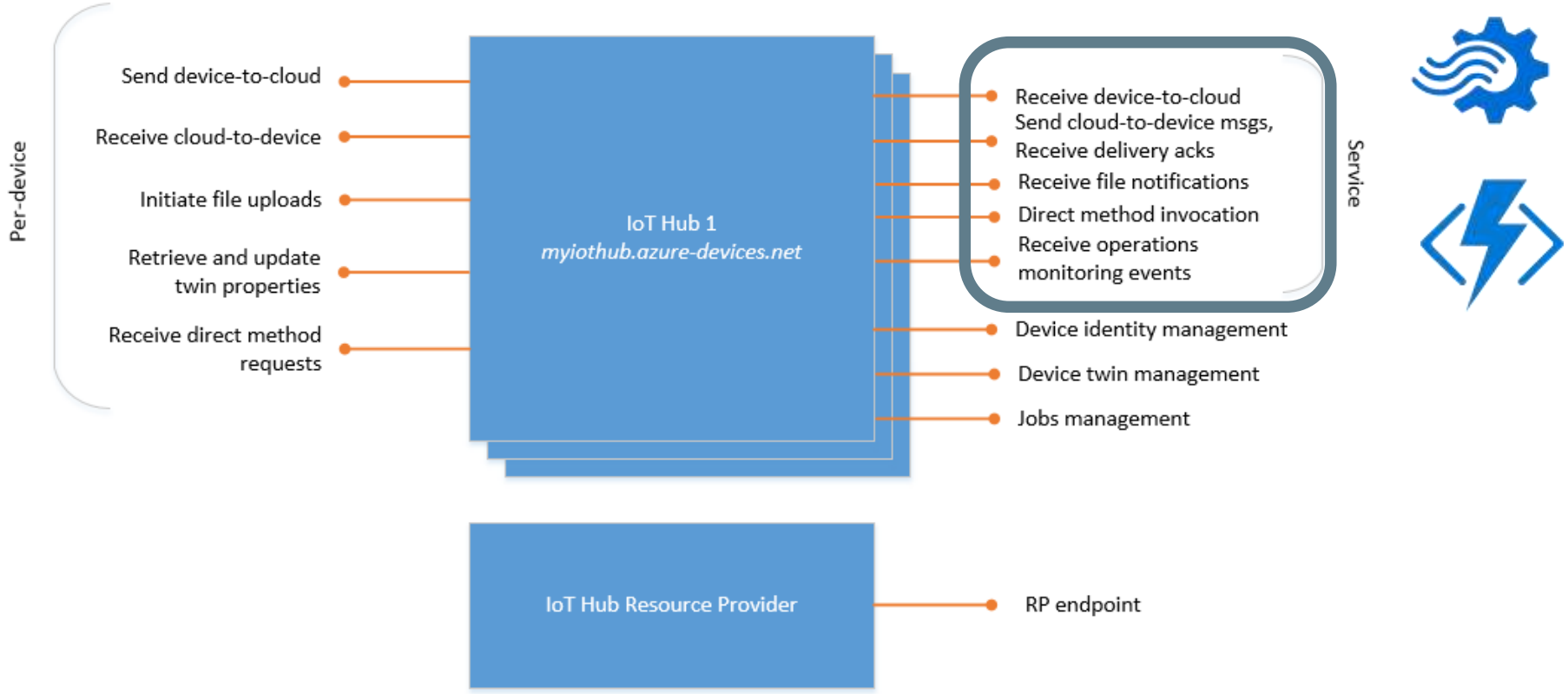
Used to authorize services

Is a permission to services or devices to access some endpoints

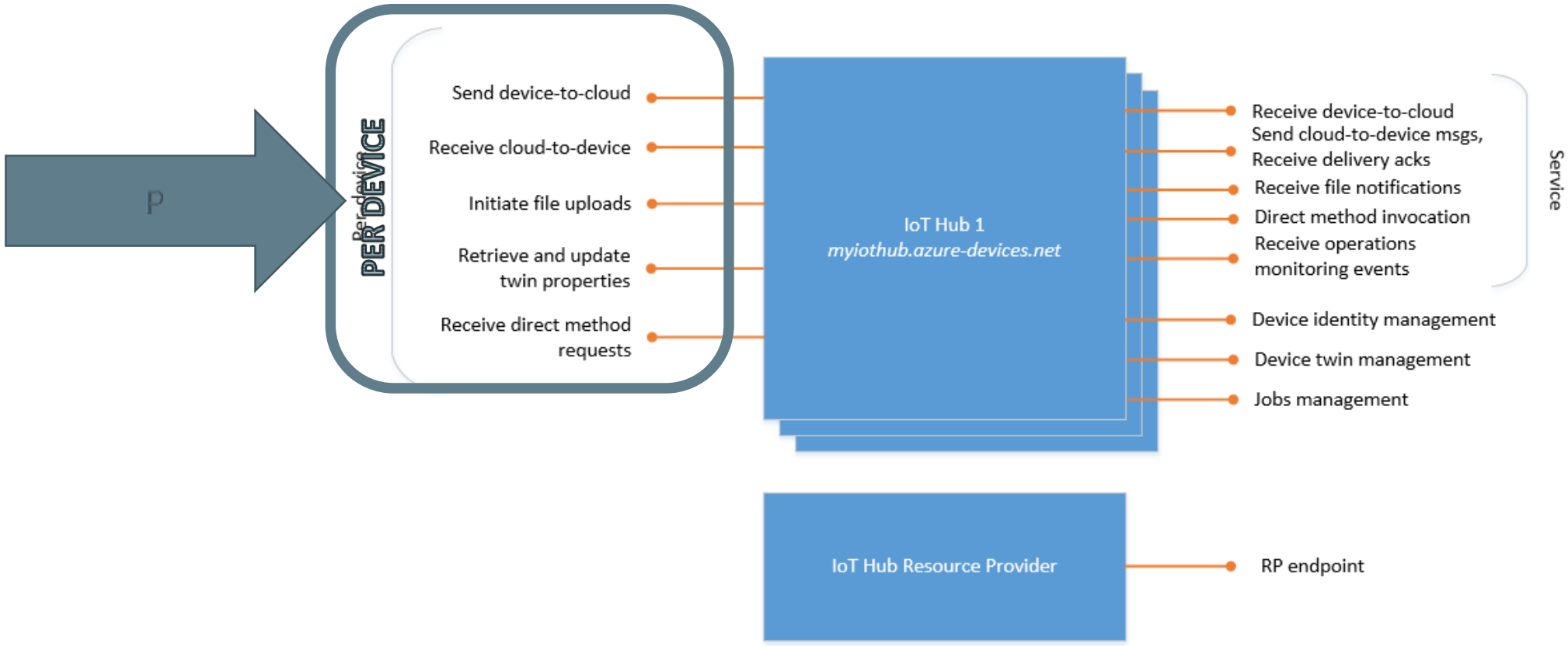
Uses symmetric key encryption technology for token authorization

Good practice: 1 policy, 1 service

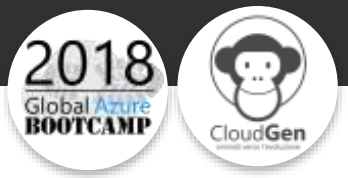
Service Connect Permission



Device Connect Permission



Device Registry Operations



Create identity

Update identity

Retrieve identity

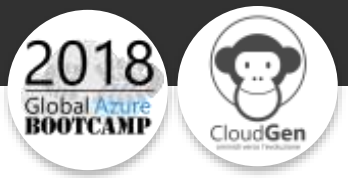
Delete identity

List identities

Export identities

Import identities

Device Registry Informations



deviceId is the name you assign to the device.

generationId is a property used to distinguish devices with the same deviceId, but that are deleted and recreated. So the real key should be deviceId+generationid

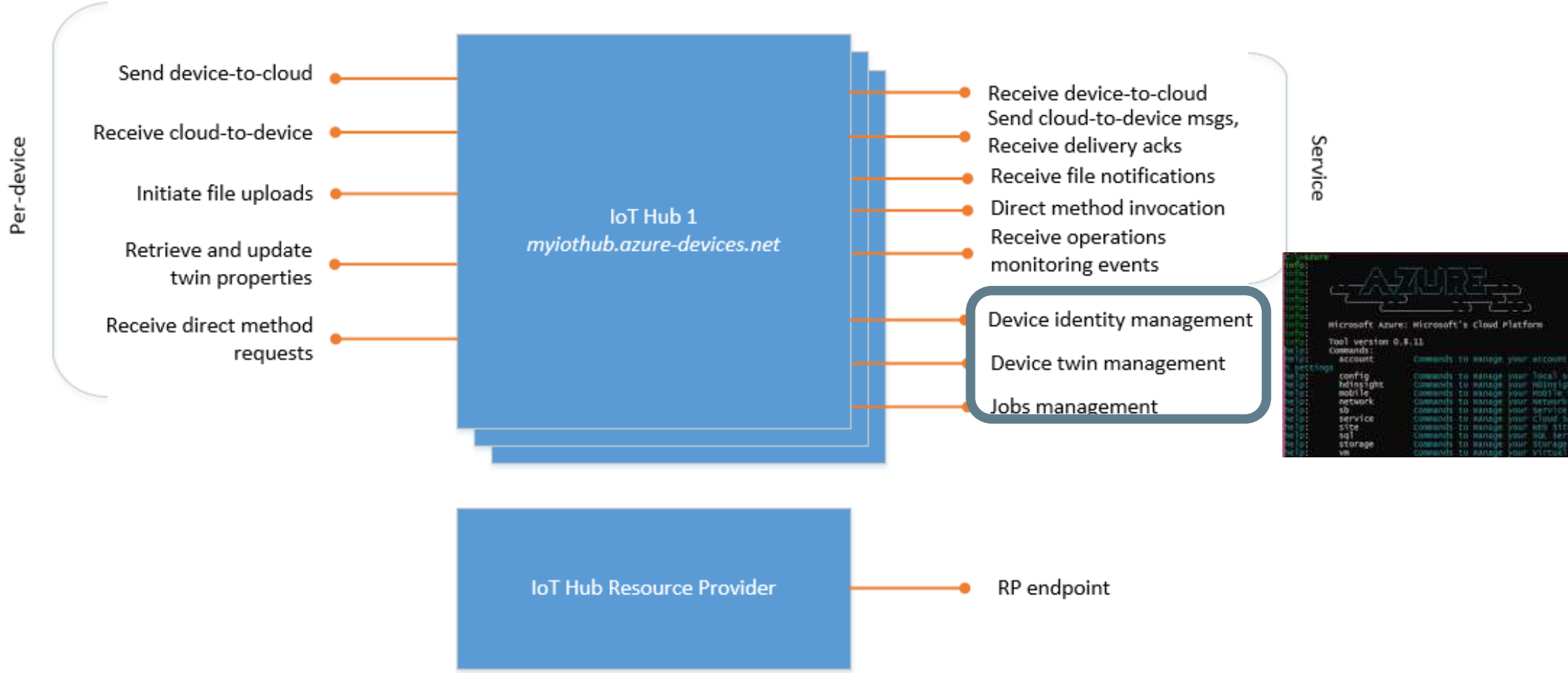
auth contains authentication information such as the symmetric keys, that are the couple of primary and secondary keys shared with IoTHUb used to secure each message. Those keys are stored in BASE64 format

Status, statusReason and statusUpdateTime are used to disable or enable the device and trace why device was disabled and when. If disabled, the device cannot use its identity to access to IoTHub.

connectionState and connectionStateUpdatedTime shows if the device is connected or not. This property is available only if you use AMQP or MQTT protocols and it is updated only every 5 minutes. So it can contain false positives and should be used only for debugging and testing purposes. You need to implement some sort of heartbeat functionality on your device to have a real feedback on connection from your device.

lastActivityTime tracks the last operation on that device

Registry Read/Registry Write Permission



Transport Level Security

TCP based protocols (HTTPS, MQTT, AMQP)

Endpoints exposes certificates with public key

Automatically handled by TCP/IP stack

Asymmetric Key encryption

Used to receive secure data by the parties

Couple of keys. Private keys, kept safe by the generator of the keys, decrypt what is encrypted by the public key,

Either self signed certificates or CA certificates (preview)

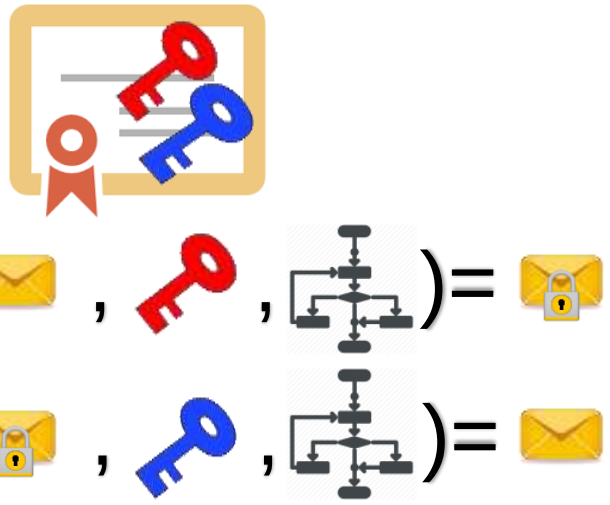
Pro

- single point of failure
- long keys (2^10 bit+) difficult to decrypt

Cons

- can encrypt small chunks of data

Used to encrypt a symmetric key at each communication



Symmetric Key encryption

Used to exchange secure data by the parties

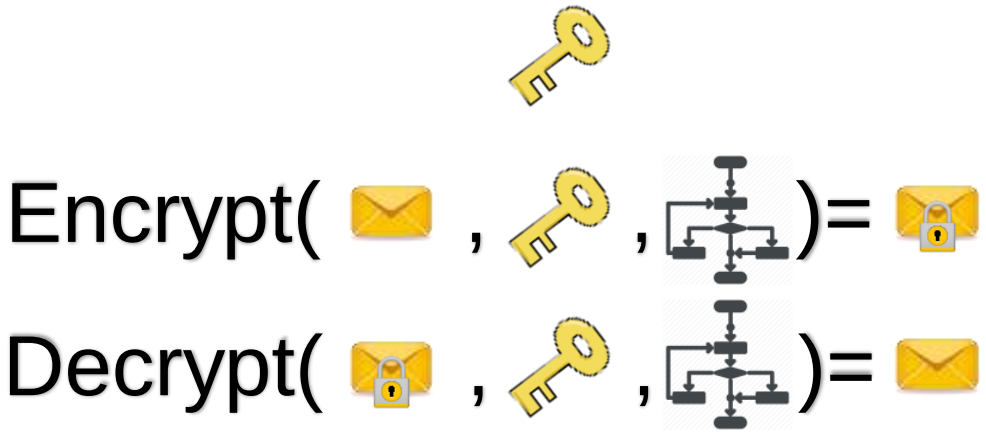
Single Key shared by the parties

Pro

- Can encrypt big blocks of data

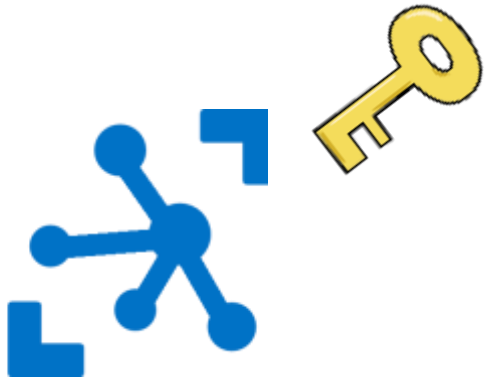
Cons

- Unsecure if one of the parties loose the key, multiple point of failures



Generate the symmetric key

DEVICE

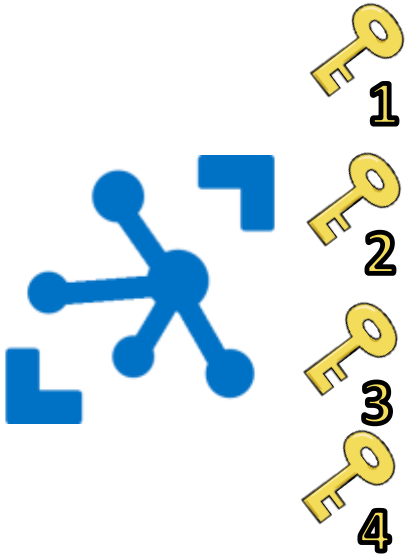


IOT HUB

```
C:\>az iot device create –hub-name <hubname> --device-id <deviceId>
```

All devices have different keys

Device Registry



Authorization with encryption

Encrypt({hostName}/devices/{device1} {expiration} , 1)

SharedAccessSignature sr={hostName}/devices/{device1}&sig={ }&se= {expiration}

Managing Devices with Azure IoT Hub

IoT Hub Device Management

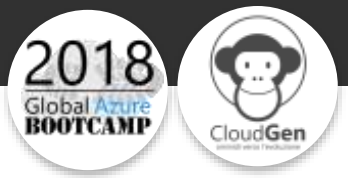
Device Twin

Queries

Methods

Jobs

Device Twin



Twin is the logical representation of the device.

It is a JSON document that stores device state information.

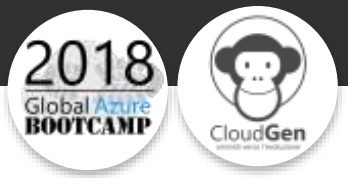
Information is properties that you can distinguish as tags, desired properties, and reported properties.

The document is stored in IoT Hub, in a CosmosDb-like eventually consistent container.

In general, all properties as just JSON properties so you can write anything you what respecting JSON rules.

It can be patched

Device Twin Limitations



Properties can have a maximum depth of 5.

The size of the property values cannot be bigger that 8Kb max .

JSON types supported: boolean, number, string, object. Arrays are not allowed

The document is updated and synchronized with device handling optimistic concurrency.

8Kb (billing size: 16 messages)

Desired properties

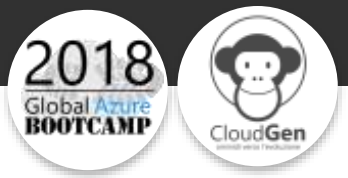
We need to configure the device. It is not cost effective to perform locally.

Desired property is a kind of property that is configured on the twin. IoT Hub handles the change queuing the update on the device endpoint.

So when it reconnects, it will update its state.

The maximum size of desired properties is 8Kb.

Reported properties



The device has a local state.

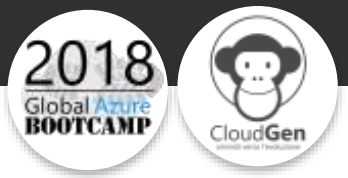
That state changes because device runs some tasks. You want to know that.

The device can send updates on these when they change.

IoTHub receives a message from the device endpoint about the update, and that is changed on the twin.

The maximum size of reported properties is 8Kb.

Tags



Desired properties and reported properties are functional for the device.

Some properties are useful only for the service and not the devices.

It’s a key/value data dictionary.

Querying Device Twins

FROM

WHERE

SELECT

GROUP BY

Sample queries

Devices located in the US configured to send telemetry less often that every minute

```
SELECT * FROM devices WHERE tags.location.region = 'US'
AND properties.reported.telemetryConfig.sendFrequencyInSecs >= 60
```

Devices which have wifi o wired connectivity

```
SELECT * FROM devices
WHERE properties.reported.connectivity IN ['wired', 'wifi']
```

Devices where reported and desired properties do not match

```
SELECT * FROM devices WHERE properties.reported.firmwareVersion <> properties.desired.
firmwareVersion
```

Devices group by status

```
SELECT * FROM devices WHERE properties.reported.firmwareVersion <> properties.desired.
firmwareVersion
```

Device Management Lifetime Operations

Reboot

Factory Reset

Firmware Update

Configuration

Reporting progress and status

Direct Methods

Immediate confirmation

Two-way data flow

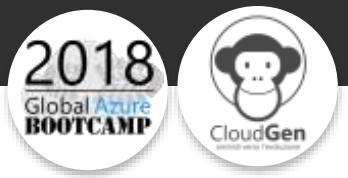
MQTT

8Kb request - 8Kb response

MQTT

Handling Events with Azure Serverless

Where storing a message



Azure SQL Database

Azure DocumentDb

Azure EventHub

Azure Queue

Azure Service Bus

Azure Storage

Alternatives to process events

Stream Analytics

Event Processor

Functions

Comparing different ways

- Data streaming
- Event correlation
- High scalability

- Single event
- Performance not critical
- Custom coding
- Flexible coding

- Special hosting requirements
- Special performance requirements

Serverless manifesto

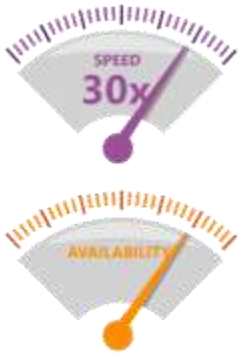
Function are the unit of deployment and scaling.

Scales per request Users cannot over- or under-provision capacity.

Never pay for idle (no cold servers/containers or their costs)

Trigger-based invocation code run because of an event happened and has to be handled

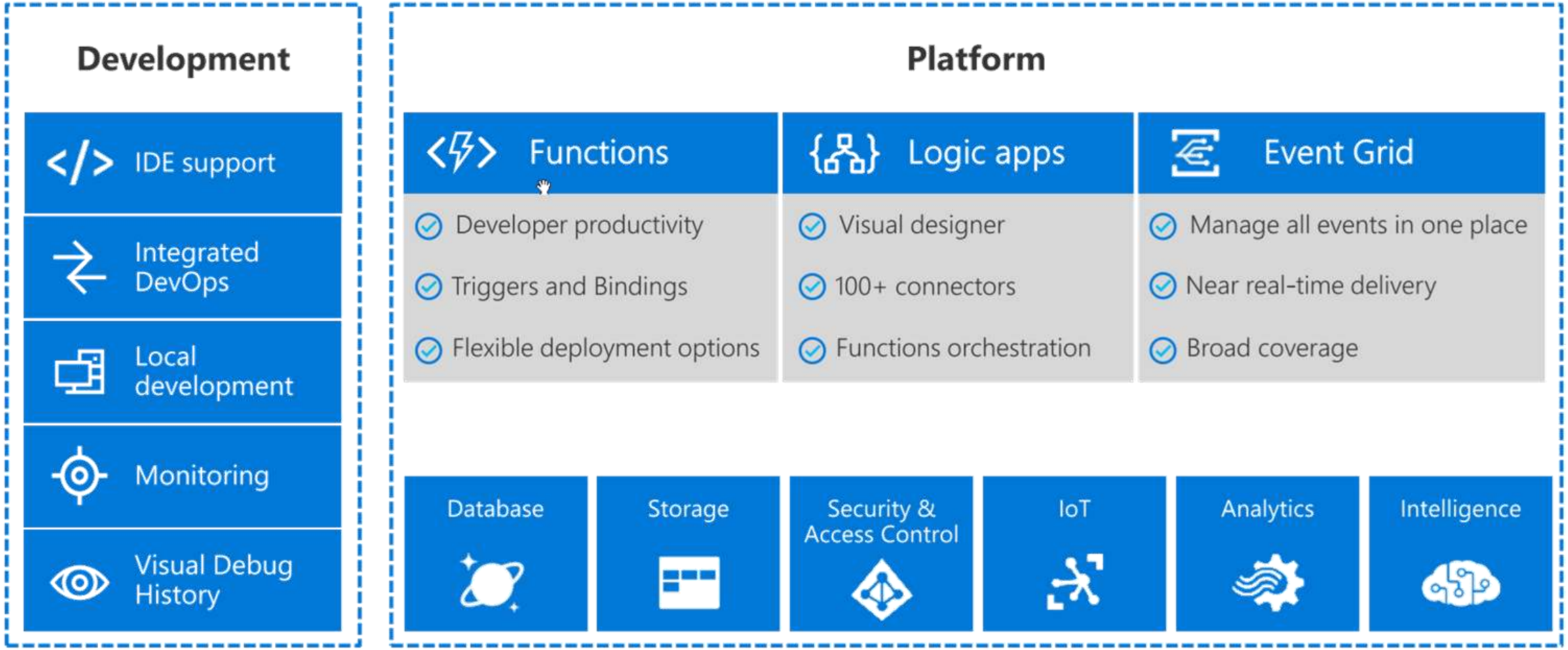
Abstraction of servers



Event-driven/ instant scale

Sub-second billing

What is Azure Serverless



https://www.geekwire.com/2017/serverless-nirvana-microsoft-azure-cto-mark-russinovich-future-cloud/
https://www.geekwire.com/2017/interview-microsofts-mark-russinovich-intersection-serverless-edge-computing/

"Ho i dati e ora devo creare una dashboard. Posso usare Power BI? No grazie, fanne una tu, così ne abbiamo completamente il controllo"

Conclusioni

Conclusioni

Approccio PaaS all'Internet of Things

Piattaforma su cui costruire la propria soluzione

Ottimo modo per cominciare e anche crescere

Ottimo modo per approcciare lo scenario Industria 4.0

Allegato A - Beni funzionali alla trasformazione tecnologica e e/o digitale delle imprese secondo il modello "Industria 4.0"

Tutte le macchine devono essere dotate delle seguenti caratteristiche:

- controllo per mezzo di CNC (Computer Numerical Control) e/o PLC (Programmable Logic Controller);
- interconnessione ai sistemi informatici di fabbrica [ERP] con caricamento da remoto di istruzioni e/o parti di programma;
- integrazione automatizzata con il sistema logistico della fabbrica o con la rete di fornitura e/o con altre macchine del ciclo produttivo;
- interfaccia tra uomo e macchina semplici e intuitive;
- rispondenza ai più recenti parametri di sicurezza, salute e igiene del lavoro.

55

Grazie

Domande?

 marcoparenzan

 @marco_parenzan

 marcoparenzan

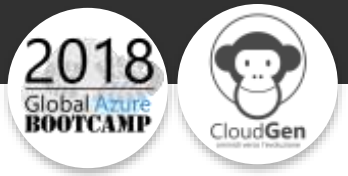
Verona

Gli sponsor Mondiali

Gli sponsor del nostro evento

Platinum Sponsor

Gli sponsor del nostro evento



Gold Sponsor

Gli sponsor del nostro evento

Basic Sponsor

Tweet della giornata

#GlobalAzure

@cloudgen_verona