Gestire devices per l'Internet of Things con Azure IoT

MARCO PARENZAN

OverNet EDUCATION

Agenda

- Device Management with Azure IoT Hub
- Azure Device Provisioning Service
- Azure IoT Edge
- Server side processing

Device Management with Azure IoT Hub

Experience
Custom Ingestion: 2 weeks up and running (and a VM)
IoT Hub: 2 hours up and running (and a PaaS service)

Connect a device for…

- Telemetry
  - Data flows in one direction from the device to other systems for conveying status changes in the device itself
- Inquiries
  - Requests from the device looking to gather required information or asking to initiate activities
- Commands
  - Commands from other systems sent to a device (or a group of devices) to perform specific activities expecting a result from the command execution, or at least a status for that
- Notifications
  - Information flows in one direction from other systems to a device (or a group of devices) for conveying status changes

Device Management

- Device Provisioning
- Device Configuration
- Invoke a task on the device
- Invoke a task on many devices
- Query for your asset
- Generalize drivers for multiple version support

Azure IoT Hub

Bi-directional communication

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

Enterprise scale and integration

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

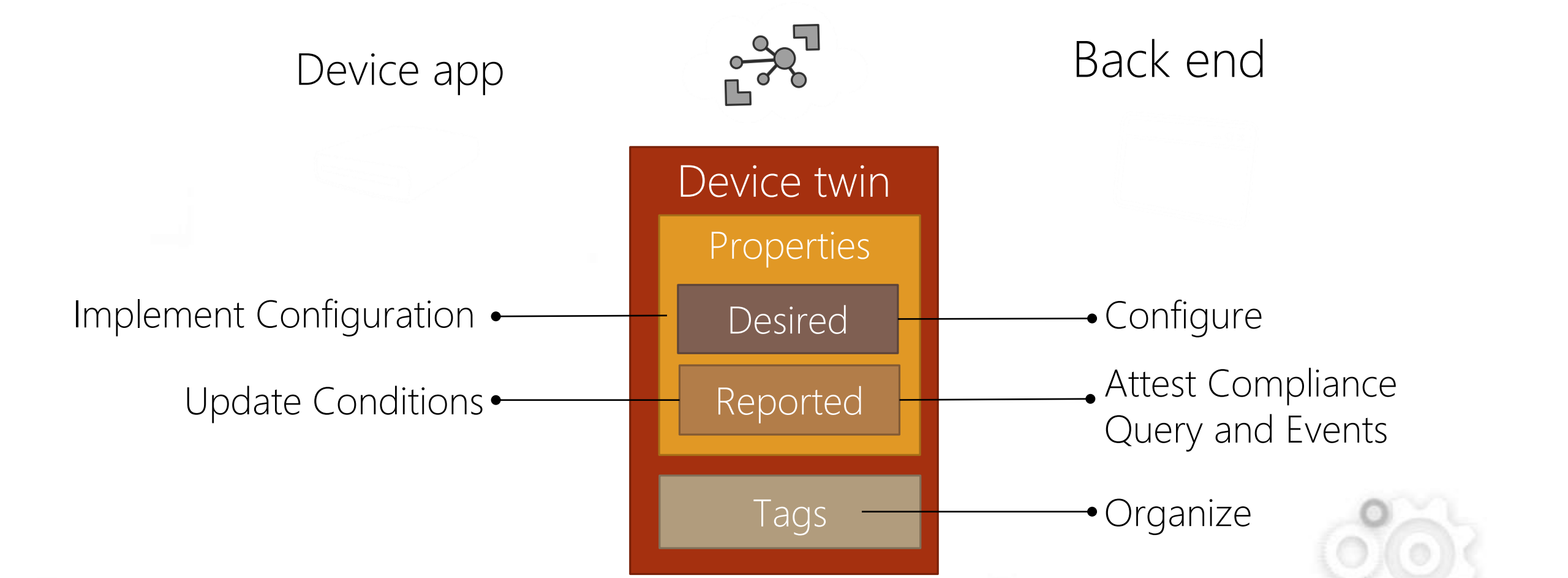
End-to-end security

- Per device certificates
- Per device enable/disable
- TLS security
- X.509 support
- IP whitelisting/blacklisting
- Shared access polices
- Firmware/software updates

IoT Hub Device Management

- Device Provisioning
  - Automatic device registration including, management enrollment and initial configuration
- Device Twin
  - Synchronize the device condition and configuration between cloud and device
- Methods
  - Perform interactive actions (e.g command &control) on devices
- Jobs
  - Broadcast and schedule device twin changes and methods at scale
- Queries
  - Dynamic reporting across device twin and jobs to attest device status and health
- Patterns, Libraries, & Implementations
  - Get started quickly with the most essential operations supporting a diverse set of platforms

Azure IoT Device Twin



Learnings over the years

- DON'T use a C2D Message for device configuration
  - TTL will never be long enough.
- DON'T use a Direct Method for device configuration
  - Direct methods are interactive (request/response).
- DO use Device Twin Desired Properties for configuration
- DO use Direct Methods for remediation
- DO use Device Reported Properties for config compliance

DEMO

Azure Device Provisioning Service

Answer these IoT questions…

- How will you connect your devices?
- How will you securely identify and enroll your devices?
- How do you scale enrollment for many devices?

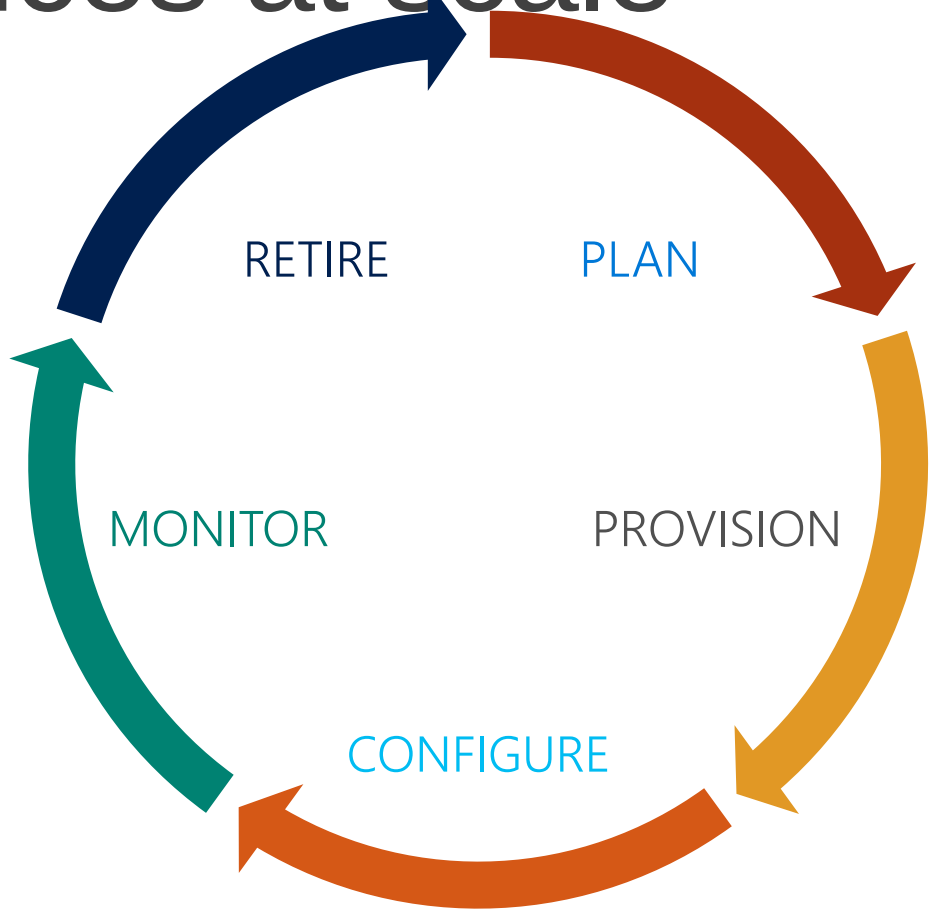
What is provisioning?

Experience
Custom Provisioning: 2 weeks up and running and 3 services
IoT Hub DPS: 2 days up and running and 1 service

Manage devices at scale

Replace or decommission devices after failure, upgrade cycle or service lifetime

Monitor device inventory, health & security while providing proactive remediation of issues



Group devices and control access according to your organization's needs

Securely authenticate devices, on-board for management and provision for service

Provide updates, configuration & applications to assign the purpose of each device

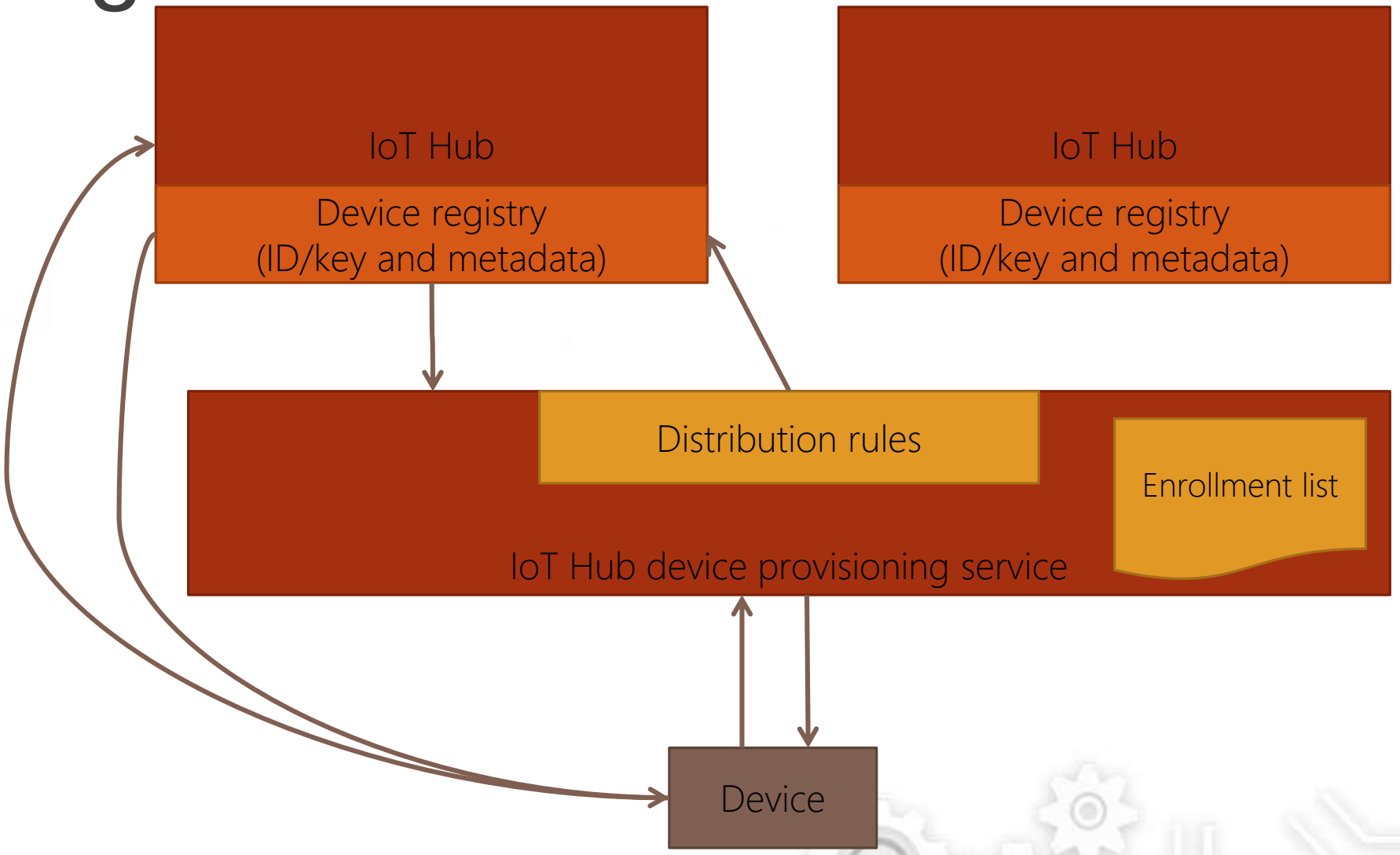
Introducing: IoT Hub Device Provisioning

- Devices are automatically and securely connected to the IoT Hub service and provisioned with initial configuration
- A single device provisioning tenant can provide service for multiple IoT hubs (in multiple regions)
- Customers provide rules and logic to assure the right device is attached to the right IoT solution (and associated IoT Hub endpoint)
- Device provisioning ability is extensible with support for several types of identity attestation patterns

Setup

- Devices already have the URI for the device provisioning service at first boot
- Device provisioning service already knows about the IoT hubs to which it can connect
- Device provisioning service already knows the type of identity attestation it is using, including connection info for each

Using an enrollment list



DEMO

Azure IoT Edge

Why Azure IoT Edge?

- Because not everything is on the cloud
  - Latency
  - Control
  - Pre-processing
- Because there are also the devices...
- ...or the gateway
- Commercial/Industrual/Ruggered
  - Look at Azure IoT Certified

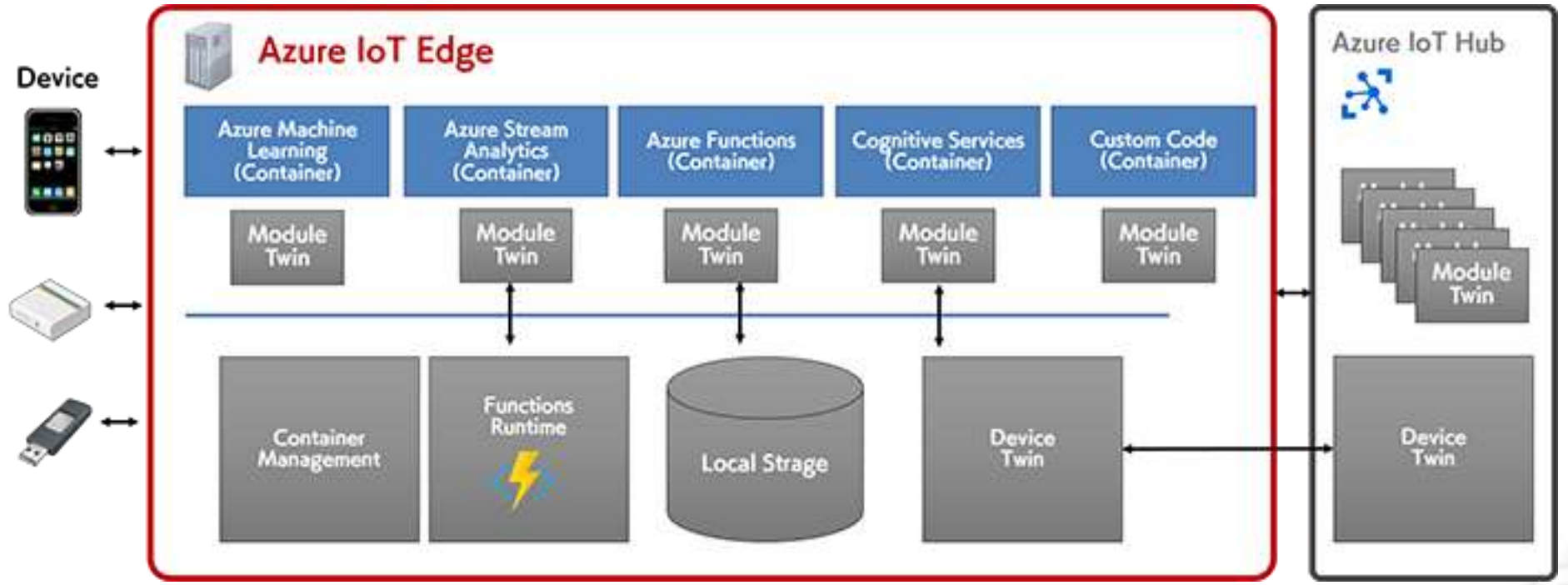
Azure IoT Edge

- It’s a device!
- Gives a structure to the edge client
- Based on docker «philosophy»
  - You need to build a container image
  - You need a container registry

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

Azure IoT Edge «Edge» Architecture



Modules

- It’s a pipeline
- Custom modules
  - C#/.NET Core
  - JavaScript
- Standard!
  - Functions
  - Stream Analytics
  - ML

Concepts – Edge Runtime

- Edge Runtime provides fundamental services
- Security
- Multiplexing
- Store and forward (Offline)
- Management for devices otherwise isolated from internet

DEMO

Telemetry Server side Processing

Which services for (IoT) events processing?

- Azure Functions
- Stream Analytics
- EventProcessorHost

Azure Function

- Good for low ingestion
  - Moving over EventGrid will improve performances
- No event correlation
  - Durable Functions not for this...
- EventGrid will change, but it’s not the same

Stream Analytics

- Great semantics
- Hyperscale
- But
  - Not cost effective in the mid-low
  - No SQL friendly
  - Difficult to test
  - Slow to start
  - shooting fish in a barrel (equivalente di «sparare a un topo con un cannone»)
    - Just to make time window aggregation

EventProcessorHost

- Where it is?
- How does it scale with EventProcessorHost?
- EventHub is «a queue» (not really, but Receive exists!)

Summer testing of...

- Experimenting with IoT protocols directly (AMQP and MQTT) by the device
- Consuming EventHub with AMQP protocol directly
- Consuming Redis with raw protocol
- Want to experiment containers
  - No more VMs for workers
  - Never liked WebJobs for that
  - No functions…Event Grid destiny…

...and a 1SU (3 subqueries) Stream Analytics process hanging with just 40 messages at the same time every two minutes...

“Serverless Streaming At Scale with Cosmos DB” by Davide Mauri

- October 9th, 2018 after my thought...
- https://medium.com/@mauridb/serverless-streaming-at-scale-with-cosmos-db-e0e26cacd27d
- «…But if you don’t need time-aware features, like Hopping or Tumbling Window, complex data processing capabilities, like stream joining, aggregates of streams and the likes, a more lightweight solution can be an option….”

Containers

- Containers = Love
- IaaS = Hate
- AKS .... Love?
- Serious Answer
  - Containers are the ~~future~~present
  - I don’t implement an AKS cluster in a proposed solution, not PaaS
  - Only if already existing or requested
  - Service Fabric is the same

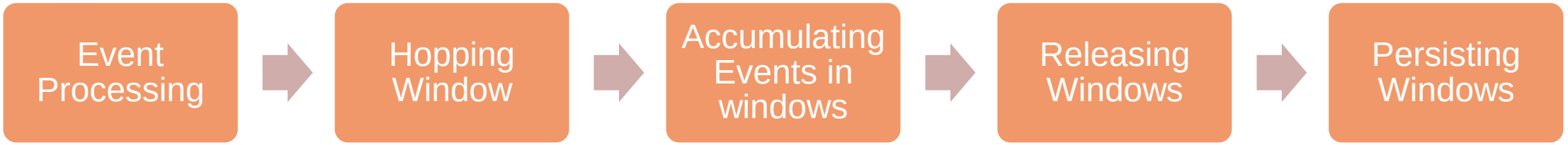
Great news

- Azure Container Instances
- AKS as a Service (PaaS)
- Not completly equivalent with AKS
  - No orchestrator
- Fits well for a fixed partition-based model (one container instance per partition)

Stateful IoT processing: how does it work?

Architecture

Replacing Stream Analytics

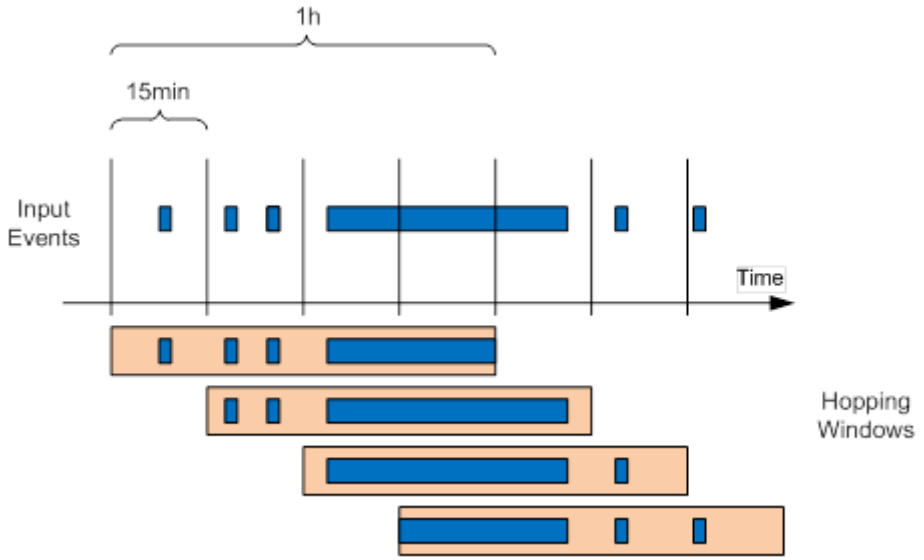


Event Processing

- Accessing EventHub API (also for IoT Hub)
- Using AMQPLite lib
- also Microsoft.Azure.EventHub
  - But no Microsoft Azure.EventHub.Processor

Hopping Window

- Fixed size time window
- Overlapping window
- Extreme condition: zero overlap➔Tumbling Window
- Class for window identification



Accumulating events in windows

- Collections
- High availability?
- Redis!
- Using Lists

Releasing data as time windows

- How long windows stay in memory?
- SA uses the notion of «unordered events»
- No time critical
  - Example 15minutes window➔after 20+ minutes
- How?
  - Redis caching expiration!

Persisting Time Windows

- Direct SQL pressure? No!
- Using queues
- Then a worker can off load queue and store in SQL
  - In the demo it’s just a thread, but it can be another container...

Hyper-scaling

- All these steps can be implemented in different containers (and probably SA does the same ☺)
- And remember that you can replicate it for each partition

Building an event processor

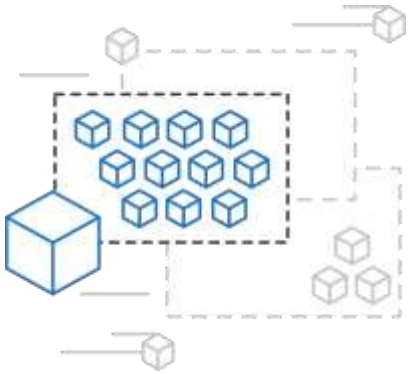
- .NET Core
- ASP.NET Core
- IHostedService
- Make it observable (at least in testing)
  - MVC
  - SignalR

DEMO

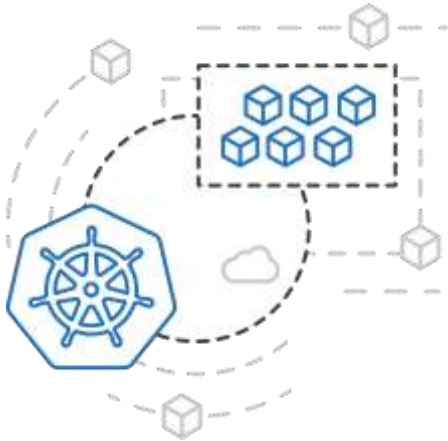
Azure Container Instances (ACI)

Easily run containers without managing servers

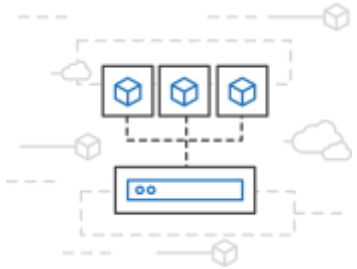
Now GA!



Run containers without managing servers



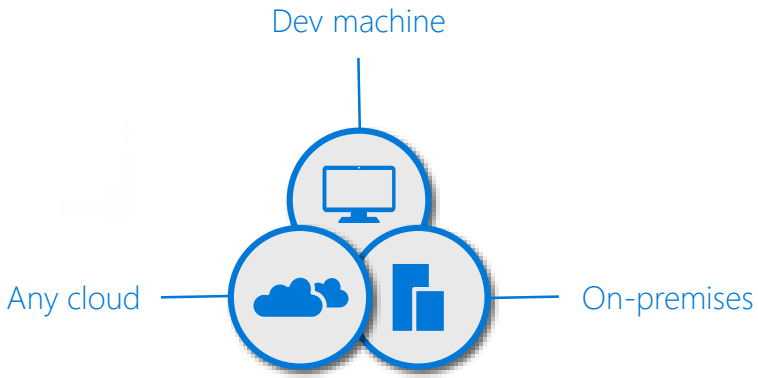
Containers as a primitive billed per second



Secure applications with hypervisor isolation

Service Fabric

Dedicated



Service Fabric Standalone

Bring your own infrastructure

Service Fabric Cluster

Dedicated Azure clusters

Serverless

Service Fabric Mesh

Fully managed by Azure

No more Redis
(that is nothing about better or worst)
Just one thing less...

Conclusions?

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

https://aka.ms/7principles

What is Azure Sphere?

- A new Azure Sphere OS secured by Microsoft for the devices 10-year lifetime to create a trustworthy platform for new IoT experiences
- The Azure Sphere Security Service guards every Azure Sphere device; it brokers trust for device-to-device and device-to-cloud communication, detects emerging threats, and renews device security.
- A new Azure Sphere class of MCUs, from silicon partners, with built-in Microsoft security technology provide connectivity and a dependable hardware root of trust

Q&A

- Domande e risposte

Contatti

ROZZANO (MI)
BOLOGNA
ROMA
GENOVA
TORINO
NAPOLI

- OverNet Education
- Info@OverNetEducation.it
- www.OverNetEducation.it
- Rozzano (MI)+39 02 365738
- Bologna +39 051 269911

- www.wpc-overneteducation.it