MASTER OF SCIENCE
IN ENGINEERING

Filière Computer Science - Internet of Things

DomoHeat

Projet Report

Étudiant : Adrien Dino Balleyguier
adriendino.balleyguier@master.hes-so.ch
Hugo Mendes
hugo.mendes@master.hes-so.ch

Professeurs : Nabil Abdennadher
nabil.abdennadher@hesge.ch

Hes·so FRIBOURG FREIBURG
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts
Western Switzerland

Lausanne, 18.12.2024

Glossaire du projet

IoT	: Internet of Things
WoT	: Web of Things
W3C	: World Wide Web Consortium
TD	: Thing Description - set of properties, actions and events associated to a device within the WoT
WHATWG	: Web Hypertext Application Technology Working Group
IRI	: International Resource Identifiers
URL	: Uniform Resource Locator
CSA	: Connectivity Standard Alliance

Table des matières

1 Introduction **1**

2 Contexts **1**
2.1 Edge context . . . 1
2.2 Cloud context . . . 2

3 Messages **2**
3.1 SYNC - edge context . . . 2
3.2 SET - edge context . . . 2
3.3 SYNC - cloud context . . . 2
3.3.1 SYNC . . . 3
3.3.2 SYNC_ACK . . . 3
3.4 SET - cloud context . . . 3

4 Alternative for edge communication **4**
4.1 Web of Things (WoT) . . . 4
4.1.1 Deployment pattern . . . 4
4.1.2 Thing Description . . . 5
4.1.3 Compliance certification . . . 6
4.1.4 Usage for Domoheat . . . 6
4.2 Thread . . . 8
4.2.1 Compliance certification . . . 9
4.2.2 Usage for domoheat . . . 9
4.3 Matter . . . 10
4.3.1 Fabric . . . 10
4.3.2 Node . . . 10
4.3.3 Compliance certification . . . 10
4.3.4 Usage for domoheat . . . 10
4.4 Comparison of solution for the Domoheat project . . . 11
4.4.1 Class deployment . . . 11
4.4.2 Industrial deployment . . . 11

5 Figures **i**
5.1 Liste des figures . . . i

6 Bibliographie **ii**

1 Introduction

During cold temperature season, some remote site are not visited each day. Such site would be for instance cottage or construction site hut. The room would be required to be heated for the visiting time. As of today a way to achieve that would be to let it activated all the time. Another way would be to start heating it when the visit start, leaving the first hours as quite cold.
The DomoHeat project propose a solution to this with IoT. An edge communicates with a cloud informing it of the current temperature. This communication also will be used to set a target temperature at a given time. This setting will be sent to the edge which will manage the heater to reach the target. The global architecture is as illustrated by the figure 1.

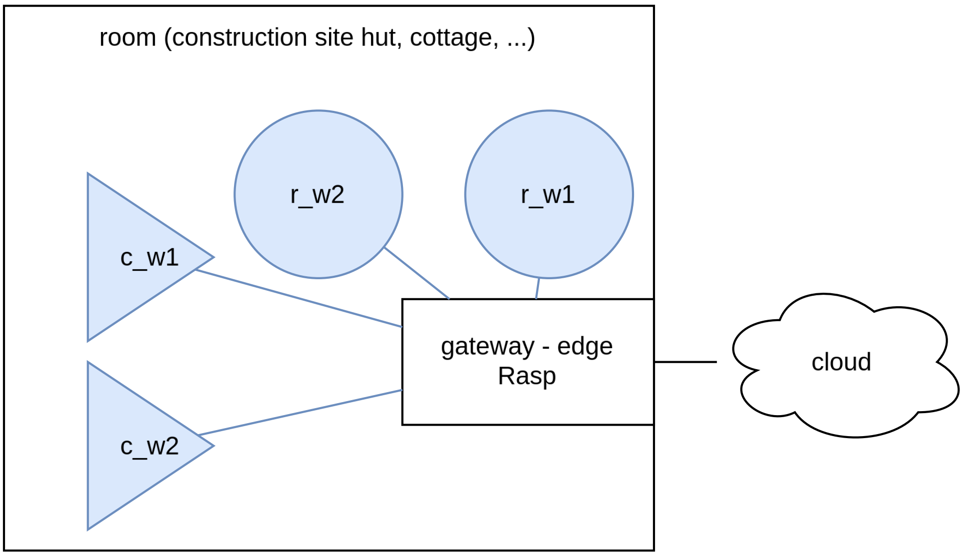

Figure 1: DomoHeat global architecture diagram

The following are meant:

- Circles symbolises heaters
- Triangles symbolises temperature sensors

The source code for this project is available in the following github repository :
https://github.com/HugoMendes98/MA_IOT-DomoHeat

2 Contexts

The project can be divided in two contexts, each interfaced with the other by the edge device.

2.1 Edge context

This context consists in a group of end device with a single edge device. The communication is done following the MQTT protocol over the Wi-Fi line. The following are expected for the exchanged data:

message	producer	subscriber	comment
SYNC	end-device	edge	contains end-device data (see section 3.1)
SET	edge	heaters (end-device)	contains target for the end-device (see section 3.2)

The MQTT broker is held by the edge device.
Every end device is given an unique identifier (end_device_uid).

2.2 Cloud context

This context consist in an edge device communicating with a remote cloud. The communication is donw following the MQTT protocol. The following are expected for the exchanged data:

message	producer	subscriber	comment
SYNC	edge	cloud	contains data held by the edge (see section 3.3)
SYNC_ACK	cloud	edge	acknowledgement for a given SYNC message (see section 3.3)
SET	cloud	edge	contains target, with associated target time (see section ??)
SET_ACK	edge	cloud	acknowledgement for a given SET message (see section ??)

3 Messages

A collection of messages will be exchanged over the time. This messages are all held in JSON files and are as described below.

3.1 SYNC - edge context

As indicated prior, this message is produced by any end device and consumed by the edge. This message is produced every 5 minutes and contains the data held at the time by the end device. The aforementioned data consist in the recorded temperature alongside the set thermostat. The topic is /<end_device_uid>/sync with payload described by the following JSON:

```
{
    "<end_device_uid>" : {
        "temp": value,
        "thermostat": value
    }
}
```

"value" is a floating point value describing the recorded temperature [°C], with a single decimal precision.

3.2 SET - edge context

As indicated prior, this message is produced by the edge and consumed by a target end device able to influence the temperature - a heater. This production time of this message is managed by the edge based on a user input. It contains the target data to set the heater at. The topic is /<end_device_uid>/set with payload described by the following JSON:

```
{
    "<end_device_uid>" : {
        "date": "<current_time>",
        "thermostat": <temp_value>
    }
}
```

- "current_time" is a string containing the producing time that follows the ISO 8601 definition [1] :
  "YYYY-mm-ddTHH:MM:SSZ"
- "temp_value" is floating point value describing the recorded temperature [°C] with a single decimal precision.

3.3 SYNC - cloud context

The SYNC message, in the cloud context, is used by an edge to send its held data to the cloud

3.3.1 SYNC

As indicated prior, this message is produced by an edge and consumed by the cloud. It contains every data produced by the edge's connected end devices and that are not yet received by the cloud. The topic is /sync with payload described by the following JSON:

```
{
    "<edge_name>":{
        "nodes":{
            "end_device_1_uid":{
                "date": "timestamp",
                "temp": recorded_temperature,
                "thermostat": value
            },
            "end_device_2_uid":{
                "date": "timestamp",
                "temp": recorded_temperature,
                "thermostat": value
            }
        }
    }
}
```

Please note that the example above only lists two end devices but more could be present.

- "recorded_temperature" are the last value, for each device, consumed by the edge device at the timestamp time.
- "timestamp" is a string containing the time at which the recorded_temperature has been observed. It follows the ISO 8601 definition [1]:
  "YYYY-mm-ddTHH:MM:SSZ"

The edge is expected to try sending its held data every 30 minutes. Should a timestamp fail to be acknowledged, it will be part of the next sending. This stays true until a timestamp is acknowledged.

3.3.2 SYNC_ACK

Once the cloud consumes a sync message, it is expected that it produces a SYNC_ACK as acknowledgement to the edge. The topic is /ack with payload described by the following JSON:

```
{
    "<edge_name>":{
        "date": ["timestamp"]
    }
}
```

"timestamp" is an array of string containing the time held by the SYNC messages that are acknowledged.

3.4 SET - cloud context

As indicated prior, this message is produced by the cloud and consumed by an edge. It contains a target temperature that the edge is tasked to reach at an associated date. The topic is /set with payload described by the following JSON:

```
{
"<edge_name>":{
    "date": "timestamp",
    "thermostat": target_thermostat
    }
}
```

- "timestamp" is a string containing the time at which the recorded_temperature has been observed. It follows the ISO 8601 definition [1]:
  "YYYY-mm-ddTHH:MM:SSZ"
- "target_thermostat" is a floating point value describing the temperature [°C] with a single decimal precision

4 Alternative for edge communication

The current section describes other way to setup the communication in the edge context. It is followed by a summary section comparing briefly the candidates and the one presented above. In the field, a system - a group of devices - will be heterogenous in regards to the protocols, data models and security. This three elements are to be considered unique for each device provider. For a system with 4 distinct provider there will be 4 protocols at stake, 4 data models and 4 security approaches.
On one hand this will increase the security breach risk as a given approach may have a vulnerability. This vulnerability may even be used to attack the other devices. On the other hand, developing a new device, or adding one to the system, may become impossible if conflicts arose with existing protocols.
The Thread and Matter section are inspired from a presentation given by Antoine Delabays at the 27th Linux Seminar [2].

4.1 Web of Things (WoT)

The W3C [3] proposes a way to homogenise this protocol, data model and security approach with the Web of Things [4].
In this context, each device is given a set of properties, a set of actions and a set of events.
The following sections details the elements that would be used in the domoheat projet if using this solution

4.1.1 Deployment pattern

The definition for in the WoT for "edge device" does not change from the usual definition : It is a device that acts as barrier between multiple end-devices and a remote controller - a cloud.
The proposed deployment pattern for edge device is as illustrated in the figure 2.

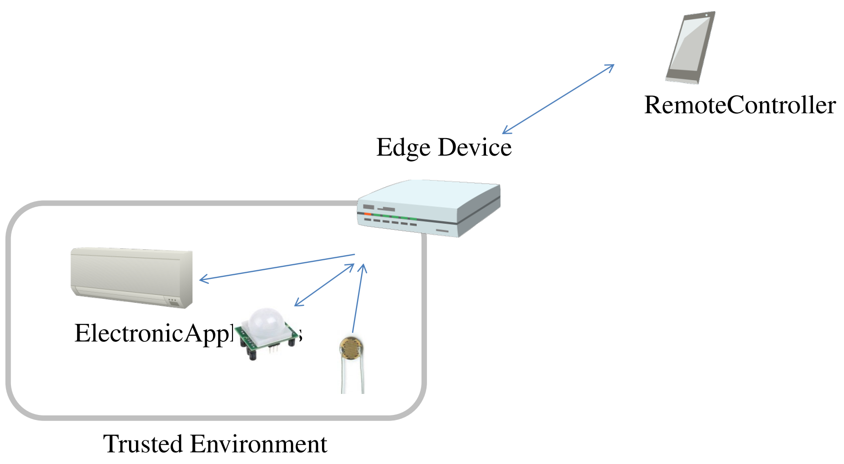

Figure 2: Edge device deployment in the Wot, from [5]

Digital twins are close to shadows. The distinction lies in the shadow indicating the live, respectively last known, state of a device. Digital twins can hold the history of values for a device and can even be used in some test cases. Digital twin or shadows can be used by the user in the cloud to set the target temperature and date.
The proposed deployment pattern for digital twins is as illustrated in the figure 3.

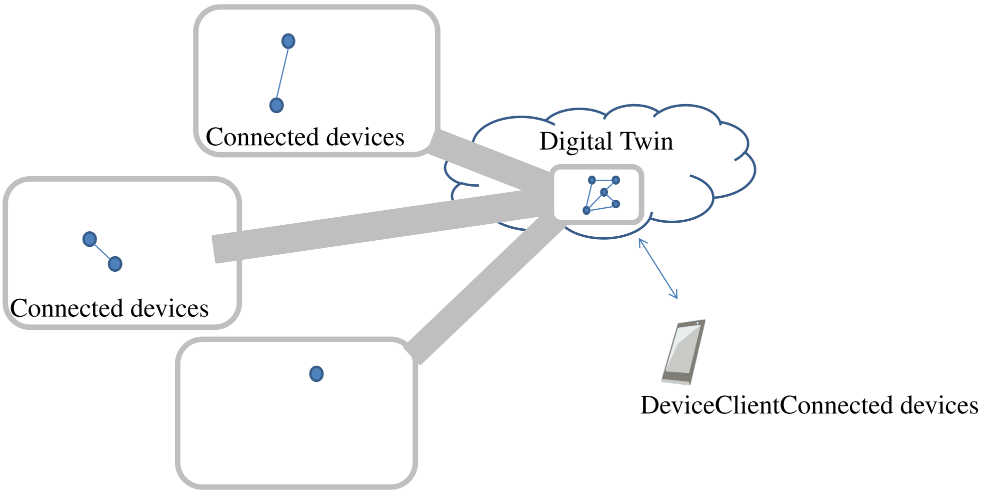

Figure 3: Digital twin deployment in the Wot, from [6]

4.1.2 Thing Description

The WoT building block is called Thing Description (TD). Each device is associated a TD which describes how to interact with and how it interacts with the rest of the system. Every TD will contain the following categories:

- Textual metadata, like device unique id or name
- Interaction Affordances, listing all properties, actions and event that can occur for the device
- Data Schema, to be used for machine-understandability and indicating how the data are exchanged with the thing
- Security definition describing how to access the thing for **and only for** Consumers that have already authorisation. The security definition is explicitly not meant to hold keys nor passwords.
- Web links to be used either to reach the thing or to access resources related to it

All TD entry are optionnal but the @context (link to definition of terms used in the TD), the title (human-readable thing name) and all in the security category.
A typical TD can take the following shape as suggested by the WoT documentation [7]:

```
{
    "@context": "https://www.w3.org/2022/wot/td/v1.1",
    "id": "urn:uuid:0804d572-cce8-422a-bb7c-4412fcd56f06",
    "title": "MyLampThing",
    "securityDefinitions": {
        "basic_sc": {"scheme": "basic", "in": "header"}
    },
    "security": "basic_sc",
    "properties": {
        "status": {
            "type": "string",
            "forms": [{"href": "https://mylamp.example.com/status"}]
        }
    },
    "actions": {
        "toggle": {
            "forms": [{"href": "https://mylamp.example.com/toggle"}]
        }
    },
```

```
    "events": {
        "overheating": {
            "data": {"type": "string"},
            "forms": [{
                "href": "https://mylamp.example.com/oh",
                "subprotocol": "longpoll"
            }]
        }
    }
}
```

This TD describes a dummy lightbulb that have some properties (status on/off), actions (toggling the status on/off), events (overheating) alongside mandatory security and title entries.
The starting @context links to a generic dictionary [8] containing the basic set of key and associated schema. This dictionary describes for instance fields like *readOnly*, *required* and many more.

4.1.3 Compliance certification

Having a device comply with the WoT must validate the following:

- A TD with, at least, the aforementioned "mandatory" entries.
- To be hosted on a networked system component with software stack to allow for communication between existing things.
- Url must be compared as string using ASCII case-insensitive [9].
- Form context and submission to be IRIs [10](which includes URLs).
- Request method to be included in the standard set for the identified protocol
- Indicate at least one protocol to use for communicating with the thing
- All data exchanged are in the MIME section [11] should the used protocol have such. Otherwise there is only *SHOULD* clauses.
- No private security data is present in the TD.
- Protect the thing by secure transport (TLS, DTLS, ...) especially if the thing is available on the public internet When using TLS or DTLS, their version are to be equal or higher than the version 1.3

All of this elements are recognised in the documentation with the pattern *MUST*.
Certifying the compliance, for a thing, in the WoT can be achieved by using tools proposed directly by the W3C [12]. Another way to certify the compliance is to put it to test by independent organisations like TUV [13] or UL [14].

4.1.4 Usage for Domoheat

In the context of the domoheat project, the following steps would be required. As it is only for exploratory purposes, TD are not explicit for each things the project would include.

End-device
The heaters end-device would require a TD that would contain the following:

```
{
    "@context": "https://www.w3.org/2022/wot/td/v1.1",
    "id": "<end_device_uid>",
    "title": "<end_device_name>",
    "securityDefinitions": {
        "basic_sc": {"scheme": "basic", "in": "header"}
    },
    "security": "basic_sc",
```

```
    "properties": {
        "temp": {
            "type": "number"
        },
        "thermostat":{
            "type": "number"
        }
    },
    "actions": {
        "set_thermostat": {
            "type": "number"
        }
    }
}
```

The action *set_thermostat* could have been named *thermostat* at the risk of adding confusion with the property of the same name.
The temperature sensor would require a TD that would contain the following:

```
{
    "@context": "https://www.w3.org/2022/wot/td/v1.1",
    "id": "<end_device_uid>",
    "title": "<end_device_name>",
    "securityDefinitions": {
        "basic_sc": {"scheme": "basic", "in": "header"}
    },
    "security": "basic_sc",
    "properties": {
        "temp": {
            "type": "number"
        }
    }
}
```

Edge

The edge would be described with the following TD:

```
{
    "@context": "https://www.w3.org/2022/wot/td/v1.1",
    "id": "<edge_device_uid>",
    "title": "<edge_device_name>",
    "securityDefinitions": {
        "basic_sc": {"scheme": "basic", "in": "header"}
    },
    "security": "basic_sc",
    "properties": {
        "heaters": {
            "title": "connected heaters",
            "type": "array",
            "items": {
                "type": "object",
                "items": {
                    "properties":{
                        "date": {
                            "type": "string",
                            "pattern": "YYYY-MM-DDTHH:mm:ssZ"
                        },
                        "temp": {
```

```
                            "type": "number"
                        },
                        "thermostat": {
                            "type": "number"
                        }
                    },
                    "required": "false"
                }
            }
        },
        "thermometers" {
            "title": "connected thermometer",
            "type": "array",
            "items":{
                "properties":{
                    "date": {
                        "type": "string",
                        "pattern": "YYYY-MM-DDTHH:mm:ssZ"
                    },
                    "temp": {
                        "type": "number"
                    }
                },
                "required": "false"
            }
        }
    },
    "actions": {
        "set_thermostat": {
            "title": "set thermostat",
            "type": "object",
            "items": {
                "properties": {
                    "target_date": {
                        "type": "string",
                        "pattern": "YYYY-MM-DDTHH:mm:ssZ"
                    },
                    "thermostat":{
                        "type": "number"
                    }
                }
            }
        }
    }
}
```

The implementation would respect the elements indicated in the section 4.1.3.

4.2 Thread

Thread, as proposed by the Thread Group Alliance [15], tries to solve the aforementioned multi-protocol issue observable in a system.
The network topology can be described by the figure 4.

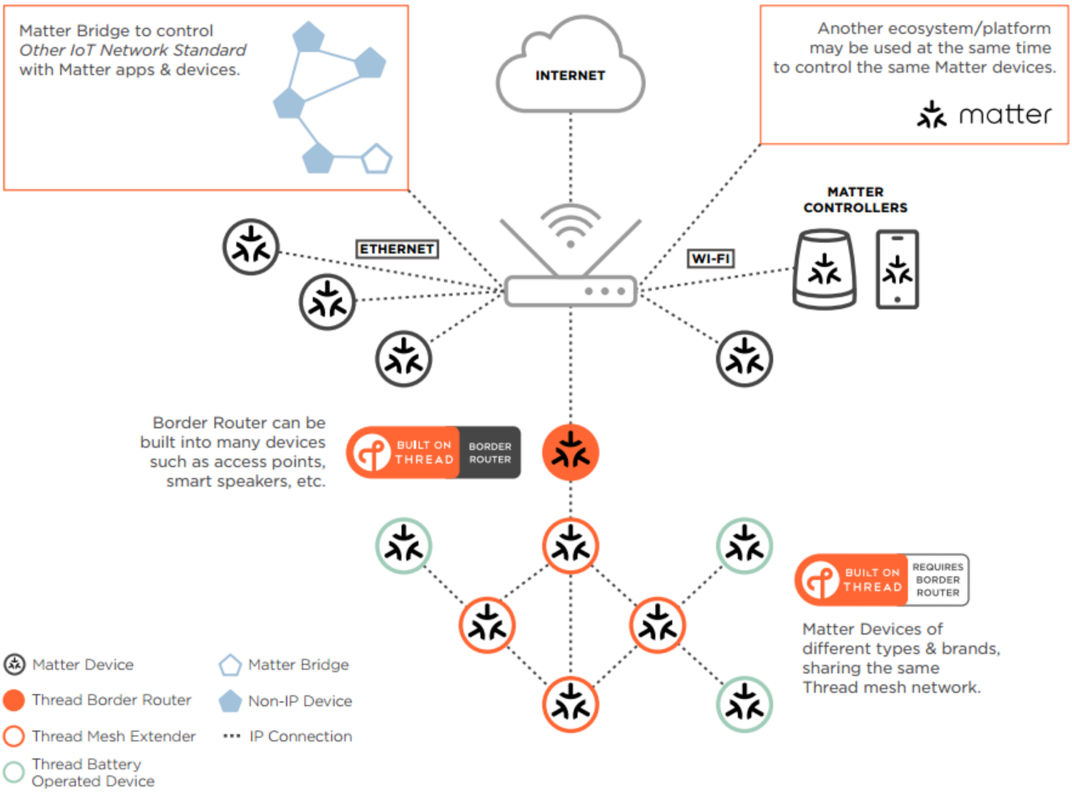

Figure 4: Thread network topology, as presented by [16]

This construction offers the following benefits, especially in the IoT context:

- The network can self-heal itself should one of it's point fail (unlike wifi for instance where the complete network goes down if the router does so). It also allows for independent communication between nodes.
- Low-power, allowing for AAA batteries to power the devices.
- IPv6 networking

However, this construction is not suited for streaming applications. Thread also follow the IEEE 802.15.4 [17]. Thread has been constructed with the recommended usage of Matter in mind, described in section 4.3. However it is not mandatory.

4.2.1 Compliance certification

To get a certification for a device compliant with Thread, one will require to join the Thread Group. Joining the Thread Group requires a fee going from 1000$ to 2500$. Afterward the device will be put to the test by authorised third-party laboratories such as the TUV [13] or UL [14]. Having a device certified allows for advertisement by the Thread Group.

4.2.2 Usage for domoheat

For the domoheat project, the use of MQTT would be put aside. The edge, considering the figure 4, would be the border router. The end devices would become Thread Devices.

4.3 Matter

Matter is a data model and security approach proposed by the CSA [18] (formerly named the Zigbee Alliance). It has been built to be use alongside Thread which is described in section 4.2 but can be used with other protocols.

4.3.1 Fabric

In the matter context, a system is called a Fabric that will be composed of nodes. Nodes can be classified in the following:

- Device: End-device of any kind
- Controller: Any device able to control other devices
- Administrator: Device creating and managing the security privileges within the Fabric
- Commissioner: Device used to setup other devices within the Fabric. It can be a device connected to the fabric for the setup time only like a smartphone.

Any device in a given fabric can communicate with the other device in the fabric. Two device that are not in the same fabric cannot communicate even if sharing the same physical network.
Each device can only be part of a single fabric at a time.

4.3.2 Node

Each node is associated endpoint, describing its hierarchy. An endpoint can contain a cluster, linking it to other nodes through actions. A cluster will contain client(s) and server(s) describing the information direction. Clients will always produce data consumed by server. Considering as an example a light-switch manipulating a single lightbulb, each endpoint can be described by:

```
node: light-switch
|-endpoint 0: root node
|-endpoint 1: toggle light switch
    |-cluster: light_toggle [client]

node: lightbulb
|-endpoint 0: root node
|-endpoint 1: on/off light
    |-cluster: light toggle [server]
```

As indicated prior, Matter has been developed alongside Thread. This is especially observable in the communication which is not built to work with broadcasting.

4.3.3 Compliance certification

To get a certification for a device compliant with Matter, one will required to join the CSA. Joining the CSA can be done in three colors, based on one goal:

- Contributor, [5000$; 10000$] per year : Allows to contribute to the development of the standard
- Adopter, [2500$; 5000$] per year : For companies that only want to use the standard
- Promoter, announced to be 30000$: Reserved for large company that will steer and promote the standard

Once joining the CSA, the device that is to be certified will be put to tests by authorised third-party laboratories such as TUV [13] or UL [14].

4.3.4 Usage for domoheat

For the domoheat project, the following would be required. As it is only for exploratory purposes, Nodes are not explicit for each device the project would include

End-device
The heaters end-device would require a node description going as follows:

```
node: heater_<uid>
|-endpoint 0: root node
|-endpoint 1: set thermostat <uid>
    |-cluster: set_thermostat_<uid> [server]
|-endpoint 2: read temperature <uid>
    |-cluster: get_temp_<uid> [client]
```

The temperature sensor would require a node description going as follows:

```
node: heater_<uid>
|-endpoint 0: root node
|-endpoint 1: read temperature <uid>
    |-cluster: get_temp_<uid> [client]
```

Edge The edge would be described with the following node description:

```
node: edge_<uid>
|-endpoint 0: root node
|-endpoint 1_<end_device_uid>: set thermostat <end_device_uid>
    |-cluster: set_thermostat_<end_device_uid> [client]
|-endpoint 2_end_device_uid: read temperature <end_device_uid>
    |-cluster: get_temp_<end_device_uid> [client]
```

4.4 Comparison of solution for the Domoheat project

The adjustments required to make the Domoheat compliant with either the WoT, Thread and Matter are described in their respective sections 4.1.4, 4.2.2 and 4.3.4.
The current section analyse what solution would be the most interesting technically going forward between the Domoheat proposed one, the WoT and the association of Thread and Matter.
Two types of deployment are considered. The first one is the class context, which can be estimated to have a maximum of one edge and 4 end-device. The second one is the industrial deployment, where the number of edge, respectively end-devices, can grow to more than 1000.

4.4.1 Class deployment

This context has the following features:

- Single edge
- Between a single and 4 end-devices
- Simulated cloud
- No necessity for certification

Considering the limited deployment numbers, developing a complex solution is too heavy. This points out that the proposed solution, SET/SYNC with MQTT, has been the one finally implemented.
However, for academic purposes, one could start the project over with the goal of implementing the WoT or Thread/Matter. Even, two project could propose each solution implemented. This would improve the comparison results.

4.4.2 Industrial deployment

This context has the following features:

- Multiple edge for multiple client. A single client could have many edge that could even be on the same location.
- Important amount of end-devices, with an estimation of 5 to 20 per edge.

- Actual cloud with solution proposed by AWS or Azure.
- Devices certification is required
- Strong security is mandatory, at the very least between the edge and the cloud

Efficient maintenance and low technical debt would be a strong point when considering solutions.

The Domoheat presents use-case indicates a deployment in remote location. Such location are usually without other IoT device or with a small amount of them. Implementing a solution created to answer situation with various IoT devices in the same location is thus not mandatory. However, nothing can prevent a deployment being done in living sites. Moreover, having the Domoheat certified in regards to a standard is a great industrial argument. The solution choice for industrial deployment would then be between the WoT and Thread/Matter.
Each approach has its pros and cons listed in their respective sections. Considering the companies behind each proposition, it becomes clear that Thread/Matter is expected to be deployed in smart homes while the WoT is more suited to industrial contexts.

In the end the Thread/Matter solution would be more suited to cover the deployment in smart-homes. Nonetheless a proof of concept for each solution should be developed before settling the choice once and for all.

5 Figures

5.1 Liste des figures

1 DomoHeat global architecture diagram 1
2 Edge device deployment in the Wot, from [5] 4
3 Digital twin deployment in the Wot, from [6] 5
4 Thread network topology, as presented by [16] 9

Listings

1 SYNC - edge context JSON construction 2
2 SET - edge context JSON construction 2
3 SYNC - cloud context JSON construction 3
4 SYNC_ACK - cloud context JSON construction 3
5 SET - cloud context JSON construction 3
6 WoT TD example 5
7 heater WoT TD 6
8 temperature sensor WoT TD 7
9 edge WoT TD 7
10 matter node example 10
11 heater matter node description 11
12 temperature sensor matter node description 11
13 edge matter node description 11

6 Bibliographie

[1] ISO/TC 154. ISO 8601 definition. ://www.iso.org/standard/70907.html. [Online, last access December 10, 2024].

[2] Antoine Delabays. Presentation on Thread and Matter - 27th Linux Seminar. https://gitlab.forge.hefr.ch/fribourg-linux-seminar/seminars/-/raw/master/24.11_handout_27th_seminar/04_FLS-27-MakingHomeAutomationSimplyHappen.pdf, 11 2024.

[3] World Wide Web Consortium homepage. ://www.w3.org/. [Online, last accessed December 10, 2024].

[4] Web of Things homepage. ://www.w3.org/WoT/. [Online, last accessed December 10, 2024].

[5] W3C. WoT edge deployment. ://www.w3.org/TR/wot-architecture/images/wot-use-cases/edge-device.svg. [Online, last accessed December 10, 2024].

[6] W3C. WoT digital twin deployment. ://www.w3.org/TR/wot-architecture/images/wot-use-cases/digital-twin-multiple-devices.svg. [Online, last accessed December 10, 2024].

[7] W3C. WoT TD example 1. https://www.w3.org/TR/wot-thing-description/#simple-thing-description-sample. [Online, last accessed December 11, 2024].

[8] W3C. WoT TD example 1 @context. https://www.w3.org/2022/wot/td/v1.1. [Online, last accessed December 11 2024].

[9] WHATWG. ASCII case-insensitive standard. https://infra.spec.whatwg.org/#ascii-case-insensitive. [Online, last accessed December 12 2024].

[10] W3C. RFC 3987 : IRI. https://www.rfc-editor.org/rfc/rfc3987. [Online, last accessed December 12 2024].

[11] Innosoft. RFC 2046 : MIME. https://www.rfc-editor.org/rfc/rfc2046. [Online, last accessed December 12 2024].

[12] Jan Lauinger. WoT W3C test bench. https://github.com/tum-esi/testbench. [Online, last accessed December 12 2024].

[13] TUV Rheinland Deutschland homepage. https://www.tuv.com. [Online, last accessed December 12 2024].

[14] Underwriters Laboratories homepage. https://www.ul.com/. [Online, last access December 12 2024].

[15] Thread group alliance. Thread Group Alliance. https://www.threadgroup.org/thread-group. [Online, last accessed December 15 2024].

[16] Thread Group Alliance. Thread network topology. https://threadgroup.org/BUILT-FOR-IOT/Smart-Home#threadinhome. [Online, last accessed December 15 2024].

[17] IEEE. IEEE 802.15.4. https://standards.ieee.org/ieee/802.15.4/7029/. [Online, last access December 15 2024].

[18] Connectivity Standard Alliance homepage. https://csa-iot.org/. [Online, last accessed December 15 2024].

[19] Thread Group Alliance. Thread certification. https://www.threadgroup.org/What-is-Thread/Certification. [Online, last accessed December 15 2024].