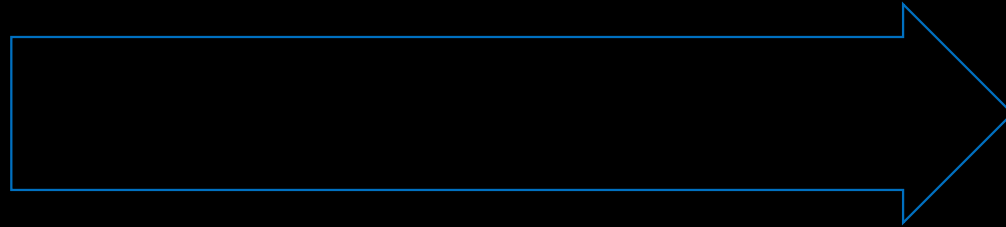


# WoT-Connectivity

Enabling Automatic Asset Onboarding  
with OpenAI

# Industrial Asset Onboarding: OPC UA as key enabler!



Normalized, standardized,  
open data model & interface for  
all assets, including security!

# Industrial Asset Onboarding: Data Model

## 1. Discoverable (~10%)

a) OPC UA-enabled (PLC) (~4%)

-> No ind. conn. software required!

b) Non-OPC UA-enabled (PLC) (~6%)

-> Automatic mapping by ind. conn. software

## 2. Non-Discoverable (~90%)

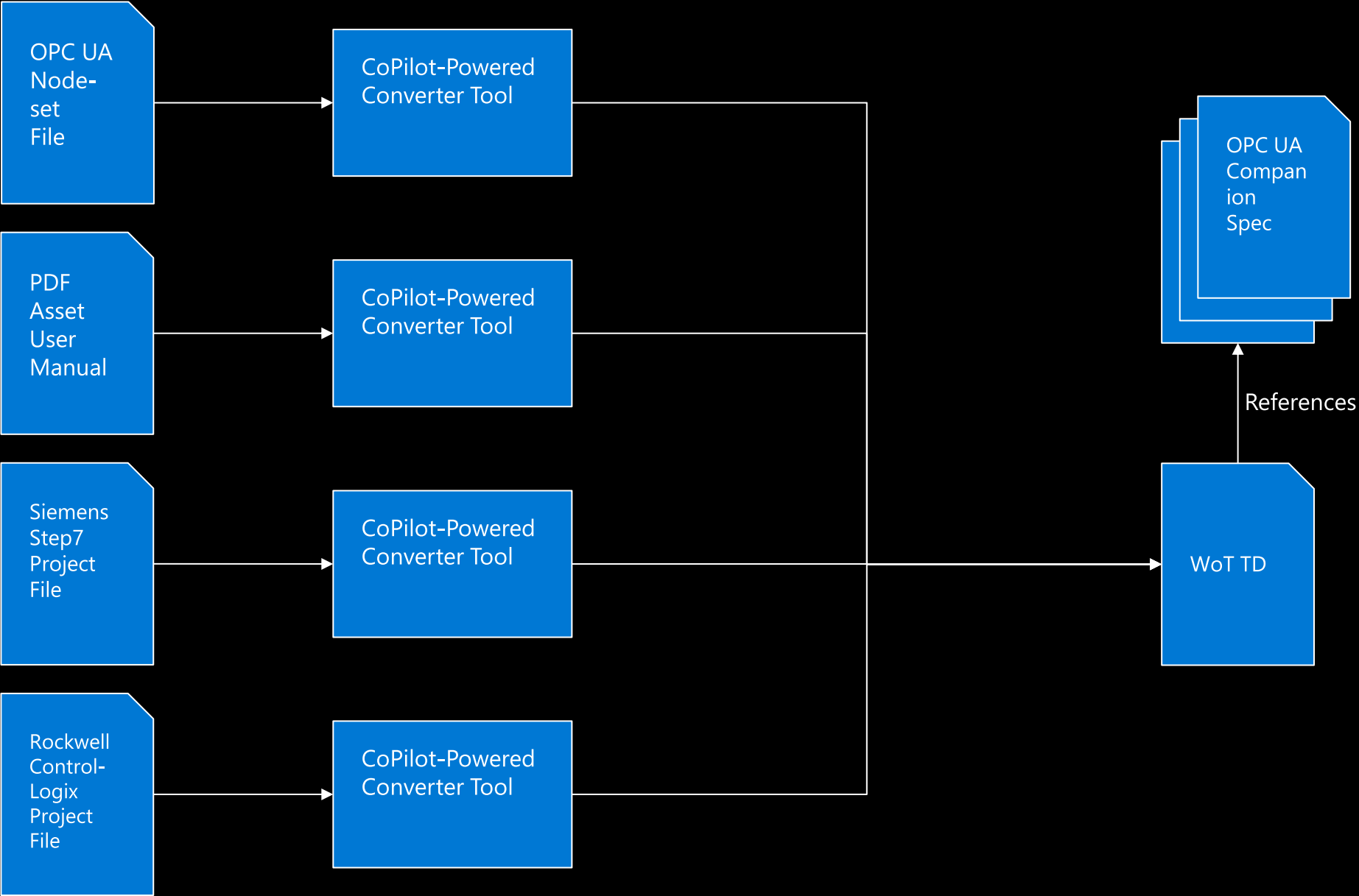
a) Fixed function/data model (~63%)

-> Automatic mapping based on WoT Thing Description sent to ind. conn. software

b) Programmable (PLC) (~27%)

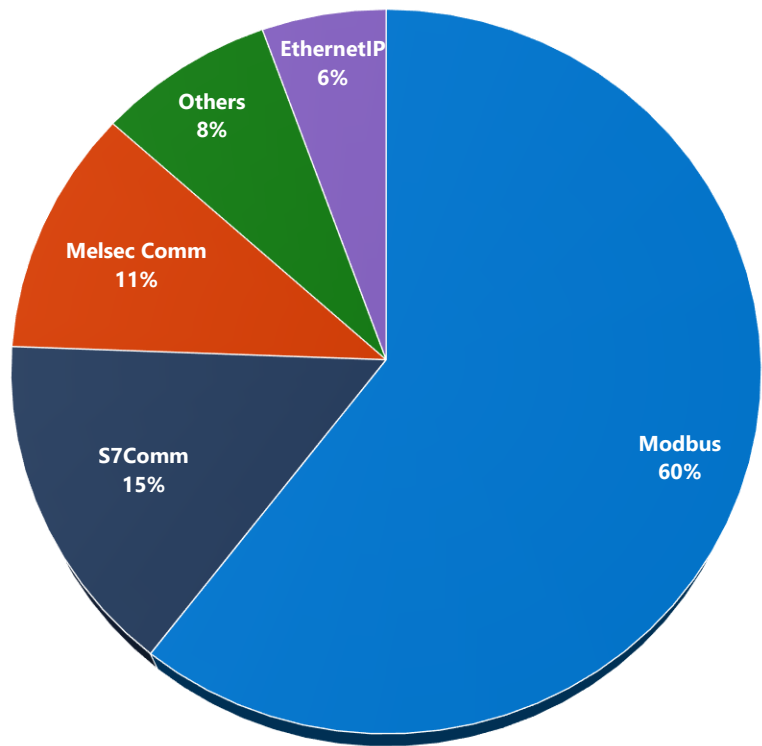
-> Automatic mapping based on project file converter tool to WoT Thing Description sent via ind. conn. software

# Auto-Generating WoT Thing Description from Other Sources

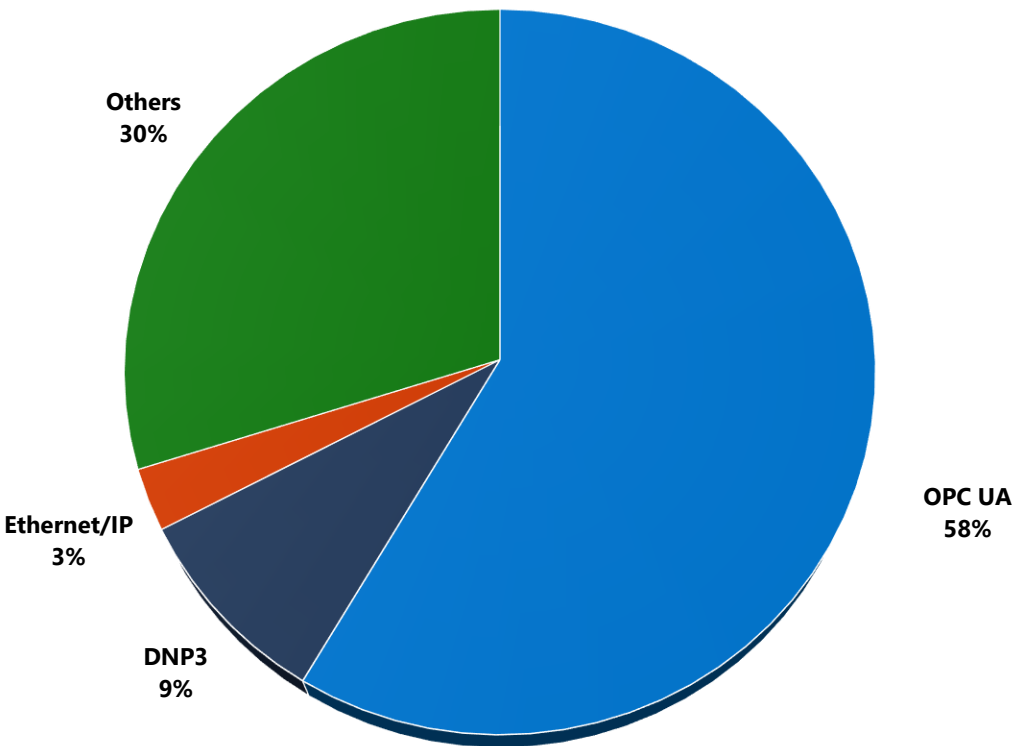


# Industrial Asset Onboarding: Interface (PLCs only)

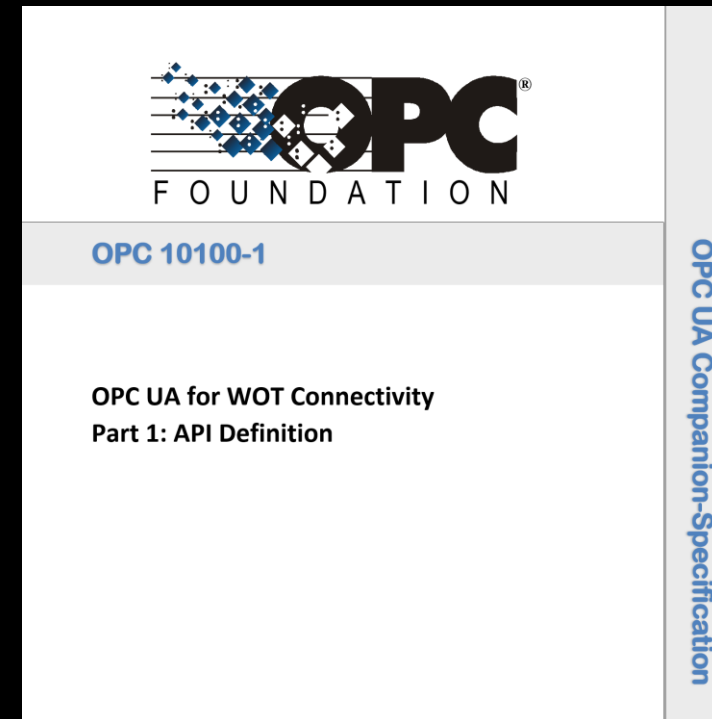
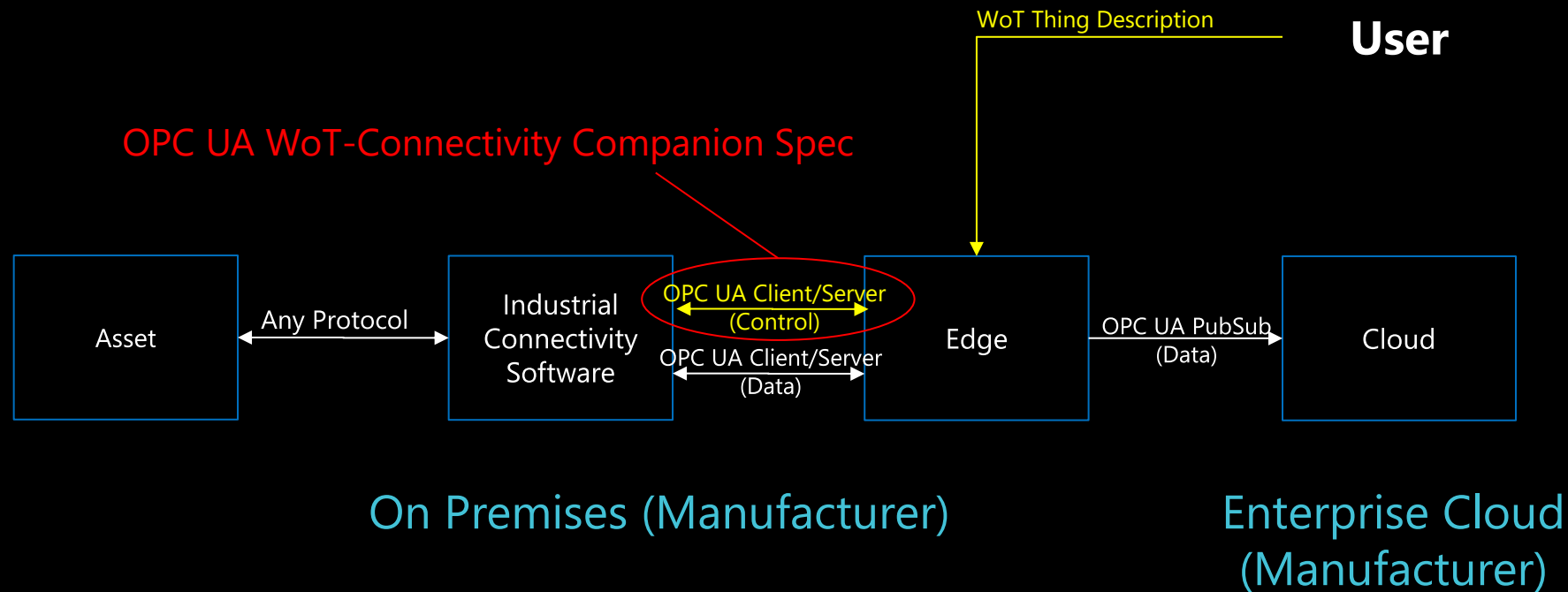
Primary PLC Interface



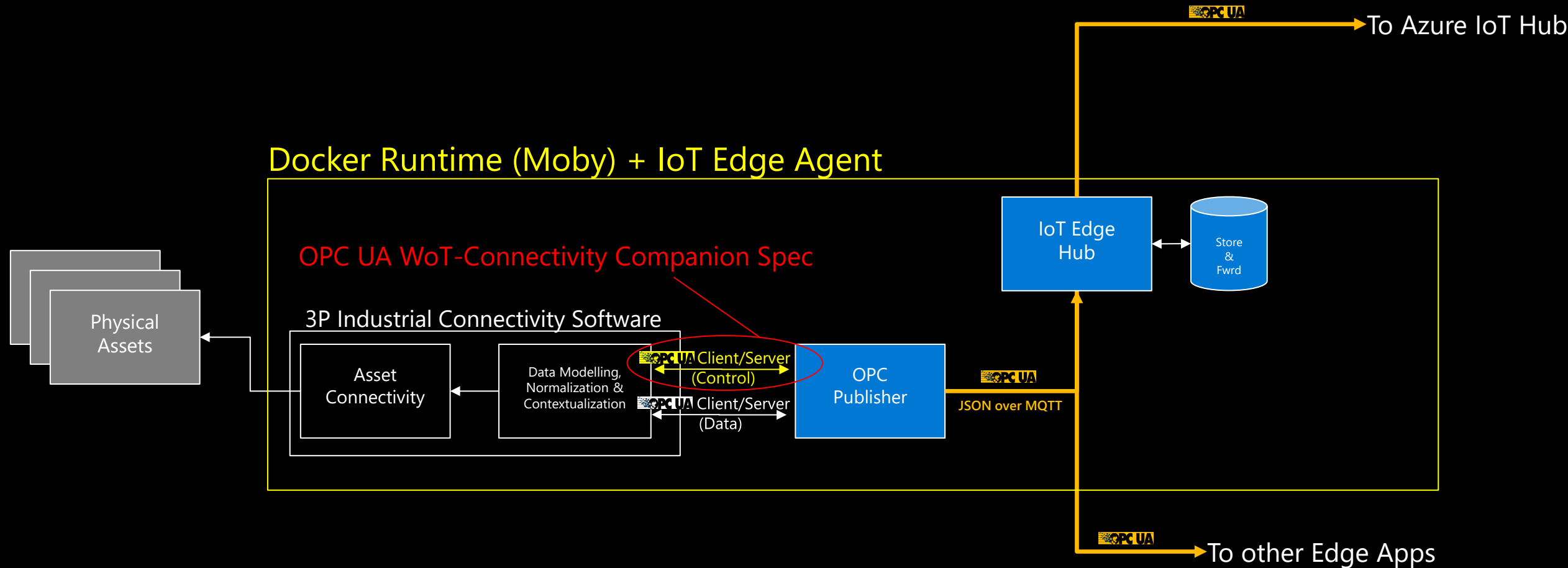
Secondary PLC Interface



# Automatic Asset Onboarding via WoT-Connectivity



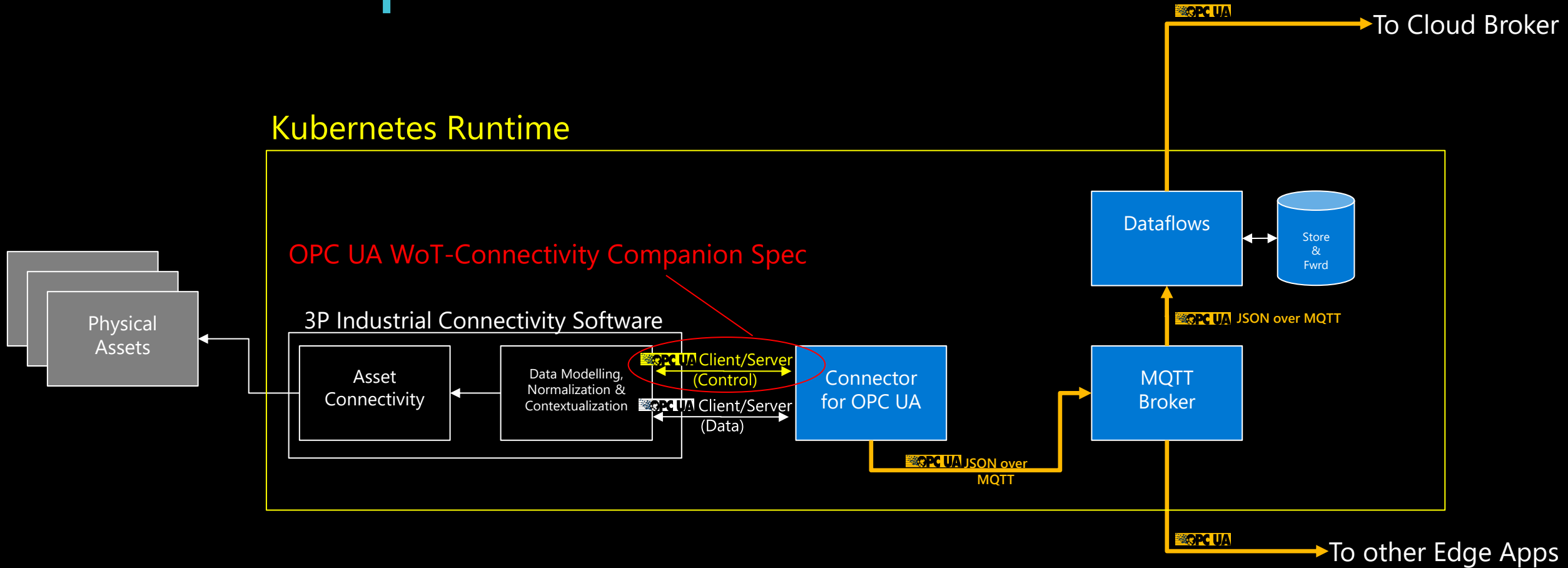
# Azure IoT Edge – The Existing Microsoft Edge Data Plane



For all modules:

- Observability (health monitoring, diagnostics, perf statistics)

# Azure IoT Operations



For all modules:

- **New:** High Availability
- Observability (health monitoring, diagnostics, perf statistics)



# OpenAI auto-generated WoT TD File

(Including support for OPC UA Companion Specifications, manually added)



```
pac4200.jsonld
Schema: https://json.schemastore.org/jsonld.json
1 {
2   "@context": [
3     "https://www.w3.org/2019/wot/td/v1",
4     "https://si-ra.github.io/ontologies/td-context.jsonld",
5     "http://opcfoundation.org/UA/PNEM/"
6   ],
7   "id": "urn:pac4200",
8   "securityDefinitions": {
9     "nosec_sc": {
10       "scheme": "nosec"
11     }
12   },
13   "security": [
14     "nosec_sc"
15   ],
16   "@type": [
17     "Thing"
18   ],
19   "name": "modbus-pac4200-sn324",
20   "base": "modbus://192.168.10.100:502",
21   "title": "Siemens SENTRON PAC4200",
22   "properties": {
23     "VoltageL1-N": {
24       "type": "number",
25       "readOnly": true,
26       "observable": true,
27       "forms": [
28         {
29           "href": "/1?address=1&quantity=2",
30           "op": [
31             "readproperty",
32             "observeproperty"
33           ],
34           "opcua:type": "nsu=http://opcfoundation.org/UA/PNEM/;i=6098",
35           "modbus:type": "float",
36           "modbus:entity": "holdingregister",
37           "modbus:pollingTime": 2000
38         }
39       ]
40     },
41     "VoltageL2-N": {
42       "type": "number",
43       "readOnly": true
44     }
45   }
46 }
```

# User edits auto-generated WoT Thing Description in online editor

W3C  
THINGS

ediTDor

GPT

Share

Discover

New

Open

Save

Persist As File

JSON Validation ✓

JSON Schema Validation ✓

SiemensSENTRONPAC4200

idurn:pacPVDAS1

@context >

namePac4200

mlfb7KM3220-1BA01-1EA0

basemodbus://host:port

securityDefinitions >

securitynosec\_sc

> Forms

Add Top Level Form

> Links

Add Top Level Link

Properties

> VoltageL1-N

> VoltageL2-N

> VoltageL3-N

Actions

Search Properties

↓

Add

Search Actions

↓

Add

Properties: 3

Actions: 0

Events: 0

Size: 1.034 KiB

SiemensSENTRONPAC4200

```
1 {
2   "@context": ["https://www.w3.org/2019/wot/td/v1"],
3   "title": "SiemensSENTRONPAC4200",
4   "id": "urn:pacPVDAS1",
5   "name": "Pac4200",
6   "mlfb": "7KM3220-1BA01-1EA0",
7   "base": "modbus://host:port",
8   "securityDefinitions": {
9     "nosec_sc": {
10       "scheme": "nosec"
11     }
12   }
13 }
```

Ask Your AI Assistant

Send a message and it will be forwarded to a GPT instance. It will assist you in writing your TDs

Generate me a Thing Description for a Siemens Sentron PAC 4200

```
{
  "@context": ["https://www.w3.org/2019/wot/td/v1"],
  "title": "SiemensSENTRONPAC4200",
  "id": "urn:pacPVDAS1",
  "name": "Pac4200",
  "mlfb": "7KM3220-1BA01-1EA0",
  "base": "modbus://host:port",
  "securityDefinitions": {
    "nosec_sc": {
      "scheme": "nosec"
    }
  },
  "security": "nosec_sc",
  "properties": {
    "VoltageL1-N": {
      "description": "voltage between L1 and N IV1"
    }
  }
}
```

Copy last message to editor

Start typing...

↩

You have unsaved changes. - version: 0.0.3 | [view on Github](#)

[eclipse.github.io/editdor](https://eclipse.github.io/editdor)

# Reference Implementation (open-source, MIT licensed)

[github.com/OPCFoundation/UA-EdgeTranslator](https://github.com/OPCFoundation/UA-EdgeTranslator)

## Supports:

1. WoT-Connectivity
2. Docker & Kubernetes
3. Separate WoT file generator tool
4. Loading OPC UA Companion Specs
5. UA Cloud Library downloads
6. Mapping to OPC UA complex types
7. Reading from and writing to asset tags
8. OPC UA Server interface (northbound)
9. GDS Server Push certificate management & provisioning mode
10. Southbound protocols: Modbus TCP, BACnet, S7Comm, Ethernet/IP, Mitsubishi MC, Beckhoff ADS

# UA Edge Translator Configuration

You can use the **Azure OpenAI** service to generate and download a Web of Things (WoT) Thing Description for the asset you want to configure automatically.

[Here](#) is a good online editor for WoT files.

After validating its contents, choose the file below.

Finally, send it to UA Edge Translator via OPC UA to configure the asset.

---

1. ChatGPT prompt to automatically generate the description file for your asset (e.g. enter "Siemens Sentron PAC4200"):

Generate & Download File

---

2. Load the manually validated asset description file:

Choose File No file chosen

---

3. Send the loaded asset description file to UA Edge Translator. UA Translator address format: `opc.tcp://ipaddress:port`

Address: