Microsoft Azure

Simplifying IoT with IoT Plug and Play

Berry Tsai
2020/12/08

Agenda

IoT Plug and Play 介紹

Digital Twins Define Language介紹

Model file介紹,製作及發佈流程演示

Azure 裝置認證流程演示

Hands on Lab – Azure認證實作

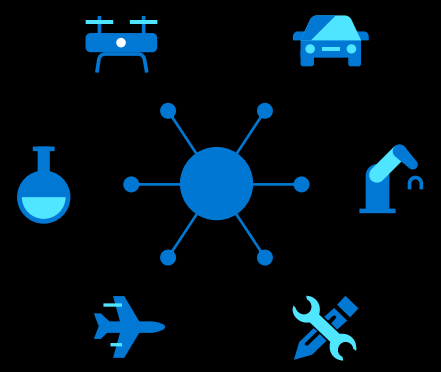
Microsoft Azure

IoT Plug and Play 介紹

Evolving to a future of connected IoT ecosystems

IoT Plug and Play is foundation to shift from connected assets to connected environments

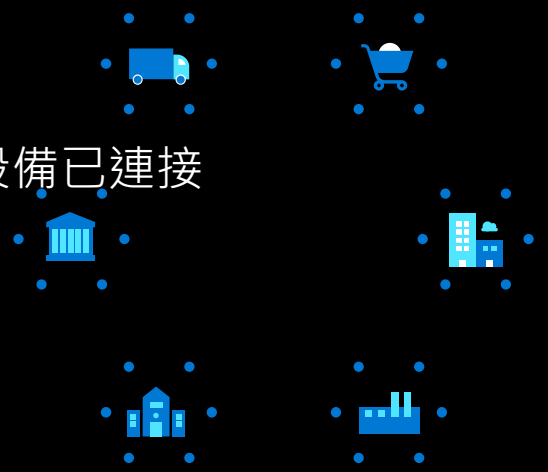
Today



Connected assets

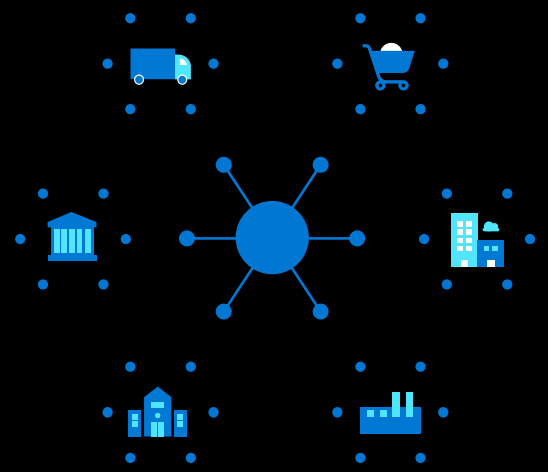
如今，許多設備已連接

Emerging



Connected environments

Future



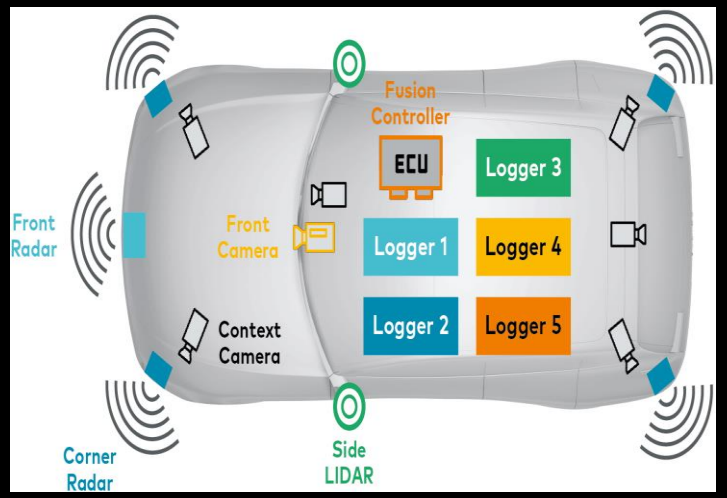
Connected ecosystems

雪山隧道假日塞車

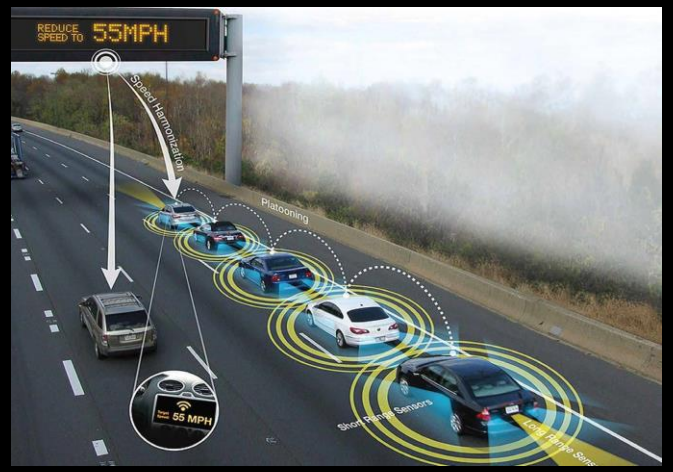
車聯網的解決方案

Today

Emerging



Future

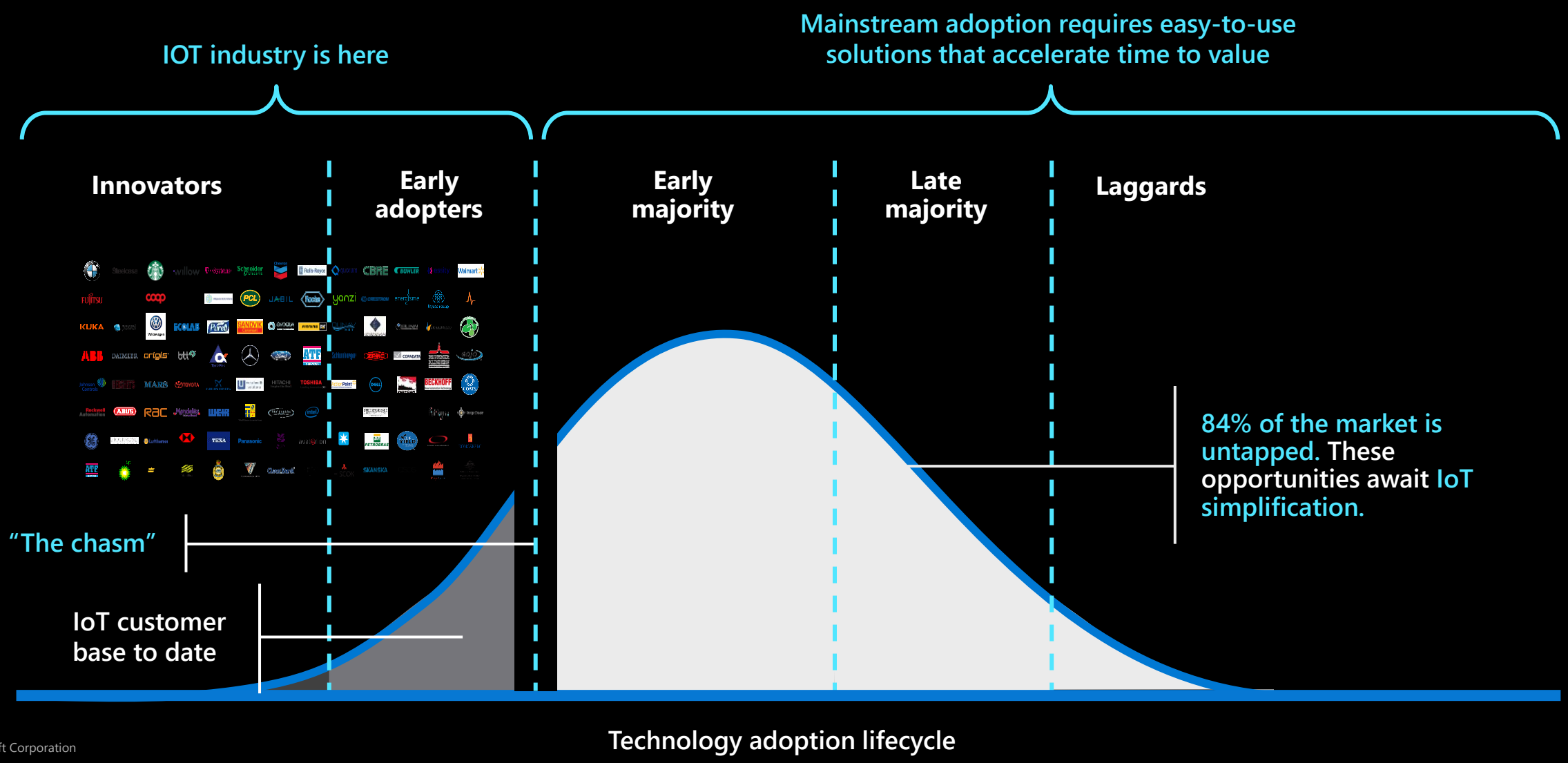


Connected assets

Connected environments

Connected ecosystems

Simplifying IoT key to mainstream adoption



IoT Ecosystem

IHV → Device → Connectivity Provider → ISV → Service Provider → SI → Professional Services → Consulting → Solutions → Customer

Developer Community

Visual Studio, GitHub, hadoop, docker, php, nodeJS, Powershell, eclipse, python, ARM mbed, MySQL, Ruby, Java, mongoDB, Chef + Puppet

Partner Ecosystem

System Integrators & Advisors

accenture, CGI, HCL, EY, TATA Consultancy Services, HARMAN, eBECS, T-Systems, KPMG, Cognizant, Infosys, Tech Mahindra, SOGETI, avanade, pwc, DXC.technology, Larsen & Toubro, Reply, wipro, Capgemini, robotron, MOQdigital

Solution Providers

Schneider Electric, GE, ABB, esri, OSI, ICONICS, Rockwell Automation, ActionPoint, conplement, codit, Johnson Controls, Honeywell, SAP, ptc, IoT, energisme, SIEMENS, Atos Worldline, DUNAV NET, Hitachi Solutions, Schlumberger, COPADATA, relayr

Solution Aggregators

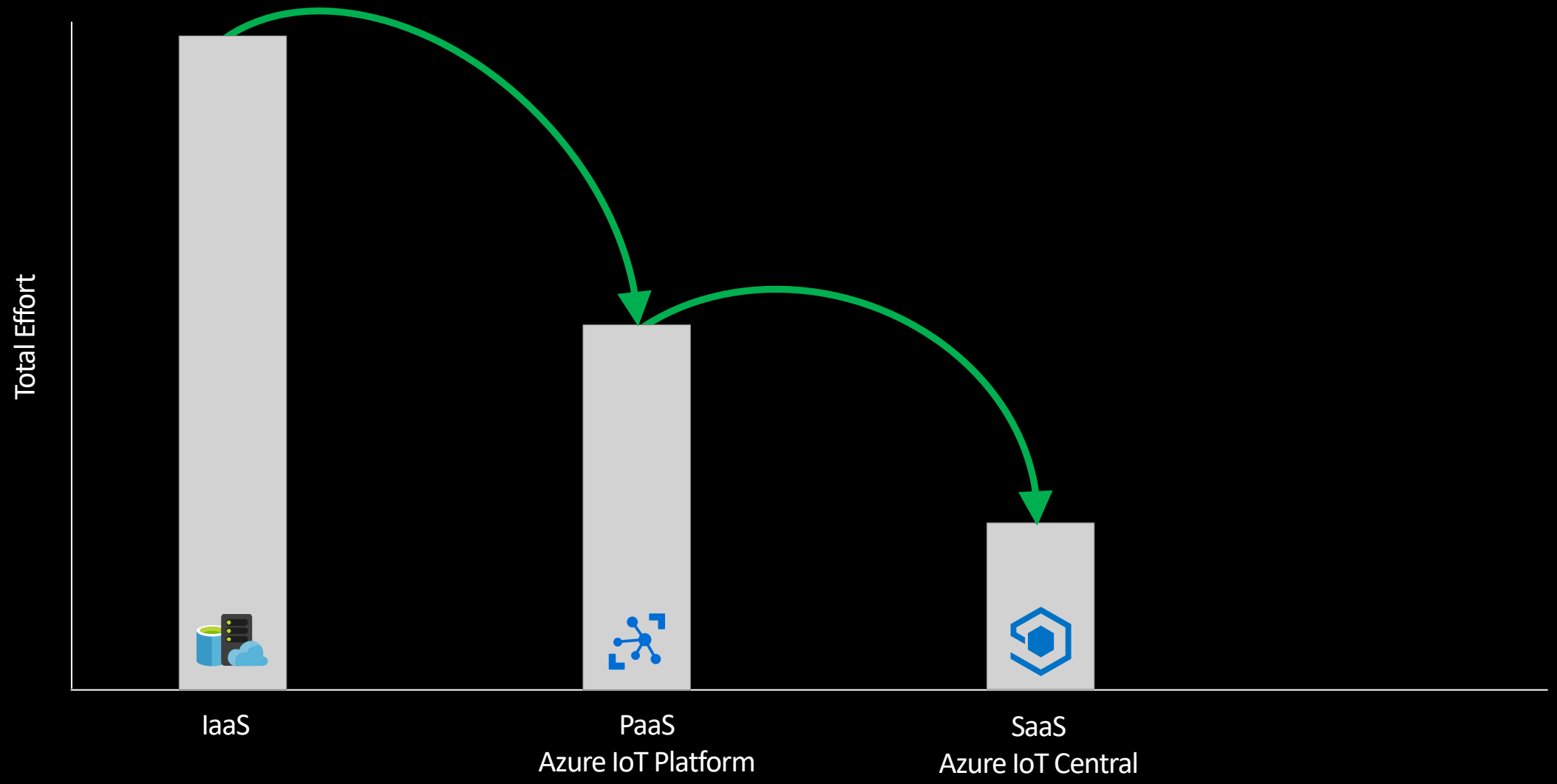
ARROW ELECTRONICS, INC., SYNNEX, MESHSYSTEMS, CII, Insight, happiest minds, ICT, AVNET, Tech Data, INGRAM, Mobiliya, SIGMA, at&t, M

Devices

ST, RENESAS, CISCO, kontron, RaspberryPi, DELL, libelium, Itron, intel, MOXA, Qualcomm, cradlepoint, BECKHOFF, HITACHI, embedded systems, TOSHIBA, Panasonic, NEXCOM, Hewlett Packard Enterprise, ADVANTECH, FUJITSU, Toradex, ARBOR

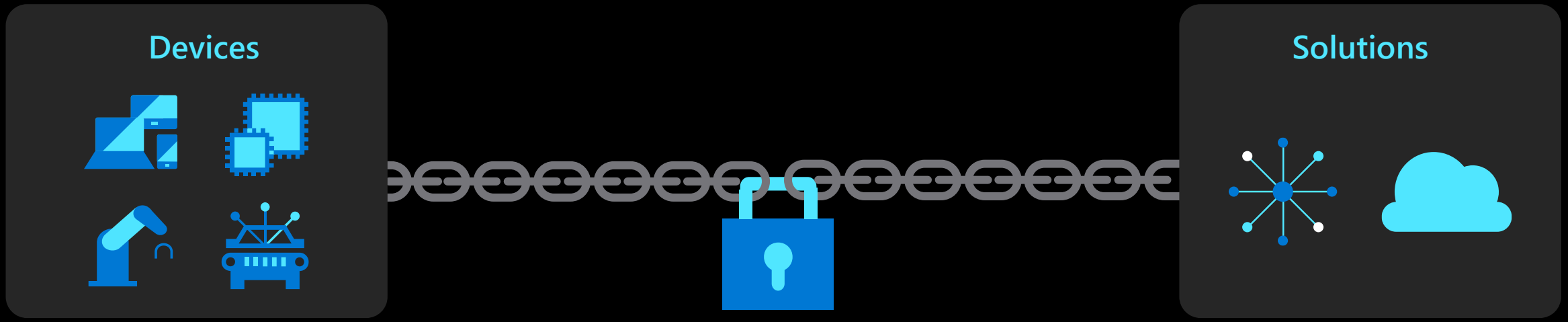
Accelerating IoT for mainstream adoption

The total effort to build and operate an IoT Solution is rapidly decreasing

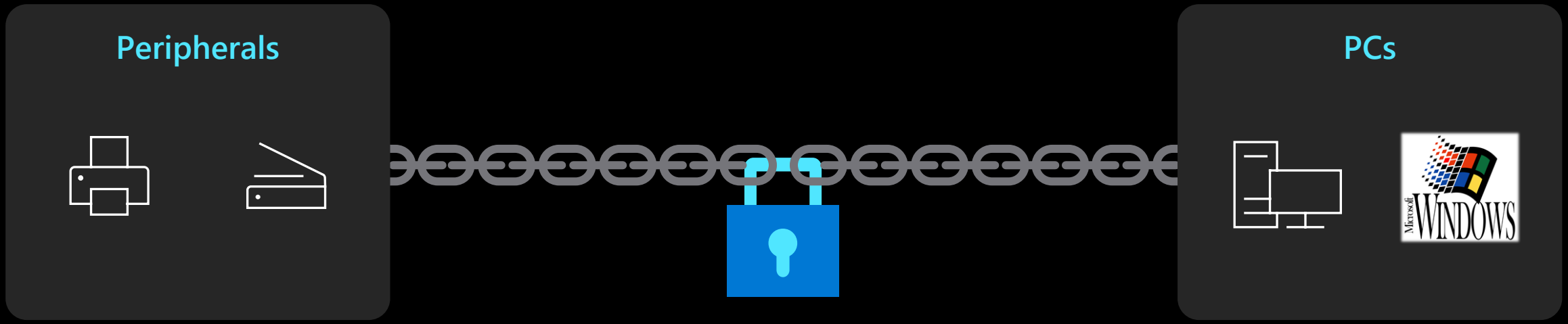


IoT Today

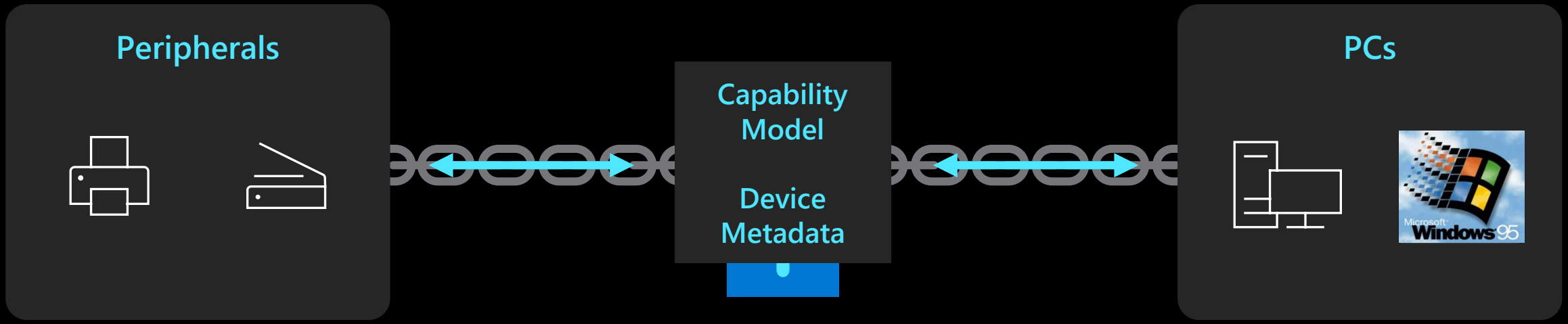
Tight coupling between software on device and IoT solution in the cloud



We had a similar challenge in the past



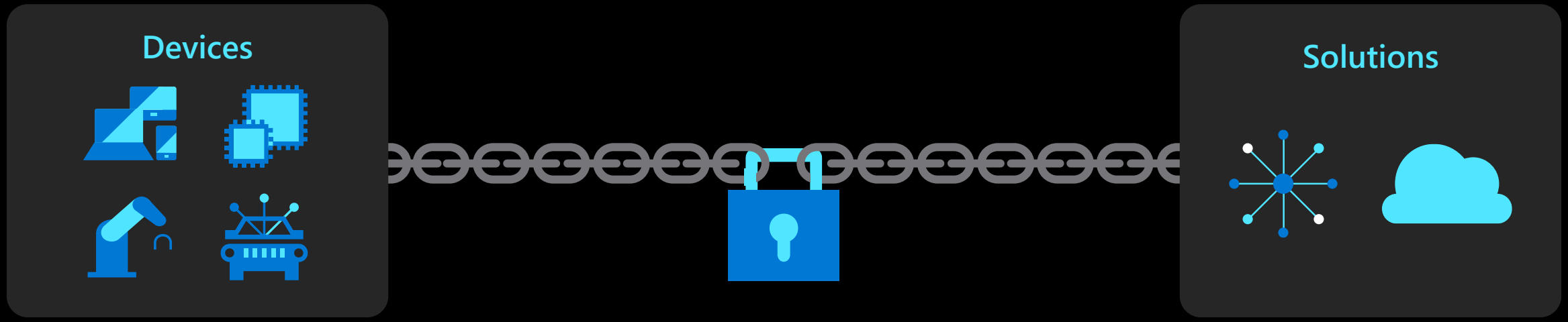
That was solved in Windows with Plug and Play



Devices published their **capability models** and adhered to them
Windows used the capability model to know how to **interact** with them

Introducing IoT Plug and Play

Simplifies device interactions in IoT solutions with an open modeling language

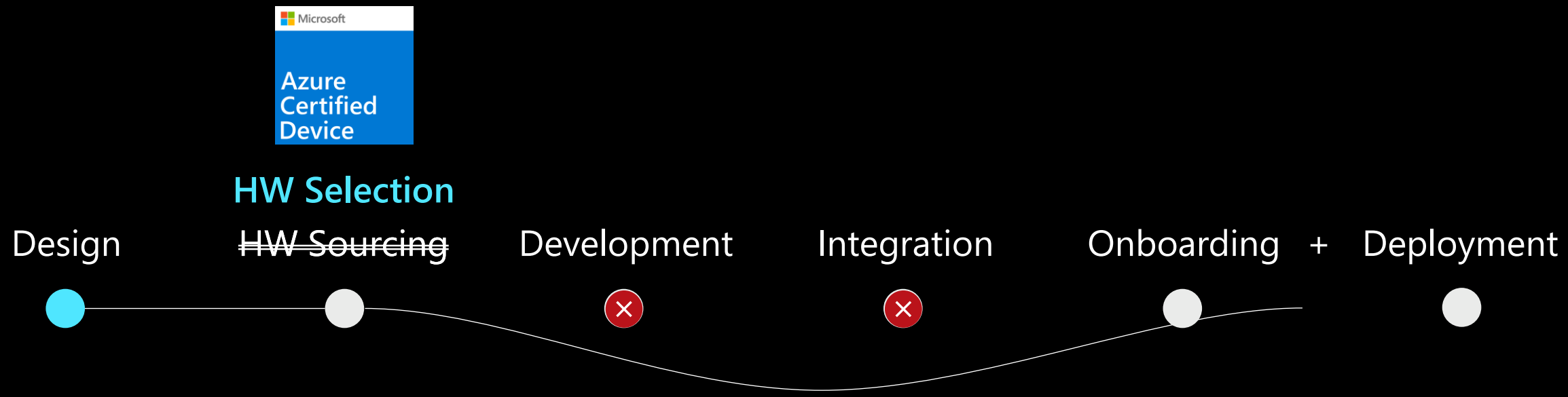


Devices self-describe capabilities based on open Digital Twins Definition Language.

Solutions can *automatically* adapt to devices

All **without custom code**

Building IoT solution with IoT Plug and Play



Building IoT solution with IoT Plug and Play

Microsoft
Azure Certified Device

Design — **HW Selection** — Onboarding + Deployment

Enables solution developers to **focus on** solution development.

Simplifies device developers to ship a single firmware for **all solutions**.

Simplifying IoT with IoT Plug and Play

Microsoft

IoT Plug and Play simplifies IoT

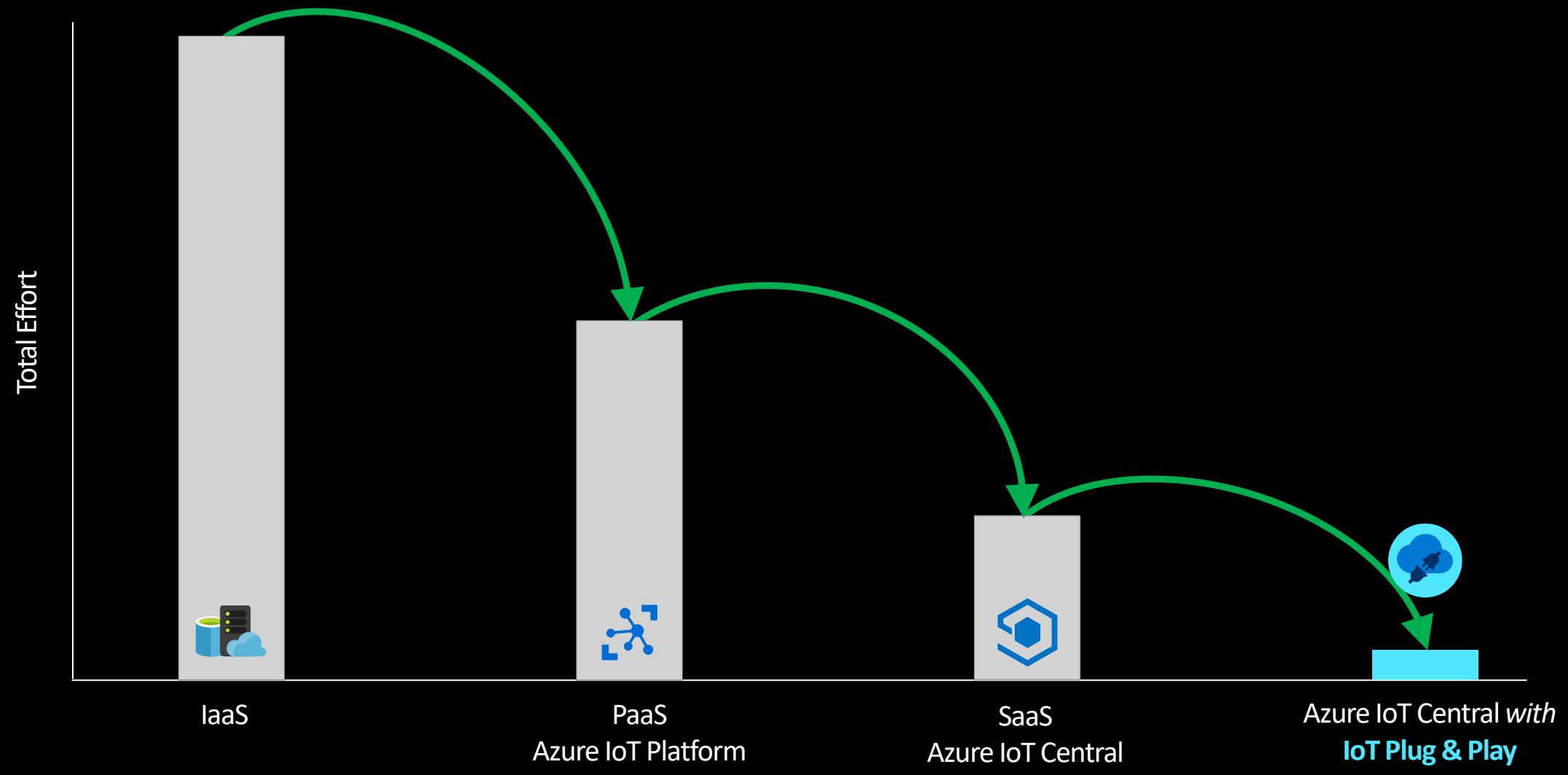
Powerful

Fast

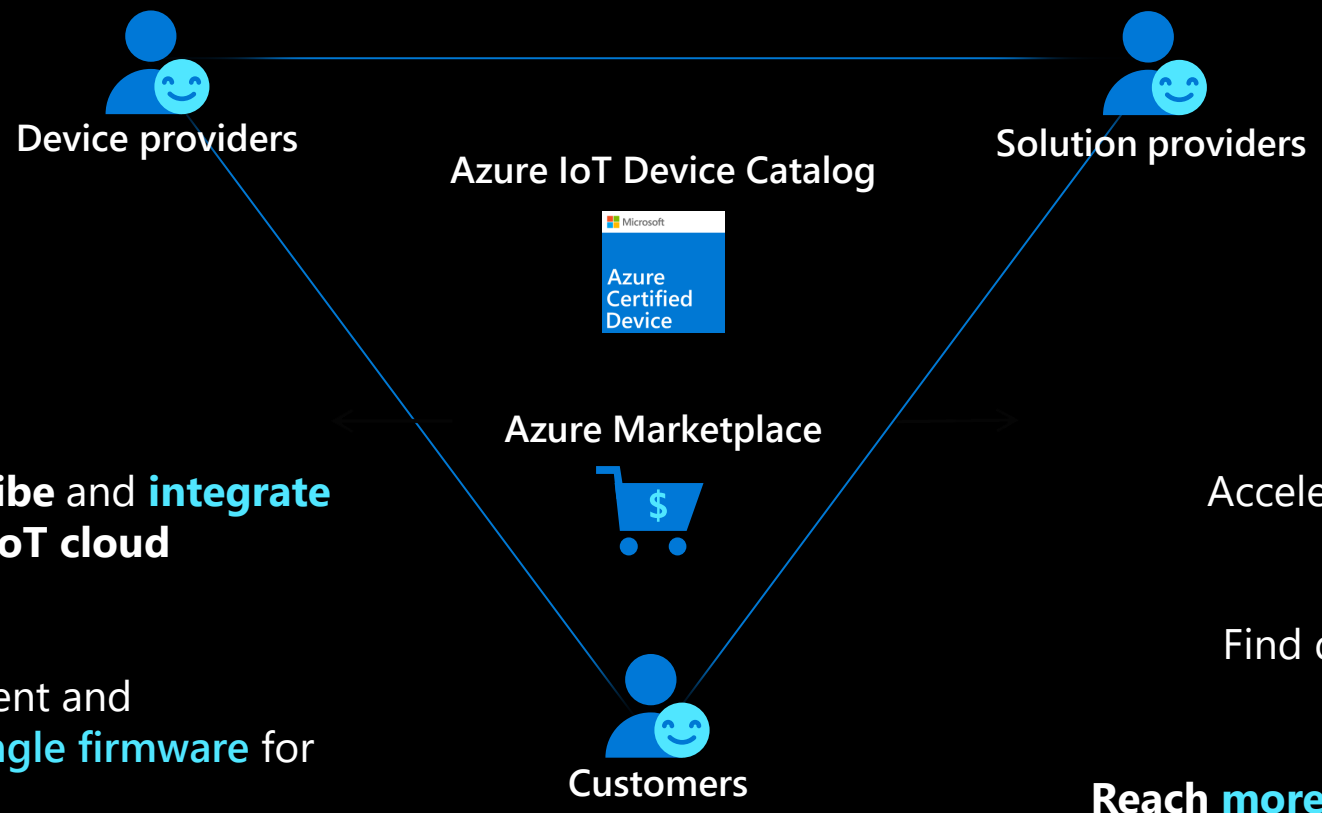
Easy

Accelerating IoT

IoT Plug and Play + Azure IoT Central reduce overall integration time from days/months to **minutes**



Building the IoT Plug and Play ecosystem – 2021 and beyond



Device providers

Build devices that **self-describe** and **integrate seamlessly** with **any Azure IoT cloud solution.**

Reduces firmware development and maintenance effort with **a single firmware** for all solutions.

Increase visibility to solution developers via the **Azure Certified Device Catalog.**

Solution providers

Accelerate solutions **without writing embedded code.**

Find devices with the **capabilities to fit your solution needs.**

Reach more customers by promoting your solutions in the **Azure Marketplace.**

Digital Twin Consortium

Founders and Groundbreakers

Founders

Ansys
DELL Technologies
GE
lendlease
Microsoft
NG Northrop Grumman

Groundbreakers

Air Force Research Laboratory
Animated Insights
Asset Management Lab, LLC
Association of Asset Management Professionals
Autiosalo Ltd
BEC - Blockchain Engineering Council
BIM6D Consulting
Bandora Systems
Bentley Systems
Building 4.0 CRC
Chain Technology Development Co. Limited
CodeData
Connector Geek Ltd
ConstruWise, Inc.
CumuloCogitus Inc.
Cybertwin
DIGIOTAI
DataCities
e-Magic Inc.
Executive Development
Gafcon, Inc.
Geminus.AI
Healthskouts
IIMBE
IOTA Foundation
IOTIFY
Idun Real Estate Solutions AB
ieLabs
IoT Management
imec
Itus Digital
Jitsuin, Inc.
LINQ Ltd.
LUNO UAB
Lux Modus Ltd.
Monash University
NSW State Government
Neural Concept
Padi LLC
Piprate
PropTechNL
Resonai
Ricardo
Slingshot Simulations
Systems Analytics Solutions
Transforma Insights
Trendspek
Twin Building GmbH
University of Melbourne
UrsaLeo Inc.
WSC Technology
Willow Technology Corporation Pty Ltd
Ynomia
YoGeo, Inc.

IoT Plug and Play overview

Open modeling language

Digital Twin Definition Language v. 2 (DTDL)

Device model

Digital Twin model to describe IoT device

Azure Digital Twins alignment

Seamless integration with Azure Digital Twins

Open modeling language

Model IoT device with Digital Twin Definition Language v. 2

- Open language based on **JSON-LD** and **RDF**: https://json-ld.org/
- **Used in publishing** and using information in the internet (i.e., search)

Common language between IoT device and IoT application via device model

- **Device to communicate** its functionalities and attributes to IoT application
- **IoT application to understand** device's functionalities and attributes

Describe interaction model in DTDL

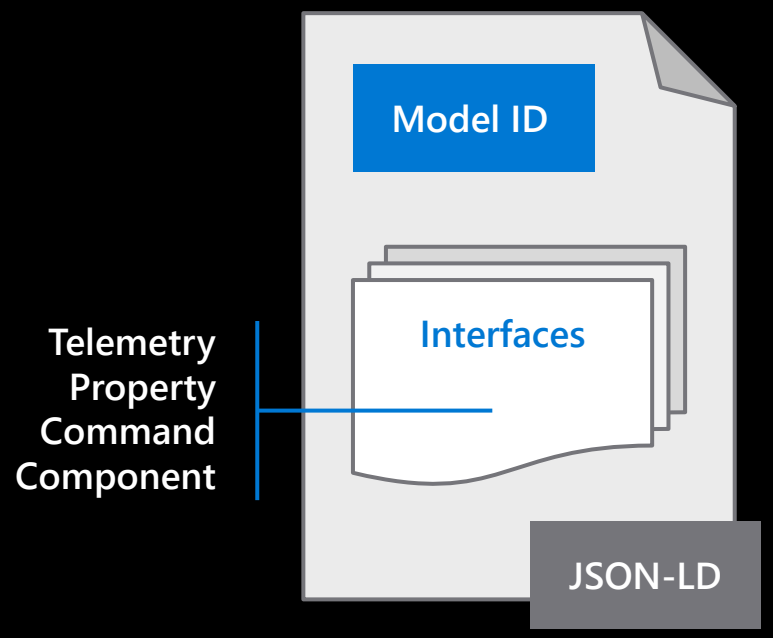
Describe device capability and interaction model

Each device model has a unique ID: **Digital Twin Model ID (DTMI).**

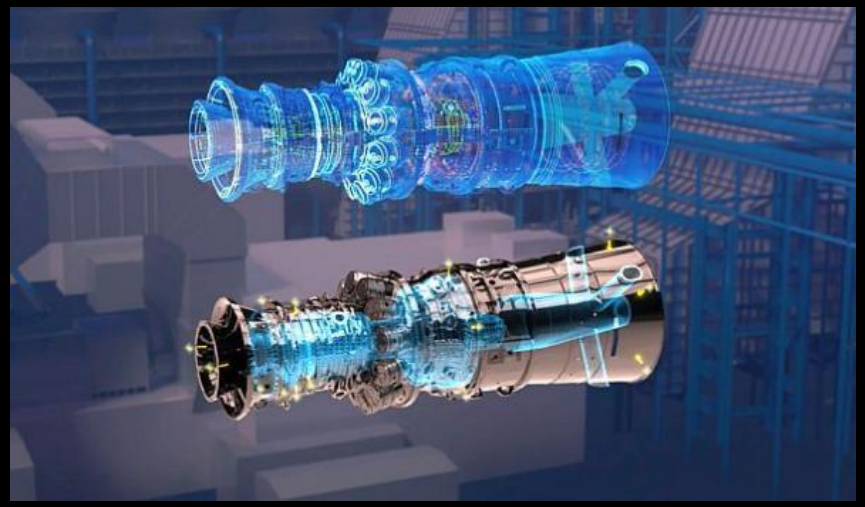
Each device model consists of a set of **interfaces.**

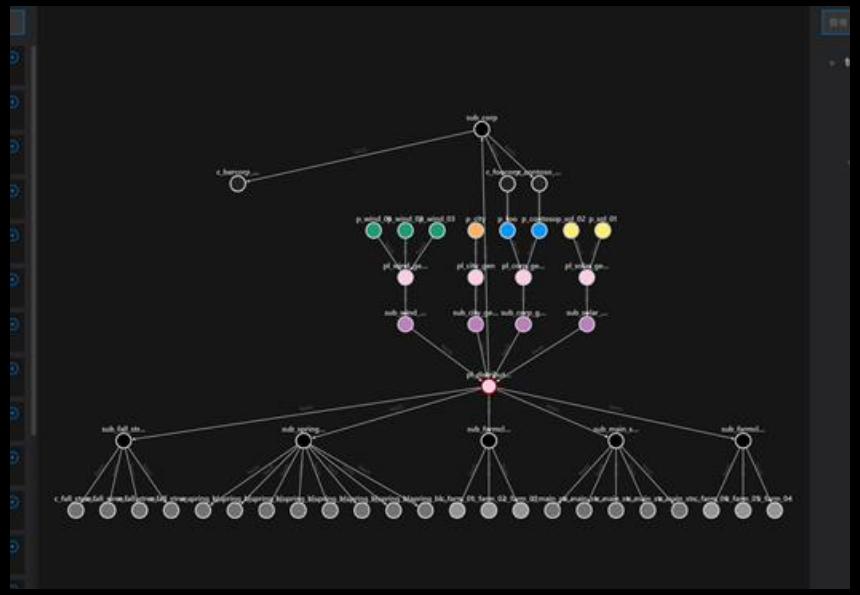
Interfaces describe attributes of the device:
telemetry, property, command, and component.

Solution can query and parse device model to understand interaction model.



Azure Digital Twins alignment





Digital Twin enables the creation of knowledge graphs based on digital models of physical environment.

Allows creation of relationships among twins.

Enriches twins by adding metadata and attributes.

IoT Plug and Play is fully aligned with Azure Digital Twins.

The IoT Plug and Play device model becomes one of the twins in the graph.

Allows addition of (plug) IoT device into the graph.

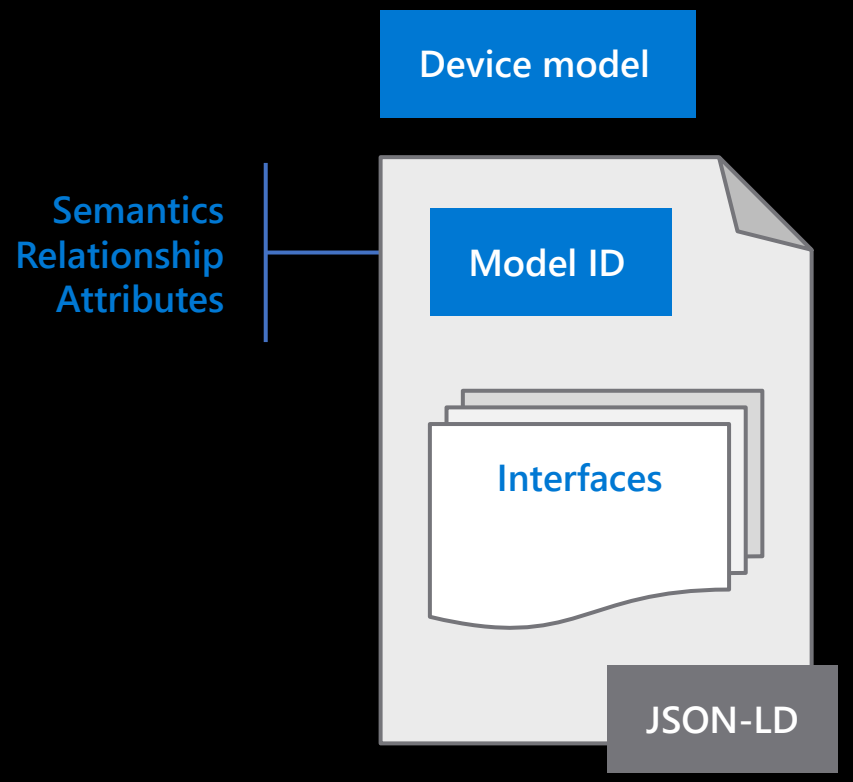
Allows interaction with (play) IoT device and data from it in the graph.

Device model key to understanding interaction model

IoT Plug and Play provides the ability to add metadata to devices and data.

The digital twin model is written with **semantics, relationship, and attributes**, enabling solution builders to understand the interaction model, so they can **innovate experiences** and **use cases**.

The **value of data** comes from interpreting, processing, and interacting with devices through the capabilities that IoT Plug and Play enables.



Example: Evolution of devices with metadata

Cassette → CD → MP3

Analog

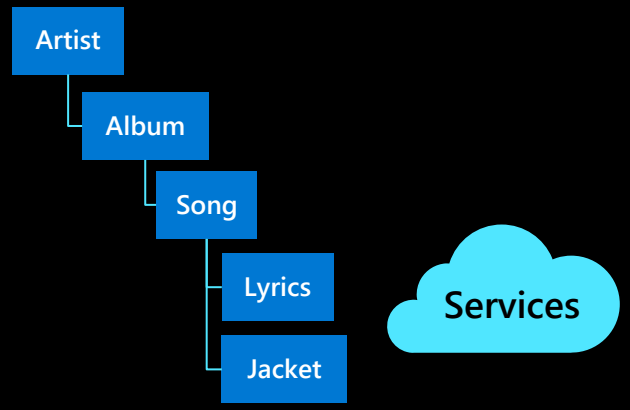
- Only **physical** goods
- **No metadata**
- Shelf to organize

Partial digitalization

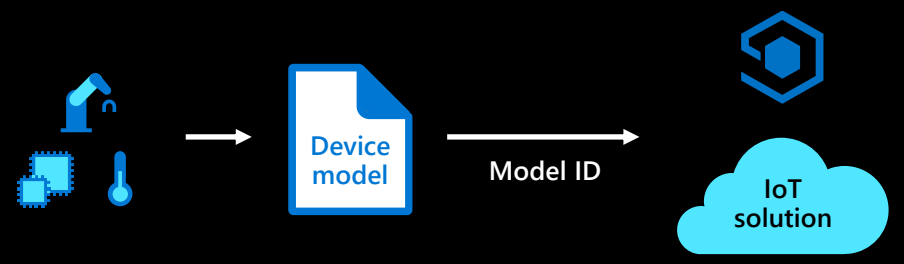
- **Physical** assets and **digital** content
- **Some metadata** - disk ID
- Enabled **internet-based services** through metadata
- Still **somewhat limited**

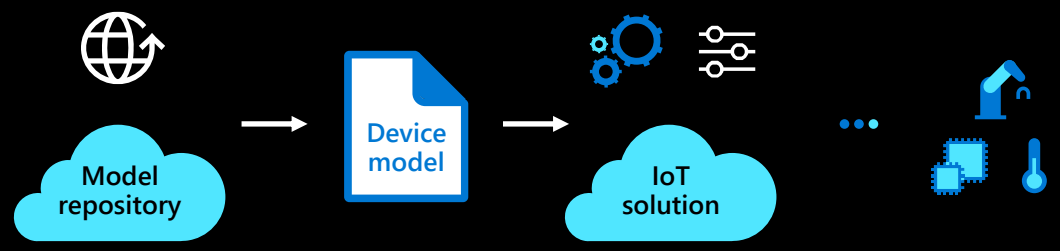
Full digitalization

- **Digital** assets' metadata attached
- **Variety of metadata**
- Enabled **new solutions and services** using metadata in structured way
- Many **new use cases**



How IoT Plug and Play communicates from device to solution





Device builder

Plug device into IoT solution with model ID. IoT Plug and Play communicates device capabilities and attributes to the cloud.

- Author a device model using Digital Twin Definition Language version 2.
- Implement IoT Plug and Play convention based on device model.
- Announce model ID to IoT Hub / Device Provisioning Service (DPS) during connection.

Solution builder

Play device with interaction model through device model. The model ID enables the IoT solution to understand the interaction model.

- Retrieve model ID as metadata to the IoT device and IoT data.
- Retrieve device model using model ID from model repository.
- Parse device model to understand interaction model.

IoT Plug and Play support at a glance

IoT Plug and Play support in Azure

We are enabling IoT Plug and Play support in multiple Azure IoT Platform services.

Content to onboard partners

Collection of online content to get started.

Blogs, **tutorials**, **quick starts**, and **videos.**

Certification program to build confidence

Build IoT Plug and Play devices with high confidence and reach our global audience through the online marketplace.

We are **accepting** certification submissions!

SDKs to empower developers

SDKs in seven supported languages to enable development.

Device SDKs for device builders.

Service SDKs for solution builders.

Tools and samples to support developers

Sample code for each SDK with documentation.

Azure IoT Explorer to interact with devices.

DTDL extension and **parser** to support device model authoring.

Partner engagement to shape the ecosystem

Working in partnership to build IoT Plug and Play ecosystem.

Enabling IoT developers with Software Development Kit

Simplify and accelerate the development of IoT solutions built with Azure.

- Not mandatory for IoT Plug and Play but highly encouraged.

Device builders

Device SDKs General availability

- C
- . NET
- Python
- Java
- Node.js

SDKs for resource constrained devices General availability

- Embedded C
- Azure RTOS

Solution builders

Service SDKs General availability

- .NET
- Java
- Python
- Node.js

*Including **sample code** + associated **documentation** show how to implement IoT Plug and Play conventions.

Empowering developers with tools

Azure IoT Explorer (new)

Azure IoT Explorer is a graphical tool for interacting with and testing your IoT Plug and Play devices.

Azure IoT CLI extension

The Azure CLI is an open-source, cross platform command-line tool for managing Azure resources such as IoT Hub.

DTDL parser library

These tools support device model authoring and consuming device models in IoT applications.

DTDL extensions

Take advantage of extensions for Visual Studio and Visual Studio Code with syntax validation and Intellisense.

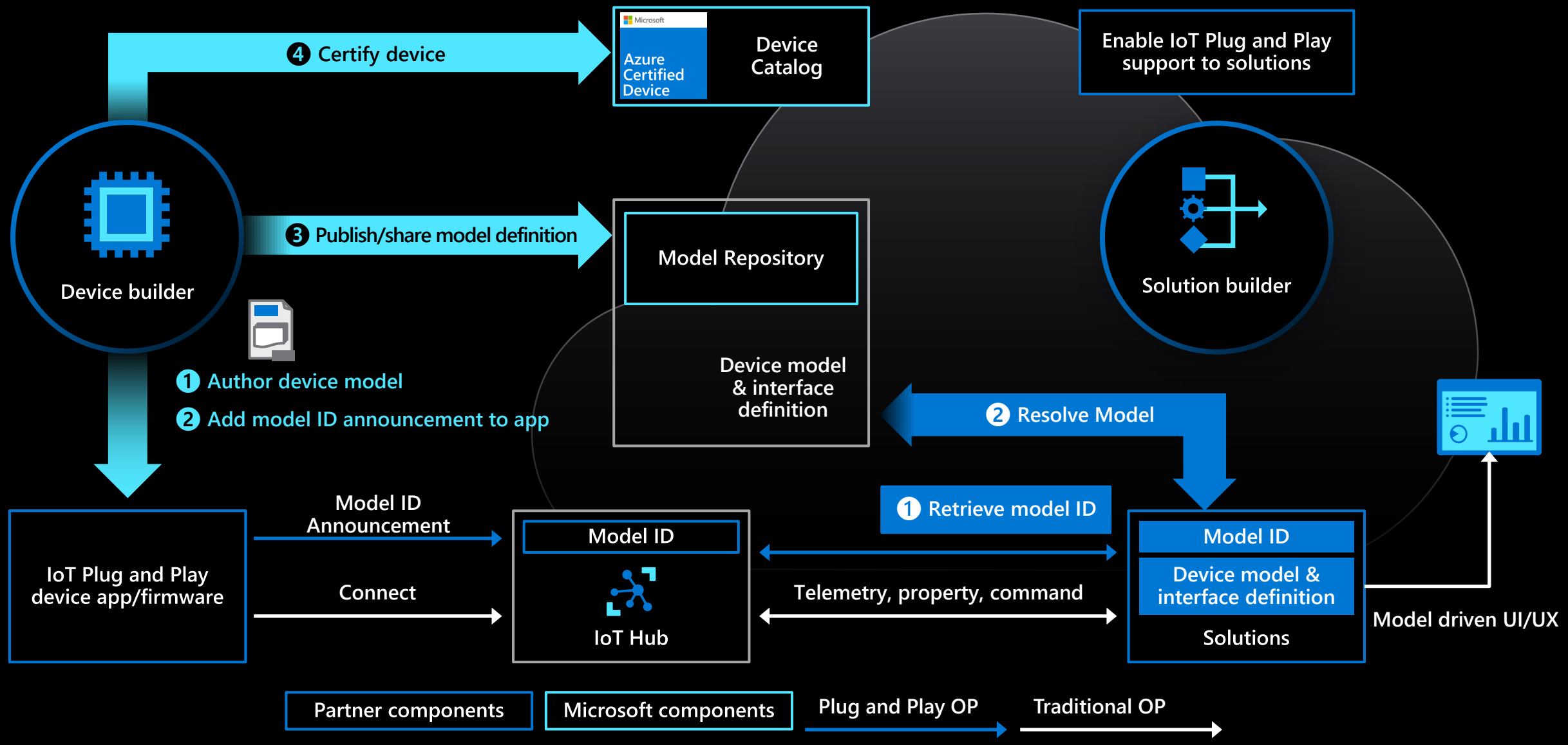
Microsoft Azure

Demo

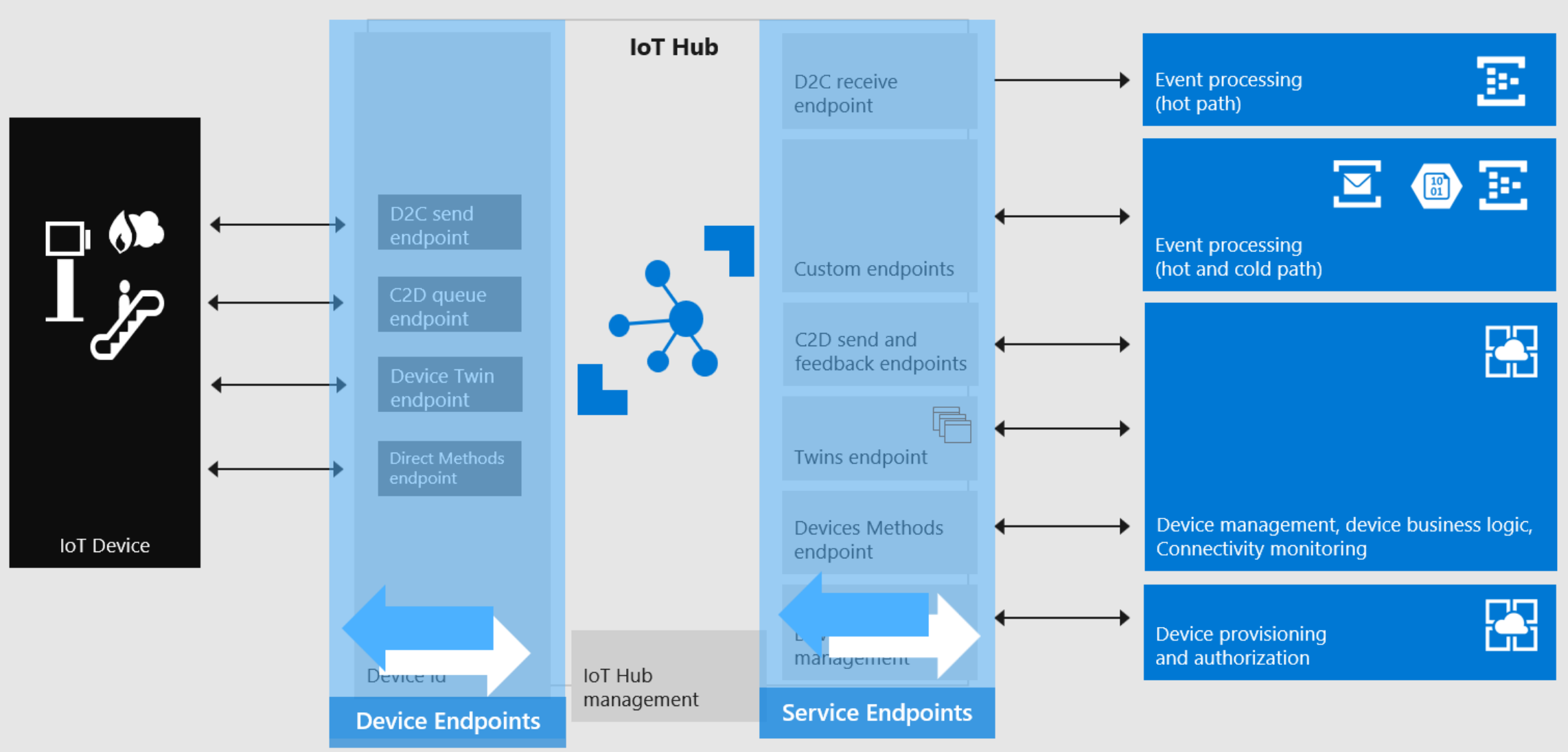
Microsoft Azure

IoT Plug and Play technical deep dive

IoT Plug and Play process overview



Azure IoT Hub Feature



❶ Author device model

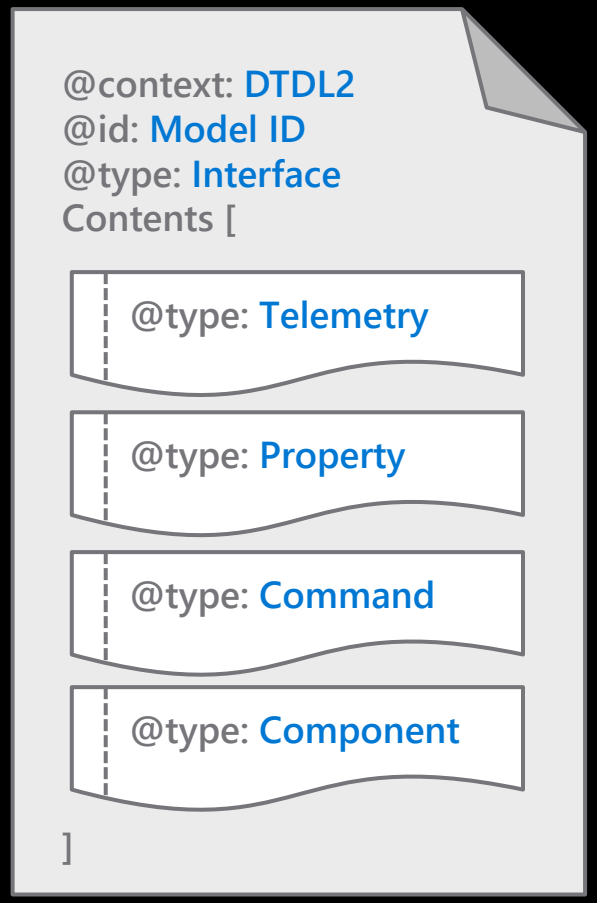
IoT Plug and Play device model

Device model

- Describes a device's **capabilities** and **attributes** using **types**, **schemas**, and **semantics**
- Consists of **one or more interfaces**
- Uniquely identified with **Digital Twin Model ID (DTMI)**

Interface

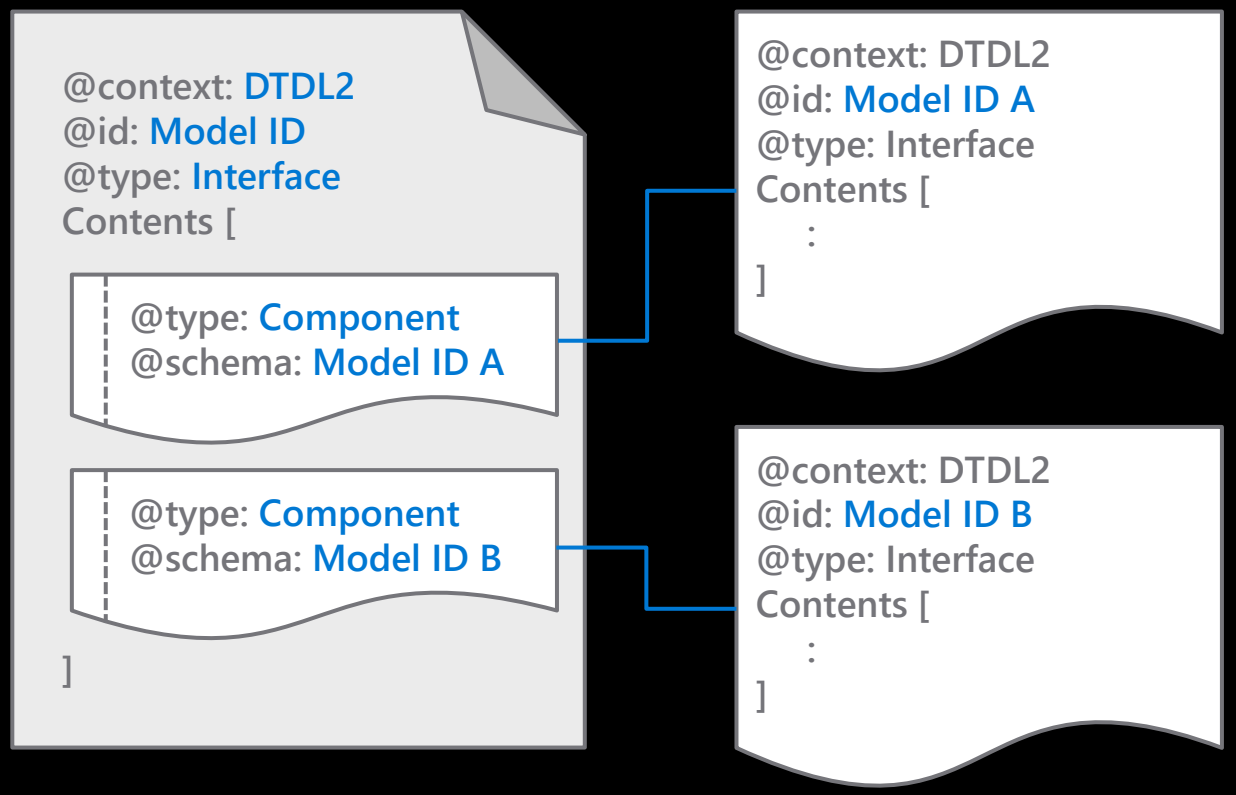
- The **top-level item** in a device model
- **Telemetry**, **property**, **command**, and **components** describe the device



Componentizing IoT Plug and Play device model

Enables creation of a device model interface as an assembly of other interfaces

Default device model is a **single interface (or component-less)** model



IoT Plug and Play device model authoring

IoT Plug and Play device model = Digital Twin model

- A schema that describes device **capabilities** and **attributes**
- Written in **DTDL v2** based on **JSON-LD**
- **DTDL v2** for schemas, semantics

  https://aka.ms/dtdl

Tools for device model authoring

- **Visual Studio / Visual Studio Code**
  - Extensions for model authoring
- **DTDL Parser Library**
  - For syntax check

```
{
    "@context": "dtmi:dtdl:context;2",
    "@id": "dtmi:com:example:Thermostat;1",
    "@type": "Interface",
    "displayName": "Thermostat",
    "description": "IoT Plug and Play Device Model for Thermostat",
    "contents": [
      {
        "@type": [
          "Telemetry",
          "Temperature"
        ],
        "name": "temperature",
        "displayName" : "Temperature",
        "description" : "Temperature in degrees Celsius.",
        "schema": "double",
        "unit": "degreeCelsius"
      },
```

This new industry standard is a collaborative effort between Microsoft and the Digital Twin Consortium.

IoT Plug and Play device model example

Temperature sensor

Telemetry payload

{"temp" : "12.34"}

Without device model

"temp is 12.34"

With device model

"Temperature is 12.34 degrees in Celsius"

```
{
  "@context":"dtmi:dtdl:context;2",
  "@id":"dtmi:com:mycompany:Thermostat;1",
  "@type":"Interface",
  "displayName":"IoT Plug and Play Device",
  "description":"Device Model Example",
  "contents": [
    {
      "@type":["Telemetry",
                "Temperature"],
      "name":"temp",
      "displayName":"Temperature Data",
      "description":"Temperature in degrees Celsius.",
      "schema":"double",
      "unit":"degreeCelsius"
    },
```

DTDL v2

Model ID

Interface type

Telemetry type

Semantic temperature

Name of data

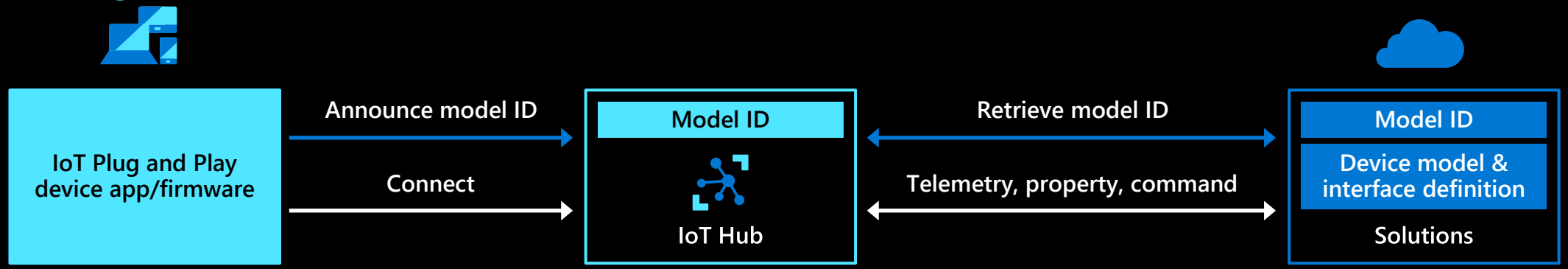
Schema

Unit

❷ Add model ID announcement to app

Model ID announcement

With existing APIs



How your device will communicate

Device communicates model ID to cloud during provisioning or connection.

During provisioning or connection to IoT Hub, the device must announce its model ID with **existing** API

Example : IoT Hub

```
IoTHubDeviceClient_LL_SetOption(deviceHandle,
                                OPTION_MODEL_ID,
                                "dtmi:com:example:IoTPnPDevice;1");
```

Example : DPS

```
Prov_Device_LL_Set_Provisioning_Payload(provDeviceHandle,
            {\"modelId\":\"dtmi:com:example:IoTPnPDevice;1\"}");
```

How solutions can retrieve model ID

IoT Hub receives the model ID and stores it in Device Twin and Digital Twin.

Device Twin

```
{
  "modelId": "dtmi:com:example:IoTPnPDevice;1",
  "deviceId": "IoTPnPDevice",
```

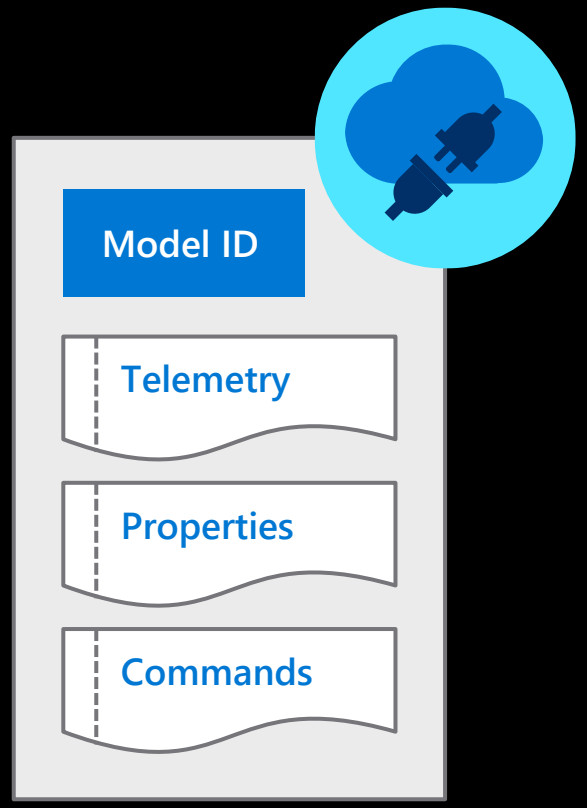
Digital Twin

```
{
  "$dtId": "pnp-thermostat136",
  "DisplayName": "Thermostat in Room 136",
  "$metadata": {
    "$model": "dtmi:iotpnpadt:DigitalTwins:IoTPnPDevice;1",
            :
```

IoT Plug and Play device app / firmware development

Implement IoT Plug and Play convention based on the device model

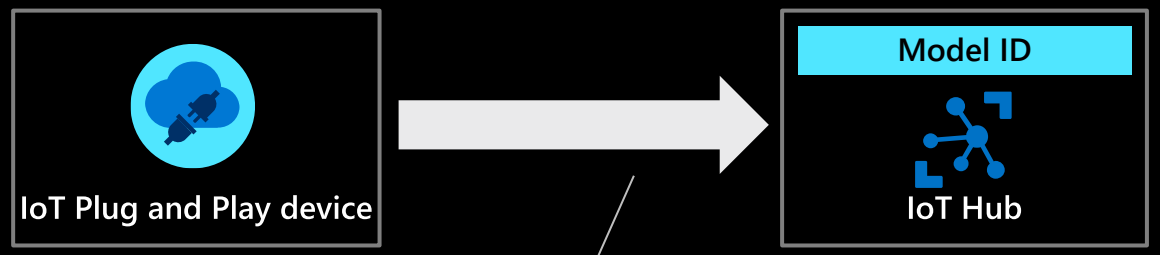
- Model ID announcement: minimum required
- Telemetry – Data from device to cloud
- Writable properties – Settings from cloud to device
- Read-only properties – Reported by device
- Commands – Invoking method on device from cloud
- Follows IoT Plug and Play convention



IoT Plug and Play conventions

IoT Plug and Play conventions

- Devices must follow the conventions when exchanging messages with IoT Hub.
- This ensures consistency in the device model and metadata in the messages.
- [Learn more about IoT Plug and Play conventions](#).




```
"reported": {
  "thermostat1": {
  "__t": "c",
  "targetTemperature": {
      "value": 23,
      "ac": 200,
      "av": 3,
      "ad": "complete“
    }
  }
}
```

❸ Publish/share model definition

Publishing model

Device model needs to be made available to IoT solutions

- Azure IoT device model repository
  Enables device builders to publish and share IoT Plug and Play device models

Model Repositories options

- Public device model repository
  Hosted by Microsoft with GitHub Pull Request validation workflow.
- Custom device model repository
  Supports custom model repository with your choice of storage.

Requirement for IoT Plug and Play device certification

- Publish your device model to Public Model Repository.
- Ensure end customers have access to your device model

Azure IoT device model repository (DMR) & Model Resolution

Solution needs to access to the model definition file (JSON file)

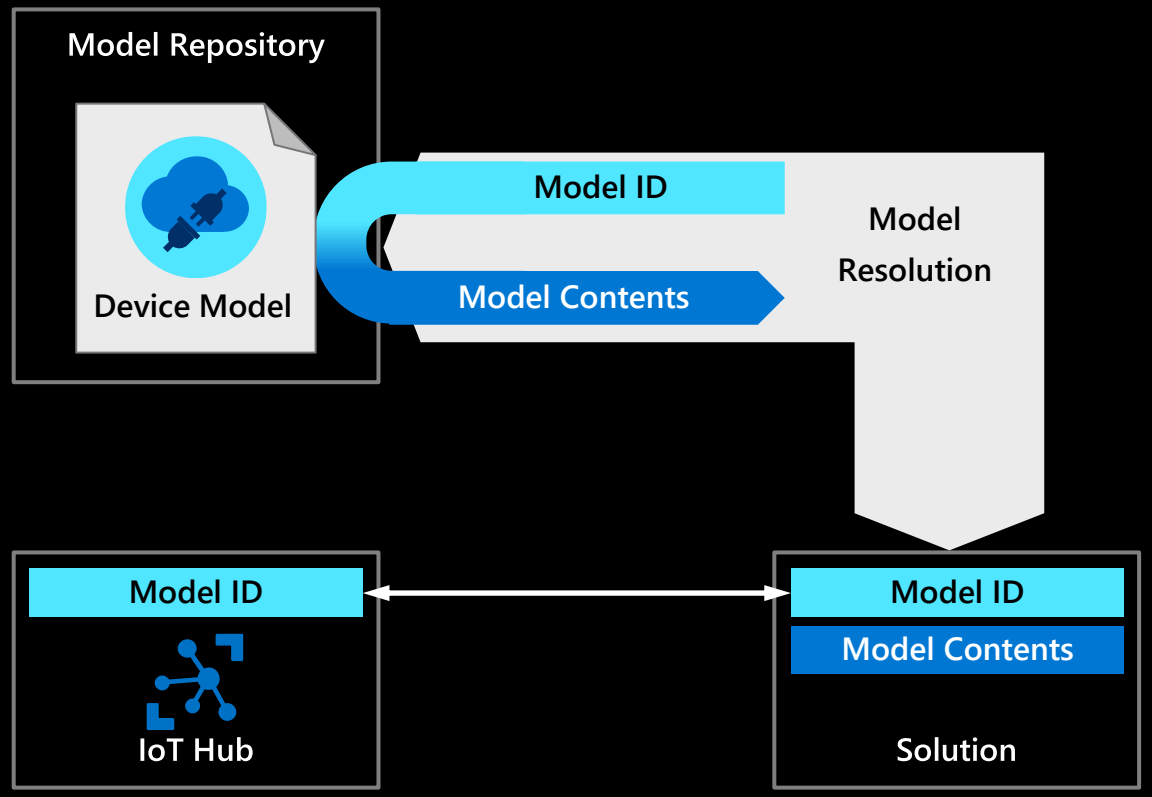
- Through Azure IoT Model Repository
- Through your own repository such as GitHub or file share

Models are immutable

- Approved models in public model repository are immutable

Access through Azure IoT model repository

- Microsoft-hosted public model repository
- Custom model repository
- Supports HTTP API and local file access
- Model Repo Client SDK



Publish a model

[Understand concepts of the device model repository | Microsoft Docs]

https://github.com/Azure/iot-plugandplay-models/tree/main/dtmi

Search or jump to... / Pull requests Issues Marketplace Explore

Azure / iot-plugandplay-models

Watch 9 Star 8 Fork 28

Code Issues Pull requests Actions Projects Wiki Security Insights

main iot-plugandplay-models / dtmi / Go to file Add file

ericmitt adding v2 of tempcontroller model (#54) 5694fae 3 days ago History

..		
advantech	Add Advantech interfaces (#39)	14 days ago
allegion	Add 56 interfaces (#22)	last month
azure/devicemanagement	Add device Information DTDL (#2)	2 months ago
azurertos/devkit	Add AzureRTOS devkit GSG models (#49)	7 days ago
benq/product	Add interfaces of BenQ interface (#25)	last month
bosch	Add Bosch interface xdk110-1.json for the Bosch XDK110 Device (#45)	11 days ago
brainium	Add 56 interfaces (#22)	last month
com	adding v2 of tempcontroller model (#54)	3 days ago
espressif	Add 56 interfaces (#22)	last month
jp/co/covia	Add 56 interfaces (#22)	last month
meshsystems/txs	initial commit for MeshSystems - TraxSmart products (#15)	2 months ago
nexcom	Commit for update deviceInfo error (#48)	10 days ago

github.com/Azure/iot-plugandplay-models/commit/d6f8291f04497117009a416724c672f5412bd41e

4 Certify device

Azure Certified Device Program

Cloud solutions

- Azure IoT Central
- Azure IoT Edge
- Azure IoT Hub
- Azure Digital Twins
- Azure Time Series Insights
- Azure Maps

Device support

Sensors + control		Sensors to interactive	Integrated platform	
Microcontroller Azure RTOS & Azure Sphere	**IoT devices** Azure IoT Device SDK	**IoT Edge devices** Azure IoT Edge	**Edge appliances** Azure Stack Edge	**Edge stack** Azure Stack Hub

Certification programs

Azure Certified Device

Edge Managed

IoT Plug and Play

Azure Certified Device Program overview

The Azure Certified Device Program currently offers three device certifications

Certification	Promise	Requirements	Targeted Devices
Azure Certified Device	Works with IoT Hub	• Device to Cloud telemetry • Device Provisioning with DPS	• Microcontroller • IoT Devices • IoT Edge Devices • Edge Appliances • Edge Stack
Edge Managed	IoT Edge RT works	• Device Provisioning with DPS • IoT Edge RT running on Moby container subsystem • Moby & Edge Security Manager preinstalled	• IoT Edge Devices • Edge Appliances • Edge Stack

IoT Plug and Play – New!	Simplify IoT solution	• IoT Plug and Play device model compliance
		• Including Azure Certified Device requirements
		  • D2C telemetry
		  • Device Provisioning with DPS
			• Microcontroller
			• IoT Devices
			• IoT Edge Devices

IoT Plug and Play certification requirements

Technical requirements

- Device model(s) describing the device and peripherals.
- Support Device to Cloud message (D2C) with the IoT Plug and Play convention.
- Support Device Provisioning Service (DPS) with one or more device authentication types:
  - Symmetric Key
  - X.509
  - Trusted Platform Module (TPM)
- Model ID announcement
  - During device provisioning through DPS
  - During the MQTT connection
- Publish device model to Model Repo provided by Microsoft

Optional features

- Cloud to Device (C2D) message
- Device method (or direct method)
- Device properties, writable and non-writable

Microsoft Azure

Demo

Microsoft Azure

Thank you