TOPIC

Azure Digital Twins

Marco Parenzan
@marco_parenzan

Thanks to all the sponsors

PREMIUM SPONSOR

4WARD

KIRATECH®
WE DEVOPS IT

BASIC SPONSOR

CODICEPLASTICO

dgroove
Tecnologia emozionale.

Who I am

Marco Parenzan

Solutions Sales Specialist @ Insight
Microsoft MVP 2018-2019 for Azure
Community Lead per 1nn0va

marco_parenzan

marco.parenzan

marcoparenzan

Approaching an IoT project

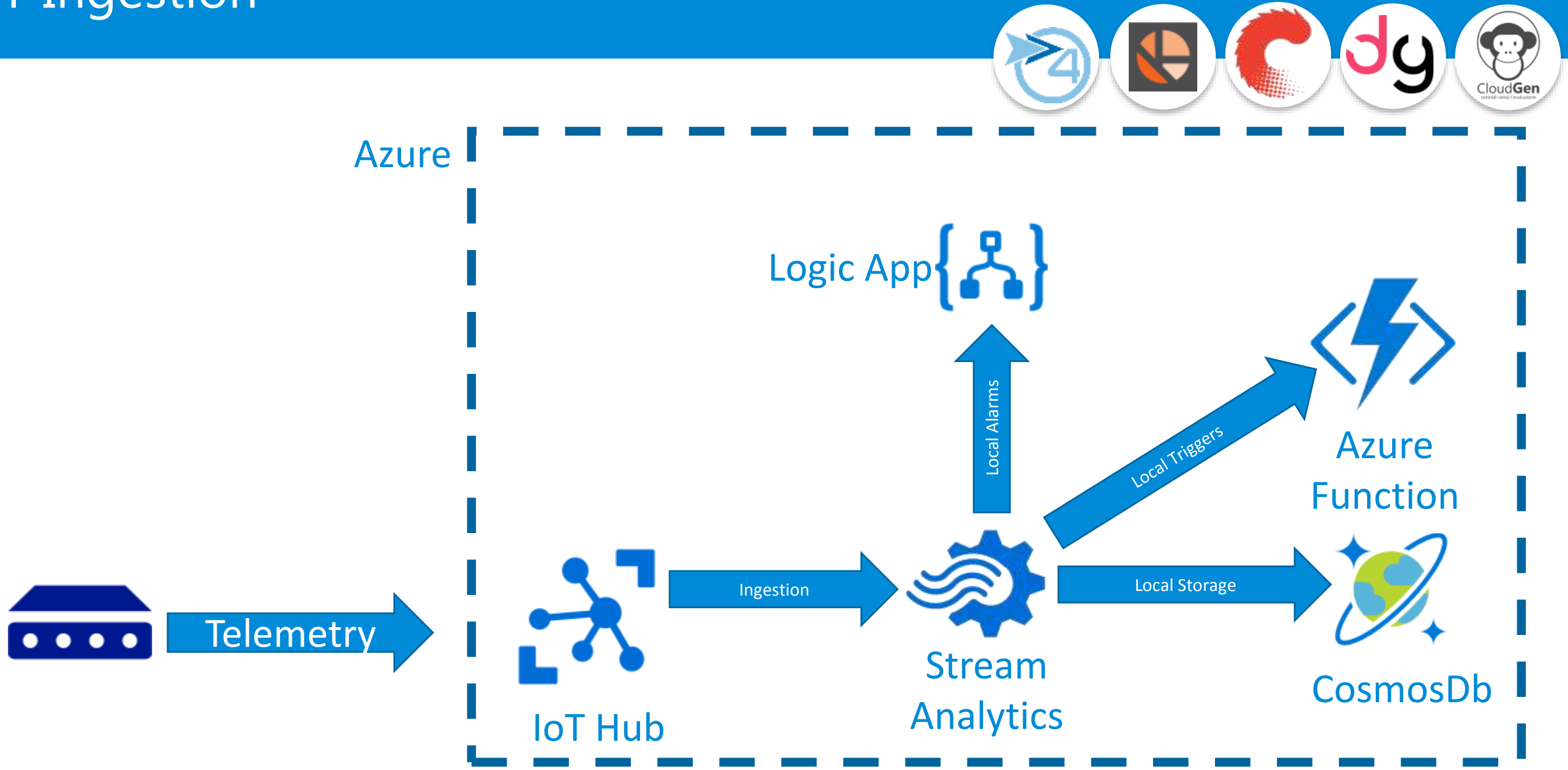
IoT Hub «state of the art»

Telemetry Ingestion and Controlling

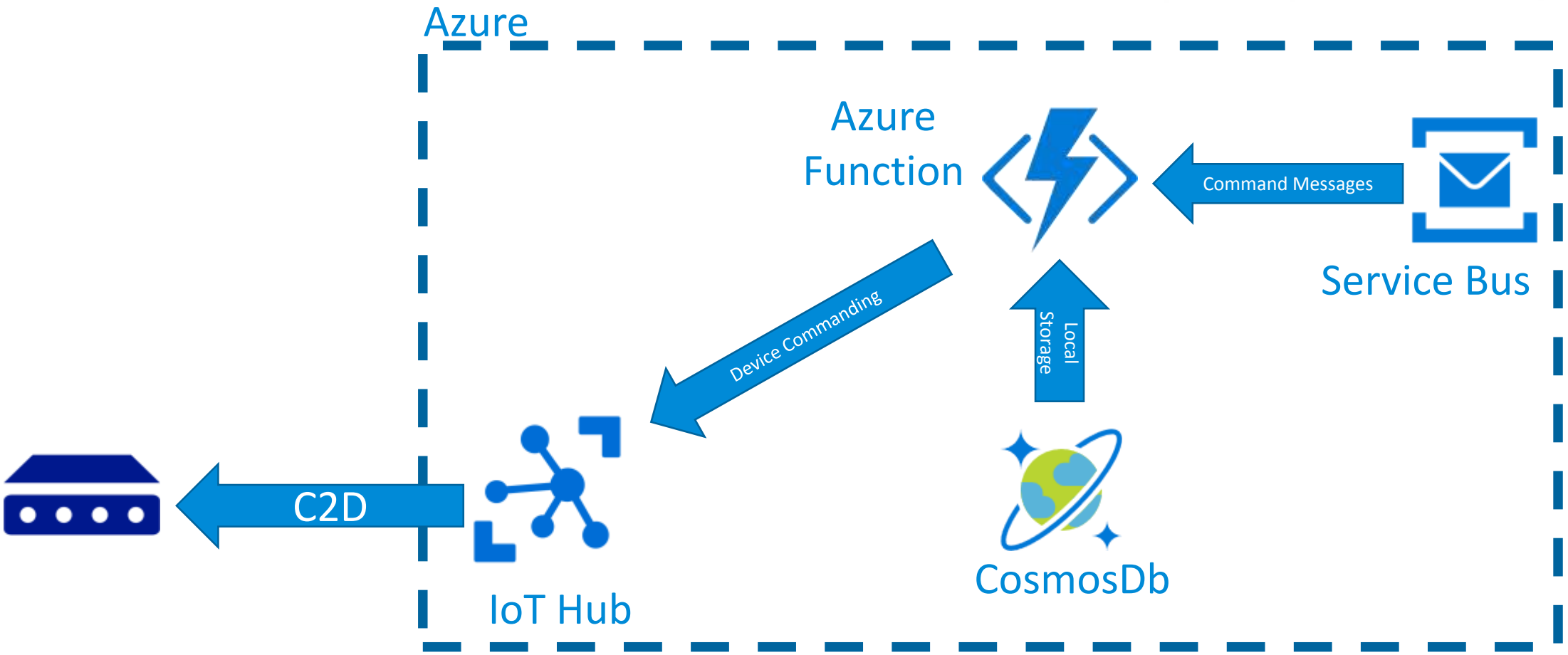
You have seen lot of times

https://www.slideshare.net/marco.parenzan/sviluppare-un-portale-per-gestire-la-tua-soluzione-iot-hub

IoT Ingestion



IoT Controlling



The IoTHub registry

Store key/value flat data to the device

Example

- CRM Key
- Spare Part Key
- ...

1Thing = 1Device

The device state (aka the twin)

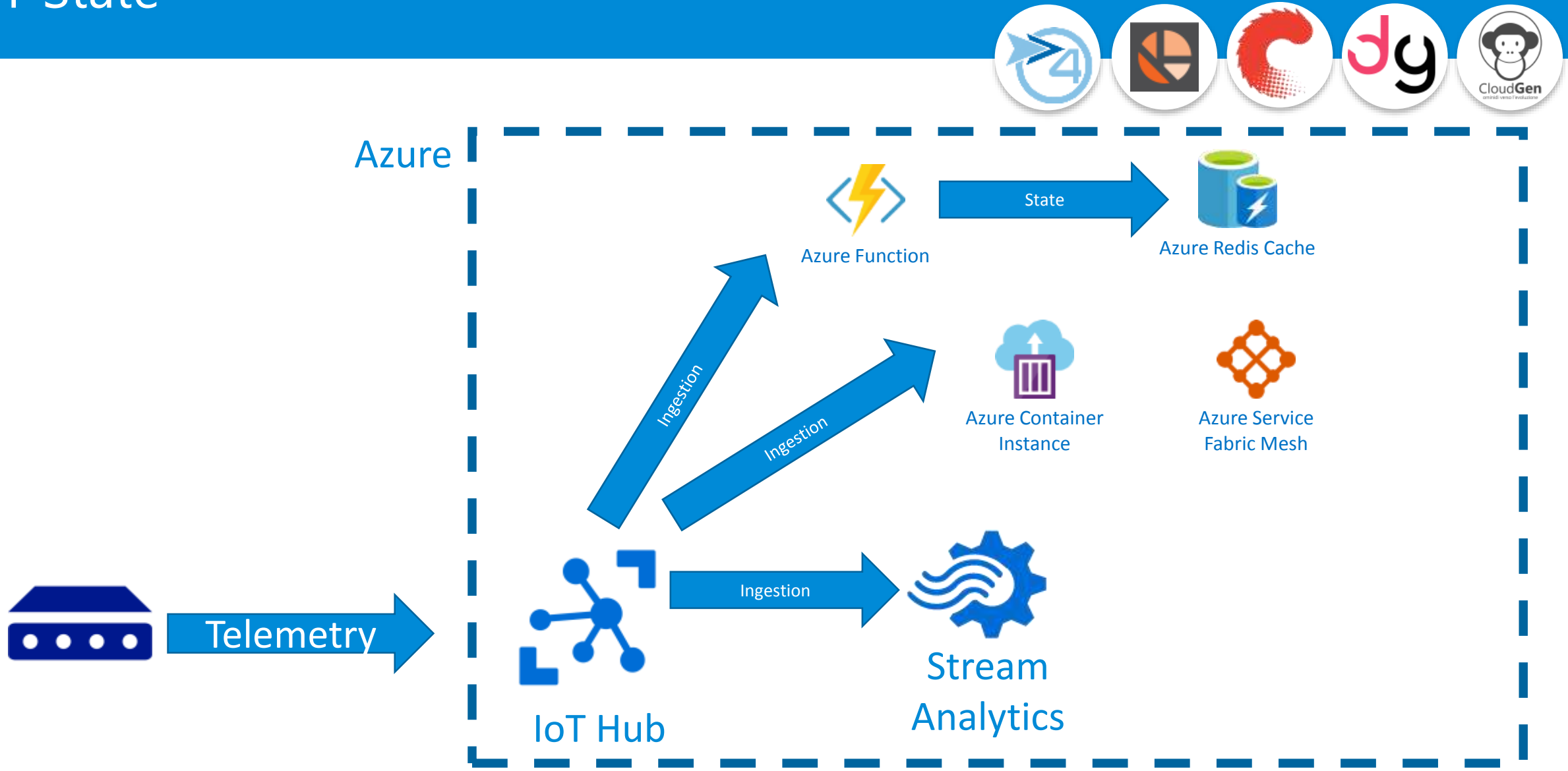
Solved (?)

- Function+Storage (persistent)
- EventGrid+WebHook (in memory)
- ...but always custom

And you have living...

- ...physical properties
- ...computed properties (javascript or Roslyn computed)
  - This is dangerously stateful
  - Stateless is expensive

IoT State


Azure
Azure Function
State
Azure Redis Cache
Ingestion
Ingestion
Azure Container Instance
Azure Service Fabric Mesh
Telemetry
Ingestion
IoT Hub
Stream Analytics

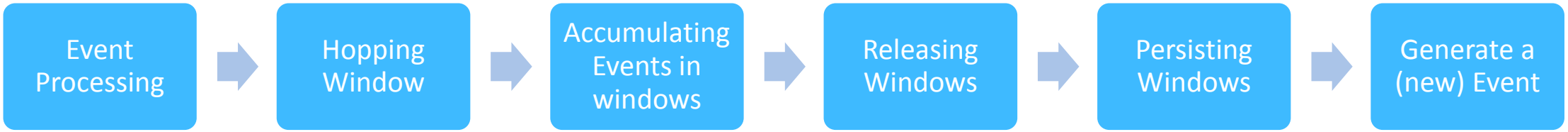
Another Research Area...

Building an event processor (ASP.NET Core, IHostedService

Make it observable, SignalR)

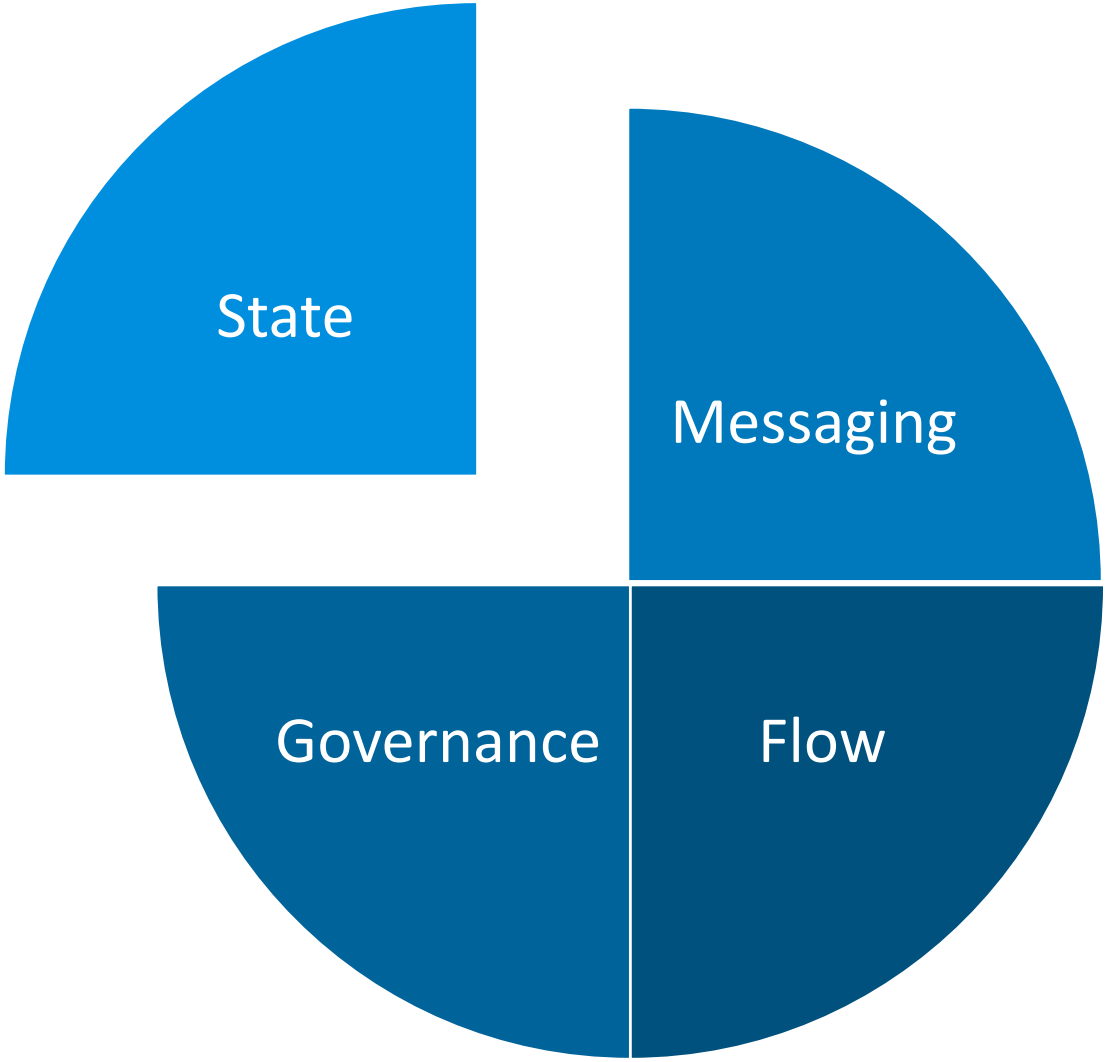
Notify with EventGrid

Run in Azure Container Instances (ACI)



https://www.slideshare.net/marco.parenzan/stateful-stream-processing-of-iiot-events-with-c-and-containers

So another point of view



Any idea?

What is Digital Twins

What is Digital Twin?

It is another point of view modeling your real infrastructure in the cloud

Focused mainly on resources, where there are the devices too…

- Not only devices!

Using «spatial intelligence» graphs

- Inheritance (security, for example)
- Filtering
- Traversing (the graph, different levels)
- Scalability (it’s a DB…)
- Extensible (with customizable enumerations and ontologies)

The difference between DeviceTwin and Digital Twin

Twin is a cloud based rapresentation of something that is remote

DeviceTwin is

- A key/value flat representation of
  - Desired configuration
  - Reported configuration
  - Keys to match to an external database

Digital Twin is

- A graph (tree…someone asking for graph)
- Richer semantics
- Also not only devices

Defining the spatial model

Spaces are virtual or physical locations

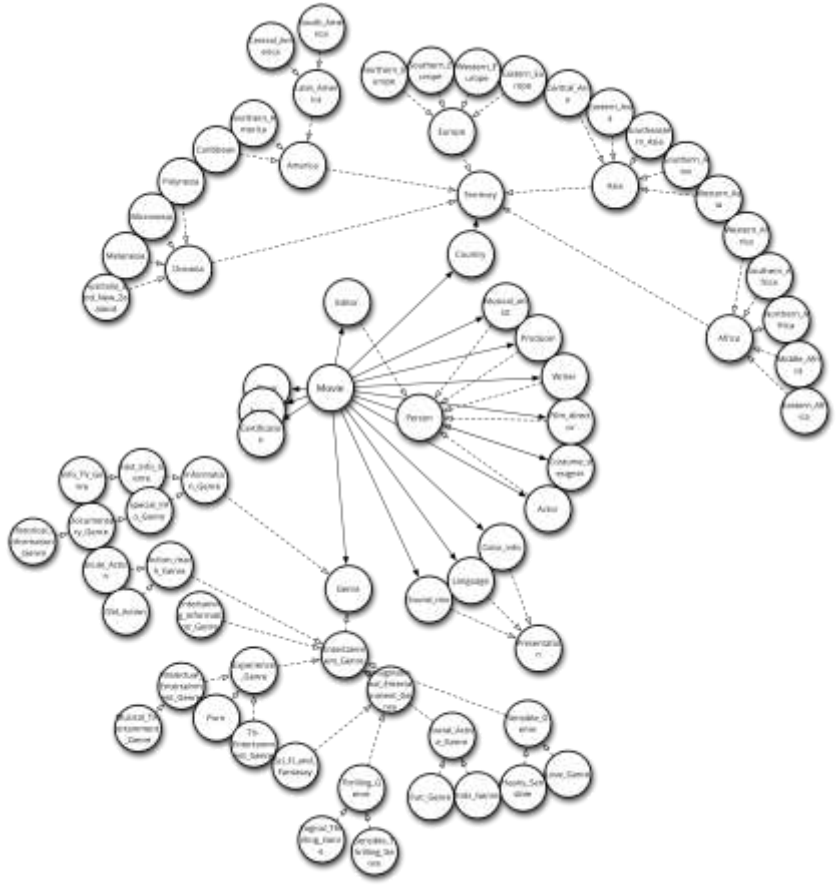
Devices are virtual or physical pieces of equipment

Sensors are objects that detect events

Users identify occupants and their characteristics.

Blobs are attached to objects (such as spaces, devices, sensors, and users)

Extended types are extensible enumerations that augment entities with specific characteristics



Defining the processing model

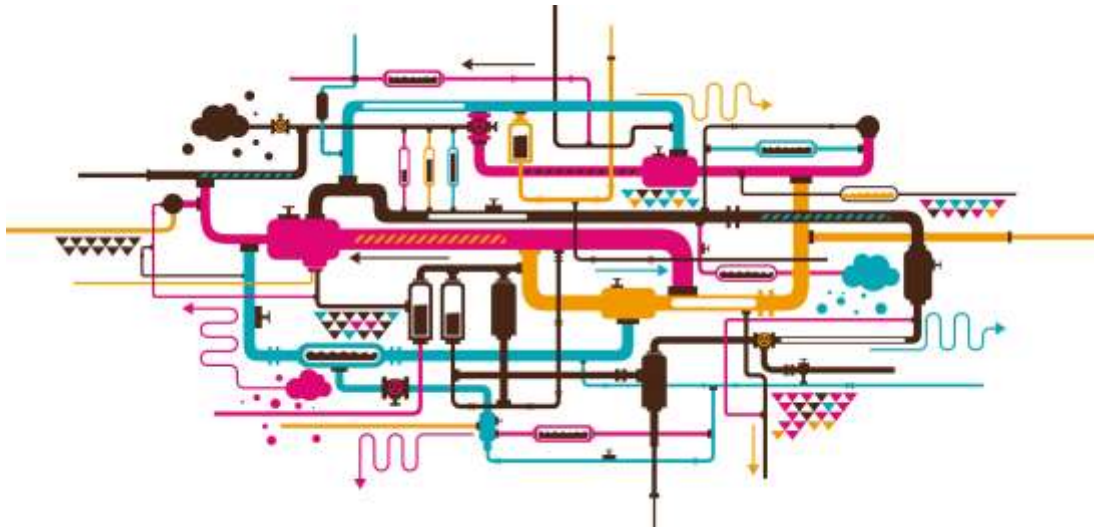
Resources are attached to a space and typically represent Azure resources to be used by objects in the spatial graph

User-defined functions (UDFs) allow customizable sensor telemetry processing within the spatial graph. Currently, UDFs can be written in JavaScript.

Matchers are objects that determine which UDFs are executed for a given telemetry message.

Role assignments are the association between a role and an object in the spatial graph

Endpoints are the locations where telemetry messages and Digital Twins events can be routed



Other part of the model

Ontologies represent a set of extended types

Property keys and values are custom characteristics of spaces, devices, sensors, and users

Roles are sets of permissions assigned to users and devices in the spatial graph

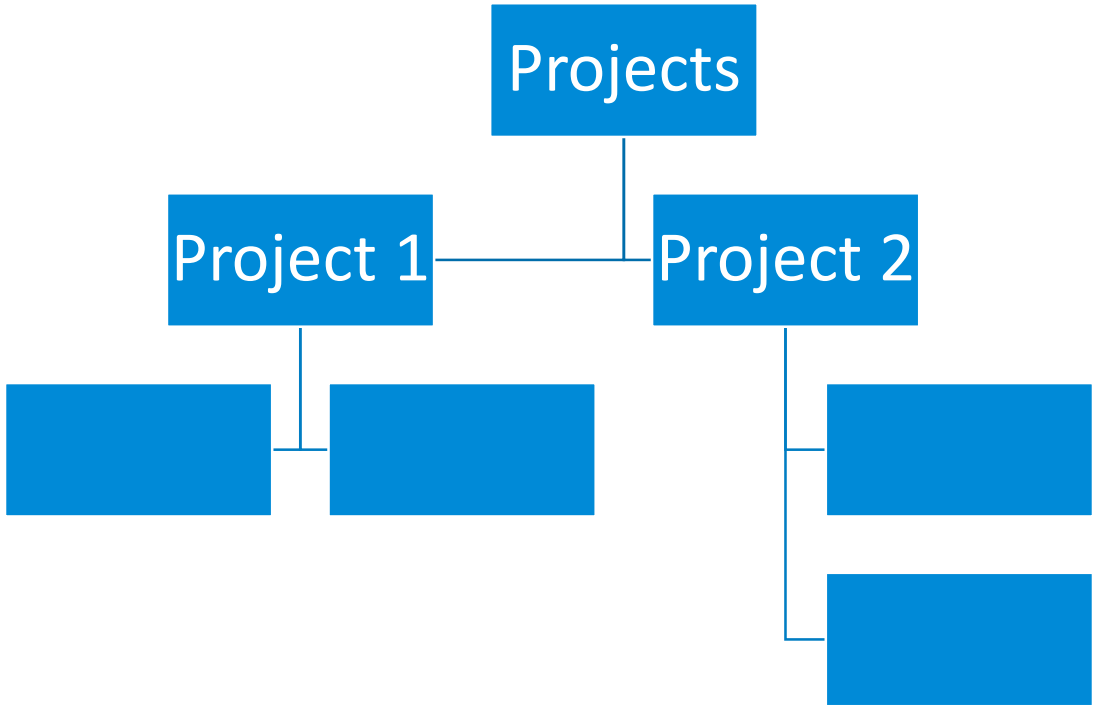
Security key stores provide the security keys for all devices in the hierarchy under a given space object to allow the device to securely communicate with Digital Twins.

Space

This is the first methaphor in Digital Twins

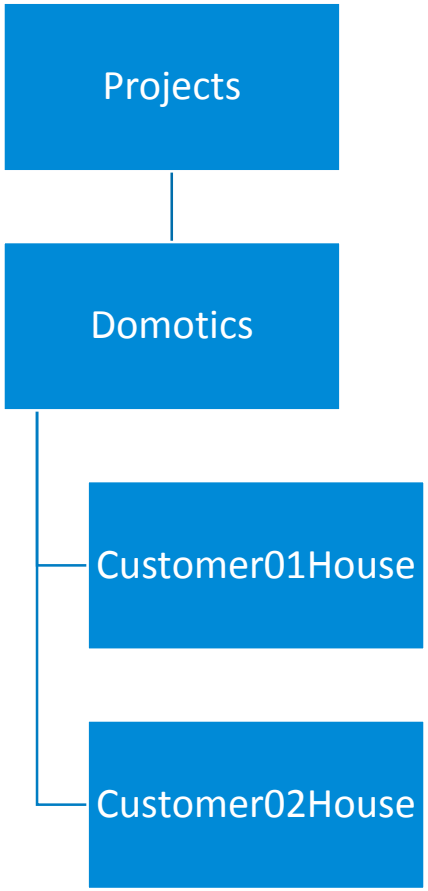
You can create hierarchies of spaces

Root space



- Preview
  - One instance of Digital Twins per subscription
- Overall, a multitenant approach
- Use a root node for all projects
- Use a child root node for each project
  - And in experimentation you can leave it there if garbage (ex. When you create iothub resources)
- Good governance practice

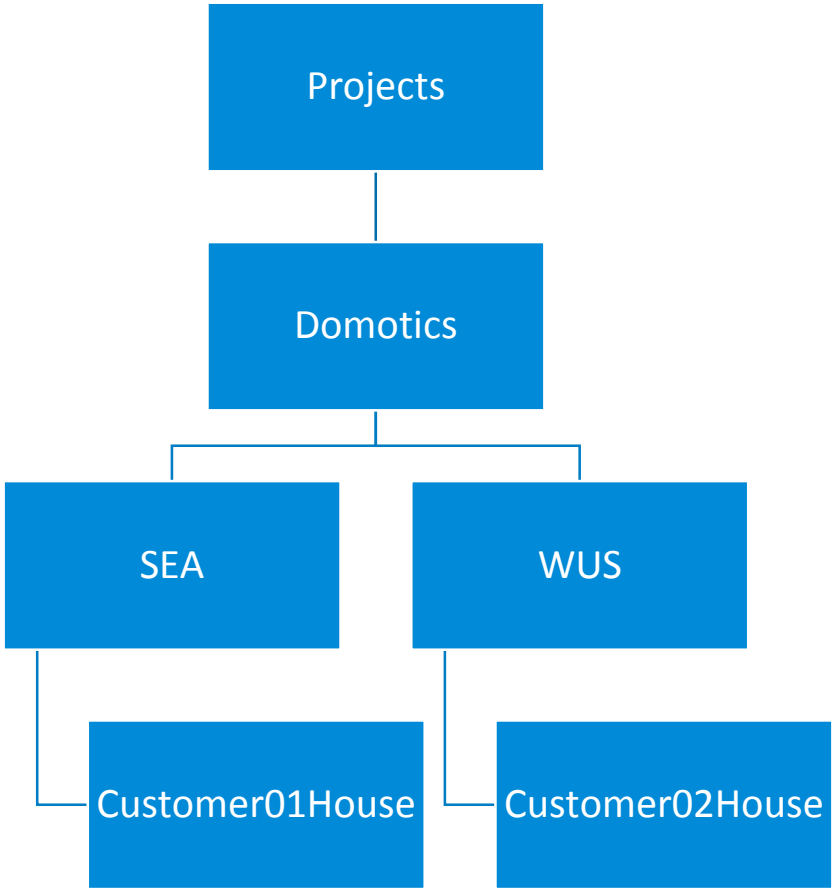
The experiment – domotics customers



Facility management

Under Domotics project space add a space for each customer

The experiment – multiregional domotics customers



If you have a multiregional business, you can introduce regional space

- Resources located under each region

Azure Digital Twins Graph Viewer

HTML+js app

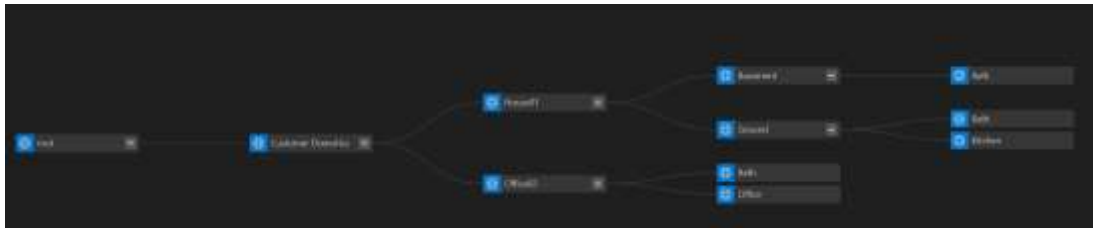
Stored in a Static Web Site

AAD authenticated

Azure SignalR

Azure Function AAD Authenticated for query and SignalR hub

https://github.com/Azure/azure-digital-twins-graph-viewer



The static website

Interacting with the API

Swagger for the API

Generate the code with NSwagStudio (for example)

- https://github.com/RSuter/NSwag/wiki/NSwagStudio

Pro

- Automatic
- Consistent

Cons

- Verbose
- Some semantics is lost (lists)

Query your devices in a multilevel tree

YOUR_MANAGEMENT_API_URL/devices?maxLevel=1 returns all devices attached to root spaces.

YOUR_MANAGEMENT_API_URL/devices?minLevel=2&maxLevel=4 returns all devices attached to spaces of levels 2, 3 or 4.

YOUR_MANAGEMENT_API_URL/devices?spaceId=mySpaceId returns all devices directly attached to mySpaceId.

YOUR_MANAGEMENT_API_URL/devices?spaceId=mySpaceId&traverse=Down returns all devices attached to mySpaceId or one of its descendants.

YOUR_MANAGEMENT_API_URL/devices?spaceId=mySpaceId&traverse=Down&minLevel=1&minRelative=true returns all devices attached to descendants of mySpaceId, excluding mySpaceId.

YOUR_MANAGEMENT_API_URL/devices?spaceId=mySpaceId&traverse=Down&minLevel=1&minRelative=true&maxLevel=1&maxRelative=true returns all devices attached to immediate children of mySpaceId.

YOUR_MANAGEMENT_API_URL/devices?spaceId=mySpaceId&traverse=Up&maxLevel=-1&maxRelative=true returns all devices attached to one of the ancestors of mySpaceId.

YOUR_MANAGEMENT_API_URL/devices?spaceId=mySpaceId&traverse=Down&maxLevel=5 returns all devices attached to descendants of mySpaceId that are at level smaller than or equal to 5.

YOUR_MANAGEMENT_API_URL/devices?spaceId=mySpaceId&traverse=Span&minLevel=0&minRelative=true&maxLevel=0&maxRelative=true returns all devices attached to spaces that are at the same level as mySpaceId.

IoTHub

Is a resource for the graph

You associate to a space

Use in descendant devices

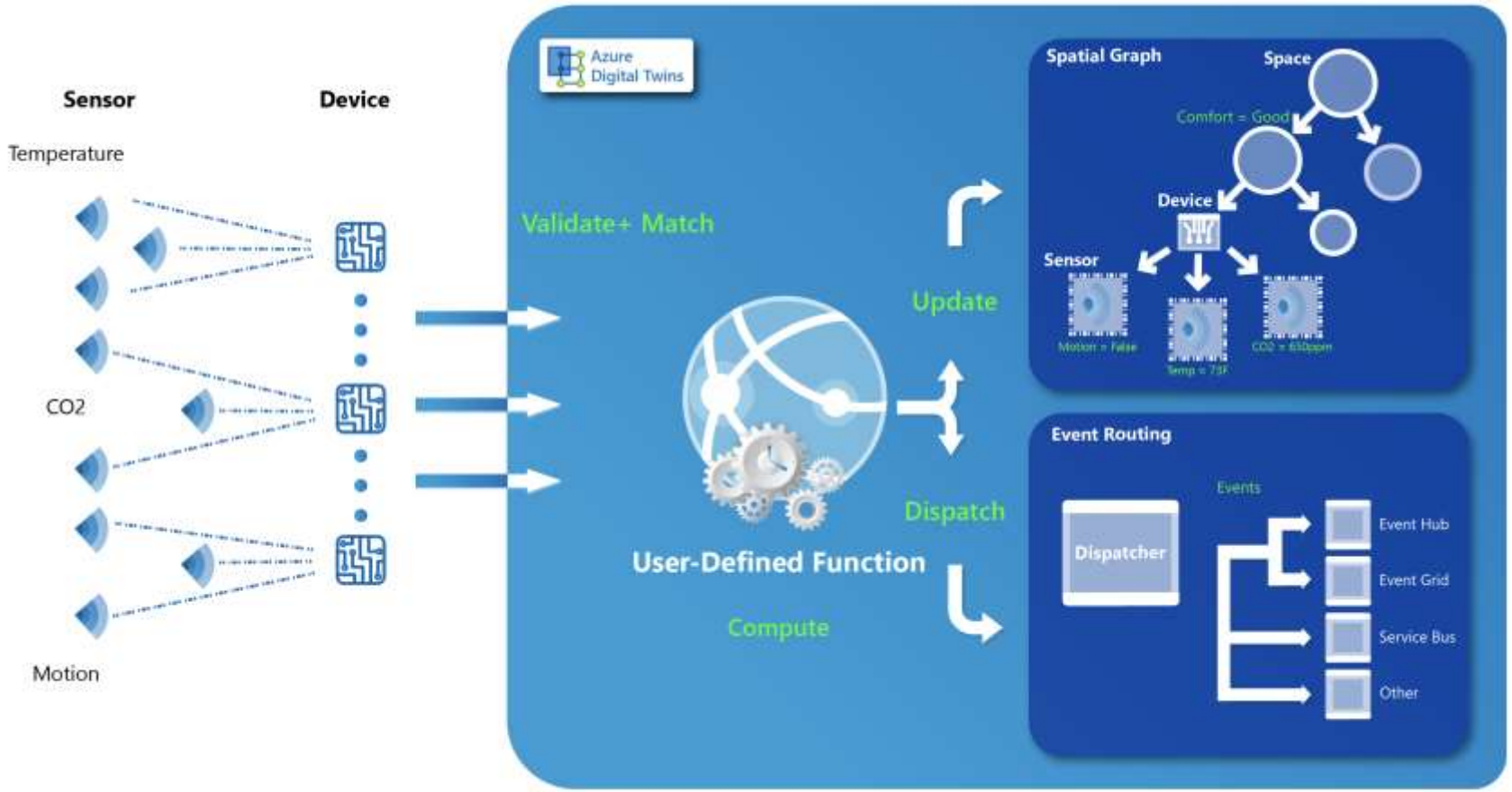
During the preview

- One per DT
- Auto provisioned, not external
- Hungs during deployment in West Europe (currently working in East US)

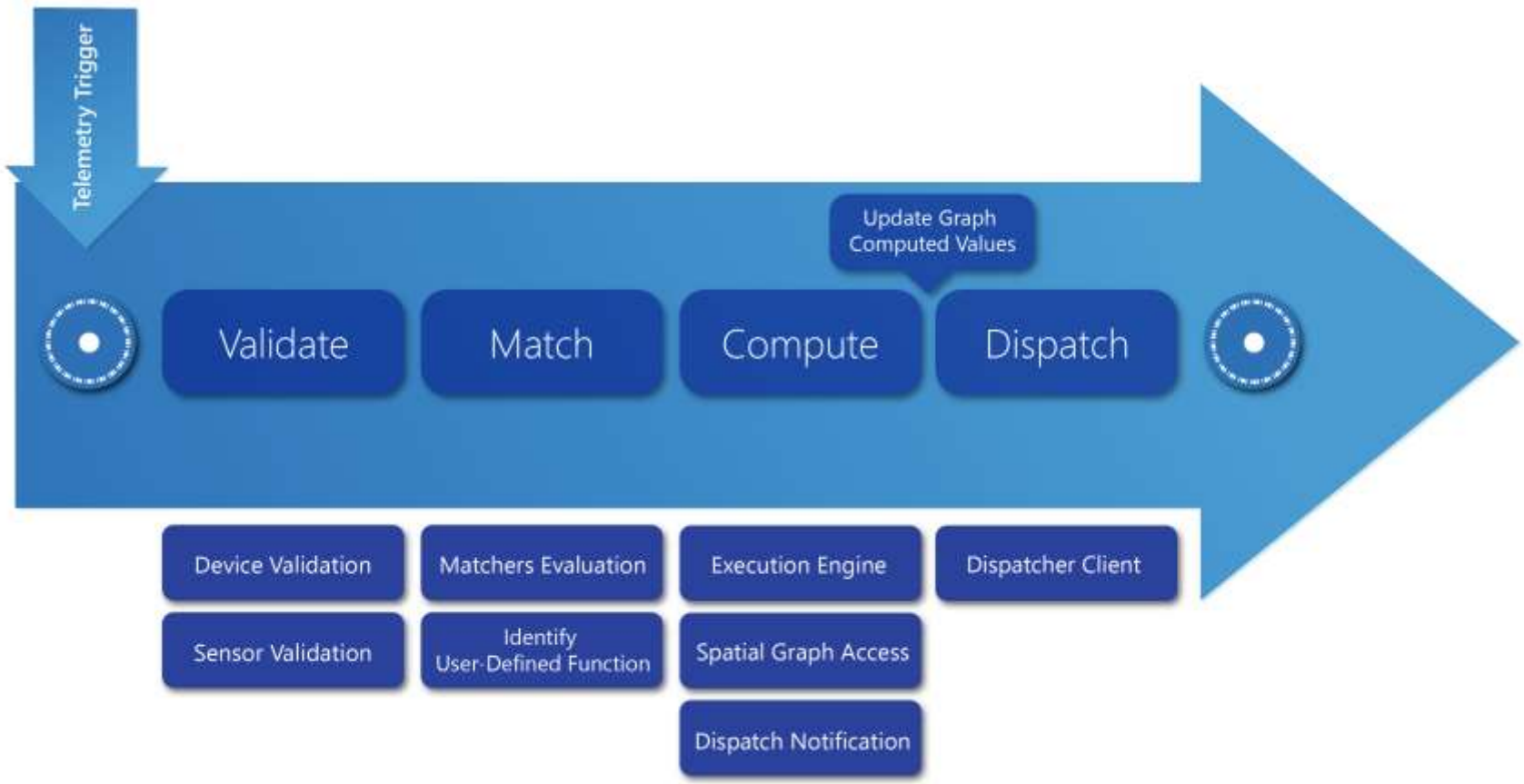
AddSpaces

The process of Digital Twins

The process of Digital Twins



The Digital Twin pipeline



Device/Sensor Validation


```
var creationTimeUtc = DateTime.UtcNow.ToString("o", Thread.CurrentThread.CurrentCulture);
var bytes = Encoding.UTF8.GetBytes($"{randomValue}");
var eventMessage = new Message(bytes);
eventMessage.Properties.Add("DigitalTwins-Telemetry", "1.0");
eventMessage.Properties.Add("DigitalTwins-SensorHardwareId", $"{sensor.HardwareId}");
eventMessage.Properties.Add("CreationTimeUtc", creationTimeUtc);
eventMessage.Properties.Add("x-ms-client-request-id", Guid.NewGuid().ToString());
eventMessage.Properties.Add("random-value", $"{randomValue}");
```

Property name	Value	Required	Description
DigitalTwins-Telemetry	1.0	Yes	A constant value that identifies a message to the system.
DigitalTwins-SensorHardwareId	string(72)	Yes	A unique identifier of the sensor that sends the **Message**. This value must match an object's **HardwareId** property for the system to process it. For example, 00FF0643BE88-CO2.
CreationTimeUtc	string	No	An ISO 8601 formatted date string that identifies the sampling time of the payload. For example, 2018-09-20T07:35:00.8587882-07:00.
CorrelationId	string	No	A UUID that's used to trace events across the system. For example, cec16751-ab27-405d-8fe6-c68e1412ce1f.

Data processing

A UDF in Javascript handles/process the value

Multiple UDFs can be defined

A set of matchers decide which UDF handles

Matchers

A sensor (and consequently all the tree branch where sensor is a leaf) is identified by DigitalTwins-SensorHardwareId property

So the matcher is relative ($) to the sensor

1. Path
2. Comparison (equal, contains, notequals)
3. Value (strings with escaped double quotes)
4. Target➔the UDF

Matcher scope…not tested

```
"target": "Sensor",
"path": "$.dataType",
"value": "\"Temperature\"",
"comparison": "Equals"
```

```
{
  "target": "SensorSpace",
  "path": "$.type",
  "value": "\"Venue\"",
  "comparison": "Equals"
}
```

```
"target": "Sensor",
"path": "$.port",
"value": "01",
"comparison": "Contains"
```

```
{
  "target": "SensorDevice",
  "path": "$.properties[?(@.name == 'Manufacturer')].value",
  "value": "\"GoodCorp\"",
  "comparison": "Equals"
},
```

The sensor state

Query for the sensors

Get the state

No DB, just sensor

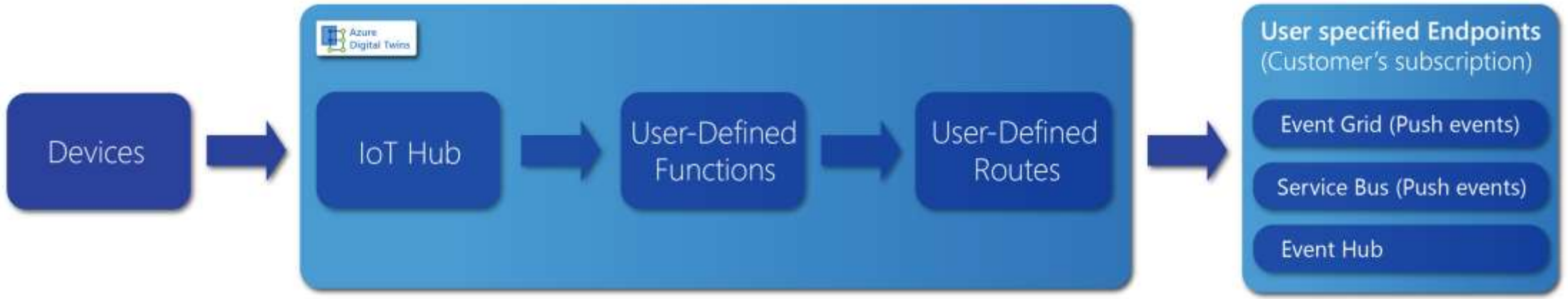
Remember?

SetTemperatureSensorValueUDF
SendFakeData
WatchData

Roles

Role	Description	Identifier
Space Administrator	*CREATE*, *READ*, *UPDATE*, and *DELETE* permission for the specified space and all nodes underneath. Global permission.	98e44ad7-28d4-4007-853b-b9968ad132d1
User Administrator	*CREATE*, *READ*, *UPDATE*, and *DELETE* permission for users and user-related objects. *READ* permission for spaces.	dfaac54c-f583-4dd2-b45d-8d4bbc0aa1ac
Device Administrator	*CREATE*, *READ*, *UPDATE*, and *DELETE* permission for devices and device-related objects. *READ* permission for spaces.	3cdfde07-bc16-40d9-bed3-66d49a8f52ae
Key Administrator	*CREATE*, *READ*, *UPDATE*, and *DELETE* permission for access keys. *READ*permission for spaces.	5a0b1afc-e118-4068-969f-b50efb8e5da6
Token Administrator	*READ* and *UPDATE* permission for access keys. *READ* permission for spaces.	38a3bb21-5424-43b4-b0bf-78ee228840c3
User	*READ* permission for spaces, sensors, and users, which includes their corresponding related objects.	b1ffdb77-c635-4e7e-ad25-948237d85b30
Support Specialist	*READ* permission for everything except access keys.	6e46958b-dc62-4e7c-990c-c3da2e030969
Device Installer	*READ* and *UPDATE* permission for devices and sensors, which includes their corresponding related objects. *READ* permission for spaces.	b16dd9fe-4efe-467b-8c8c-720e2ff8817c
Gateway Device	*CREATE* permission for sensors. *READ* permission for devices and sensors, which includes their corresponding related objects.	d4c69766-e9bd-4e61-bfc1-d8b6e686c7a8

Routing events and messages



	DeviceMessages	TopologyOperation	SpaceChange	SensorChange	UdfCustom
EventHub	X	X	X	X	X
ServiceBus		X	X	X	X
EventGrid		X	X	X	X

AddEndpoints
GetTopologyChanges
ServiceBus
SignalR

Conclusions

It’s state

It’s not a database, but it has query semantics

It has declarative semantics as a rule-based engine

It can be infinite scalable using one only services rather then more than one (AKS or SF or Redis)

It’s a first preview

- No resource management
  - Scalability not evaluable
- Define better integration with IoT Hub and offering
  - Geo/Manual Failover
  - DPS
- IoT Edge semantics missing (device pipeline)
- Difficult server-side pipeline debugging
  - JavaScript-only
  - Not observable

What is a Graph Database

A Graph Database is a Database that is modelled as a Graph

Traditional data modeling focuses on entities.

For many applications, there's also a need to model both entities and relationships naturally.

Both vertices and edges can have an arbitrary number of properties.

https://www.slideshare.net/marco.parenzan/graph-databases-in-the-microsoft-ecosystem

Thanks

Questions?


marco.parenzan


marco_parenzan


marcoparenzan

#GlobalAzure

#CloudGenVerona

@cloudgen_verona