



# TUGA IT

## SUMMER EDITION

LISBON, JULY 19-21, 2018

# THANK YOU TO OUR SPONSORS



Microsoft

GOLD SPONSOR

**bi4all**

CREATING BUSINESS INTELLIGENCE

SILVER SPONSOR

|create|**it**|

INNOVATING **LIFE**

TUGA BEER SPONSOR

**FARFETCH**

SWAG SPONSOR



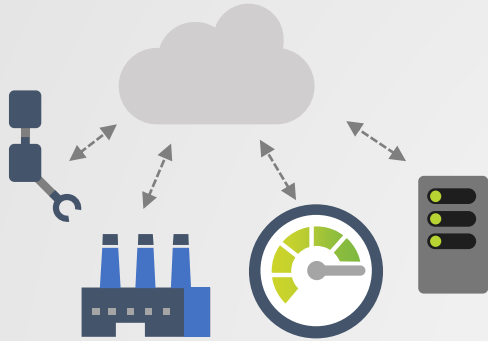


# Deploy IoT Edge in 45 minutes (and live happy)

**Marco Dal Pino**  
Microsoft WinDev MVP  
Intel® Innovator & BlackBelt  
Program Manager  
Technical Evangelist

@MarcoDalpino  
[About.me/marcodalpino](https://about.me/marcodalpino)

# IoT in the Cloud and on the Edge



## IoT in the Cloud

Remote monitoring and management

Merging remote data from multiple IoT devices

Infinite compute and storage to train machine learning and other advanced AI tools



## IoT on the Edge

Low latency tight control loops require near real-time response

Pre-process data on prem. – E.g. video streams

Intelligence at the edge – ML, AI, Analytics

Offline operations (short and long term)

Protocol translation & data normalization

Privacy of data and protection of IP

Consistency

# Azure IoT Edge has three main operating mode:

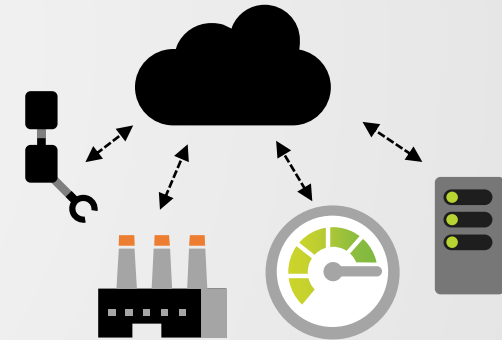
## The boardroom view

- Aggregate analysis and data from many applications (supply chain, operation, HR)
- Take system wide actions across enterprise functions
- Integrate with partners and affiliates



## The supervisory view

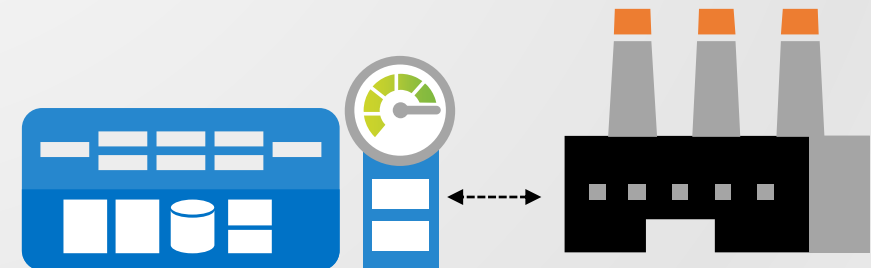
- Aggregate data and alerts from many sites and take enterprise-wide actions
- Train intelligence based on data from many sources
- Manage and update IoT Edge from cloud



## The local view

Local data, local control, local analytics and intelligence

- Send alerts and data digests to cloud
- Operate in offline state (partially or completely)



# Design



# 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, provides 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

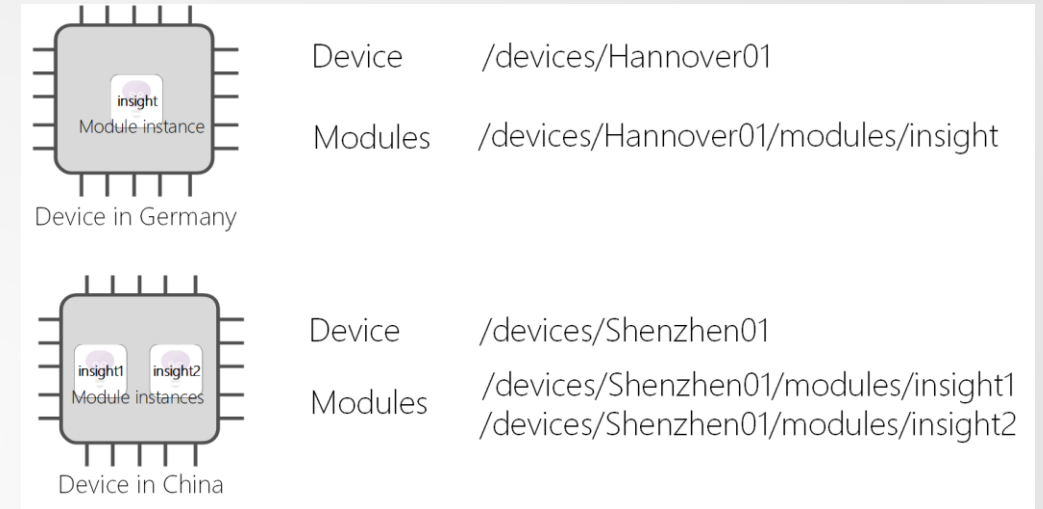
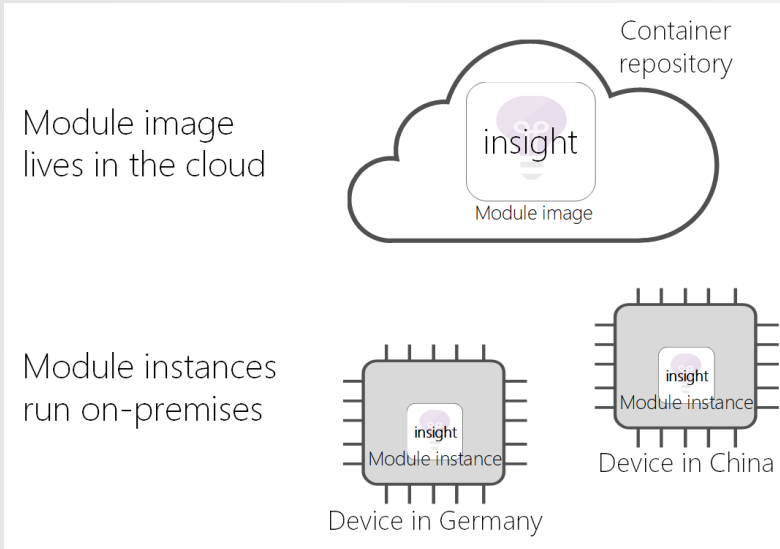
Enables seamless deployment of advanced capabilities such as AI from Microsoft, and any third party, today and tomorrow

# Azure IoT Edge deployment is made up of:

- **Modules:** containers that run Azure services, 3rd party services, or your own code. They are deployed to IoT Edge devices and execute locally on those devices.
- **Runtime:** runs on each IoT Edge device and manages the modules deployed to each device.
- **A cloud-based interface:** enables you to remotely monitor and manage IoT Edge devices.

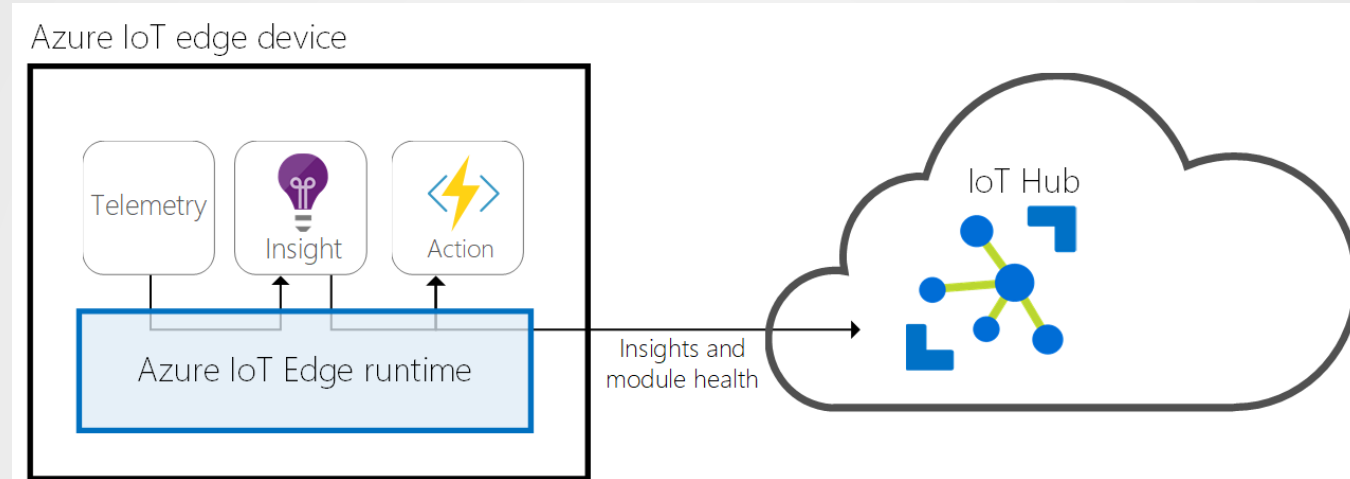


# Concept – Module



- A **module image** is a package containing the software that defines a module.
- A **module instance** is the specific unit of computation running the module image on an IoT Edge device. The module instance is started by the IoT Edge runtime.
- A **module identity** is a piece of information (including security credentials) stored in IoT Hub, that is associated to each module instance.
- A **module twin** is a JSON document stored in IoT Hub, that contains state information for a module instance, including metadata, configurations, and conditions.
- SDKs to develop custom modules in multiple languages (C#, C, Python, Java, Node.JS)

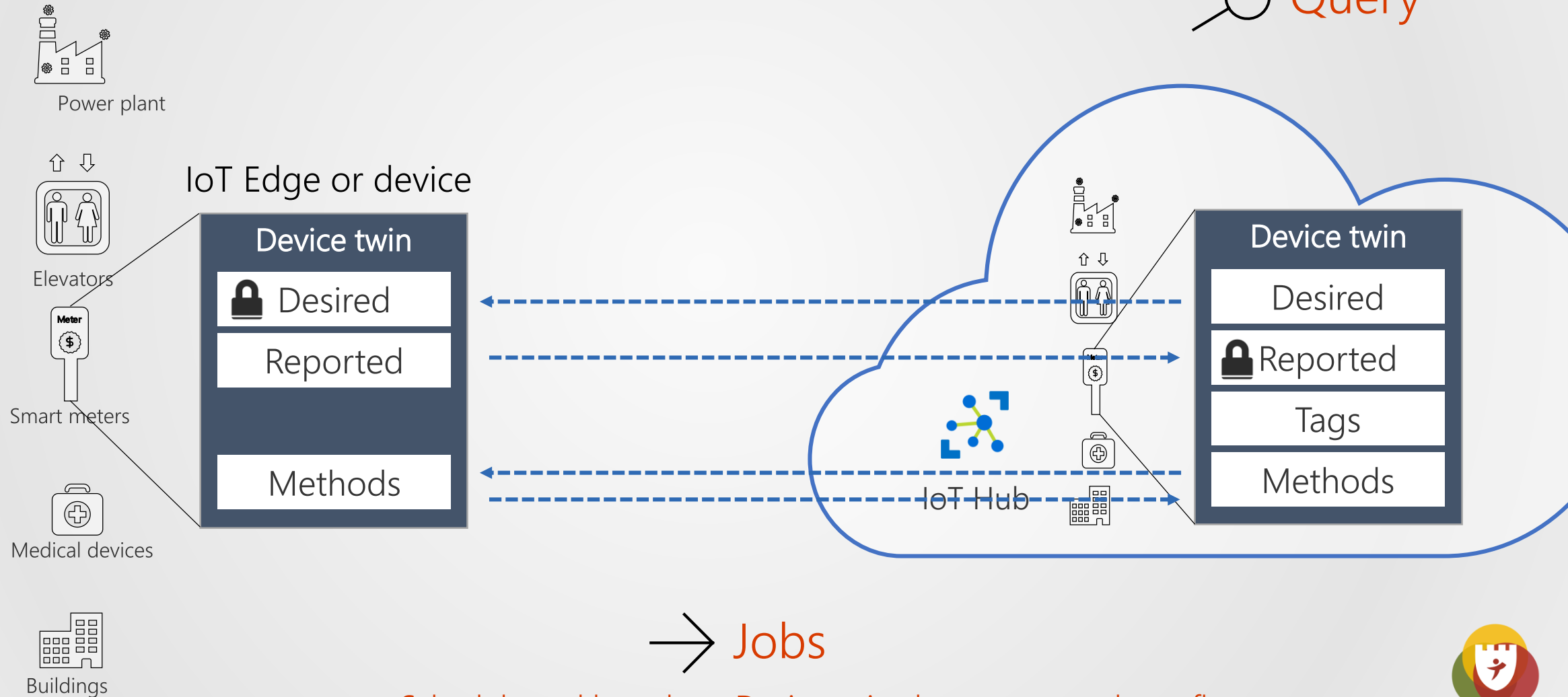
# Concept – Azure IoT Edge Runtime



- Installs and updates workloads on the device.
- Maintains Azure IoT Edge security standards on the device.
- Ensures that IoT Edge modules are always running.
- Reports module health to the cloud for remote monitoring.
- Facilitates communication between downstream leaf devices and the IoT Edge device.
- Facilitates communication between modules on the IoT Edge device.
- Facilitates communication between the IoT Edge device and the cloud

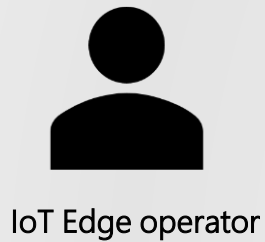
# Concept – Device Management

 Query



Schedule and broadcast Device twin changes across large fleets

# IoT Edge in action



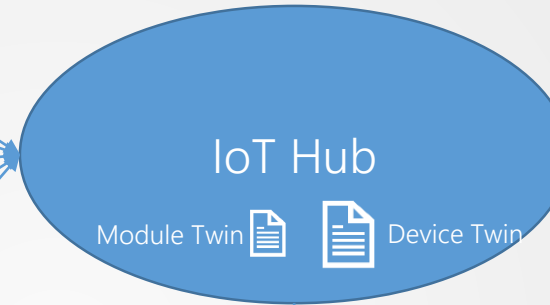
1 – Edge device provisioned with right agents for scenario

2 – Select Edge node to deploy to

3 – Define modules on Edge node via device twin

4 – Define message routes for modules on edge node via device twin

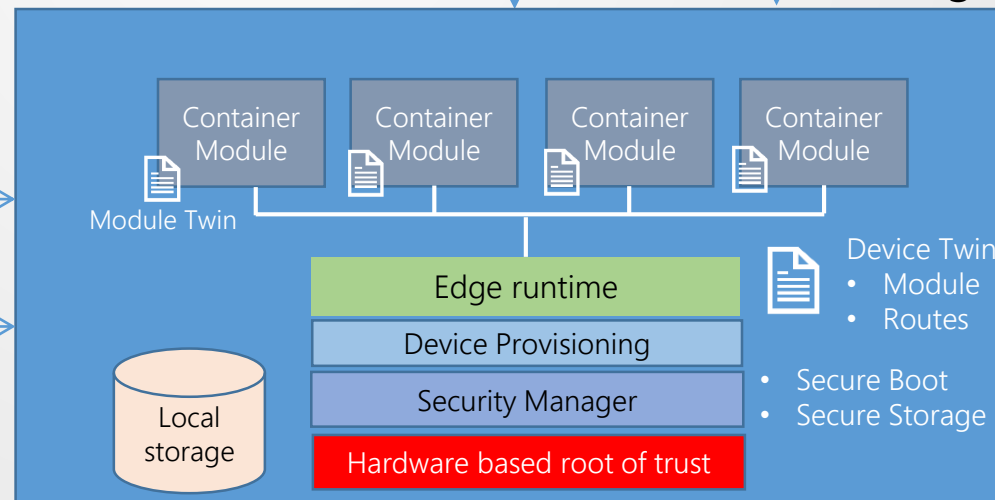
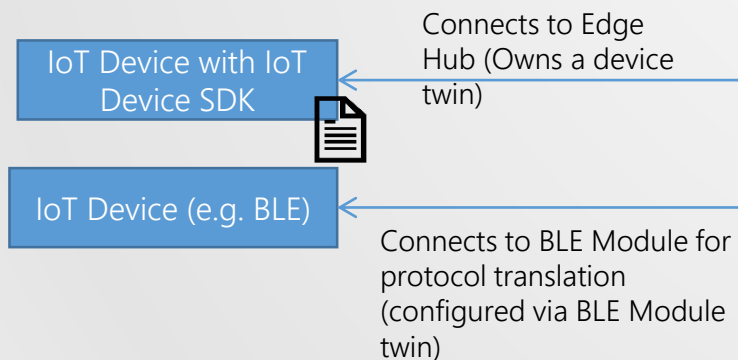
5 – Define Module twins for module configurations (parameters)



- Container based workloads
- AI Services
- Azure Functions
- Azure Stream Analytics
- Azure Machine Learning
- Your own code using module SDK

Container Modules

IoT Edge



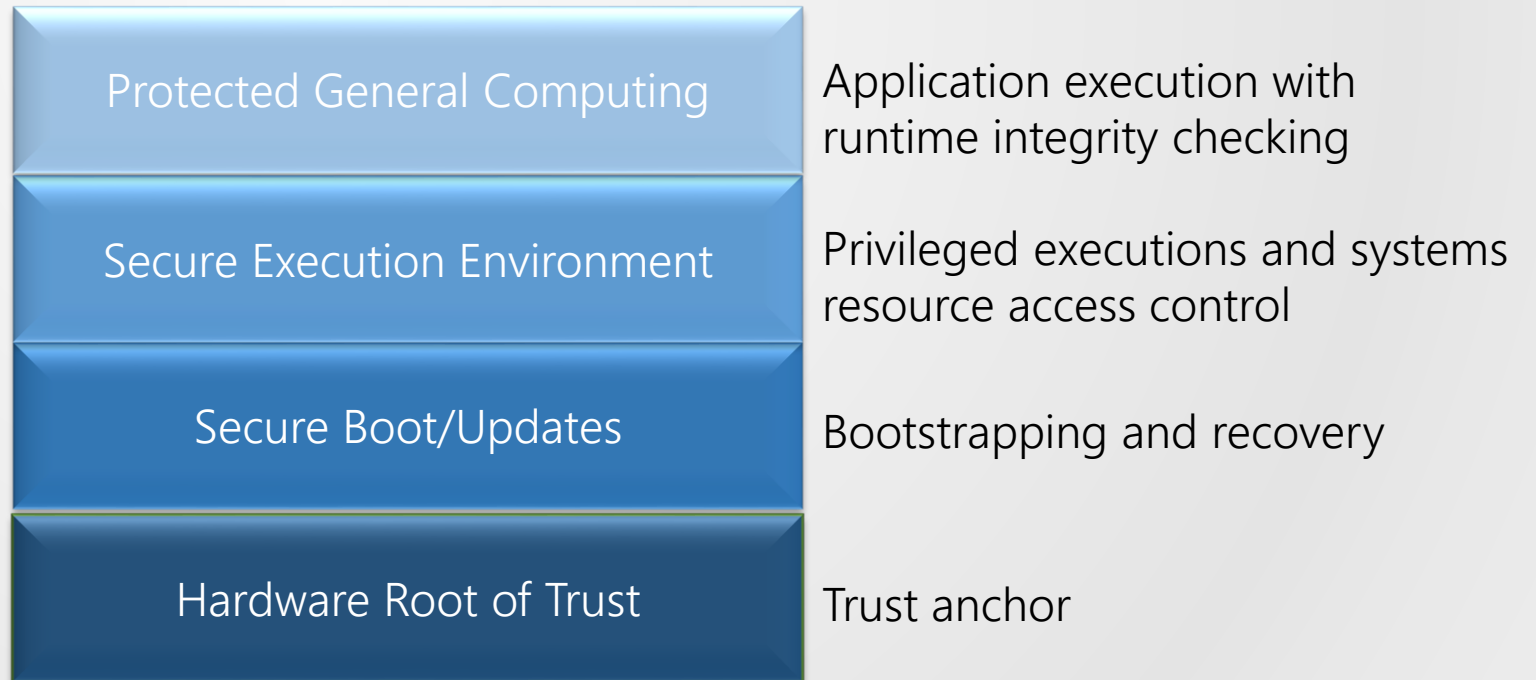
- Edge device with security requirements
- Rich OS – Linux or Windows
- Docker-compatible container management system



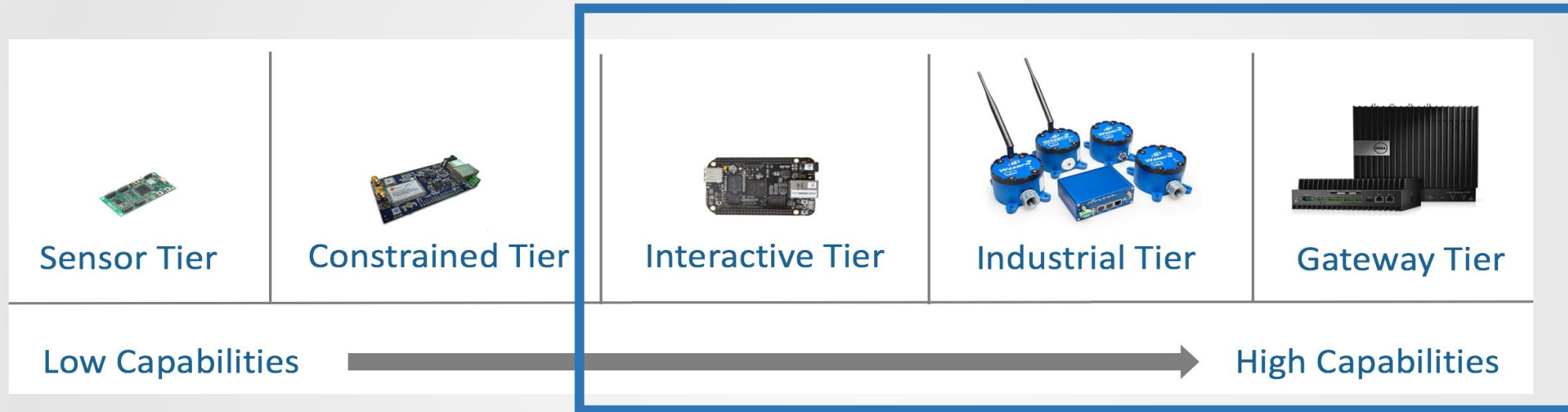
# Security

## Principles and Goals

- Cross Platform (Linux, Windows, multiple architectures)
- Standardized Protocols
- Secure technology isolation from app developer
- Availability of technology



# Hardware for Azure IoT Edge



Linux and Windows supported on x64 and ARM (support for containers required)

Hardware sizing dependent on workloads

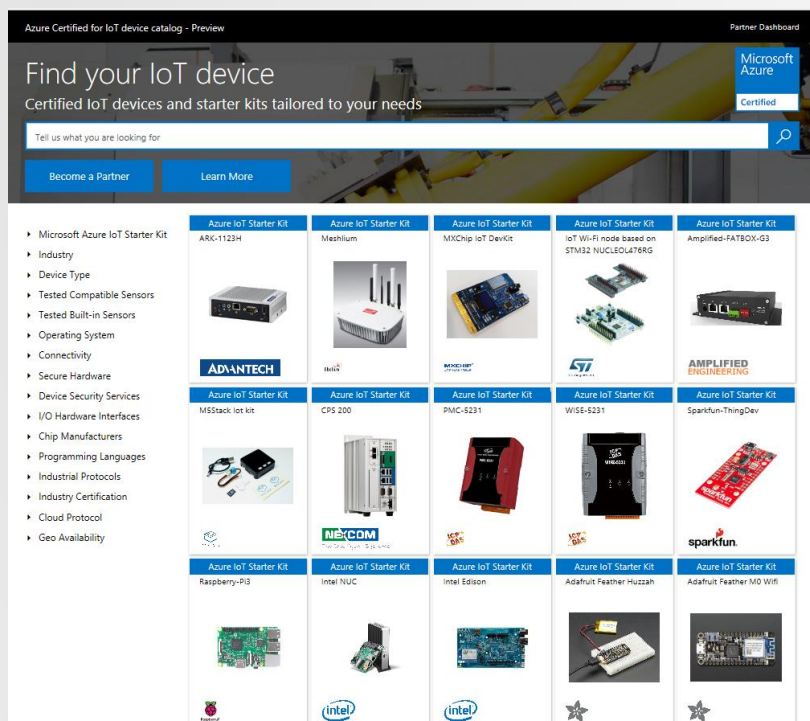
Internal tests on devices as small as Raspberry Pi 3 with four containers (on Linux)

# Certified hardware for Azure IoT Edge

## Azure Certified for IoT Device Catalog

- Provides an easy and intuitive way to discover the right IoT device for intended use case
- More than 1000 devices already listed in the device catalog
- [catalog.azureiotsuite.com](https://catalog.azureiotsuite.com)

## Expanded Device Catalog with IoT Edge certified hardware



- Capability based certification for extensibility and long-term sustainability
- Each capability has N number of levels. Level 1 being the lowest.
- Select the best device most suitable for your IoT application



Demo



# Steps to deploy a new device on IoT Edge (on Windows)

## Prerequisites

- Windows 10 or Windows Server 2016 (or more)
  - Install Docker
  - Set Docker to use Linux Containers or Windows Containers (your choice... but...)
- 
- Create an IoT Hub (portal)
  - Register an IoT Edge device (portal)
  - (deploy and) Start the IoT Edge runtime (device)
  - Deploy a module

# Deploy and start IoT Edge Runtime

1. On IoT Edge device, run PowerShell as an administrator.
2. Download the IoT Edge service package.
3. Install the vcruntime
4. Create and start the IoT Edge service
5. Add firewall exceptions for the ports that the IoT Edge service uses
6. Create a .reg file with some settings and execute it
7. To configure IoT Edge Runtime need to set:
  1. Hostname and IP of the device
  2. Set to Environment variable
  3. Add firewall rules
  4. Restart service

It (hopefully) works!



# THANK YOU TO OUR SPONSORS



Microsoft

GOLD SPONSOR

**bi4all**

CREATING BUSINESS INTELLIGENCE

SILVER SPONSOR

|create|**it**|

INNOVATING **LIFE**

TUGA BEER SPONSOR

**FARFETCH**

SWAG SPONSOR





Questions?



PLEASE FILL IN THE  
EVALUATION FORM.  
YOUR OPINION IS  
IMPORTANT!

AT THE ENTRANCE AFTER THE LAST SESSION OF THE DAY



#TUGABEER sponsored by

|create|**it**|  
INNOVATING LIFE



# References

- <https://www.microsoft.com/en-us/WindowsForBusiness/windows-iot>
- <https://docs.microsoft.com/en-us/azure/iot-edge>
- <https://catalog.azureiotsolutions.com/>





# Thanks

**Marco Dal Pino**

Microsoft WinDev MVP

Intel® Innovator & BlackBelt

Program Manager

Technical Evangelist

@MarcoDalpino

[About.me/marcodalpino](https://about.me/marcodalpino)