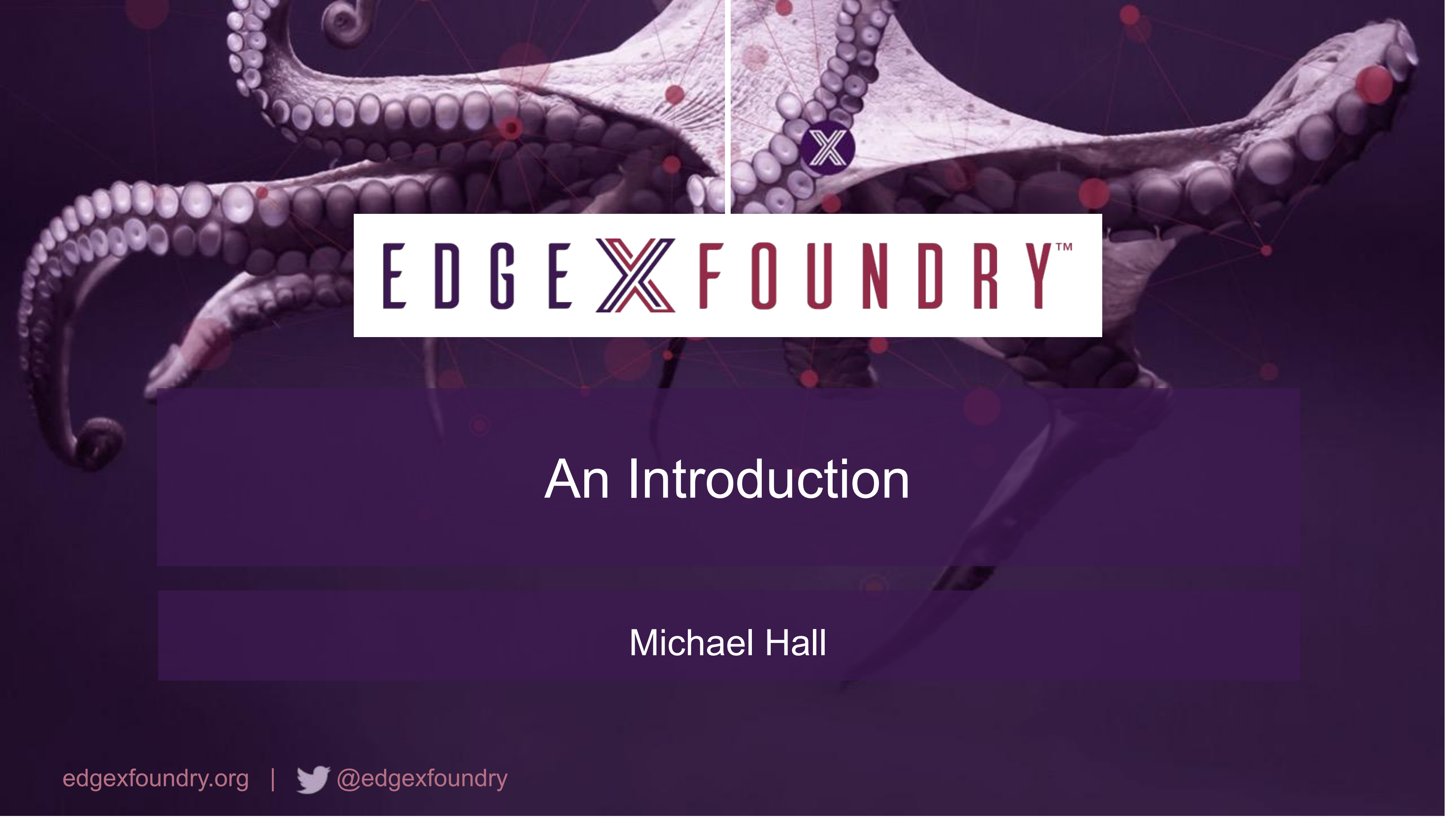


EDGEXFOUNDRY™

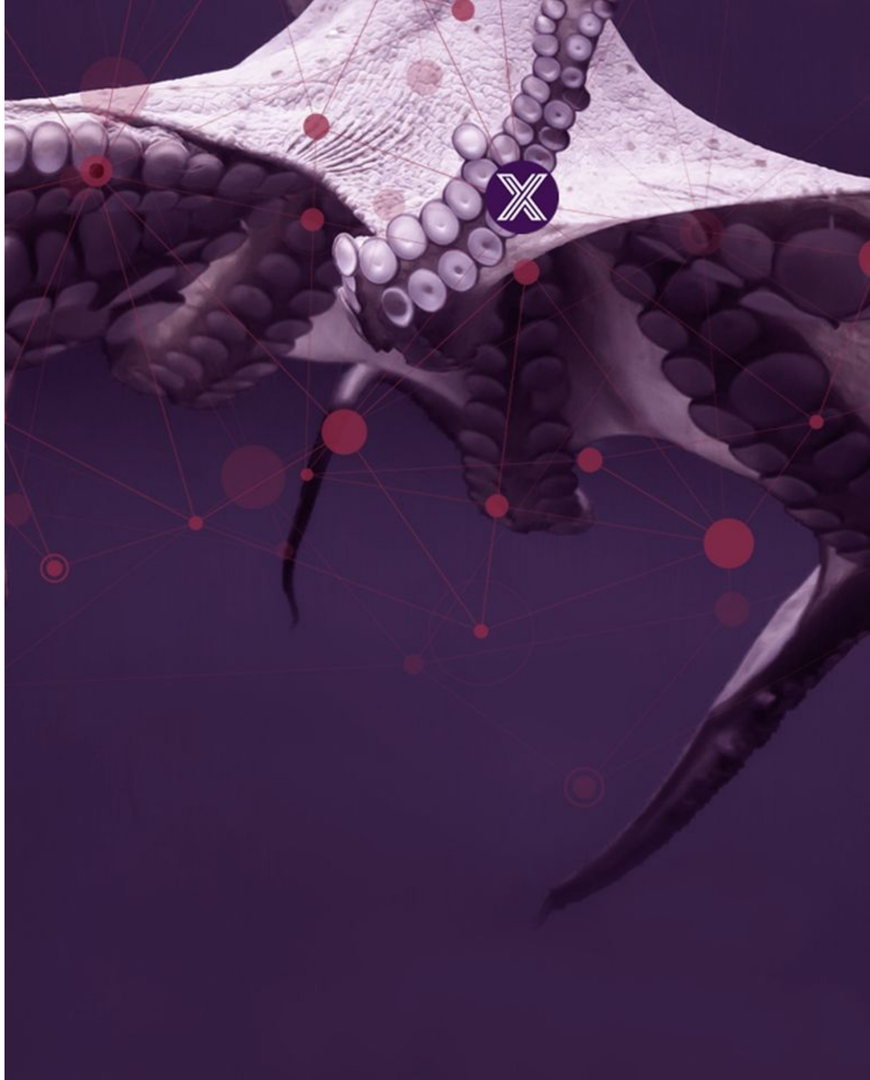
An Introduction

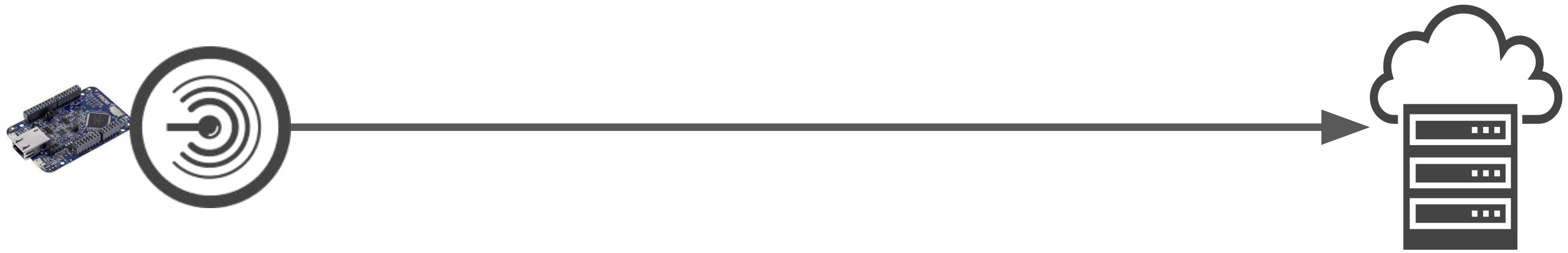
Michael Hall

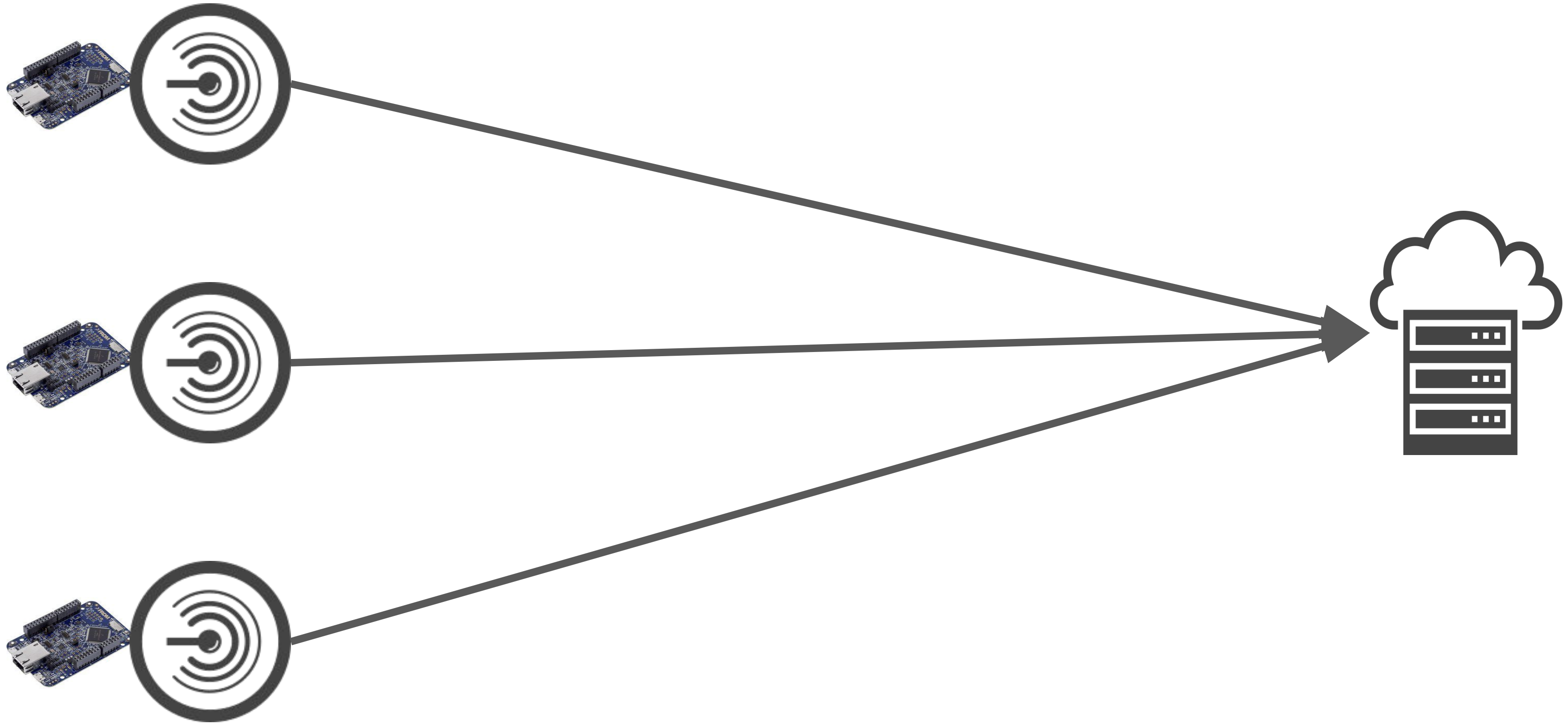
EDGEXFOUNDRY™

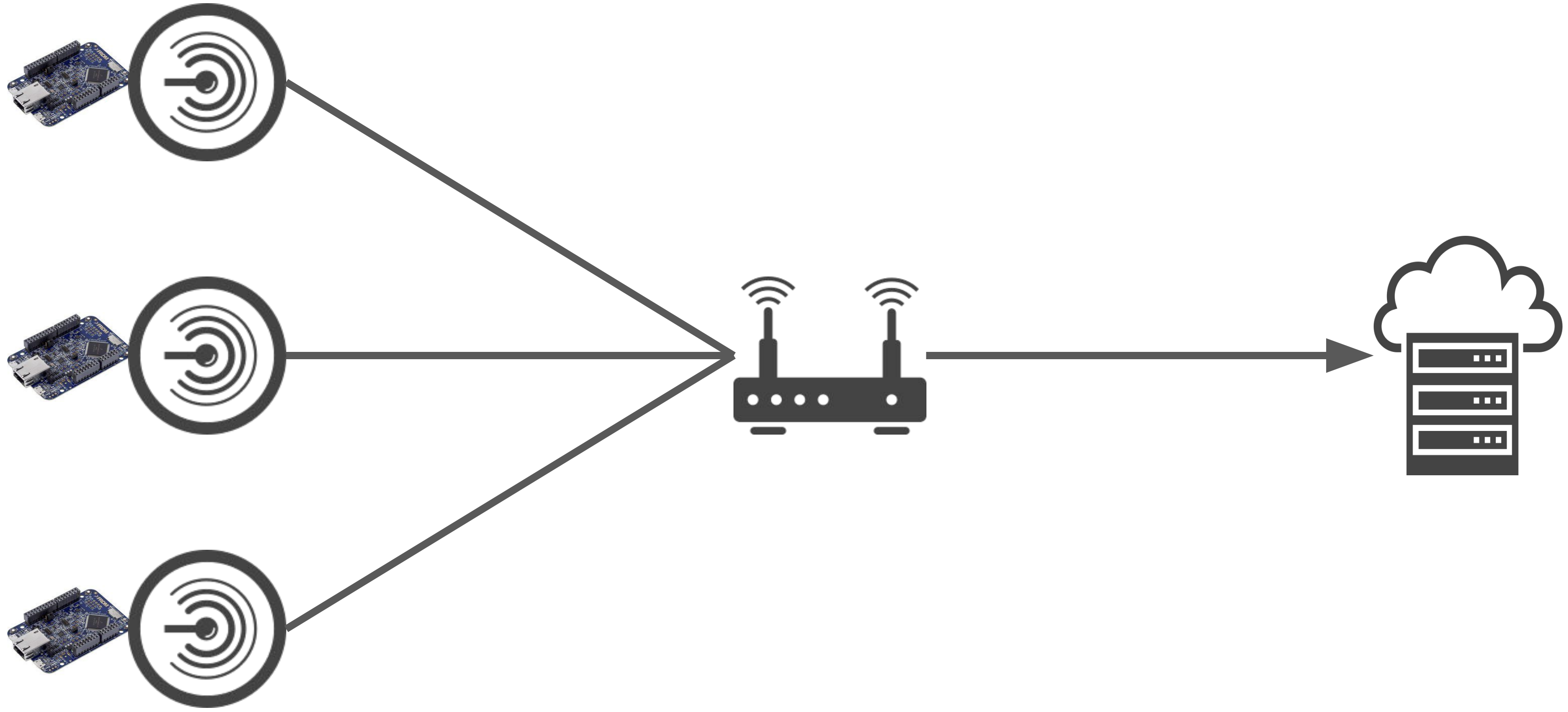
Edge Computing

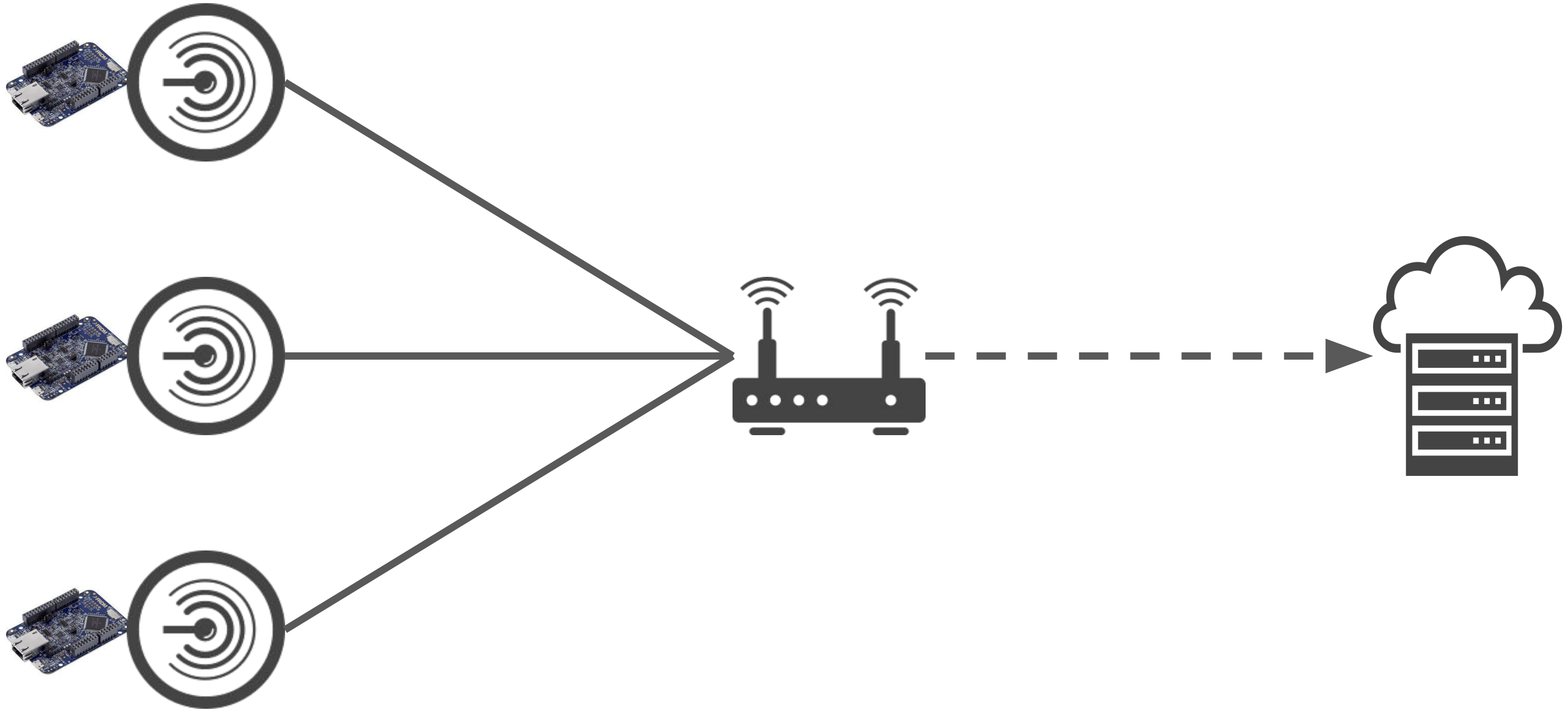
Why do I need it?

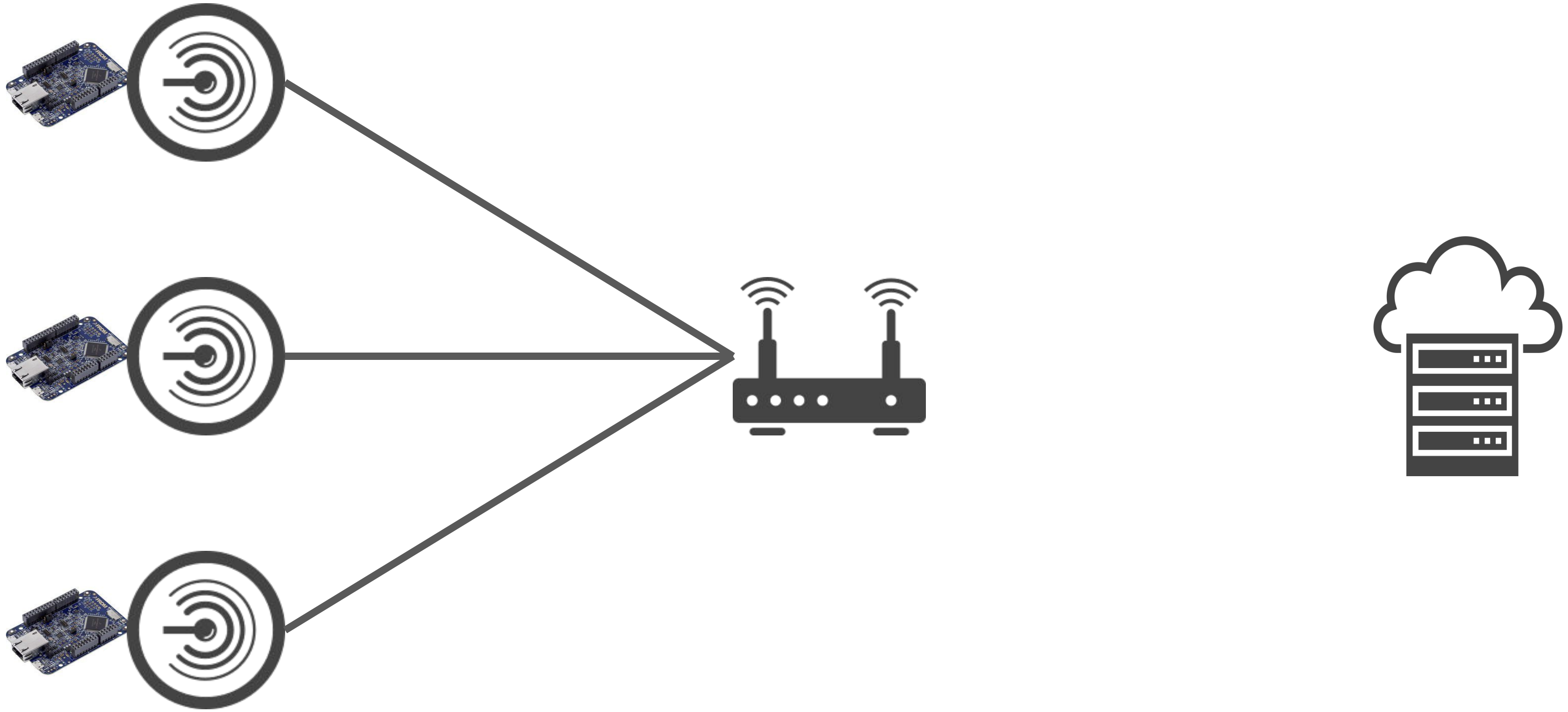


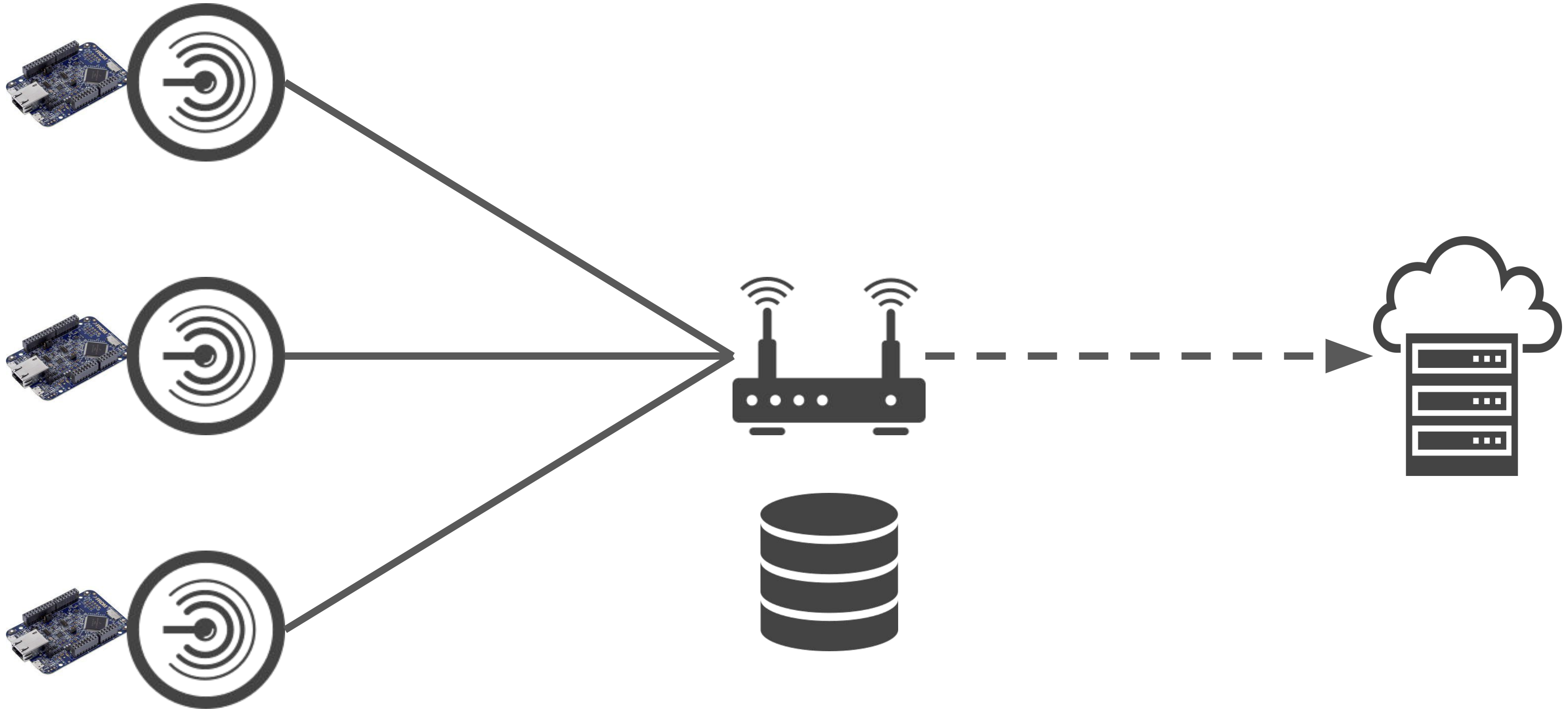


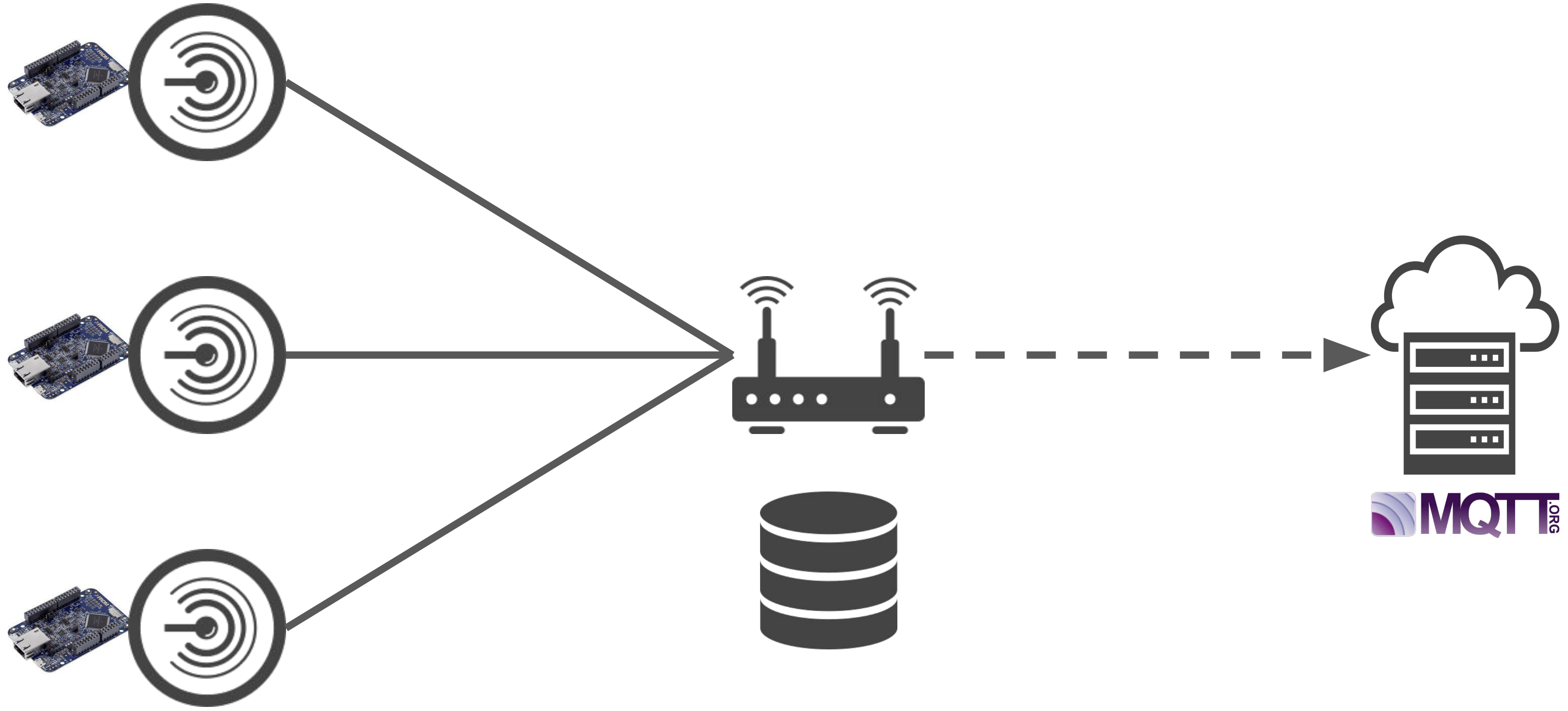


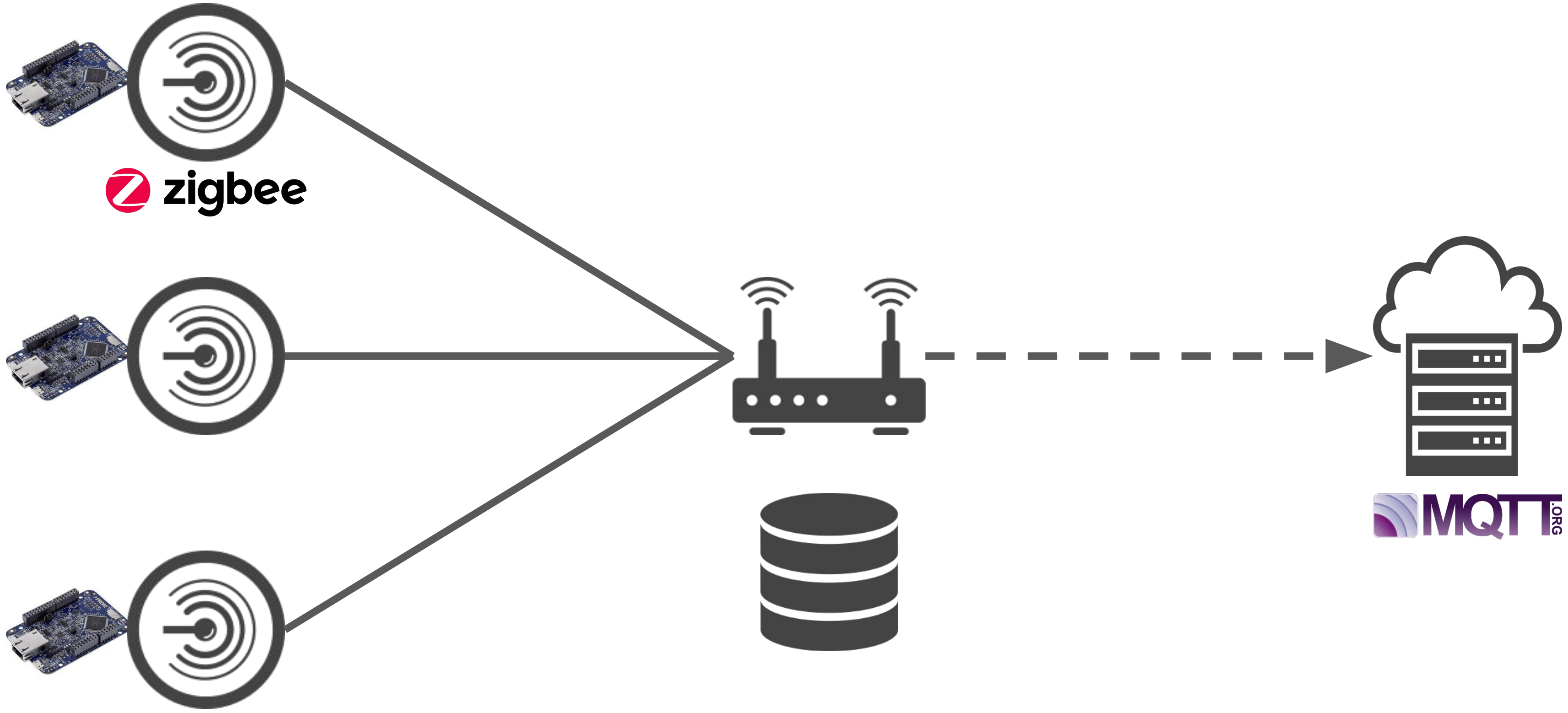


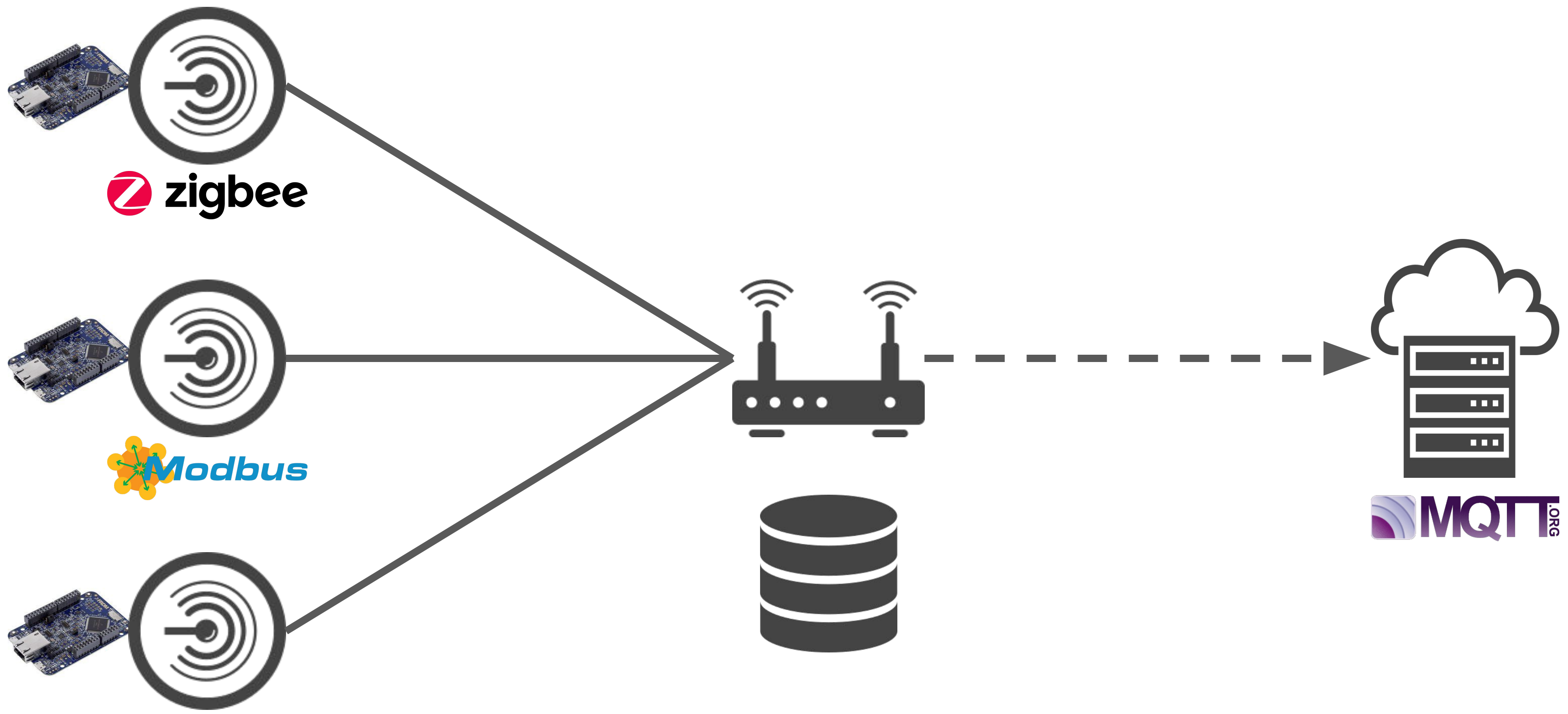


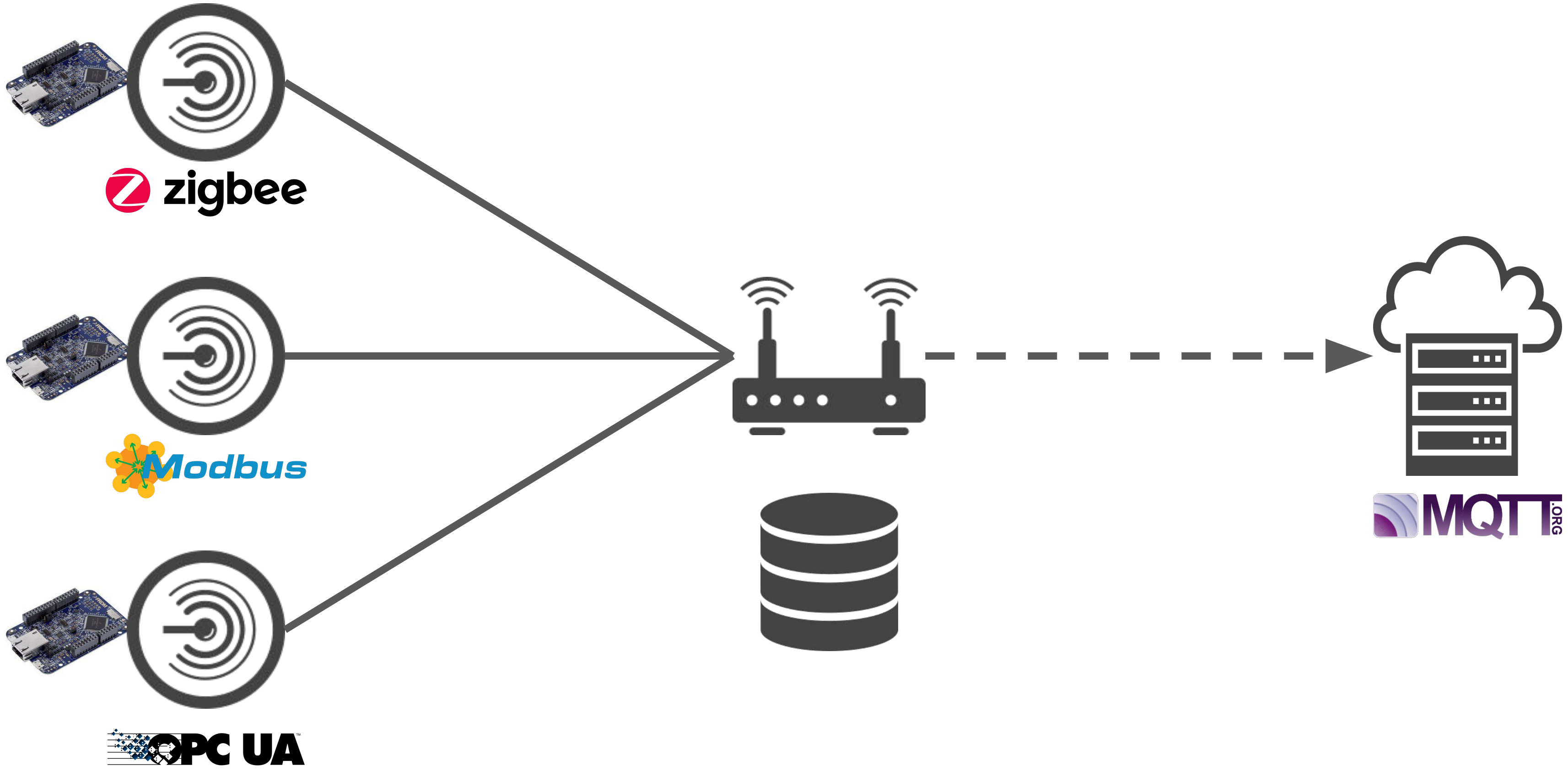


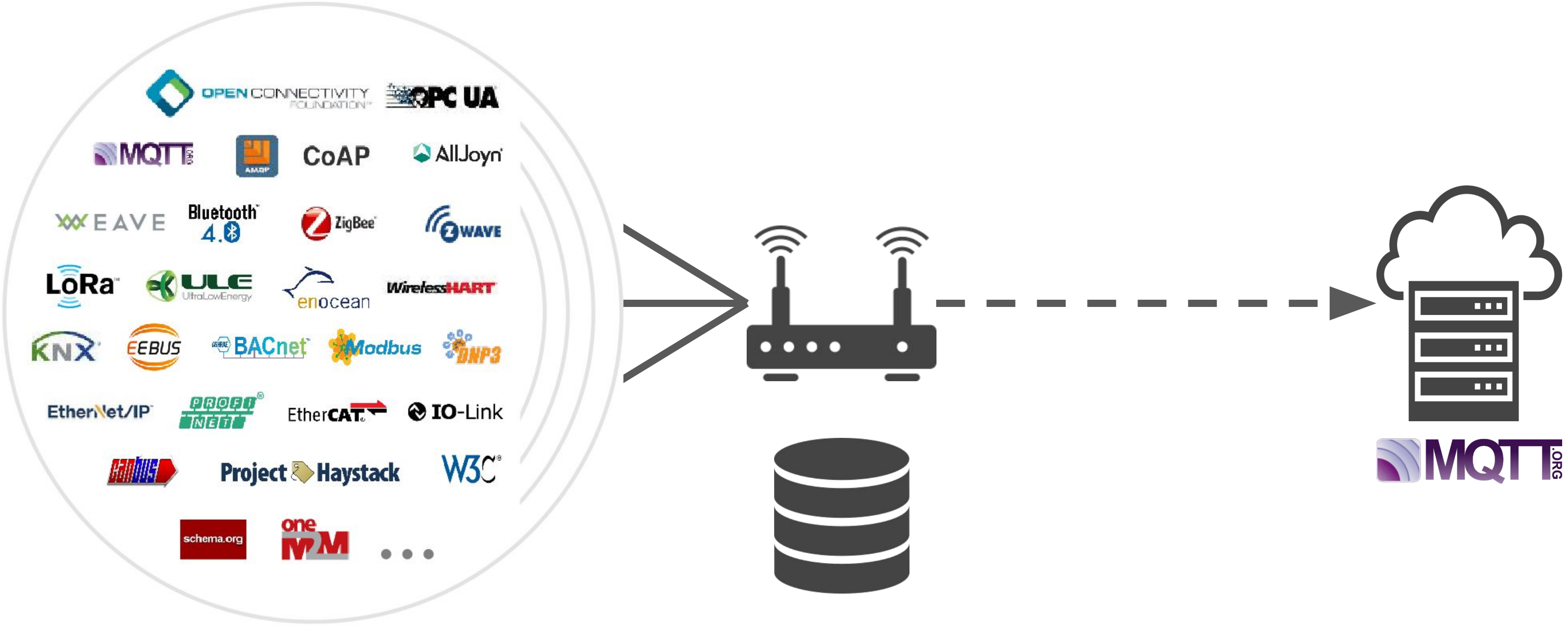


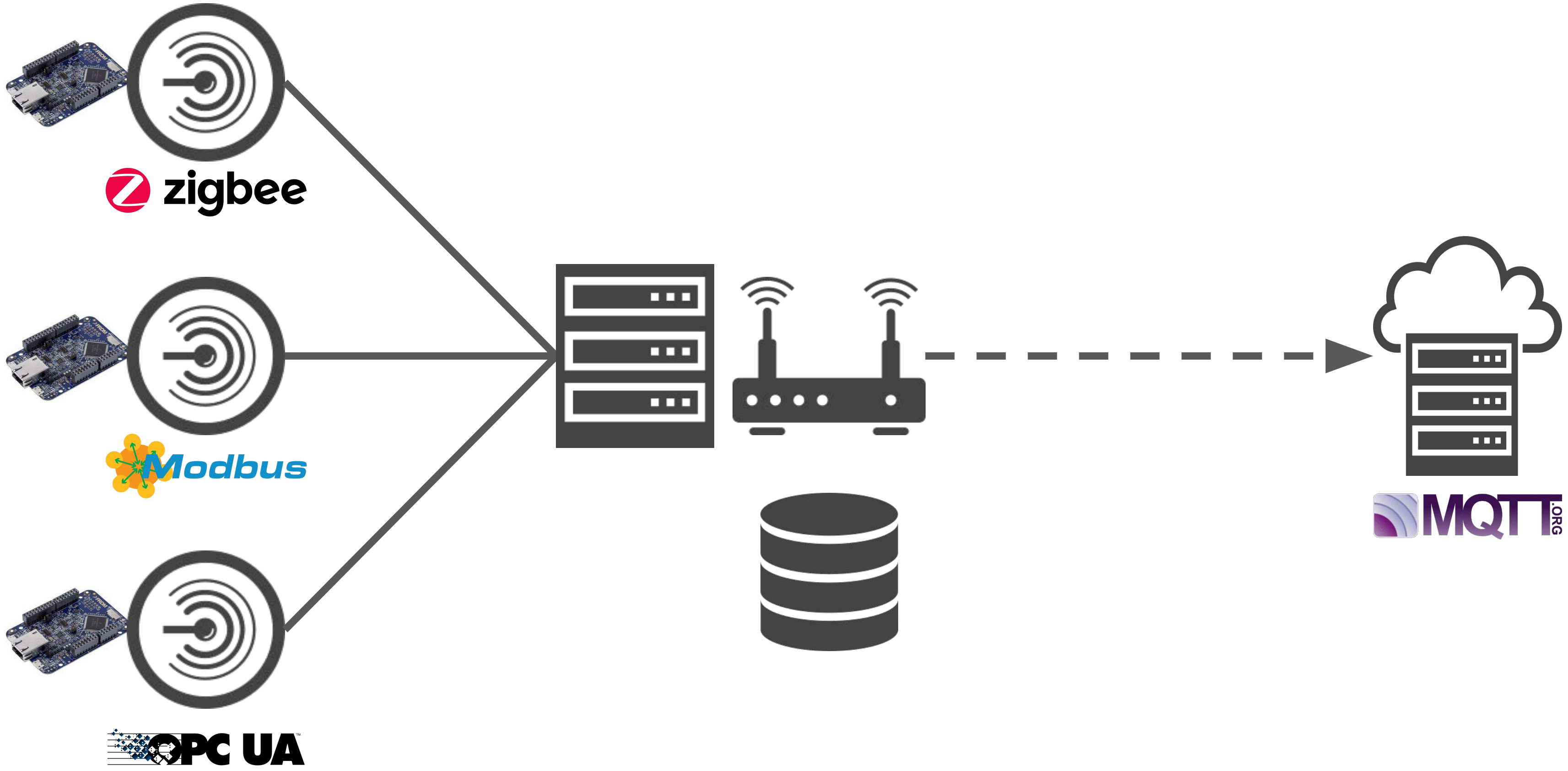

MQTT.ORG

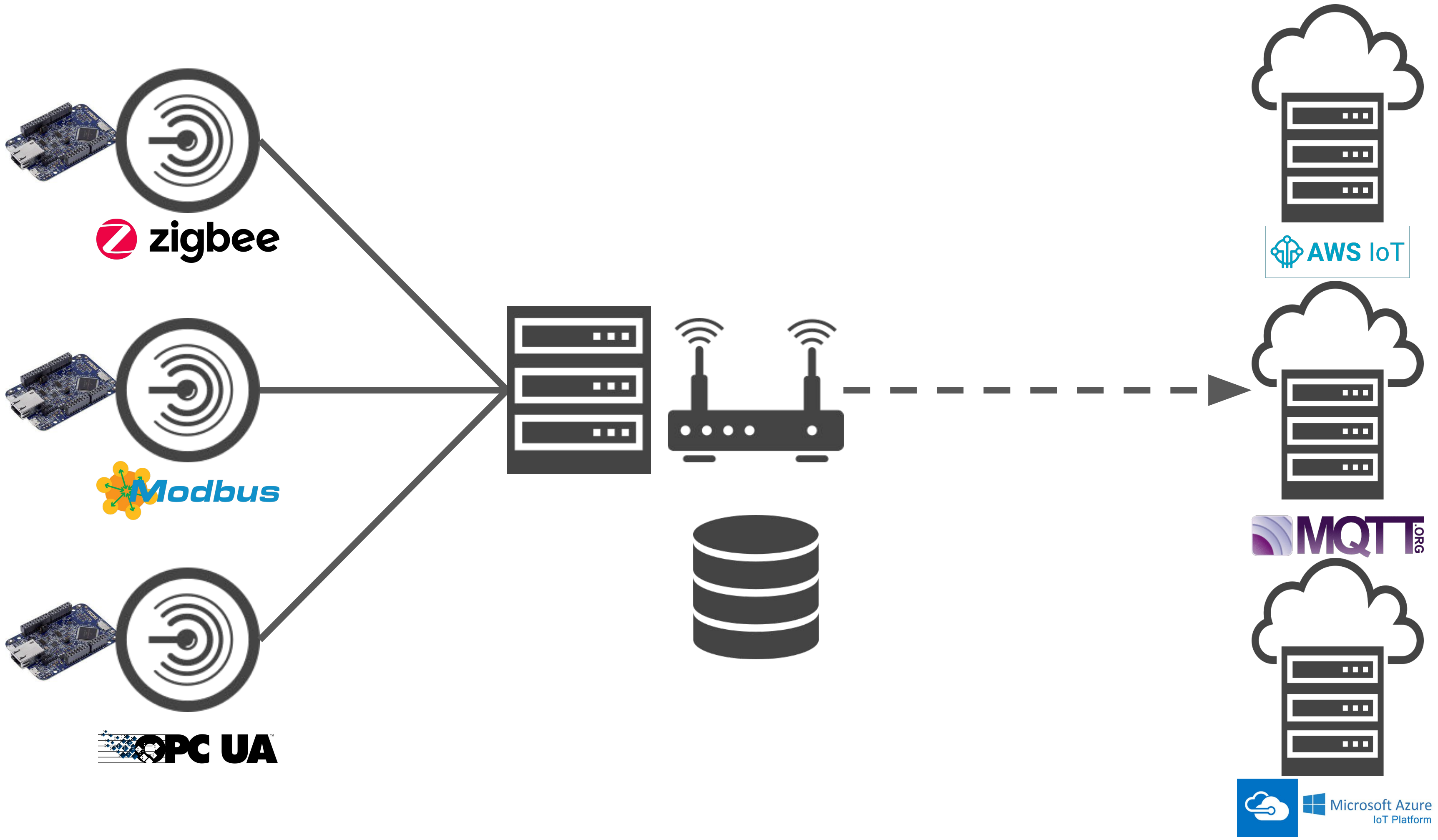

zigbee
MQTT.ORG

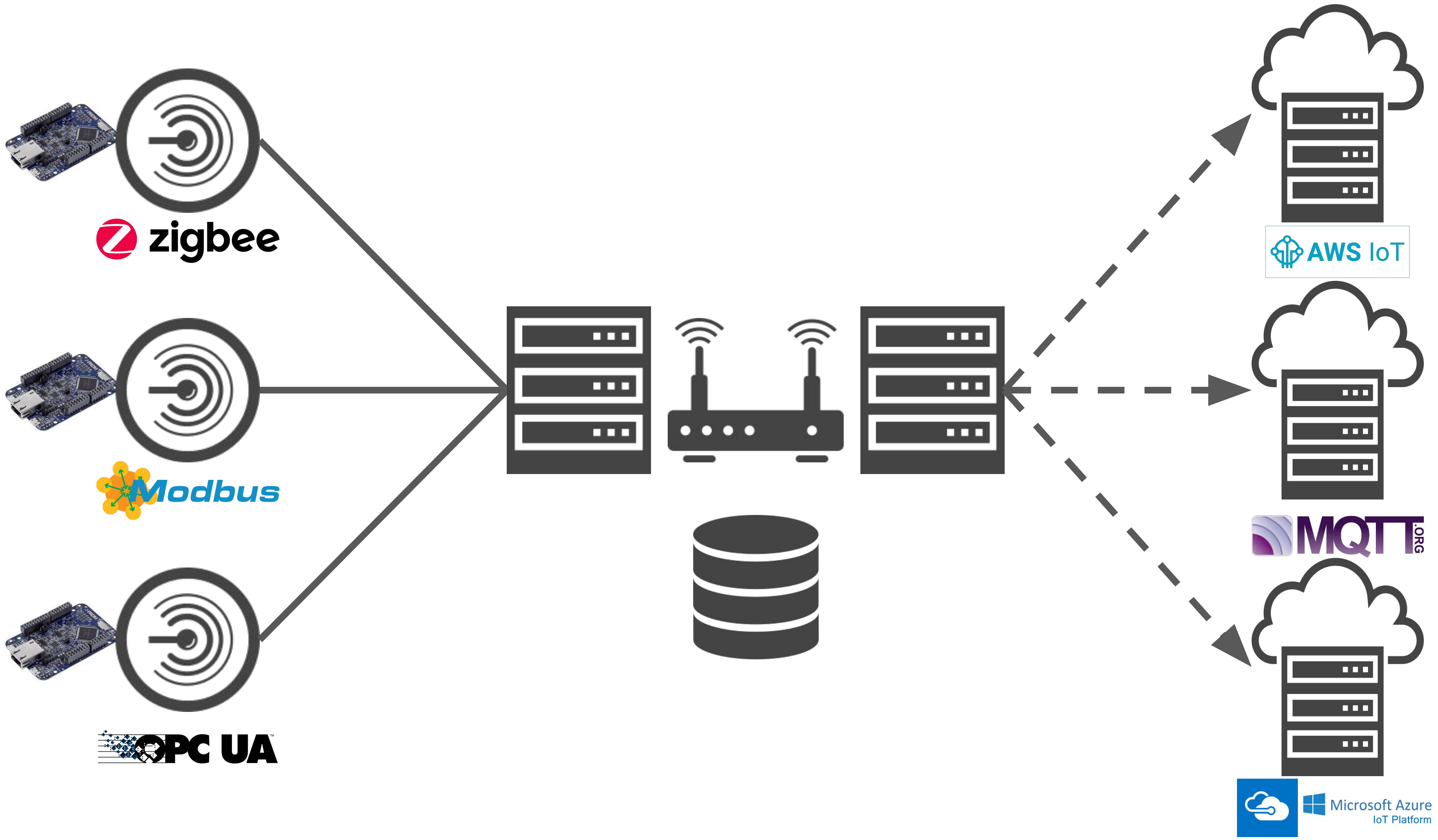

zigbee
Modbus
MQTT.ORG

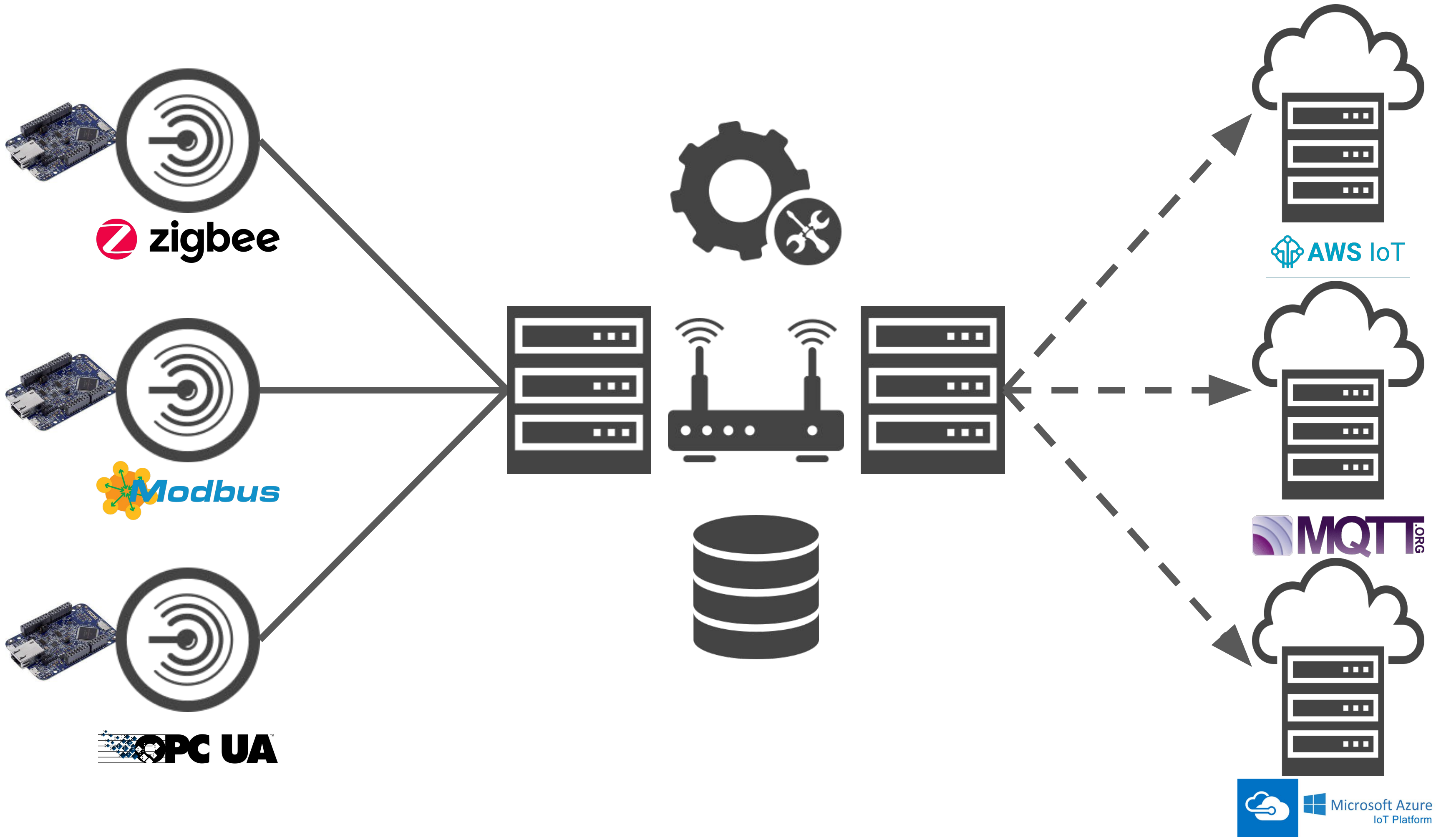

zigbee
Modbus
OPC UA
MQTT.ORG

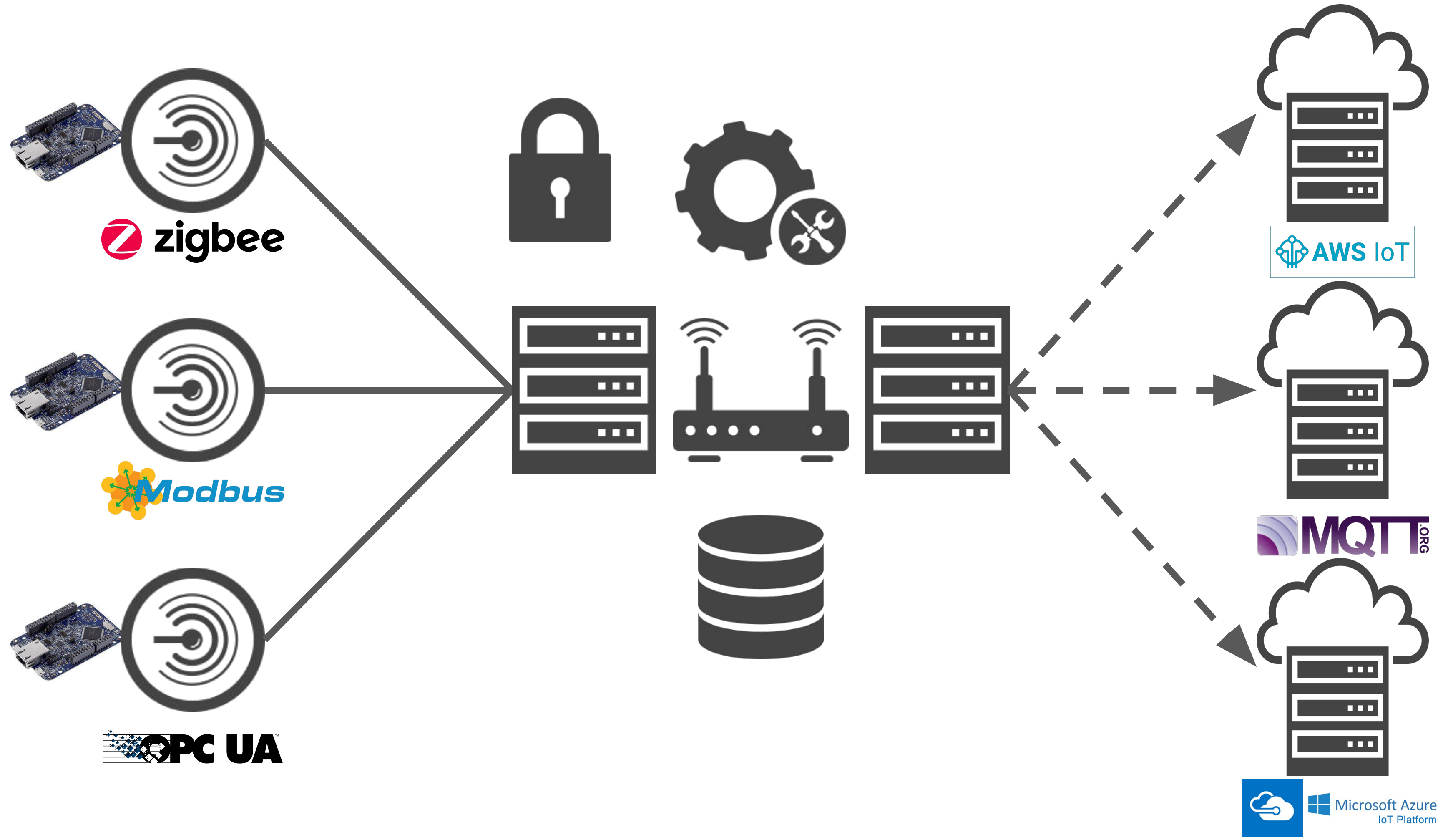

OPEN CONNECTIVITY FOUNDATION
OPC UA
MQTT.ORG
AMQP
CoAP
AllJoyn
WEAVE
Bluetooth 4.0
ZigBee
Z-WAVE
LoRa
ULE UltraLowEnergy
enocean
WirelessHART
KNX
EEBUS
BACnet
Modbus
DNP3
EtherNet/IP
PROFINET
EtherCAT
IO-Link
canbus
Project Haystack
W3C
schema.org
oneM2M
• • •
MQTT.ORG

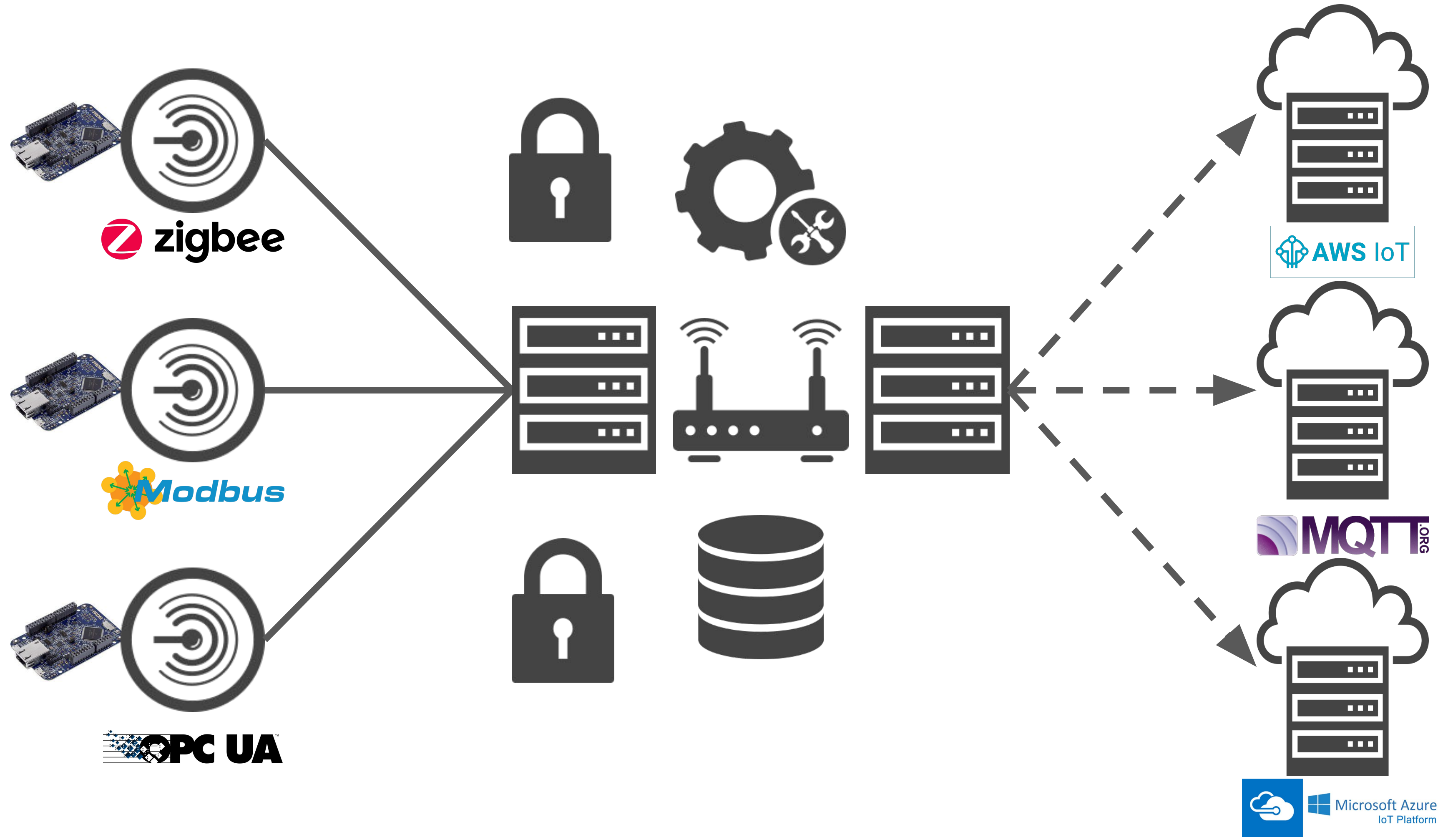

zigbee
Modbus
OPC UA
MQTT.ORG

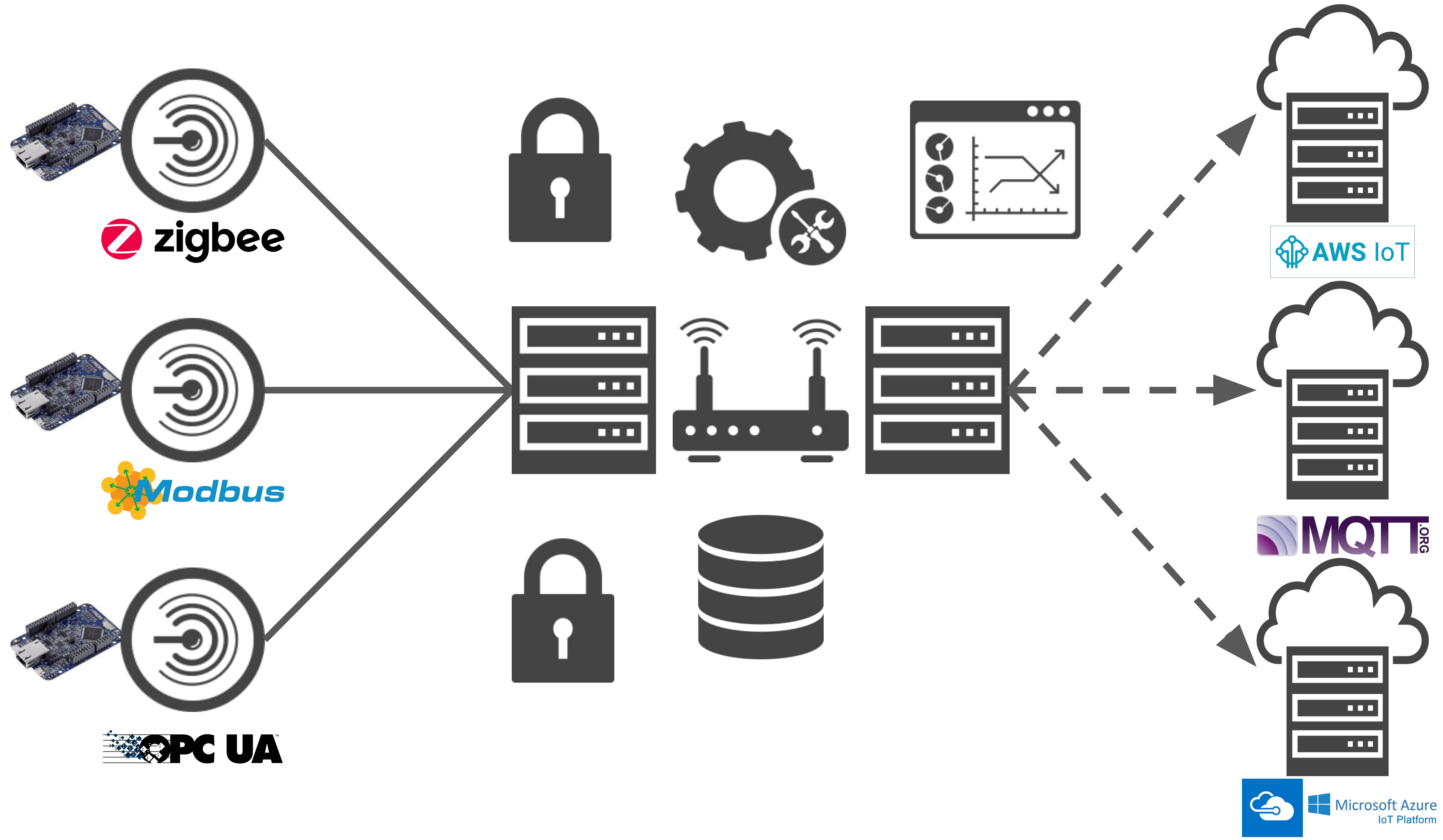

zigbee
Modbus
OPC UA
AWS IoT
MQTT.ORG
Microsoft Azure
IoT Platform

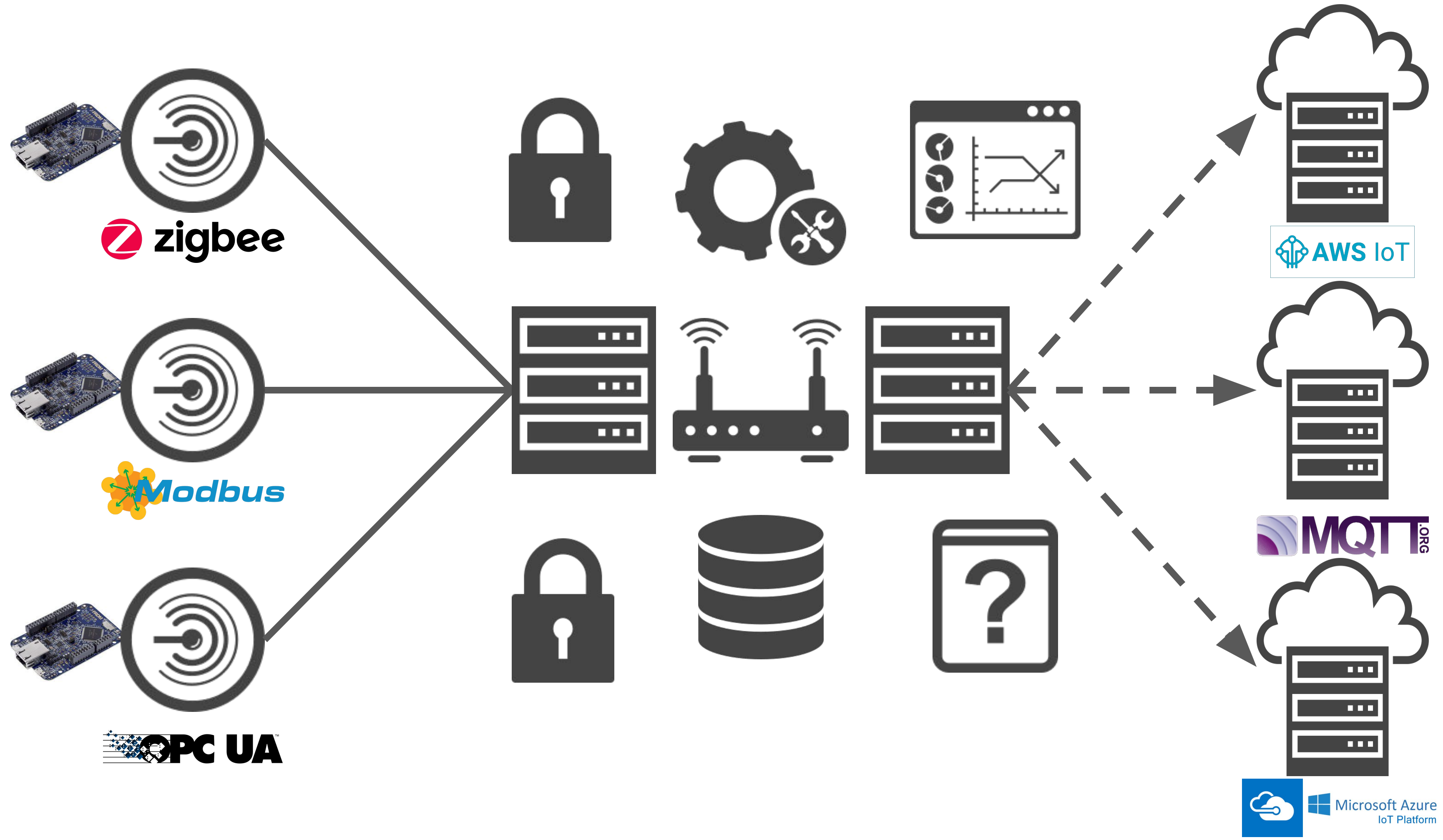

zigbee
Modbus
OPC UA
AWS IoT
MQTT.ORG
Microsoft Azure
IoT Platform

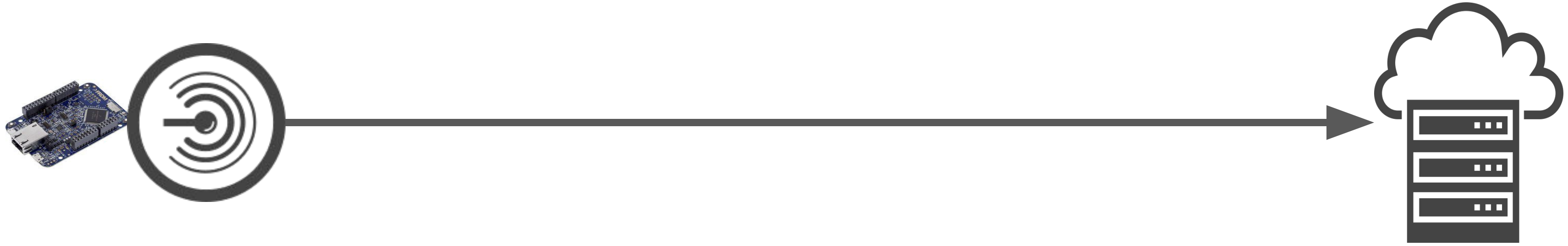

zigbee
Modbus
OPC UA
AWS IoT
MQTT.ORG
Microsoft Azure
IoT Platform

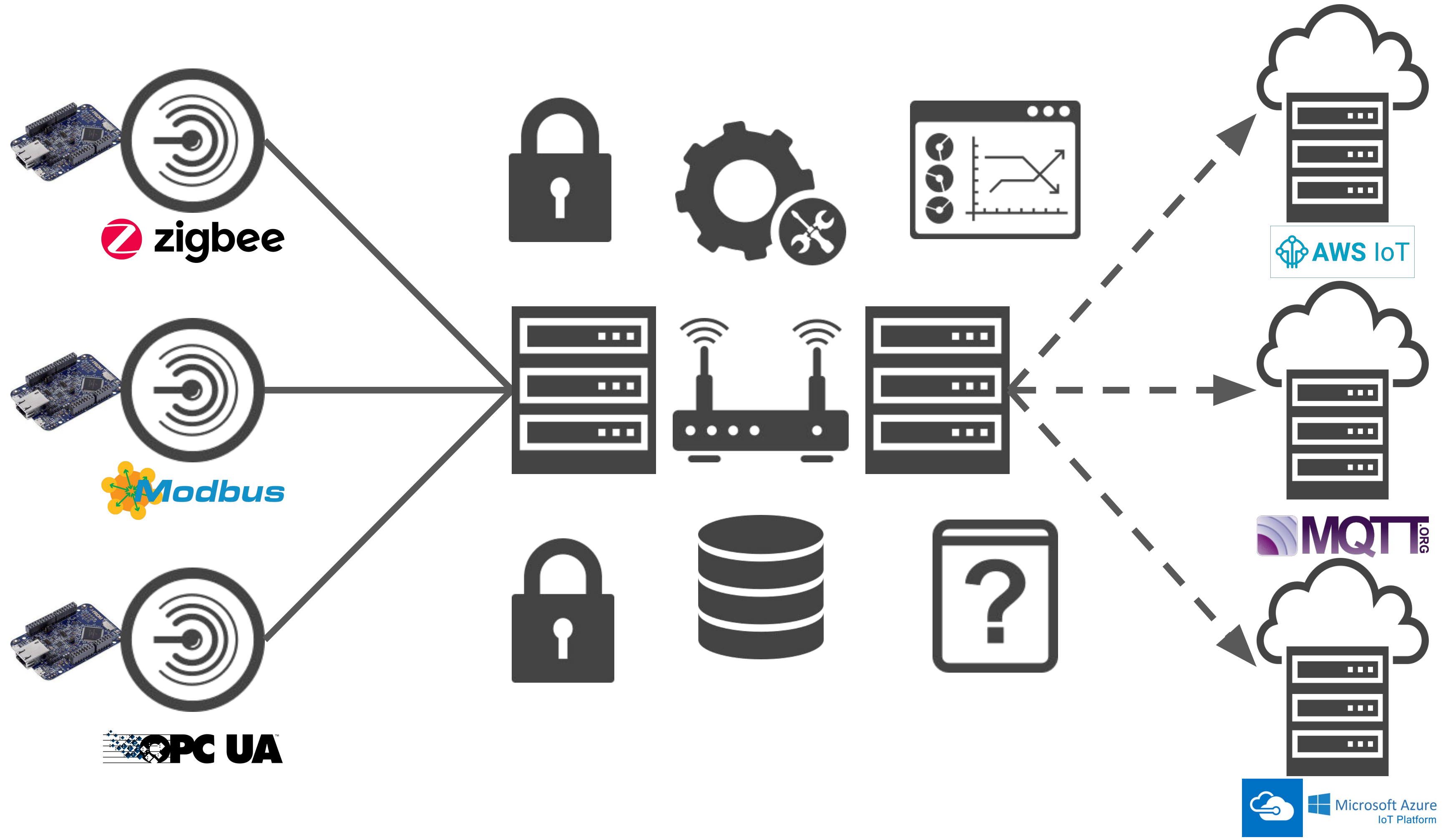

zigbee
Modbus
OPC UA
AWS IoT
MQTT.ORG
Microsoft Azure
IoT Platform


zigbee
Modbus
OPC UA
AWS IoT
MQTT.ORG
Microsoft Azure
IoT Platform


zigbee
Modbus
OPC UA
AWS IoT
MQTT.ORG
Microsoft Azure
IoT Platform

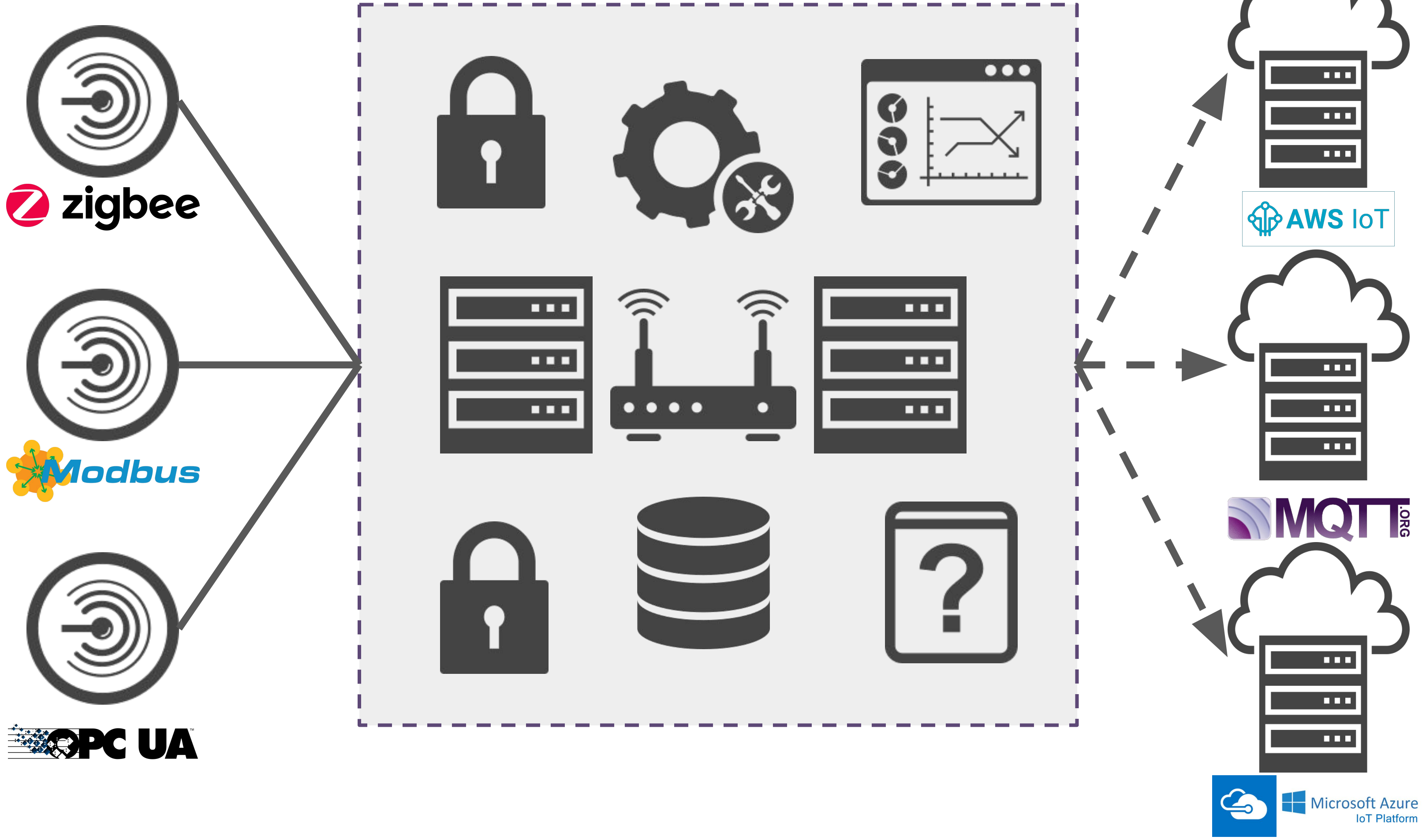

zigbee
Modbus
OPC UA
AWS IoT
MQTT.ORG
Microsoft Azure
IoT Platform




zigbee
Modbus
OPC UA
AWS IoT
MQTT.ORG
Microsoft Azure
IoT Platform

EDGE X FOUNDRY™

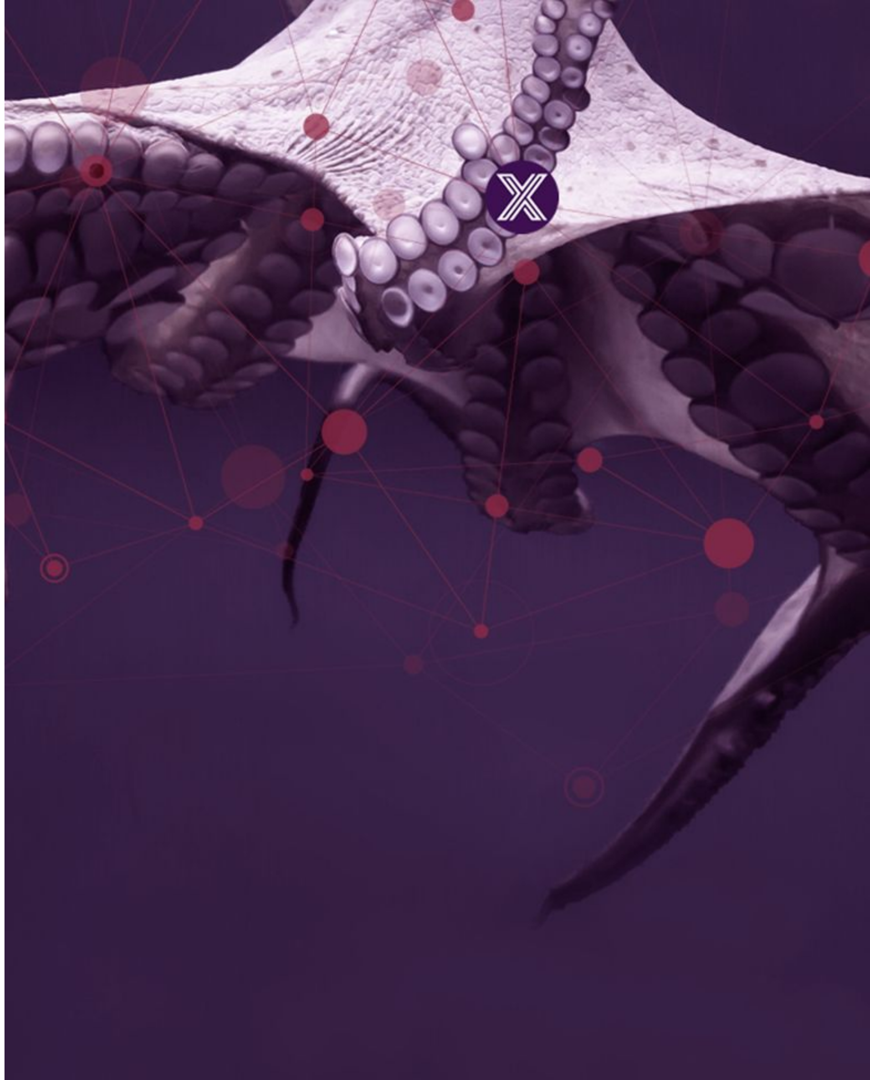
EDGE X FOUNDRY™



EDGEXFOUNDRY™

Who is EdgeX Foundry?

And how to join us



EDGE X FOUNDRY™

Vendor-neutral open source project hosted by The Linux Foundation building a common open framework for IoT edge computing.

Interoperability framework and reference platform to enable an ecosystem of plug-and-play components that unifies the marketplace and accelerates the deployment of IoT solutions.

Architected to be agnostic to protocol, silicon (e.g., x86, ARM), OS (e.g., Linux, Windows, Mac OS), and application environment (e.g., Java, JavaScript, Python, Go Lang, C/C++) to support customer preferences for differentiation

Part of the **LF Edge** project at the Linux Foundation

LF Edge Premium Members

Aricent ALTRAN GROUP · arm · AT&T · Baidu 百度 · DELL EMC · DIANOMIC

ERICSSON · hp · Hewlett Packard Enterprise · HUAWEI · IBM · intel

inwinSTACK · JUNIPER NETWORKS · MobiledgeX · NETSIA · NOKIA · NTT

OSIsoft · Qualcomm Qualcomm Technologies, Inc. · Radisys · redhat · SAMSUNG · SEAGATE

Tencent 腾讯 · WIND · wipro · ZEDEDA

LF Edge General Members

alef
Alleantia
Beechwoods
ubuntu Delivered by Canonical
CertusNet
CloudPlugs Connecting Things
DATA AHEAD
EMQ

enigmedia
EPISENSOR
FOGHORN
FORGEROCK
FOUNDRIES.IO
谐云科技 HARMONY CLOUD
IOTech
IoTium www.iotium.io

KMC CONTROLS
Linaro
MAINFLUX Internet of Things Solutions
MARVELL
MOCANA
NETFOUNDRY SPIN UP YOUR NETWORK
packet
Pluribus NETWORKS

RackN
redislabs home of redis
REPLY
Section
Vapor
vitro
Volterra EDGE SERVICES
萬向集團 WANXIANG GROUP

Associate Members

AECC AUTOMOTIVE EDGE COMPUTING CONSORTIUM
北京郵電大學 Beijing University of Posts and Telecommunications
ETRI Electronics and Telecommunications Research Institute
INFRASTRUCTURE MASONS
Project Haystack

Getting Involved

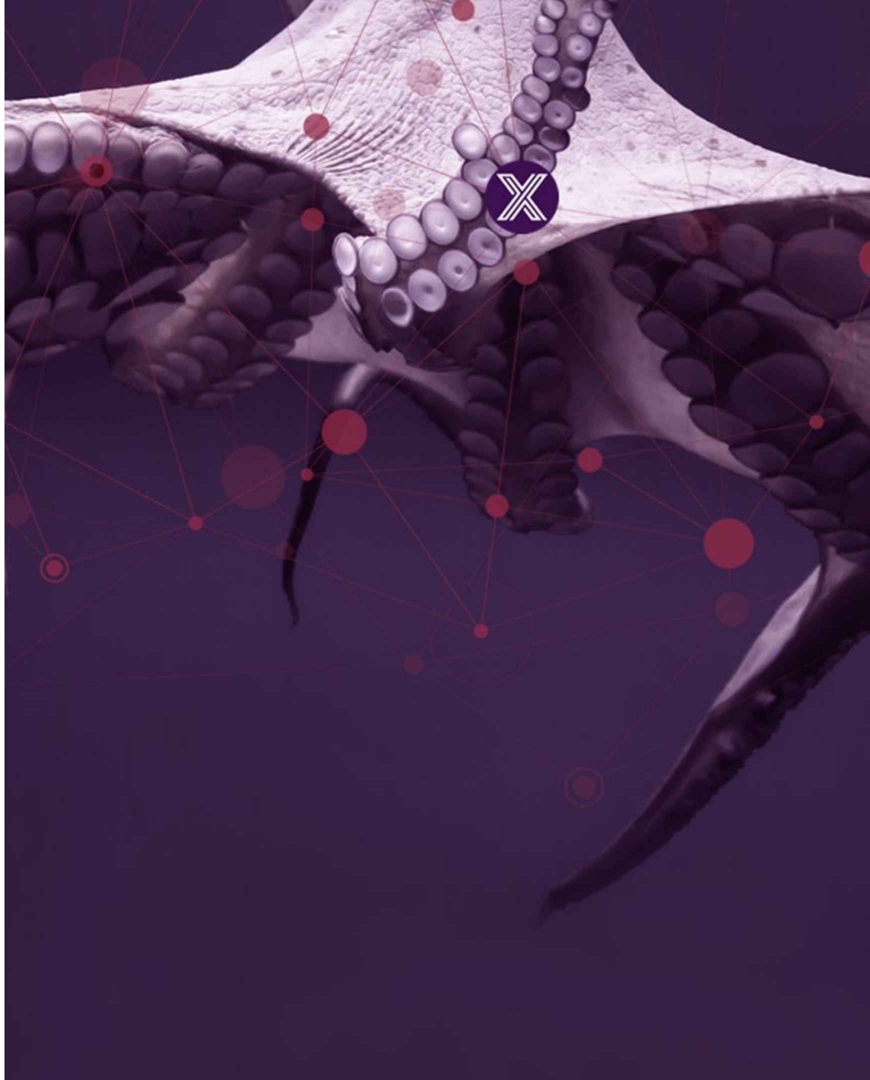
- Open Source and contributor driven, anybody can participate
- TSC and WG meetings open to public
- Technical leadership (TSC & WG chairs) elected by technical contributors

- GitHub:
  - https://github.com/edgexfoundry
- Documentation
  - https://docs.edgexfoundry.org
- Slack
  - https://slack.edgexfoundry.org
- Mailing Lists
  - https://lists.edgexfoundry.org
  - https://lists.edgexfoundry.org/calendar

EDGE X FOUNDRY™

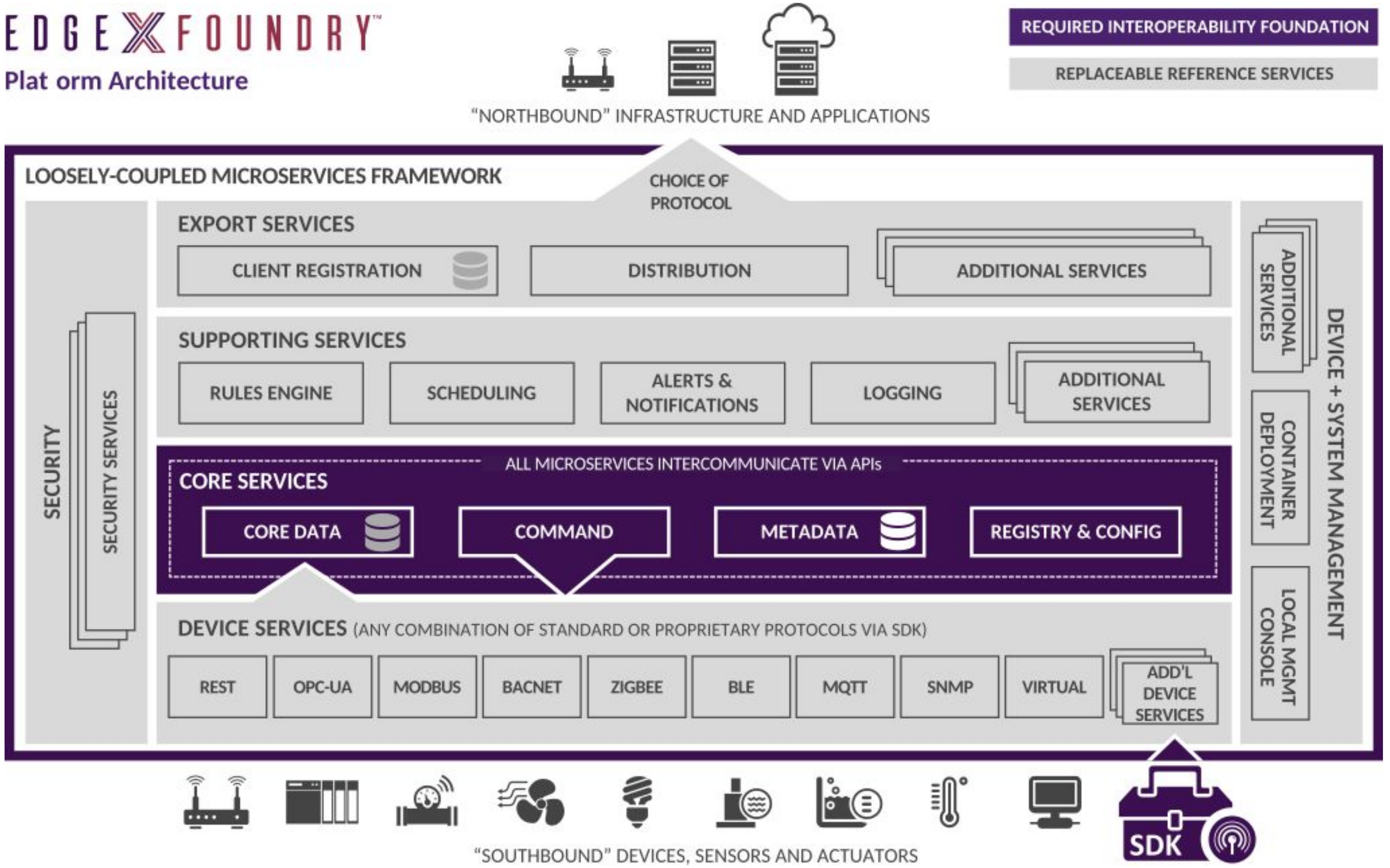
What is EdgeX?

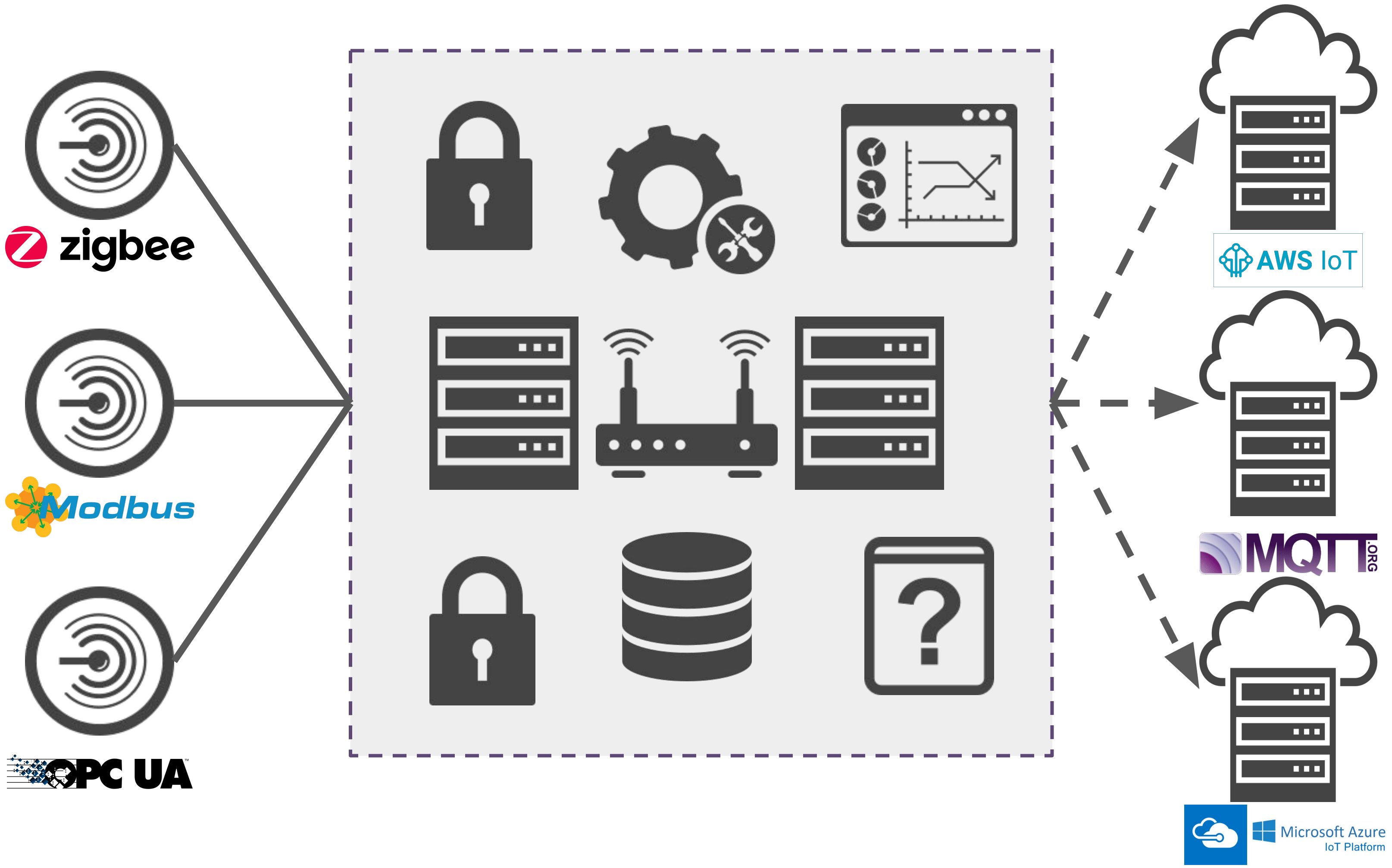
Microservices and Deployments

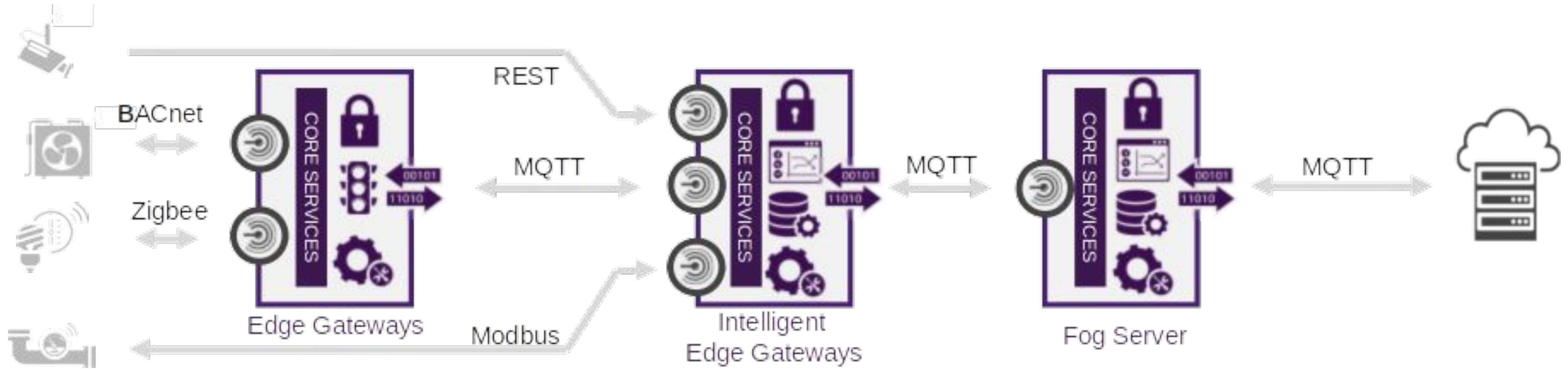


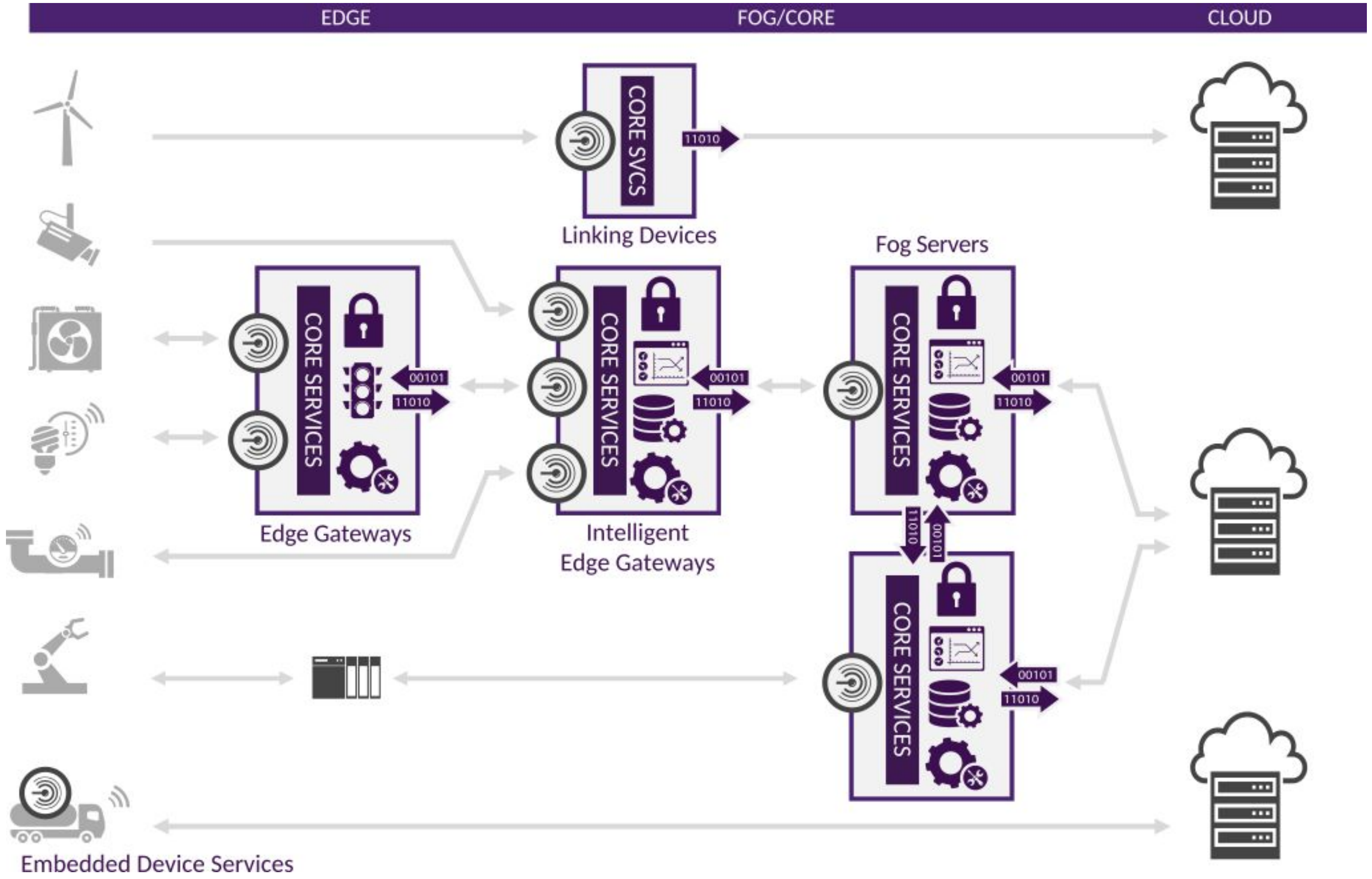
EDGE X FOUNDRY™

Plat orm Architecture




zigbee
Modbus
OPC UA
AWS IoT
MQTT.ORG
Microsoft Azure
IoT Platform


REST
BACnet
Zigbee
CORE SERVICES
00101
11010
MQTT
Edge Gateways
Modbus
CORE SERVICES
Intelligent Edge Gateways
MQTT
CORE SERVICES
Fog Server
MQTT



EDGE
FOG/CORE
CLOUD
CORE SVCS
11010
Linking Devices
Fog Servers
CORE SERVICES
00101
11010
Edge Gateways
Intelligent
Edge Gateways
Embedded Device Services

Hardware OEMs

(Examples: Controllers, Hubs, Routers, Gateways, Servers)

Scale faster with an interoperable partner ecosystem and more robust security and system management

ISVs

Interoperate with 3rd party applicatons and hardware without reinventng connectvity

Sensor/Device Manufacturers

Write a device driver with your selected protocol once using the SDK and get pull from all Solution Providers

System Integrators

Get to market faster with plug-and-play ingredients combined with your own innovatons

End Customers: Less confusion and faster time to ROI

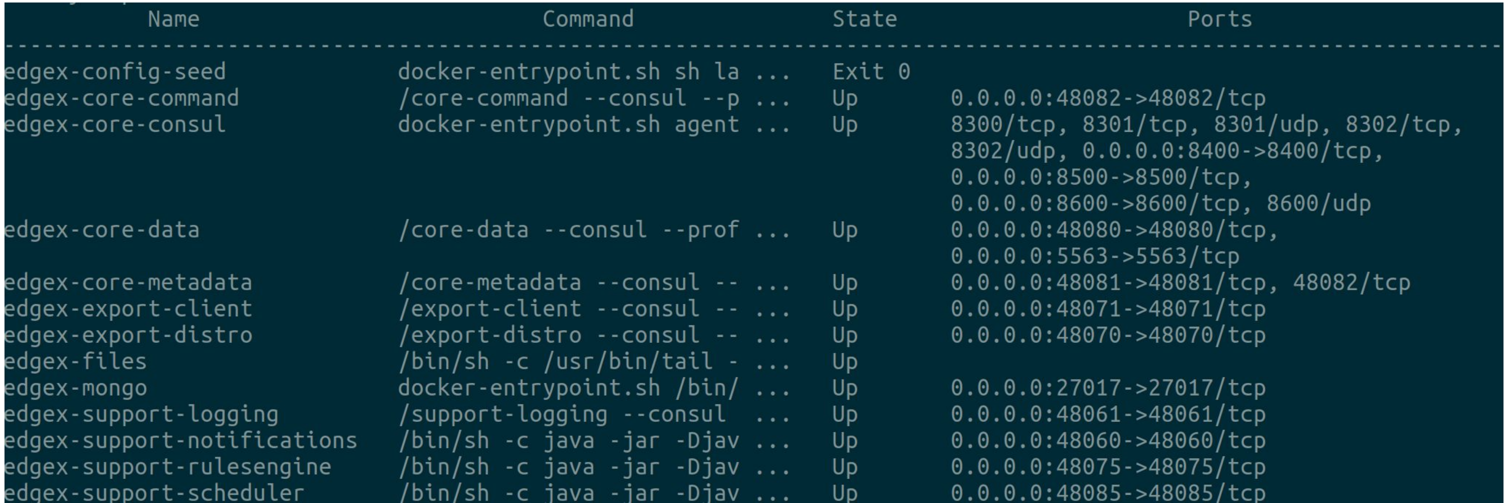
EDGE X FOUNDRY™

Walkthrough

Let’s see it in action

Deploying with Docker

- Install docker & docker-compose
- Fetch docker-compose.yml from developer-scripts repo
    - https://github.com/edgexfoundry/developer-scripts/tree/master/compose-files
- Run `docker-compose up -d`



```
          Name                          Command               State                   Ports
--------------------------------------------------------------------------------------------------------------
edgex-config-seed             docker-entrypoint.sh sh la ...   Exit 0
edgex-core-command            /core-command --consul --p ...   Up       0.0.0.0:48082->48082/tcp
edgex-core-consul             docker-entrypoint.sh agent ...   Up       8300/tcp, 8301/tcp, 8301/udp, 8302/tcp,
                                                                        8302/udp, 0.0.0.0:8400->8400/tcp,
                                                                        0.0.0.0:8500->8500/tcp,
                                                                        0.0.0.0:8600->8600/tcp, 8600/udp
edgex-core-data               /core-data --consul --prof ...   Up       0.0.0.0:48080->48080/tcp,
                                                                        0.0.0.0:5563->5563/tcp
edgex-core-metadata           /core-metadata --consul -- ...   Up       0.0.0.0:48081->48081/tcp, 48082/tcp
edgex-export-client           /export-client --consul -- ...   Up       0.0.0.0:48071->48071/tcp
edgex-export-distro           /export-distro --consul -- ...   Up       0.0.0.0:48070->48070/tcp
edgex-files                   /bin/sh -c /usr/bin/tail - ...   Up
edgex-mongo                   docker-entrypoint.sh /bin/ ...   Up       0.0.0.0:27017->27017/tcp
edgex-support-logging         /support-logging --consul  ...   Up       0.0.0.0:48061->48061/tcp
edgex-support-notifications   /bin/sh -c java -jar -Djav ...   Up       0.0.0.0:48060->48060/tcp
edgex-support-rulesengine     /bin/sh -c java -jar -Djav ...   Up       0.0.0.0:48075->48075/tcp
edgex-support-scheduler       /bin/sh -c java -jar -Djav ...   Up       0.0.0.0:48085->48085/tcp
```

Defining data - Addressable: Camera Control

POST to http://localhost:48081/api/v1/addressable

```
{
    "name":"camera control",
    "protocol":"HTTP",
    "address":"172.17.0.1",
    "port":49977,
    "path":"/cameracontrol",
    "publisher":"none",
    "user":"none",
    "password":"none",
    "topic":"none"
}
```

Defining data - Addressable: Camera 1

POST to http://localhost:48081/api/v1/addressable

```
{
    "name":"camera1 address",
    "protocol":"HTTP",
    "address":"172.17.0.1",
    "port":49999,
    "path":"/camera1",
    "publisher":"none",
    "user":"none",
    "password":"none",
    "topic":"none"
}
```

Defining data - Value Descriptors: Human Count

POST to http://localhost:48080/api/v1/valuedescriptor

```
{
    "name":"humancount",
    "description":"people count",
    "min":"0",
    "max":"100",
    "type":"I",
    "uomLabel":"count",
    "defaultValue":"0",
    "formatting":"%s",
    "labels":["count","humans"]
}
```

Defining data - Value Descriptors: Human Count

POST to http://localhost:48080/api/v1/valuedescriptor

```
{
    "name":"caninecount",
    "description":"dog count",
    "min":"0",
    "max":"100",
    "type":"I",
    "uomLabel":"count",
    "defaultValue":"0",
    "formatting":"%s",
    "labels":["count","canines"]
}
```

Defining data - Value Descriptors: Scan Distance

POST to http://localhost:48080/api/v1/valuedescriptor

```
{
    "name":"depth",
    "description":"scan distance",
    "min":"1",
    "max":"10",
    "type":"I",
    "uomLabel":"feet",
    "defaultValue":"1",
    "formatting":"%s",
    "labels":["scan","distance"]
}
```

Defining data - Value Descriptors: Duration

POST to http://localhost:48080/api/v1/valuedescriptor

```
{
    "name":"duration",
    "description":"time between events",
    "min":"10",
    "max":"180",
    "type":"I",
    "uomLabel":"seconds",
    "defaultValue":"10",
    "formatting":"%s",
    "labels":["duration","time"]
}
```

Defining data - Value Descriptors: Camera Error

POST to http://localhost:48080/api/v1/valuedescriptor

```
{
    "name":"cameraerror",
    "description":"error response message from a camera",
    "min":"",
    "max":"",
    "type":"S",
    "uomLabel":"",
    "defaultValue":"error",
    "formatting":"%s",
    "labels":["error","message"]
}
```

Defining your device - Device Profile

```
name: "camera monitor profile"
manufacturer: "Dell"
model: "Cam12345"
labels:
    - "camera"
description: "Human and canine camera monitor profile"
commands:
  -
    (Next Slide)
```

Defining your device - Device Profile - Commands

```
commands:
  -
    name: People
    get:
        path: "/api/v1/devices/{deviceId}/peoplecount"
        responses:
          -
            code: "200"
            description: "Number of people on camera"
            expectedValues: ["humancount"]
          -
            code: "503"
            description: "service unavailable"
            expectedValues: ["cameraerror"]
```

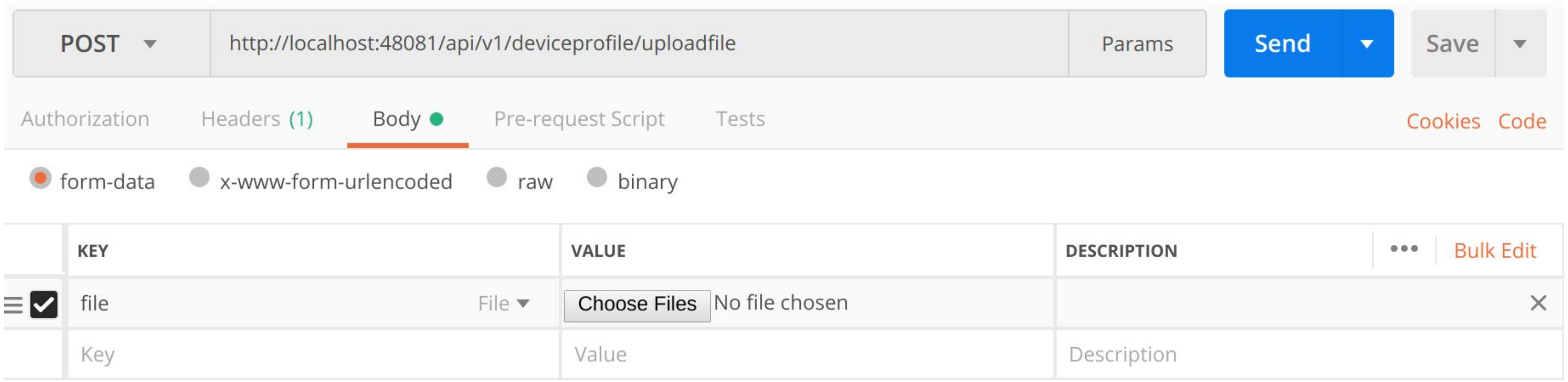
Defining your device - Device Profile - Commands

```
name: ScanDepth
get:
    ...
put:
    path: "/api/v1/devices/{deviceId}/scandepth"
    parameterNames: ["depth"]
    responses:
      -
        code: "204"
        description: "Set the scan depth."
        expectedValues: []
      -
        code: "503"
        description: "service unavailable"
        expectedValues: ["cameraerror"]
```

Defining your device - Device Profile

POST to http://localhost:48081/api/v1/deviceprofile/uploadfile

```
FORM-DATA:
    key: “file”
    value: EdgeX_CameraMonitorProfile.yml
```



POST ▾ | http://localhost:48081/api/v1/deviceprofile/uploadfile | Params | Send ▾ | Save ▾

Authorization | Headers (1) | Body ● | Pre-request Script | Tests | Cookies | Code

[✓ form-data] x-www-form-urlencoded raw binary

	KEY	VALUE	DESCRIPTION ••• Bulk Edit
✓	file File ▾	Choose Files No file chosen	×
	Key	Value	Description

curl -F **“file=@EdgeX_CameraMonitorProfile.yml”** http://localhost:48081/api/v1/deviceprofile/uploadfile

Defining a device service

POST to http://localhost:48081/api/v1/deviceservice

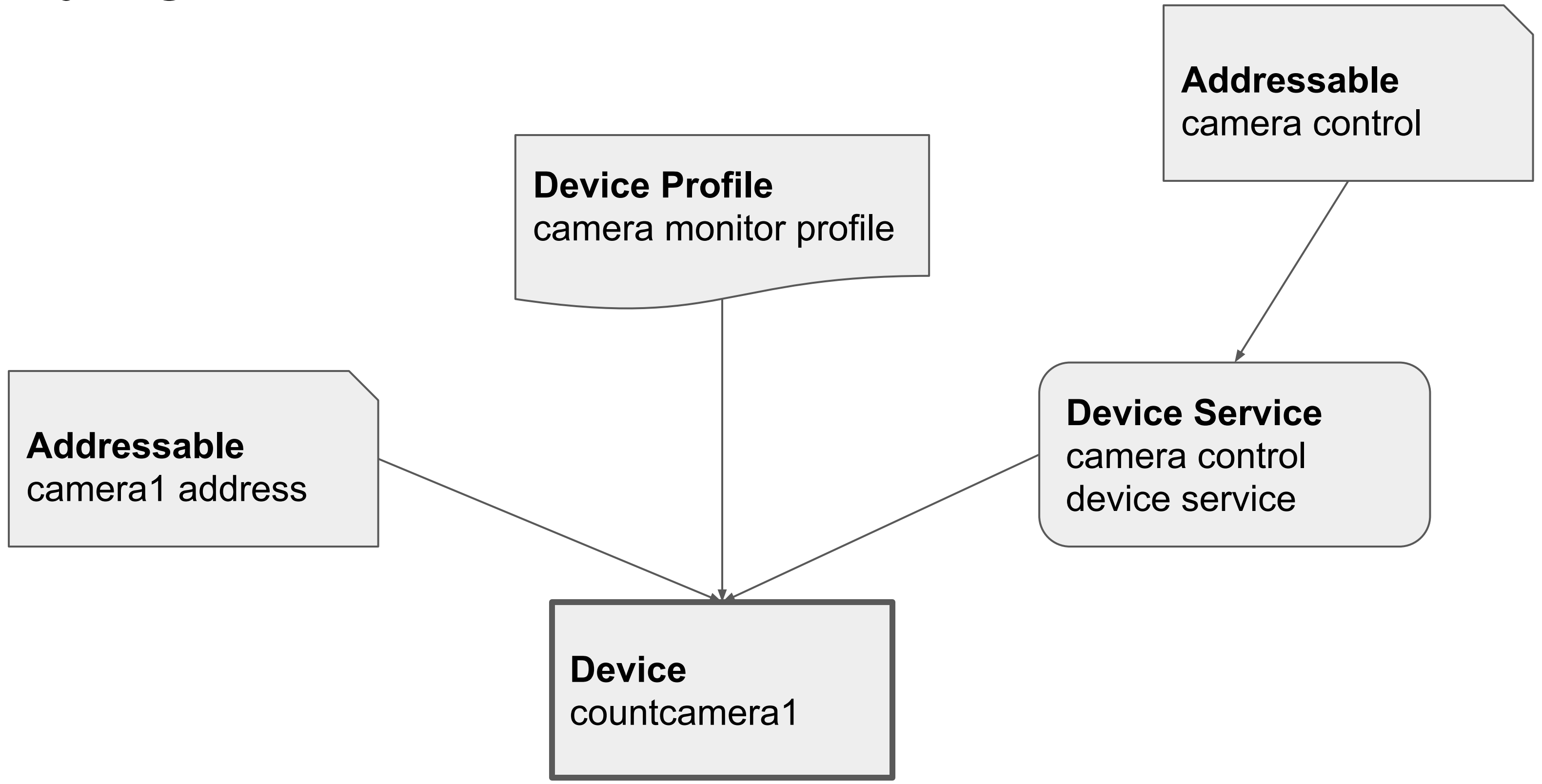
```
{
    "name":"camera control device service",
    "description":"Manage human and dog counting cameras",
    "labels":["camera","counter"],
    "adminState":"unlocked",
    "operatingState":"enabled",
    "addressable": {
        "name":"camera control"
    }
}
```

Deploying a device

POST to http://localhost:48081/api/v1/device

```
{
    "name":"countcamera1",
    "description":"human and dog counting camera #1",
    "adminState":"unlocked",
    "operatingState":"enabled",
    "addressable":{"name":"camera1 address"},
    "labels":["camera","counter"],
    "location":"",
    "service":{"name":"camera control device service"},
    "profile":{"name":"camera monitor profile"}
}
```

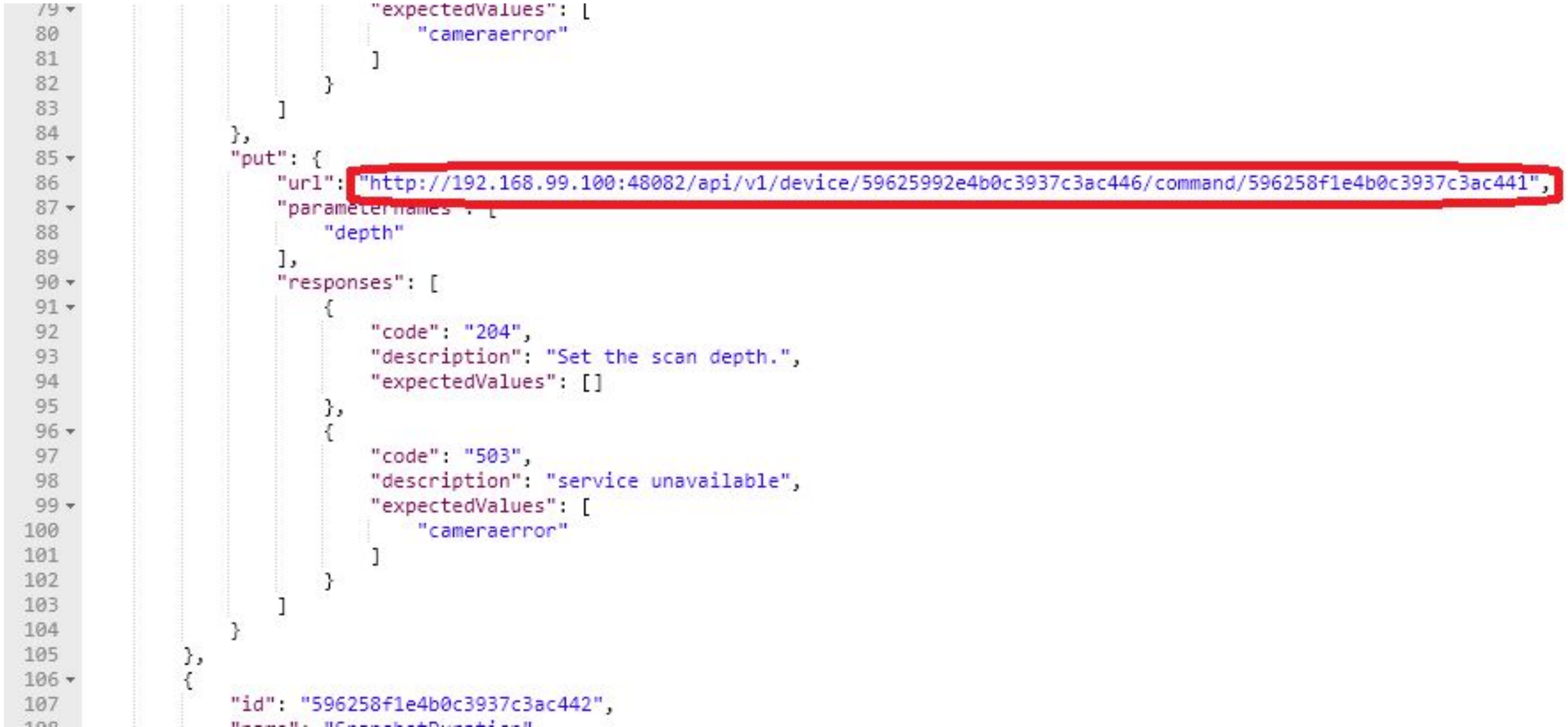
Deploying a device



Addressable camera control
Device Profile camera monitor profile
Addressable camera1 address
Device Service camera control device service
Device countcamera1

Calling device commands

GET to [http://localhost:48082/api/v1/device/name/**countcamera1**](http://localhost:48082/api/v1/device/name/countcamera1)



```
                    "expectedValues": [
                        "cameraerror"
                    ]
                }
            ]
        },
        "put": {
            "url": "http://192.168.99.100:48082/api/v1/device/59625992e4b0c3937c3ac446/command/596258f1e4b0c3937c3ac441",
            "parameterNames": [
                "depth"
            ],
            "responses": [
                {
                    "code": "204",
                    "description": "Set the scan depth.",
                    "expectedValues": []
                },
                {
                    "code": "503",
                    "description": "service unavailable",
                    "expectedValues": [
                        "cameraerror"
                    ]
                }
            ]
        }
    },
    {
        "id": "596258f1e4b0c3937c3ac442",
```

Calling device commands

PUT to http://localhost:48082/api/v1/device/**<device id>**/command/**<command id>**

```
{
    "depth":"9"
}
```

Sending events

POST to http://localhost:48080/api/v1/event

```
{
    "device":"countcamera1",
    "readings":[
        {"name":"humancount","value":"5"},
        {"name":"caninecount","value":"3"}
    ]
}
```

Reading events

GET to http://localhost:48080/api/v1/**event/device/countcamera1**/10

GET to http://localhost:48080/api/v1/**reading/name/humancount**/10

Exporting data

POST to http://localhost:48071/api/v1/registration

```
{
    "name":"MyMQTTTopic",
    "addressable":{
        "name":"MyMQTTBroker",
        "protocol":"TCP",
        "address":"tcp://m10.cloudmqtt.com",
        "port":15421,
        "publisher":"EdgeXExportPublisher",
        "topic":"EdgeXDataTopic"
    },
    "format":"JSON",
    "enable":true,
    "destination":"MQTT_TOPIC"
}
```

EDGEX FOUNDRY™

Developing & Contributing

Install Go

Get GoLang 1.11.x:

```
wget https://dl.google.com/go/go1.11.8.linux-amd64.tar.gz

sudo tar -C /usr/local -xvf go1.11.8.linux-amd64.tar.gz
```

Setup your environment

```
cat >> ~/.bashrc << ‘EOF’
export GOPATH=$HOME/go
export PATH=/usr/local/go/bin:$PATH:$GOPATH/bin
EOF

source ~/.bashrc
```

Install MongoDB

- sudo apt install mongodb-server
- systemctl status mongodb
- wget https://github.com/edgexfoundry/docker-edgex-mongo/raw/master/init_mongo.js
- sudo -u mongodb mongo < init_mongo.js

Get the EdgeX source code

- go get **github.com/edgexfoundry/edgex-go**
- cd ~/go/src/github.com/edgexfoundry/edgex-go
- sudo apt install libczmq-dev
- make build
- make run

- cd ./docs
- ./build.sh

Setup your git repository

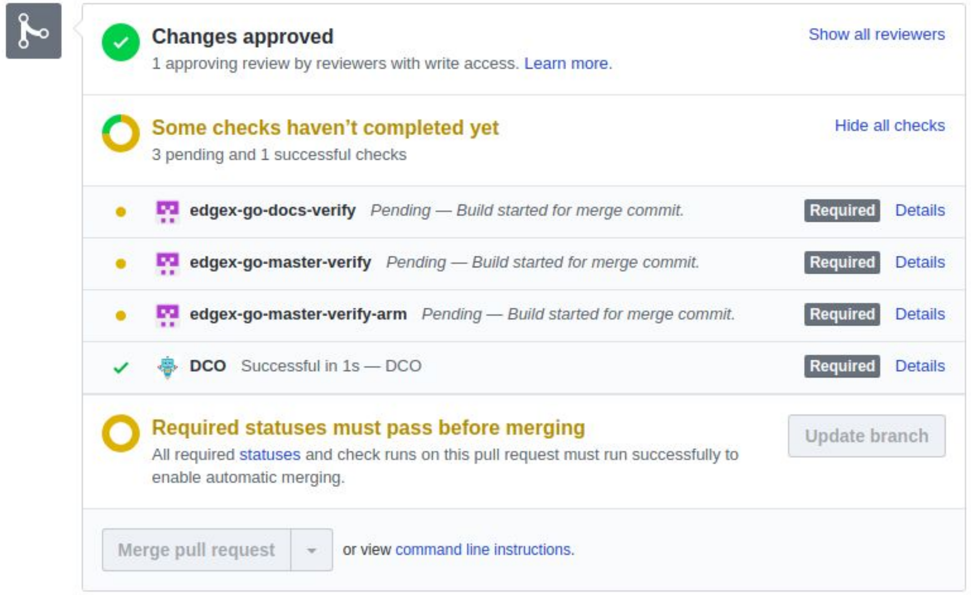
- Fork https://github.com/edgexfoundry/edgex-go
- git remote add **mygithub** https://github.com/**<your_username>**/edgex-go.git
- git config **--global.user.name** “John Doe”
- git config **--global.user.email** johndoe@example.com

Contributing changes

- git checkout **-b your_fix_branch_name**
- git add <files you changed>
- git commit **--signoff** -m “Your commit message”
- git push **mygithub your_fix_branch_name**

PR review and approval

- Pass DCO Signoff
- Pass automated tests
- Have at least one approving review



Changes approved
1 approving review by reviewers with write access. Learn more.
Show all reviewers

Some checks haven't completed yet
3 pending and 1 successful checks
Hide all checks

● **edgex-go-docs-verify**	*Pending — Build started for merge commit.*	Required	Details
● **edgex-go-master-verify**	*Pending — Build started for merge commit.*	Required	Details
● **edgex-go-master-verify-arm**	*Pending — Build started for merge commit.*	Required	Details
✓ **DCO**	Successful in 1s — DCO	Required	Details

Required statuses must pass before merging
All required statuses and check runs on this pull request must run successfully to enable automatic merging.
Update branch

Merge pull request ▾ or view command line instructions.