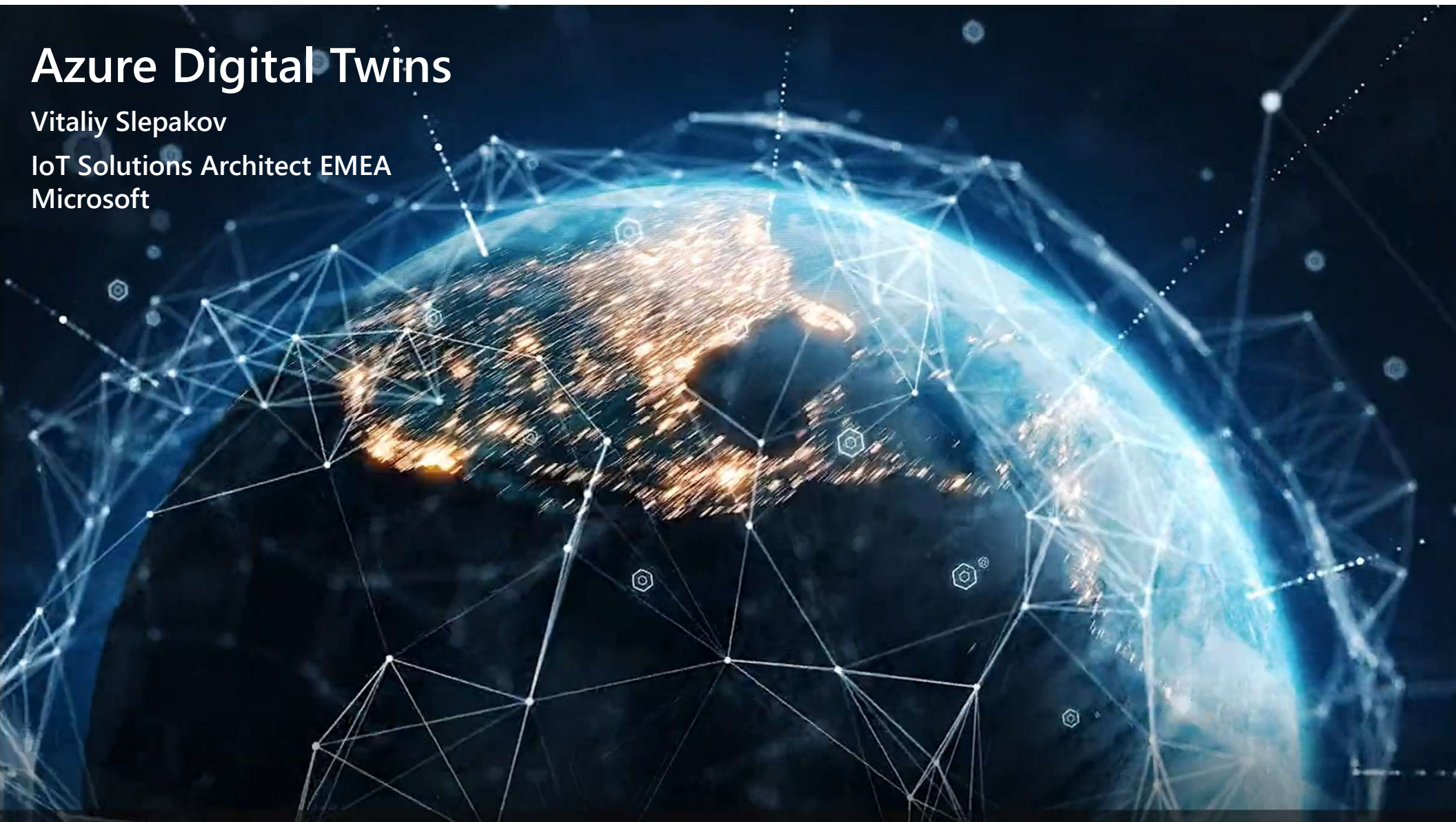


Azure Digital Twins

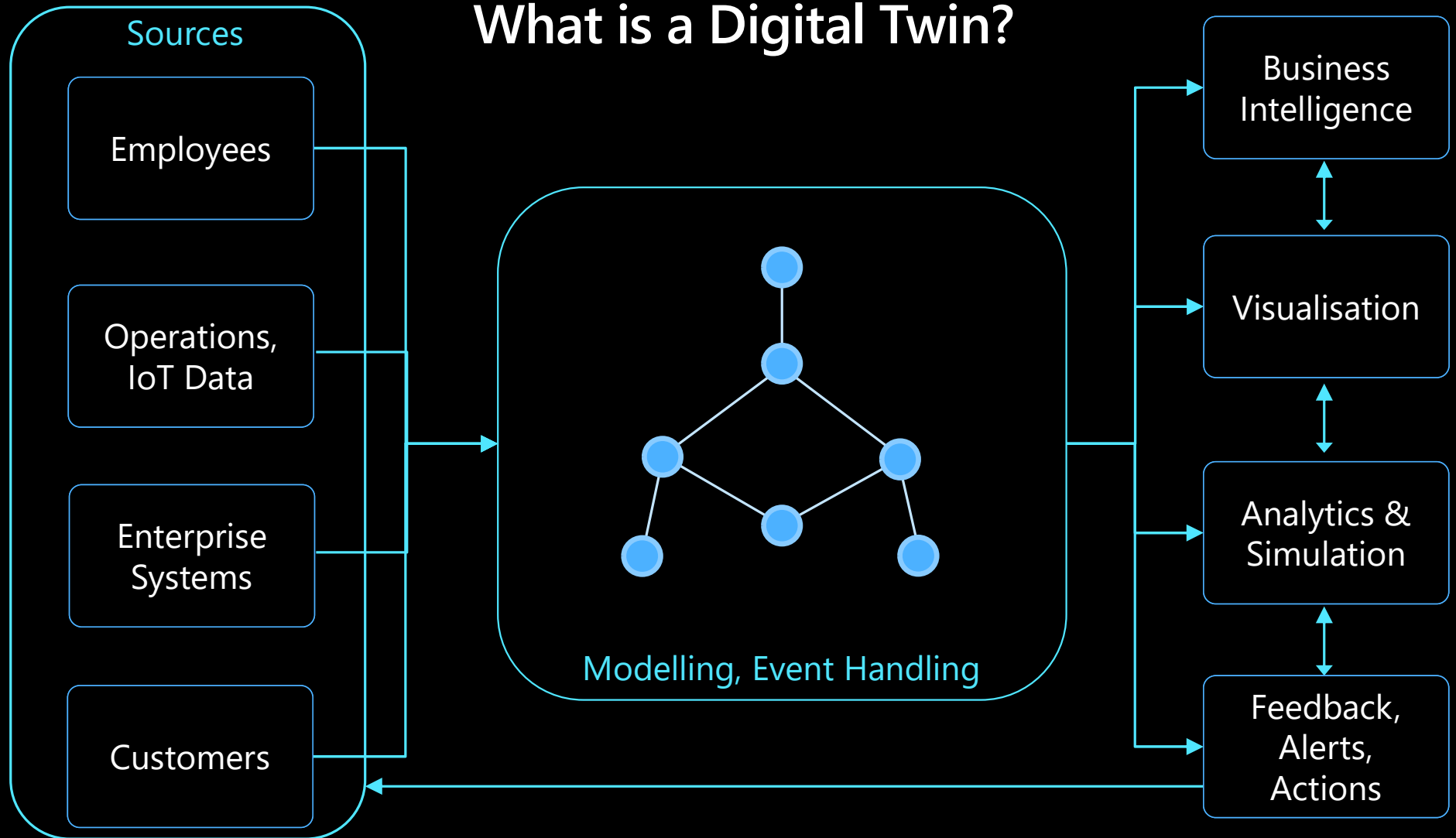
Vitaliy Slepakov

IoT Solutions Architect EMEA

Microsoft



What is a Digital Twin?



Obstacles for building a digital twin.



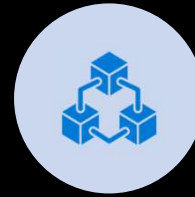
COMPLEX MODELS OF
ENVIRONMENTS



SILOED DATA ACROSS
ECOSYSTEM



COMBINING SEVERAL
BUILDING BLOCKS

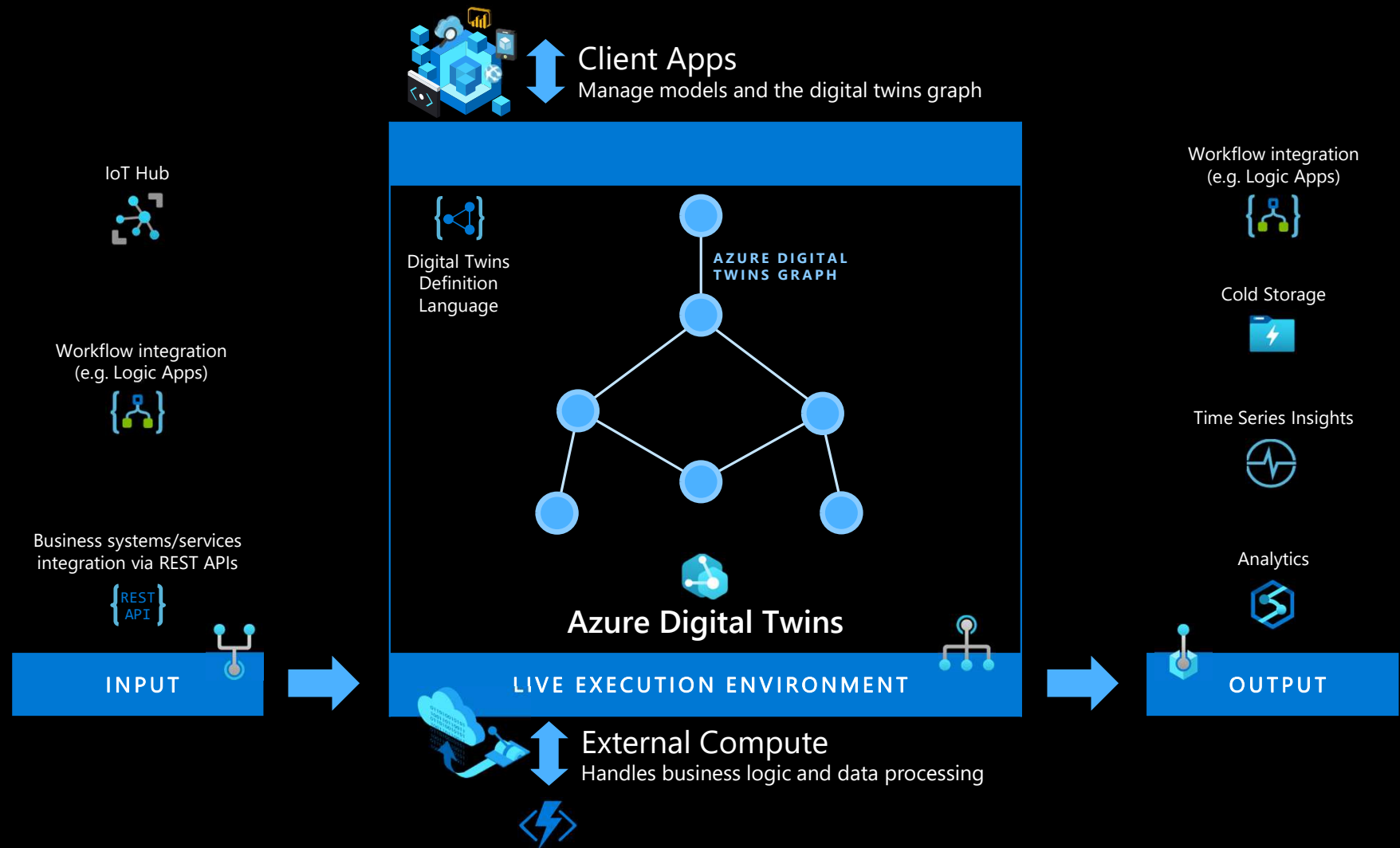


CREATING A
SCALABLE AND
SECURE DIGITAL TWIN



STAYING SYNCED
WITH REAL WORLD
UPDATES

Create next generation IoT solutions that model the real world



Journey to Digital Twin Solutions

ADT Journey:

MODELLING THE DIGITAL TWIN

IDENTIFY AND CONNECT TO
DATA SOURCES

BUILD BUSINESS LOGIC AND
INTEGRATE WITH BUSINESS
SYSTEMS

Operational Systems

Business Systems

Modeling

Driving Insights

Business Systems
&
Visualization

IT Infrastructure & Operations

Modeling the Digital Twin

MODELLING THE DIGITAL TWIN

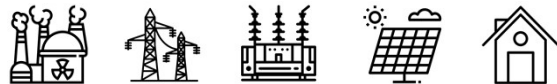
IDENTIFY AND
CONNECT DATA
SOURCES

BUILD BUSINESS LOGIC
AND INTEGRATE WITH
BUSINESS SYSTEMS

Modeling with DTDL

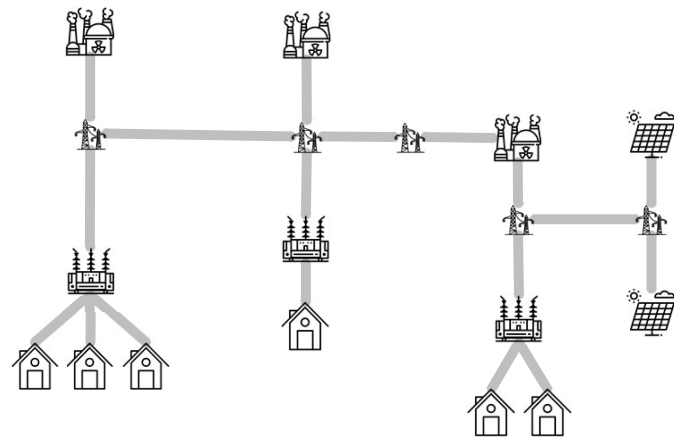
Create a domain vocabulary

- Describe the entities and concepts important for your business
- Describe how entities relate and connect to each other
- Describe data and behavior
- Digital Twins Definition Language (DTDL)
 - Open source specification
 - Programming language independent
 - Based on JSON-LD
- DTDL is also used to describe IoT devices
 - Aligned with IoT Plug and Play and Time Series Insights data model
 - Enables Plug and Play connectivity for device
 - Consistent programming model from ADT in the cloud to devices



Build a model of your environment

- Create instances of the specific entities in your real world
- Connect the instances into a topology graph that represents your environment
- Define event processing and routing for your environment



Simple DTDL Example

```
{
  "@id": "dtmi:com:example:MyModel;1",
  "@type": "Interface",
  "contents": [
    {
      "@type": "Property",
      "name": "name",
      "schema": "string"
    },
    {
      "@type": "Telemetry",
      "name": "temperature",
      "schema": "double"
    },
    {
      "@type": "Command",
      "name": "update"
    }
  ],
  "@context": "dtmi:dtdl:context;2"
}
```

- An interface with id `dtmi:com:example:MyModel;1`
 - A “name” property
 - A “temperature” telemetry
 - An “update” command

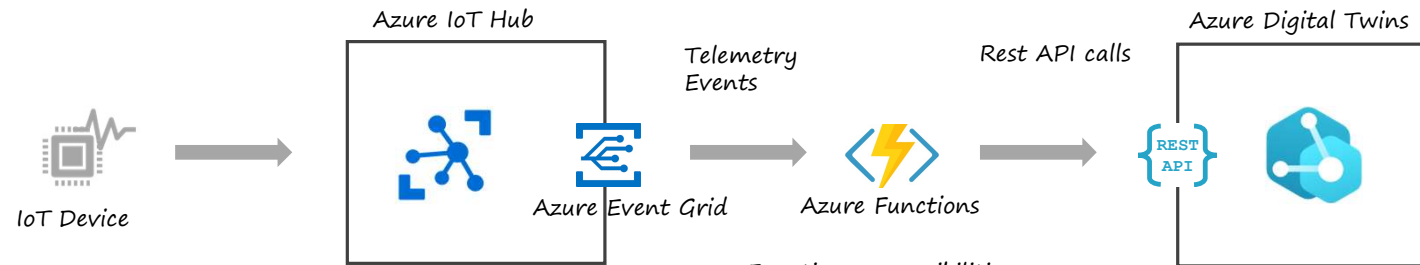
Connecting to Data Sources

MODELLING THE DIGITAL TWIN

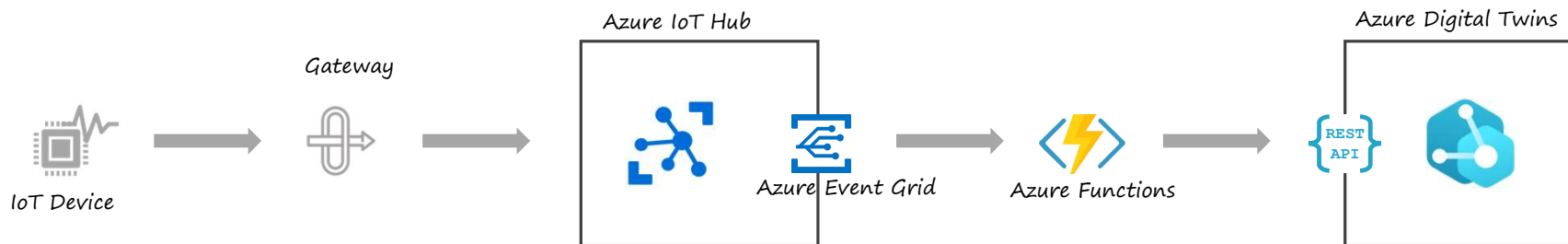
IDENTIFY AND
CONNECT TO DATA
SOURCES

BUILD BUSINESS LOGIC
AND INTEGRATE WITH
BUSINESS SYSTEMS

Basic Ingress from IoT Hub

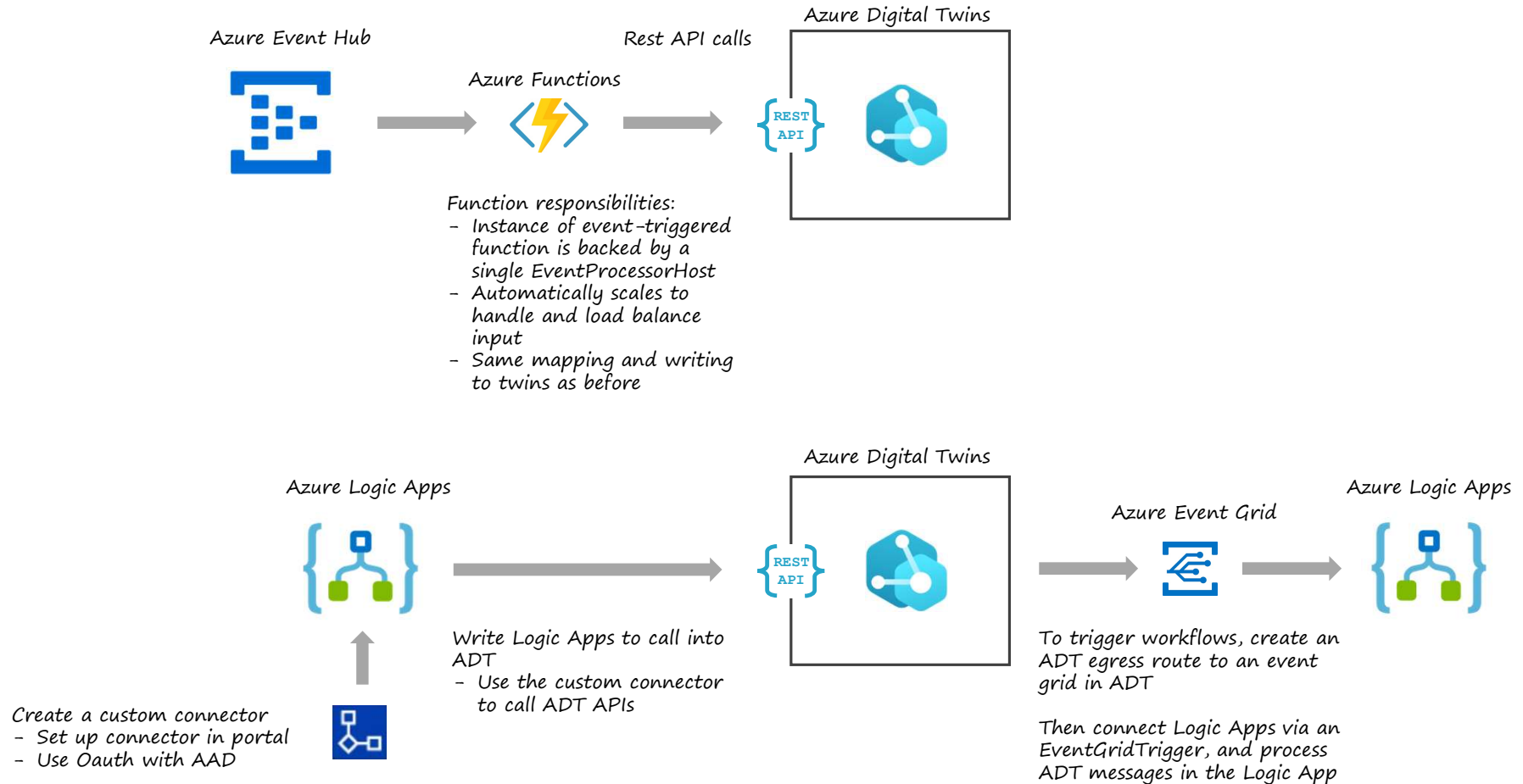


- Function responsibilities:
- Decode the telemetry messages from IoT Hub
 - Identify the twin to write to, typically using the device id as a driver for mapping
 - Transform the event as desired
 - Write to one or more twins



Typically use a message property to map to a target twin, as opposed to a Hub device id

Basic Ingress from other Sources: EventHub or Logic Apps





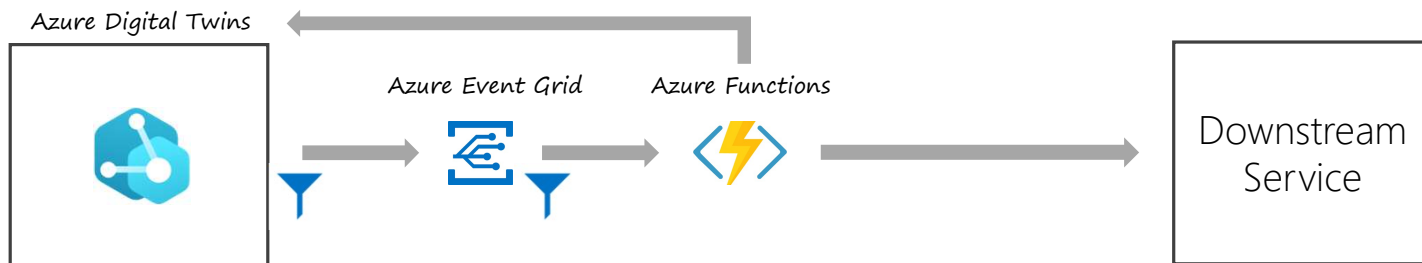
Integration, Analytics & Visualization

MODELLING THE DIGITAL TWIN

IDENTIFY AND
CONNECT DATA
SOURCES

BUILD BUSINESS LOGIC
AND INTEGRATE WITH
BUSINESS SYSTEMS

Basic Event Processing



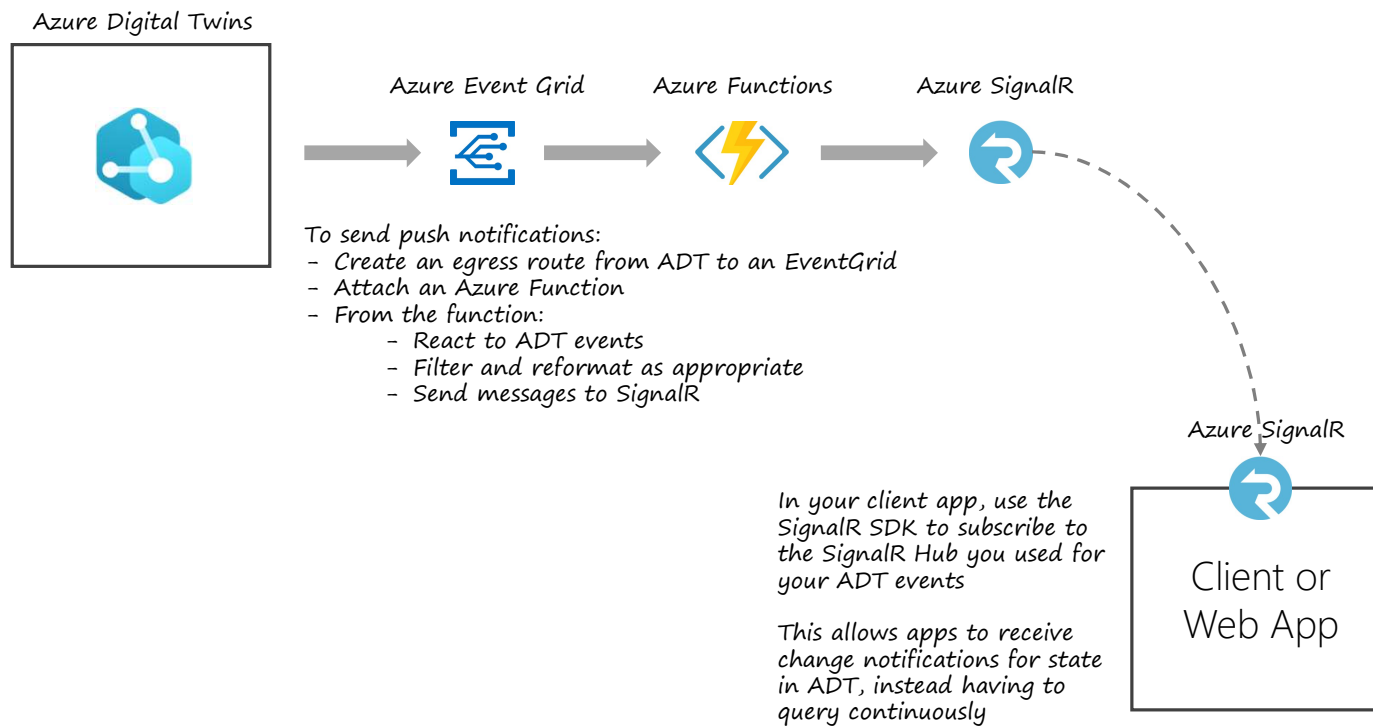
To process data with ADT:

- Create an egress route from ADT to an EventGrid
- Use route filters to select the desired events
- If desired, use event grid filters for further filtering
- Process events in an Azure Function
- Write results back into a twin
 - Use case example: Calculate an updated average temperature on a floor whenever any of the connected sensors changes
- Write results to downstream services
 - Examples: Storage, analytics, machine learning, etc.

Note:

- The compute resource used does not need to be an Azure Function. It can be any compute resource available.

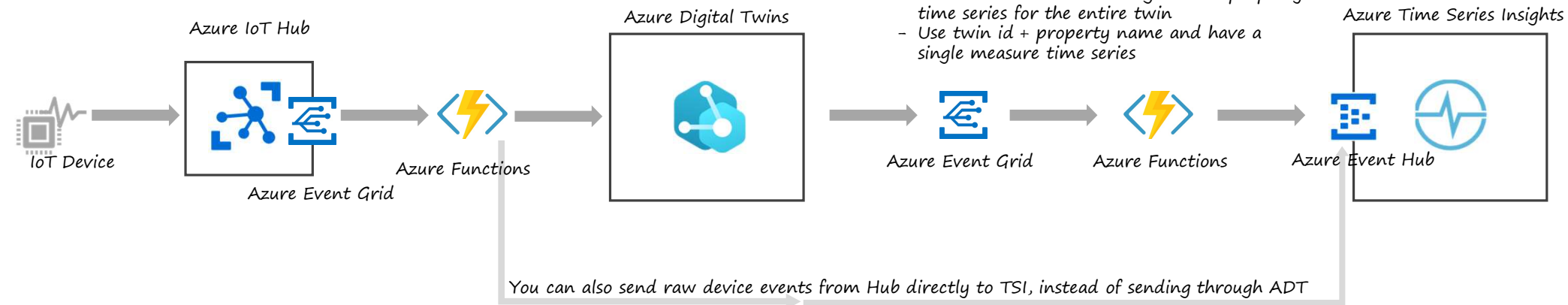
Push Notifications from ADT with SignalR



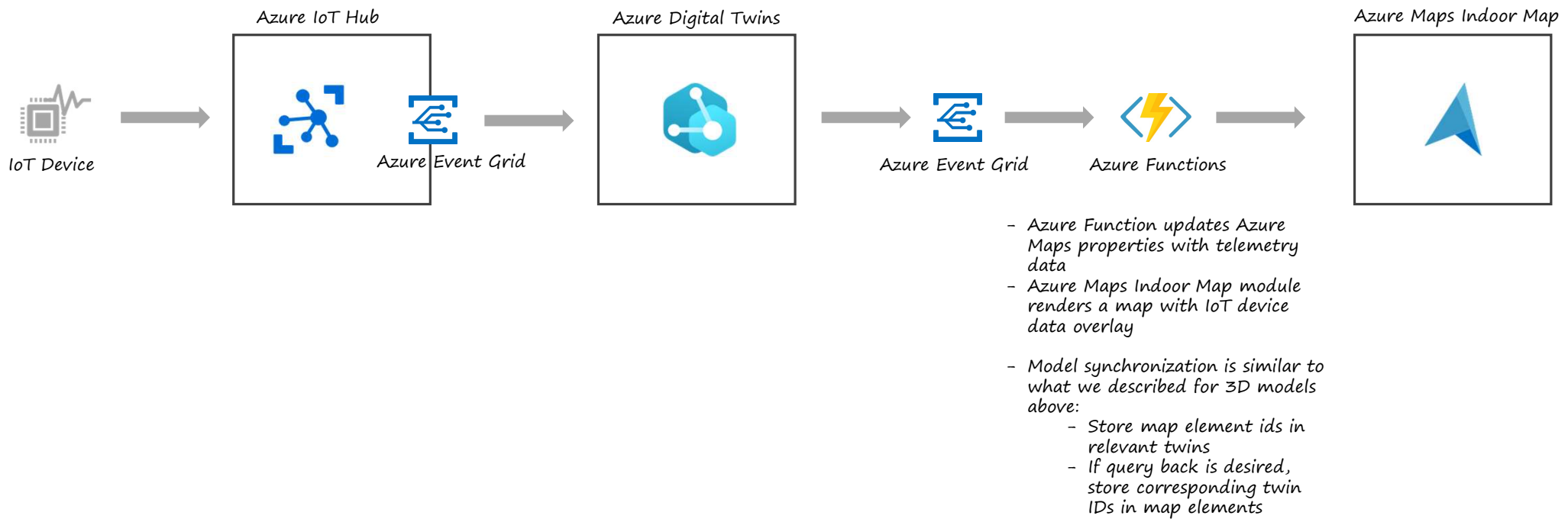
Connecting to TSI

To send data to TSI (or another time series database)

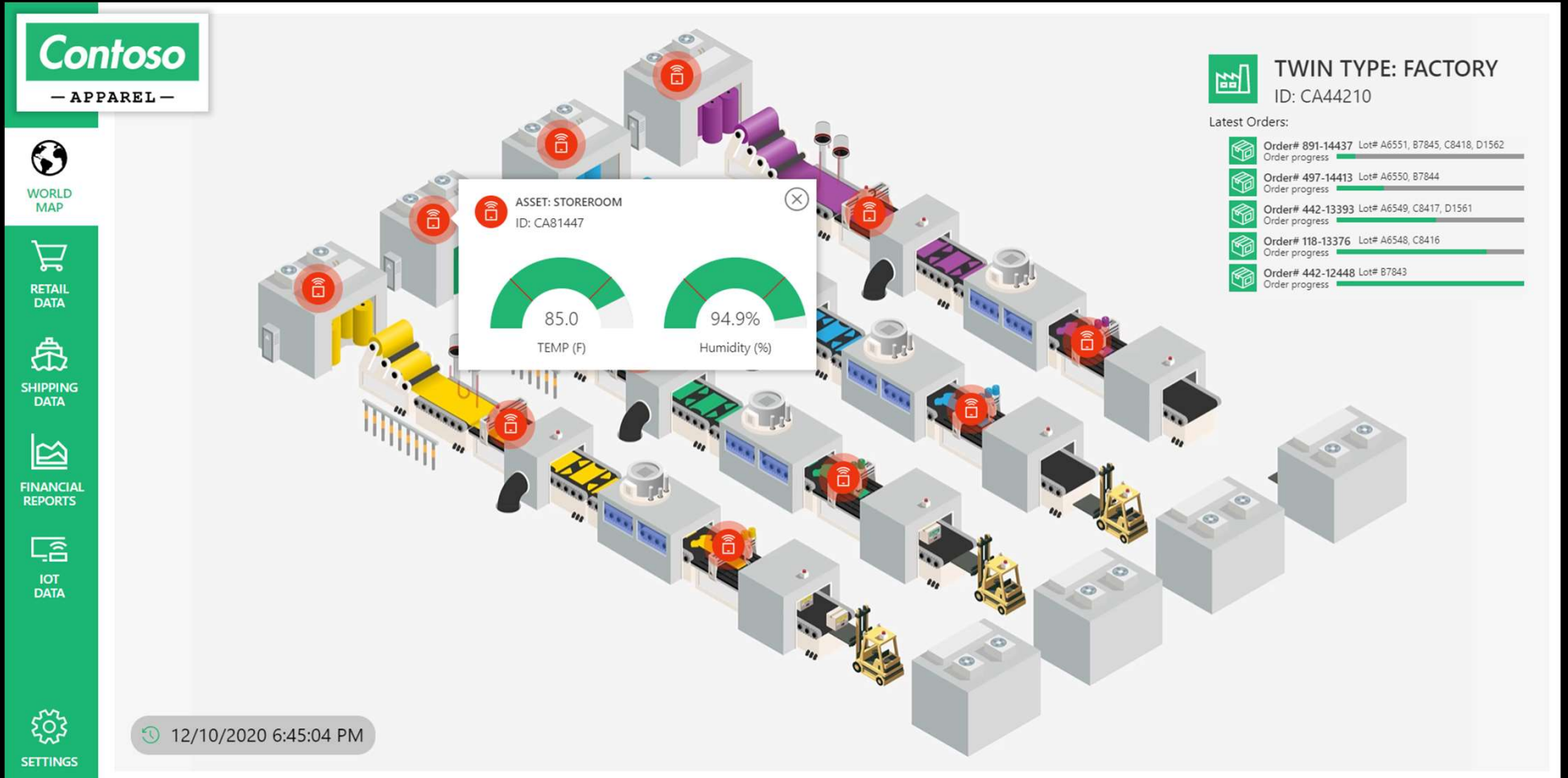
- Create an egress route from ADT to an EventGrid
- Attach an Azure Function
- From the function:
 - Filter and reformat ADT events as appropriate
 - Push messages into TSI via the EventHub SDK
- For time series ID:
 - Use twin ID and have a single multi-property time series for the entire twin
 - Use twin id + property name and have a single measure time series



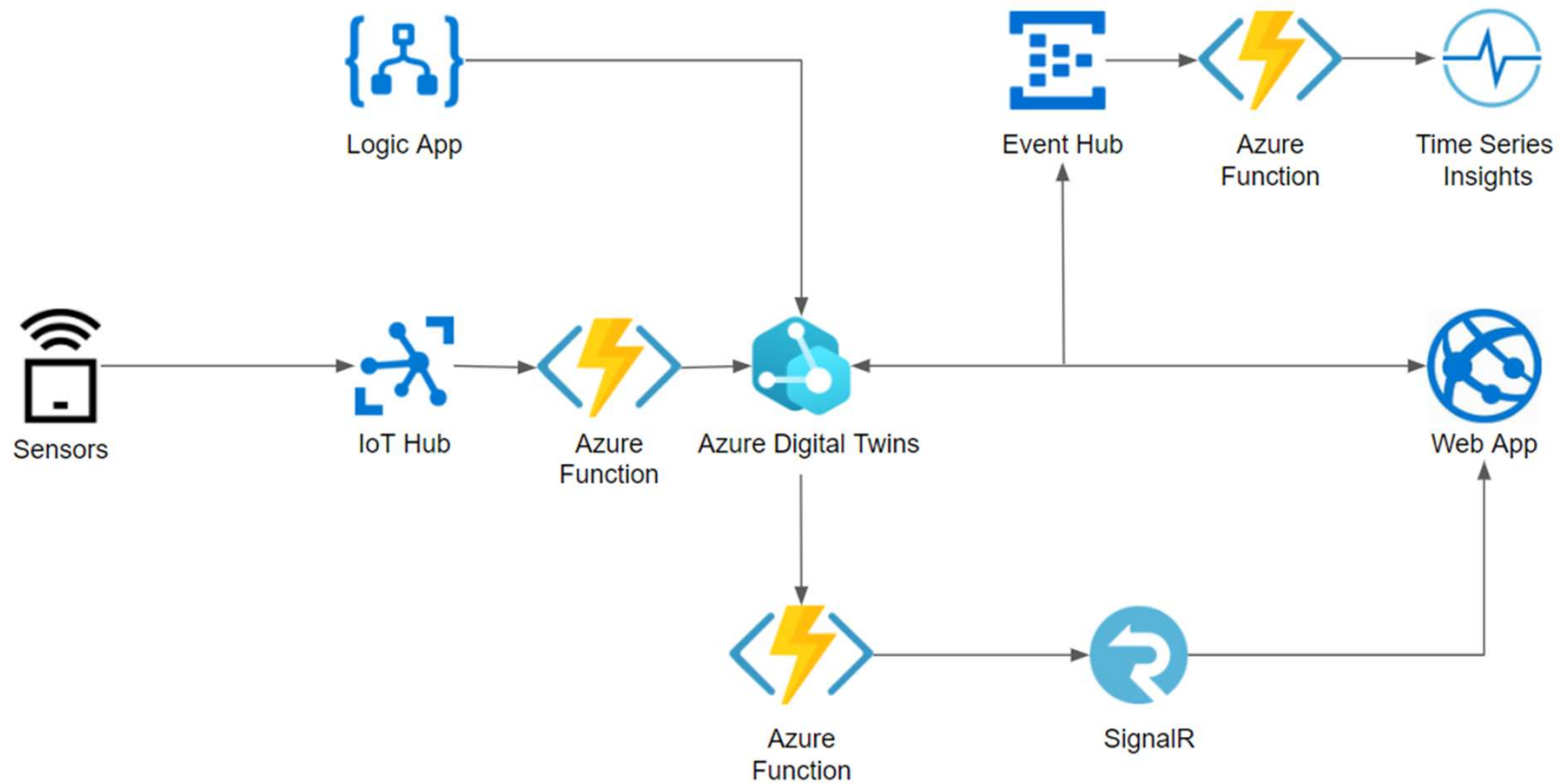
Connecting to Azure Maps Indoor Map



DEMO



DEMO





Resources

• Tools

- [Model Uploader, Model Visualizer](#)
- [ADT Explorer](#)
- [DTDl Validator](#)
- [az dt](#)
- [DTDl VS Code Extension](#)

• Learning and Demos

- [Supply Chain](#)
- [Chocolate Manufacturing Factory](#)
- [E2E Demo](#)
- [ADT at MS Learn](#)

• Ontologies

- [RealEstateCore smart building ontology](#)
- [Smart cities ontology](#)



Thank you!
Q&A

DTDL Basics

- Interface
 - Digital twin type
- Property
 - State of a digital twin
 - Read only or writable
 - Implies synchronization for remote entities
- Telemetry
 - Measurement of a digital twin
 - Implies timestamped stream of measurements
- Command
 - Method on a digital twin
 - Implies "time to start" and "time to complete" for remote entities or async operations
- Relationship
 - Reference type containment in a digital twin
 - 1:n collection
 - Circular references allowed
 - Relationships support properties
- Component
 - Value type containment in a digital twin
 - No circular references
- Schema
 - Data type definition
 - Standard primitive schemas provided for strings, numbers, dateTime, etc.
 - Complex schema types for arrays, enums, objects, and maps
- Inheritance
 - Interface subclassing
 - Twin instances have exactly one interface type
 - Interfaces can extend multiple other interfaces
- Semantic Type
 - Annotations on DTDL elements (additional "semantics")
 - Example: measurement type and unit

Build Connected Environments with Azure Digital Twins

- Accelerate time to value
- Simplify the creation of comprehensive digital models
- Track the past, simulate possibilities and help predict the future of any connected environment
- Break down silos within connected environments
- Build on a trusted enterprise-grade platform

