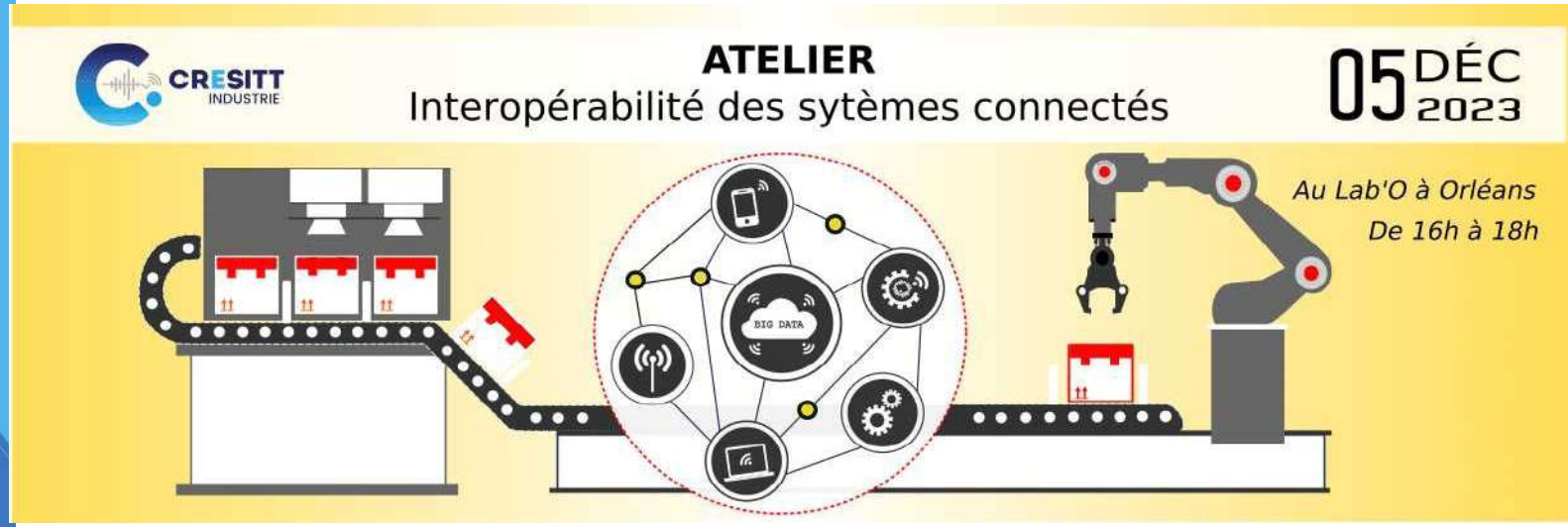
CRESITT INDUSTRIE

Centre de Ressources
Technologiques en Électronique



Le CRT CRESITT est soutenu par :

L'action de diffusion technologique est cofinancée par l'Union européenne.
L'Europe s'engage en région Centre-Val de Loire avec le Fonds européen de développement régional.

SOMMAIRE

Olivier RENCKERT

- Smart Building Gasket

Olivier BICHAUT

- Le protocole Matter

Samuel ROUXEL

- Mise en œuvre d’un device Matter (end node)

Conclusion

quelques chiffres

40,000 Employees

25 Countries

102 Sites

4,4 Billion revenue

5% of our revenue invested in R&D

4 Research & Innovation Centers

40 Development Centers

Smart Building Gasket

L’objectif est de développer un système de mesure et de transmission intégré dans les joints HUTCHINSON permettant de détecter l’état d’un ouvrant de type battant ou coulissant

- Détection intégrée directement aux joints
- Communication des informations via une liaison sans fil
- Intégration discrète et esthétique
- Temps de réaction Ⓟ ≈ seconde
- Autonomie Ⓟ ≈ 5 ans

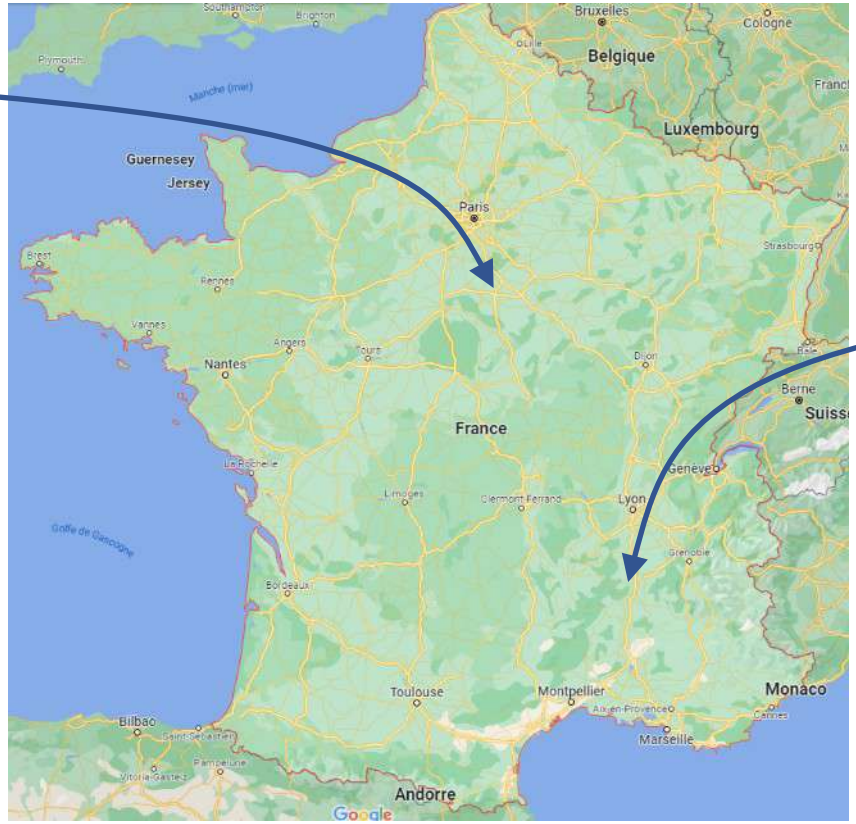
Détecteur magnétique

Smart Building Gasket

HUTCHINSON

45120 CHALETTE S/ LOING

Centre de **R**echerche et d'**I**nnovation

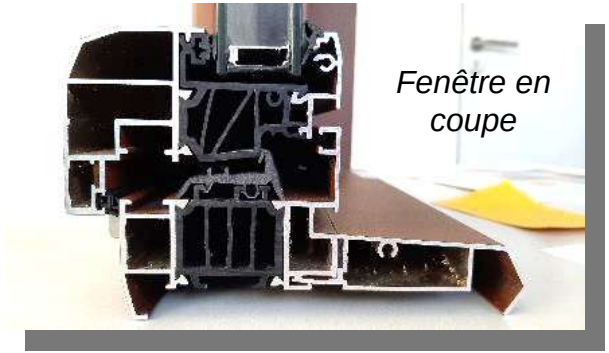


Partenaires : CRESITT INDUSTRIE, todo, PASQUET menuiseries

HUTCHINSON

38 430 MOIRANS

Étanchéité bâtiment



Fenêtre en coupe

Smart Building Gasket

2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024

Créativité
Connectivité des joints

POC 1

BATIMAT PARC DES EXPOSITIONS PARIS NORD VILLEPINTE 04-08 NOV 2019
Tout pour réussir les chantiers du bâtiment

Retours clients:
Système d'état

Projet CRI/Moirans
Partenariat

PASQUET menuiseries

TRL 2

POC 2
faisabilité 04/21

TRL 3

POC 3
technologie 02/22

TRL 4

POC 4

BATIMAT PARIS EXPO PORTE DE VERSAILLES 03-06 OCT 2022
Connectons les pros à l'innovation

TRL 5

POC 5
Gamme produit 09/23

ARTIBAT LE SALON DE LA CONSTRUCTION 18 19 20 OCTOBRE 2023

TRL 9

Produit série

Smart Building Gasket

La détection est basée sur les variations d'une capacité électrique. Le condensateur est constitué d'une simple électrode conductrice ajoutée au joint et variant suivant la permittivité du milieu environnant.

Technologie et utilisation brevetée

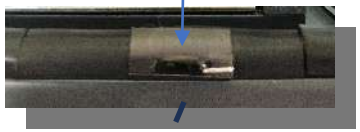
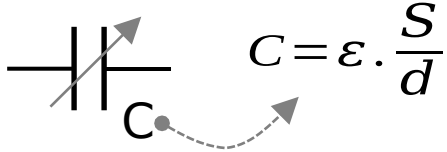
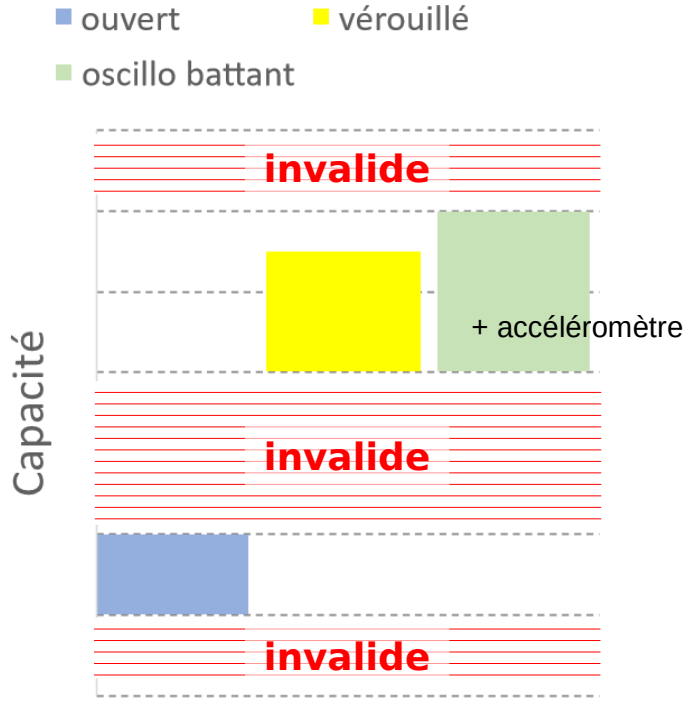


Schéma équivalent





Smart Building Gasket

1

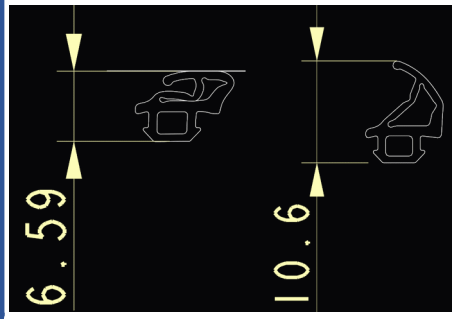
Sérigraphie encre carbone & film protecteur sur feuille plastique

2

Préparation électrode

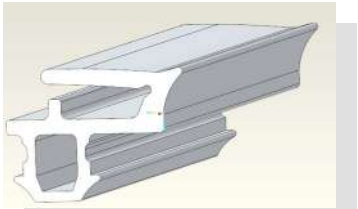
3

Collage électrode sur joint

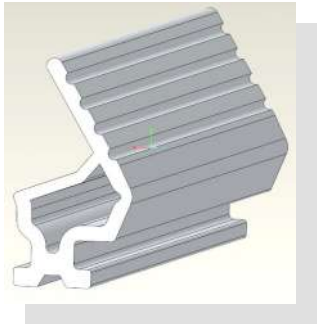


Joints TP

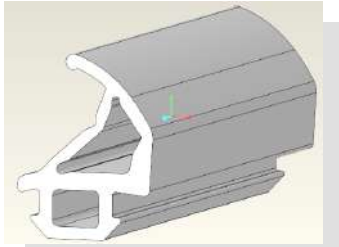
2 matières



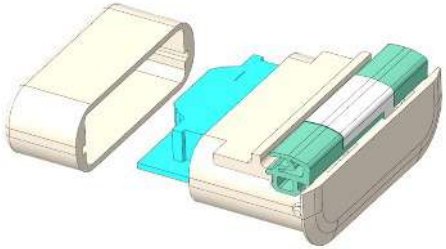
3 matières



Joint caoutchouc



Smart Building Gasket



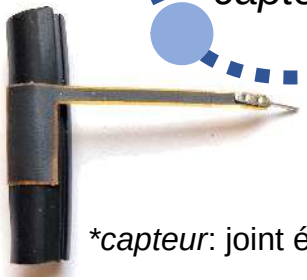
joint

boîtier

intégration

capteur*

électronique



**capteur*: joint équipé d'une électrode

état de l'ouvrant

Smart Building Gasket

Baie coulissante Aluminium

Fenêtre aluminium

Fenêtre PVC

matter THREAD

matter THREAD

matter THREAD

Routeur

RJ45

Hub MZ otodo

ARTIBAT LE SALON DE LA CONSTRUCTION

18 | 19 | 20 OCTOBRE 2023

(4G)

Cloud

zigbee

Ampoules x3

zigbee

Thermostat chauffage

zigbee

Ventilateur

zigbee

Volets roulants

(4G)

Application mobile

- État des ouvrants
- Automatisation de tâches

Données remontées

ARTIBAT LE SALON DE LA CONSTRUCTION 18 | 19 | 20 OCTOBRE 2023

Ouvrant	Fenêtre aluminium		Fenêtre PVC		Baie aluminium	
	gauche	droite	gauche	droite	gauche	droite
État	- ouvert - verrouillé - invalide	- ouvert - oscillo-battant - verrouillé - invalide	- ouvert - verrouillé - Invalide	- ouvert - oscillo-battant - verrouillé - invalide	- ouvert - fermé - verrouillé - Invalide	- ouvert - fermé - verrouillé - Invalide
	4 niveaux Ⓟ ouvert / oscillo-battant ou fermé / verrouillé / invalide					

Matter

OTODO

- La société OTODO a été créée en 2016
- OTODO commercialise en marque blanche une plateforme d'interopérabilité dédiée à la maison connectée, notamment destinée aux opérateurs de services et/ou télécoms.
- Située à Tours et Paris

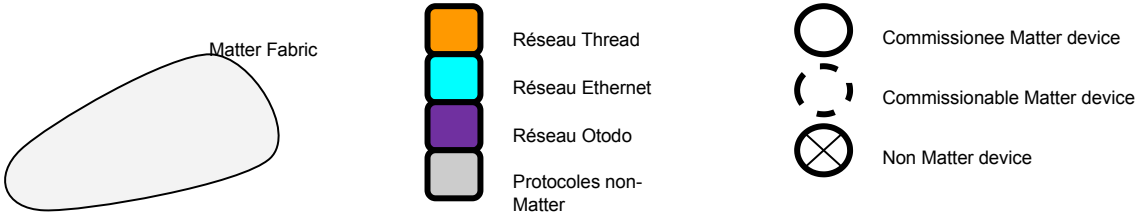
Protocole MATTER

matter

Matter c’est :

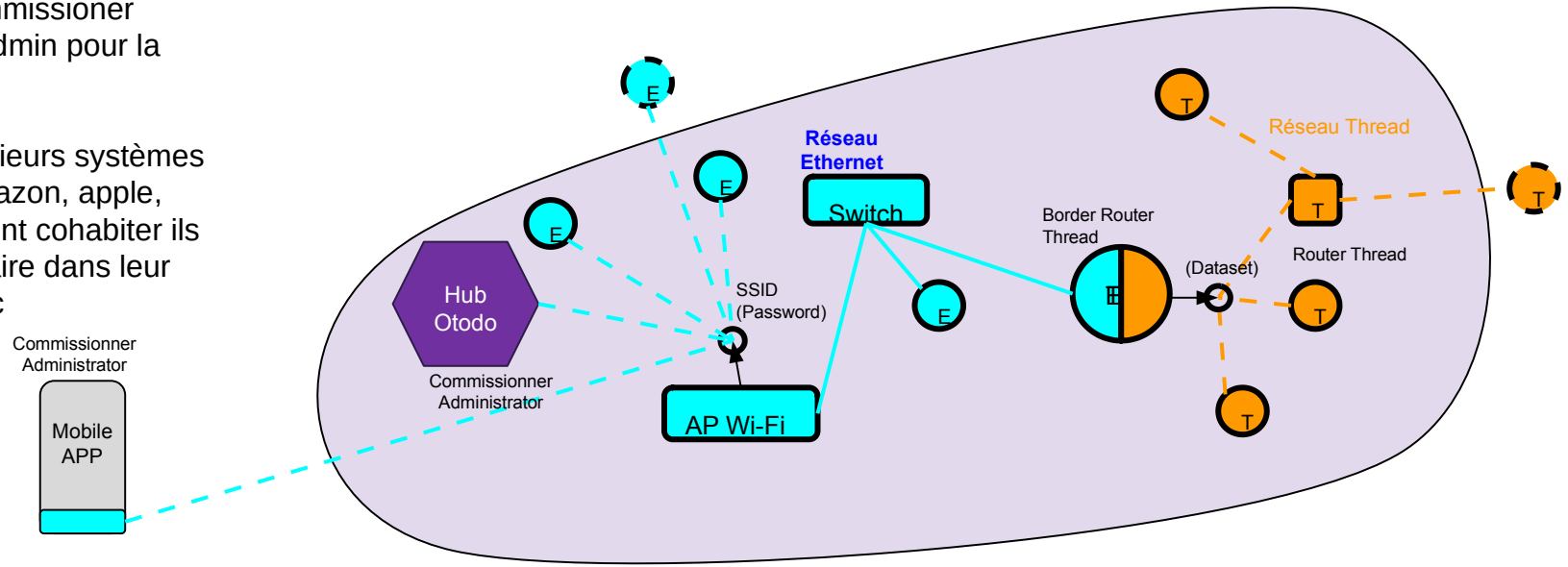
- Un standard de connectivité pour la domotique, dérivé du protocole Zigbee. Il a été conçu pour améliorer la communication entre les appareils domotiques, quel que soit leur vendeur/fabricant, et pour assurer leur interopérabilité.
- Il définit les couches OSI au-dessus de IPV6
- S’appuie sur des standards existants (protocoles sur IP (LAN), Thread, BLE, Zigbee…)
- Est géré par le consortium CSA (Connectivity Standards Alliance) ex Zigbee

Réseau MATTER



Un seul commissioner et un seul admin pour la Fabric!

Donc si plusieurs systèmes (google, amazon, apple, otodo) doivent cohabiter ils devront le faire dans leur propre fabric



Protocole MATTER

Termes à retenir :

MATTER :

- Fabric
- Commissioner : Node ayant la capacité de commissioner un autre Node.
- Administrator (Node ayant tous les droits d'accès).
- Node (~device) = Entité adressable (Attention il va falloir jongler entre ces deux termes !)

Wi-Fi

- AP (Access Point)
- SSID (Identifiant du réseau)
- Password (mot de passe du réseau)

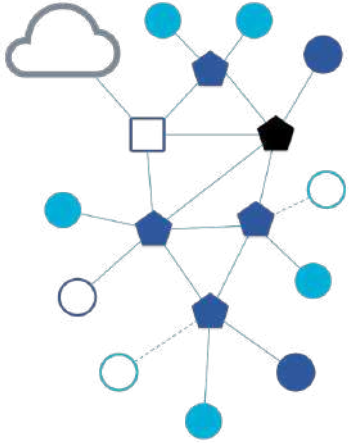
Thread

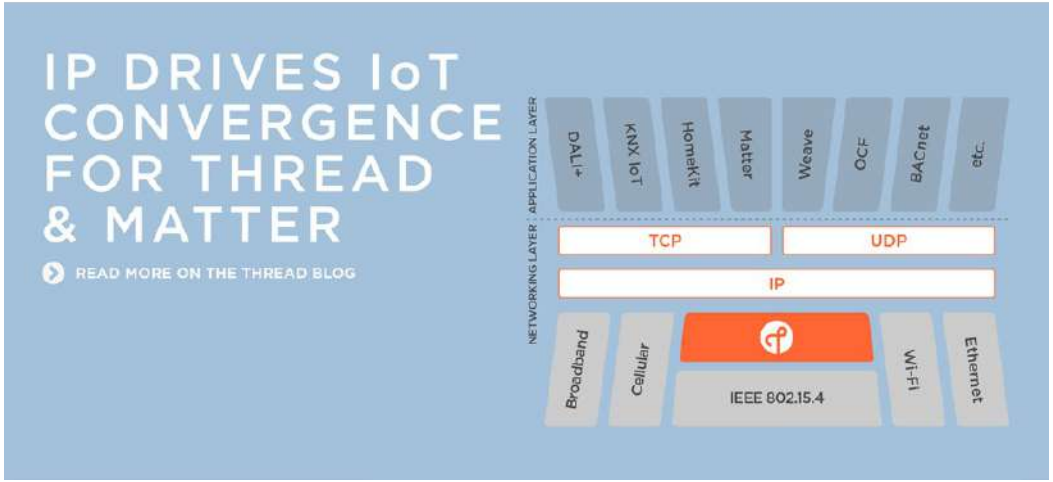
- Border Router ou OT-BR (Open Thread Border Router)
- Router (équipement alimenté en permanence (prise, lampe) permettant d'étendre le réseau Thread)
- DATASET (~ SSID/PWD Wi-Fi + numéro de canal +...)

Protocole MATTER

Thread est :

- un protocole radio (basé sur le 802.15.4 en 2.4GHz, comme pour zigbee)
- un protocole bas débit
- plutôt destiné aux objets basse consommation
- supporte le maillage
- IPV6 (avec quelques limitations) - facilite le développement de passerelles LAN-Thread





Commissioning MATTER

Les étapes du Commissioning

1. Découverte de l'objet (BLE, On-Network)
2. Obtention des infos permettant d'identifier l'objet
3. Recherche de l'objet dans l'environnement proche
4. Établissement d'une première connexion sécurisée
5. Vérification de son authenticité et sa certification
6. Configuration pour l'intégration dans le réseau local
7. Détection de l'objet sur le réseau local
8. Établissement d'une session sécurisée pour la communication "matter"

Commissioning MATTER

Comment identifier l'objet?

Les informations disponibles sur l'objet sont :

- **Vendor ID** : Permet d'identifier le fabricant (la marque) (optionnel sur le manual pairing code)
- **Product ID** : Permet d'identifier le produit (associé à la marque) (optionnel sur le manual pairing code)
- **Discovery Capabilities** (BLE, on network). Dans tous les cas le commissioner fera aussi une découverte à l'aide de DNS-SD
- **Discriminator** : entier sur 12 bits permettant de différencier plusieurs objets identiques (4 bits sur le manual pairing code)
- **Passcode** : Entier 27 bits utilisé en tant que preuve de possession. Évite que le voisin ou une personne mal intentionnée commissione l'objet à notre place.

- **Manual pairing code**

  Short : 34970112332

  Long : 749701123309050652796

- **QR Code**

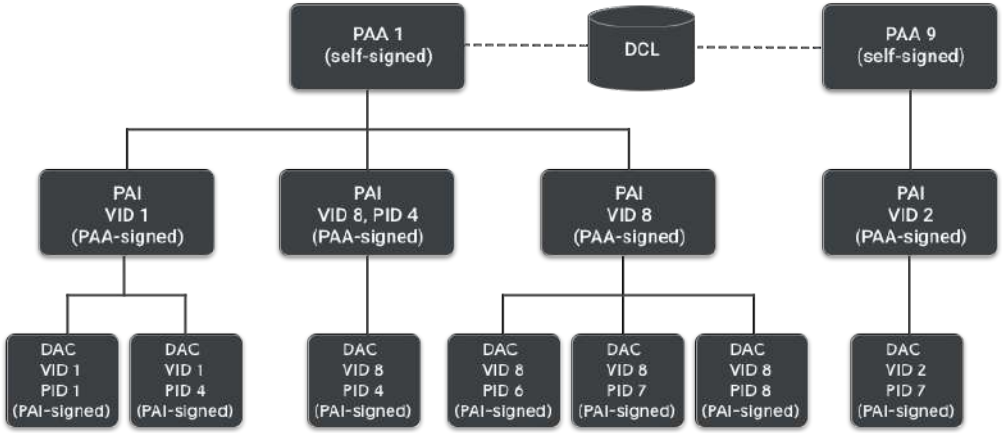
MT:C8XA0-1K010O0648G00

VID	0x235A (9050)
PID	0x4541 (17729)
Discriminator	0x0F01 (3841)
Setup Pin Code (Password)	20202022
Pairing	BLE

Commissioning MATTER

Vérification de l'authenticité et du statut de la certification de l'objet?

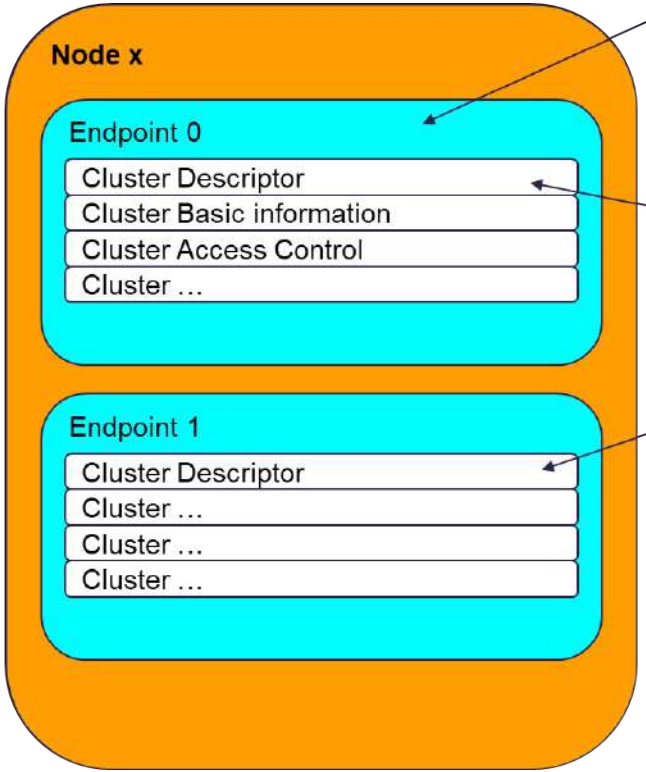
- repose sur une infrastructure à clés publiques (PKI)
- Les PAI et DAC sont sauvegardés dans chaque objet.
- Les PAA sont stockés sur le serveur DCL (qu'il faudra donc interroger périodiquement).



PAA : Product Attestation Authority, **PAI** : Product Attestation Intermediate, **DAC** : Device Attestation Certificate

Data Model MATTER

Le modèle :



Le Endpoint 0 est obligatoire pour chaque Node.
C'est le point d'entrée de la découverte de l'objet.
Il doit implémenter le device type "Root Node"

Chaque Endpoint a un cluster "Descriptor"

Il indique le(s) type(s) de device implémenté(s) par le endpoint

Les device Type sont figés pour une version de spec. Ils sont décrits dans le document "Matter Device Library"

Un device type précise quels cluster doivent êtres implémentés.

Certification MATTER

Pourquoi certifier ?

En plus de la reconnaissance de la conformité à une spécification du CSA, la certification permet l'utilisation de logos de produits certifiés et la liste du produit sur le site Web de l'Alliance à des fins de vérification. La certification indique la conformité à une spécification et montre l'interopérabilité dans le programme respectif.

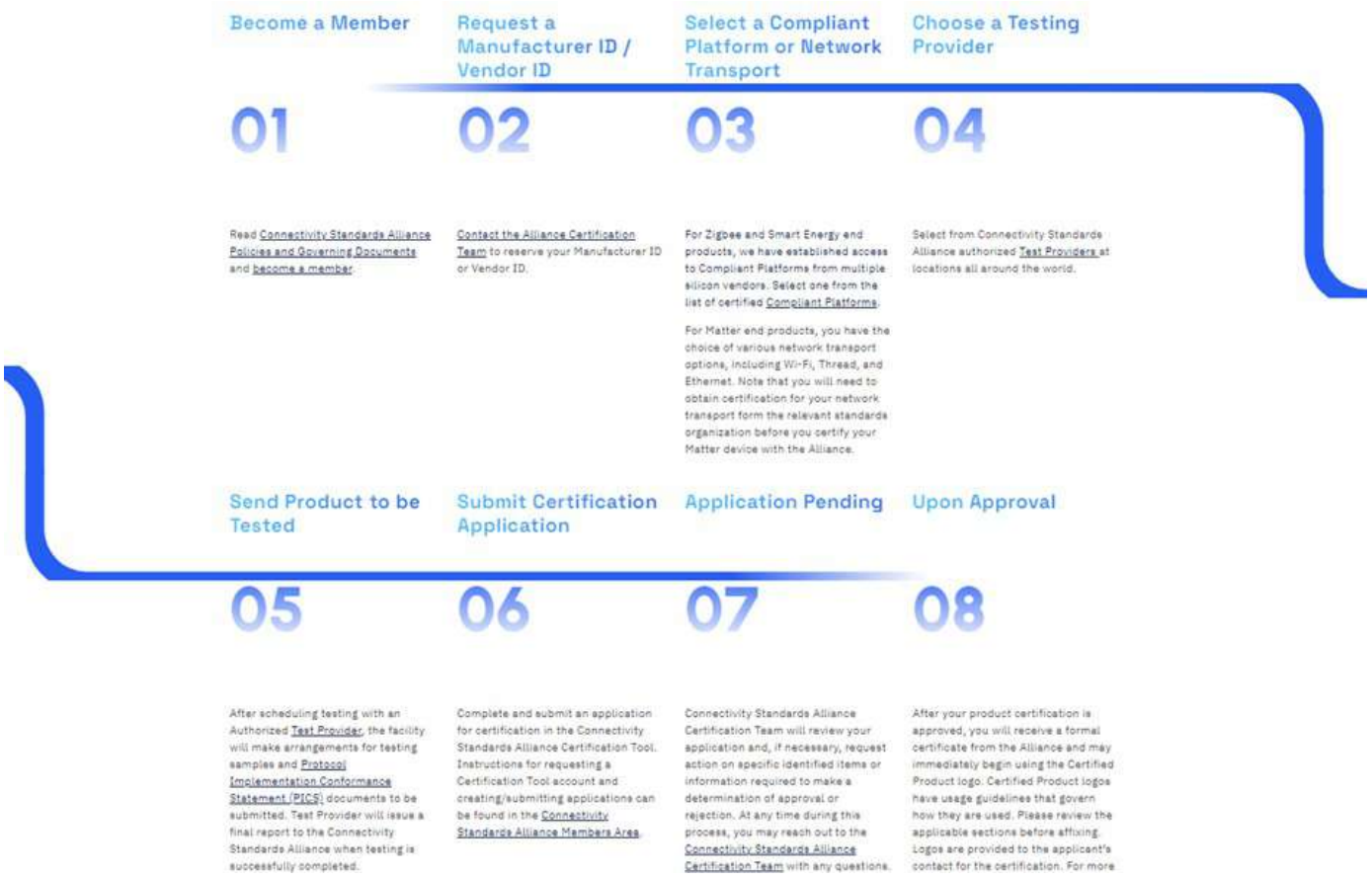
Avantages de la certification des produits :

Indique clairement et indépendamment l'interopérabilité avec les produits des plus grandes marques mondiales

Réduit les barrières à l'entrée sur le marché grâce à l'accès à un vaste écosystème technologique

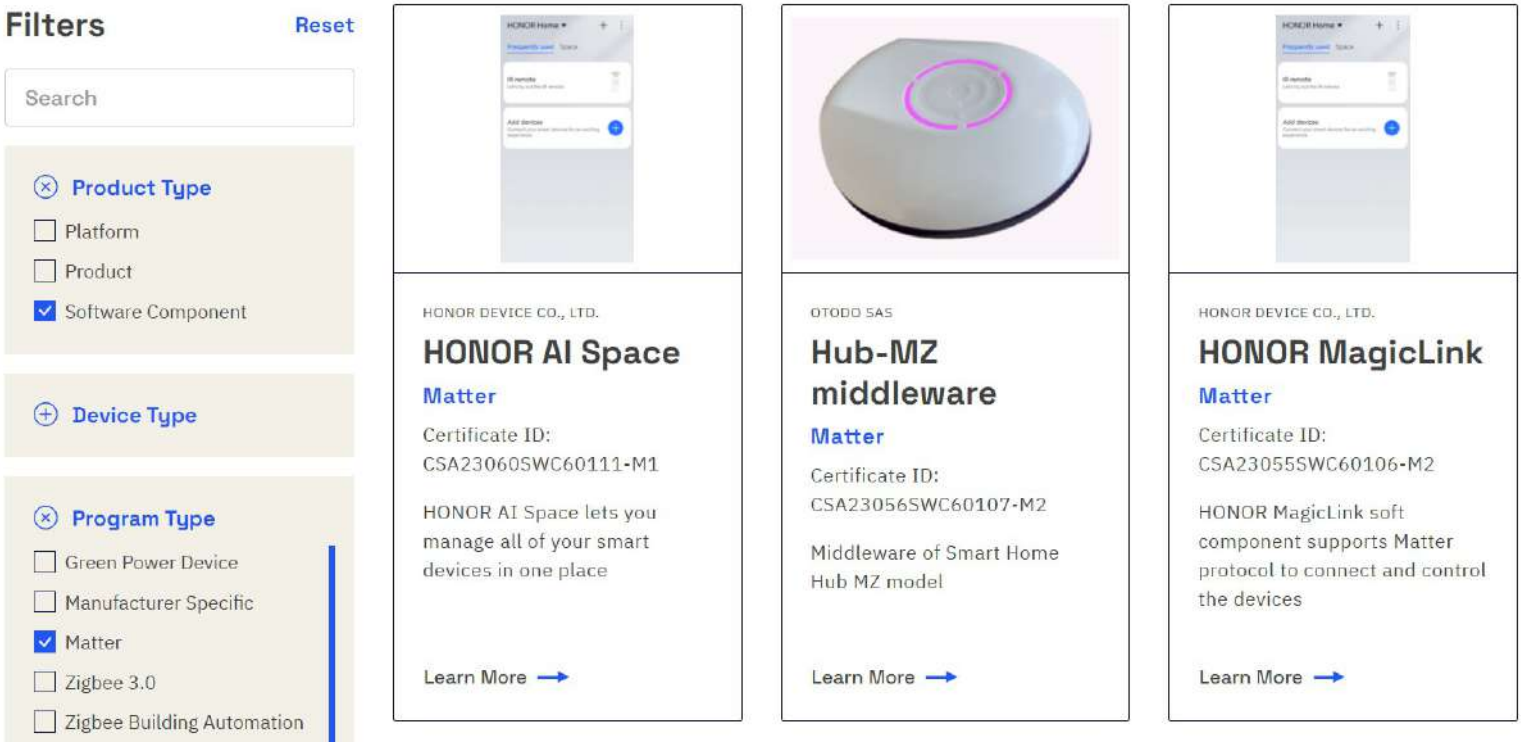
Permet d'utiliser des logos de produits certifiés exclusifs Connectivity Standards Alliance et une liste de produits sur le site Web de Connectivity Standards Alliance.

Certification MATTER



Certification MATTER

Otodo a certifié le firmware de son Hub-MZ



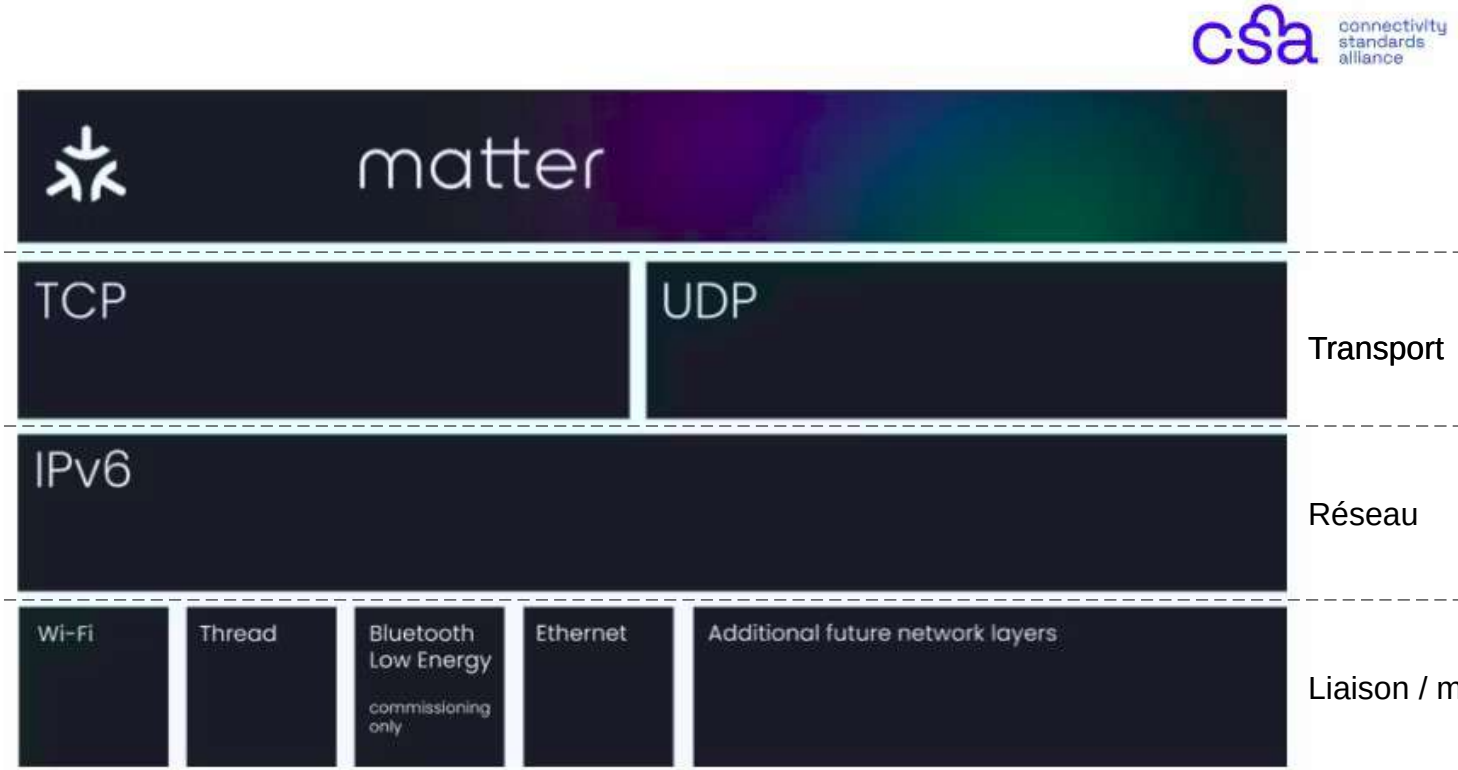
MATTER 1.2

Nouvelle version de Matter tous les 6 mois.
La liste des devices types ajoutés dans Matter 1.2 est :

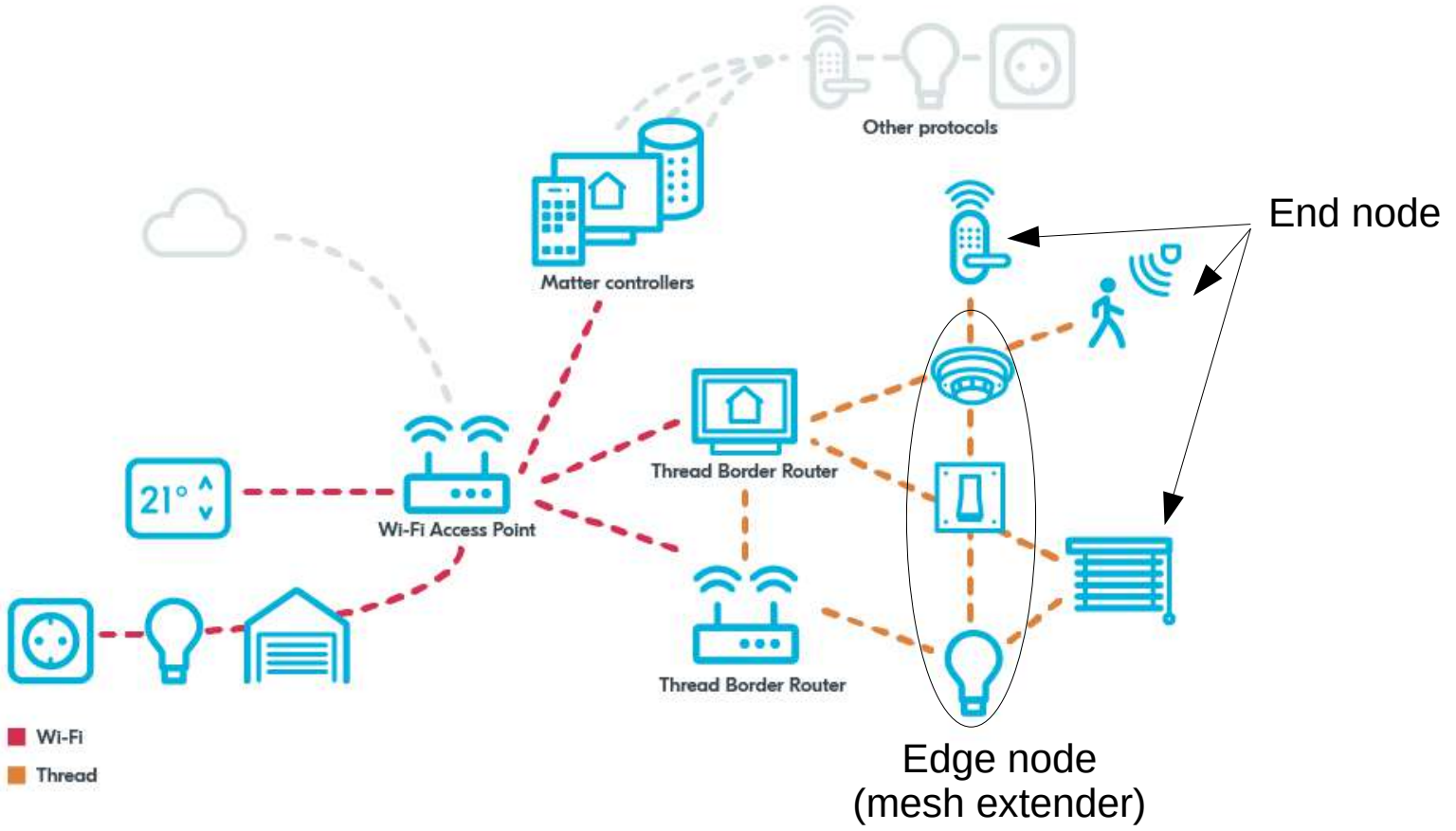
- Smoke CO Alarm
- Air Purifier
- Air Quality Sensor
- Robotic Vacuum Cleaner
- Laundry Washer
- Refrigerator
- Room Air Conditioner
- Temperature Controlled Cabinet
- Dishwasher

Mise en œuvre

Architecture Matter



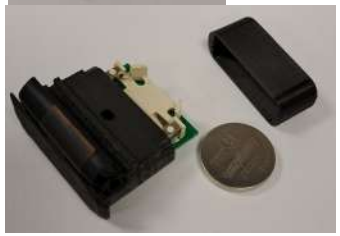
Ecosystème Matter



Src : nordic semiconductor

End node Matter

Objectif : mettre à disposition l'état de la fenêtre



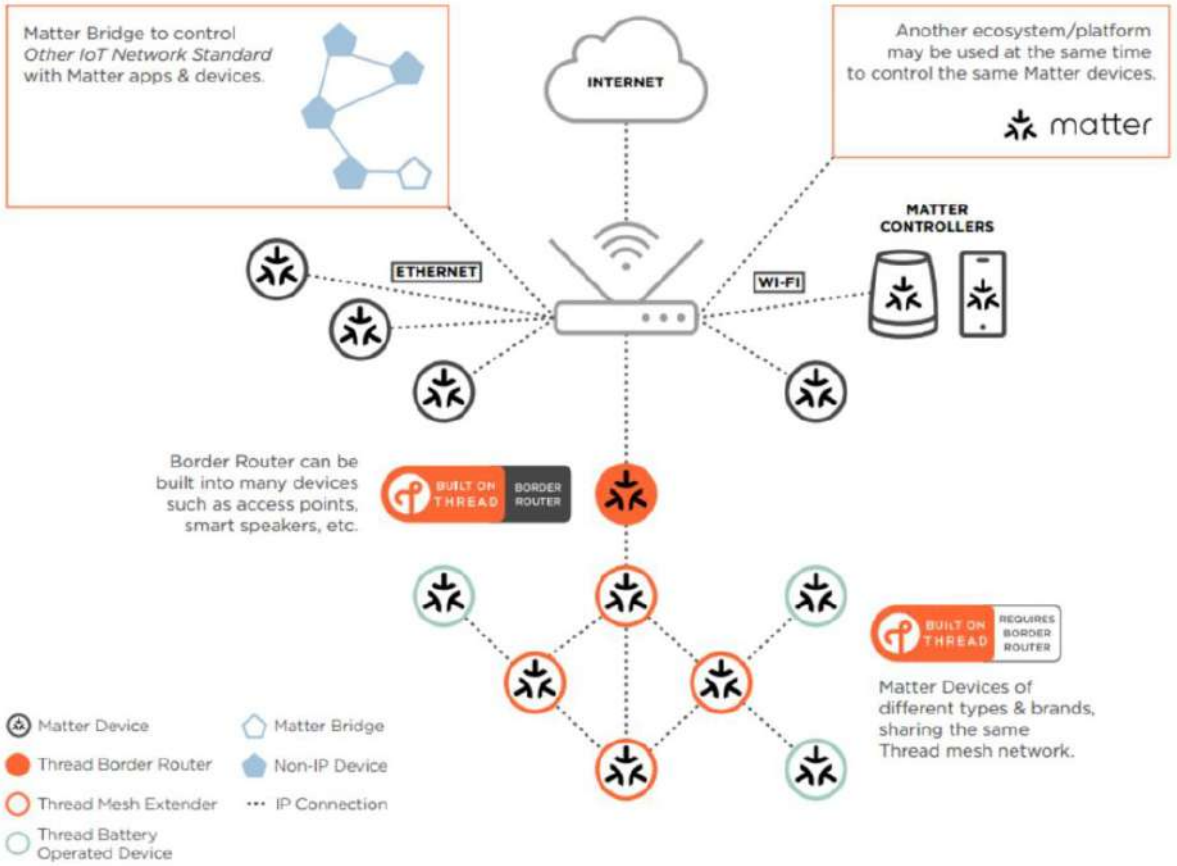
Contraintes

- Capteurs embarqués
- Longue autonomie
- Faible encombrement
- Supporte Matter

Début de solution

- Technologie radio : Thread

Thread & Matter



Src : threadgroup.org

Thread device

Device types

OpenThread distinguishes between different device types depending on what roles a device can take in the Thread network.

Full Thread Device (FTD)

An FTD can be both a router and an end device. Its radio is always on.

Minimal Thread Device (MTD)

An MTD is always an end device. It forwards all messages to its parent.

There are two important subtypes:

Minimal End Device (MED)

An MED keeps its transceiver always on.

Sleepy End Device (SED) and Synchronized Sleepy End Device (SSED)

An SED is usually off and wakes occasionally to receive messages from its parent.

An SSED is an enhanced SED. It transmits less data than an SED and relies on receiving messages from its parent only in specified time intervals.

For more information, see the Thread device types page.

Src : nordic semiconductor

Thread device

Device types

OpenThread distinguishes between different device types depending on what roles a device can take in the Thread network.

Full Thread Device (FTD)

An FTD can be both a router and an end device. Its radio is always on.

Minimal Thread Device (MTD)

An MTD is always an end device. It forwards all messages to its parent.

There are two important subtypes:

Minimal End Device (MED)

An MED keeps its transceiver always on.

Sleepy End Device (SED) and Synchronized Sleepy End Device (SSED)

An SED is usually off and wakes occasionally to receive messages from its parent.

An SSED is an enhanced SED. It transmits less data than an SED and relies on receiving messages from its parent only in specified time intervals.

For more information, see the Thread device types page.

Src : nordic semiconductor

Thread device (SED)

Minimal Thread Devices

Minimal Thread Devices (MTDs) are devices that do not maintain a routing table and are typically low-power devices that are not always on. They can only be End Devices, and they always need a parent to function. They forward all their messages to their parent.

Minimal Thread Devices can be further divided into three categories:

- Minimal End Device (MED)
- Sleepy End Device (SED)
- Synchronized Sleepy End Device (SSED)

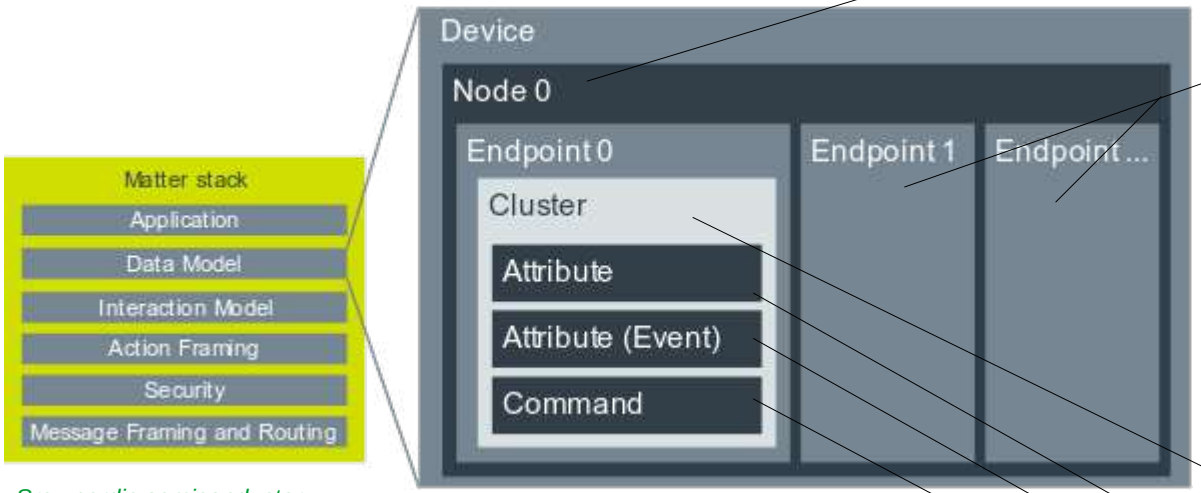
	Minimal End Device	Sleepy End Device	Synchronized Sleepy End Device
Maintains a routing table	No	No	No
Radio	Always enabled	Periodically enabled	Periodically enabled
New messages	Parent immediately forwards the messages	The SED polls for new messages when it wakes up	Parent forwards the messages during designated transmission window or when SSED explicitly polls data
No new messages	No data transmission	Parent indicates no pending messages	No data transmission*

Minimal Thread Device categories

Src : nordic semiconductor

Matter Device

- Meta-Model | data model



Src : nordic semiconductor

Ressource unique identifiable et adressable dans le réseau composée d'un ensemble de fonctions et de capacités (1 seul nœud par appareil en général)

Conforme à un ou plusieurs « device type », déterminés suivant la version de matter, qui définissent le(s) cluster(s) supporté(s)

Ex device type :
base : node, app, client, server…
utility : Root node, power source…
app : on/off light, contact sensor, temp sensor, door lock...

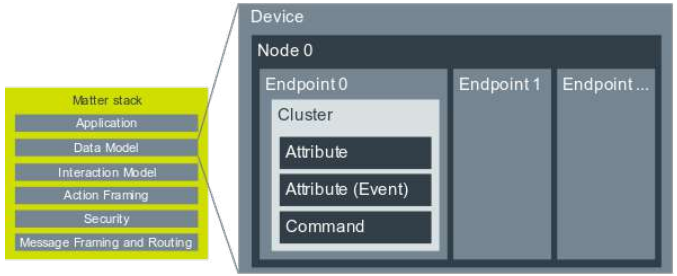
Interface fonctionnelle qui définit les interactions (client/serveur) possibles

États actuels typés (bool, int, string..)
États passés
Actions pouvant être effectuées

https://developers.home.google.com/matter/primer/device-data-model?hl=fr

Matter Device

- Exemple d'un interrupteur



ID	Device Name	Superset	Class	Scope
0x0103	On/Off Light Switch		Simple	Endpoint

Table 12. Attributes of the On/Off Server Cluster

ID	Name	Type	Constraint	Quality	Default	Access	Conformance
0x0000	OnOff	bool	all	SN	FALSE	R V	M
0x4000	GlobalSceneControl	bool	all		TRUE	R V	LT
0x4001	OnTime	uint16	all	X	0	RW VO	LT
0x4002	OffWaitTime	uint16	all	X	0	RW VO	LT
0x4003	StartUpOnOff	StartUpOnOffEnum	desc	XN	MS	RW VM	LT

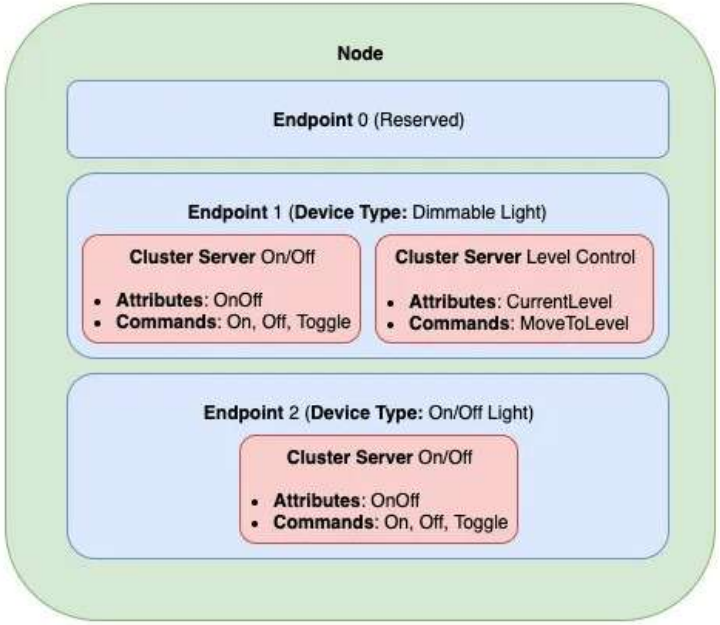
Table 10. On/Off Light Switch Cluster Requirements

ID	Cluster	Client/Server	Quality	Conformance
0x0003	Identify	Server		M
0x0003	Identify	Client		M
0x0004	Groups	Client		O
0x0005	Scenes	Client		O
0x0006	On/Off	Client		M

Src : nordic semiconductor

Matter Device

- Exemple d'une commande de lampe dimmable



Simplistic Representation of a Matter Data Model

Src : nordic semiconductor

Endpoint matter

Endpoint 0 réservé aux device type utility :
dont le Root Node :

- Shall be root node point
- Shall support the Root Node device type
- A Root Node device type should be a singleton on the root node endpoint
- Only one root node endpoint is accepted for the node
- The PartList of the descriptor cluster on the root node endpoint SHALL list all endpoints on the node, except the root node endpoint

Root Node

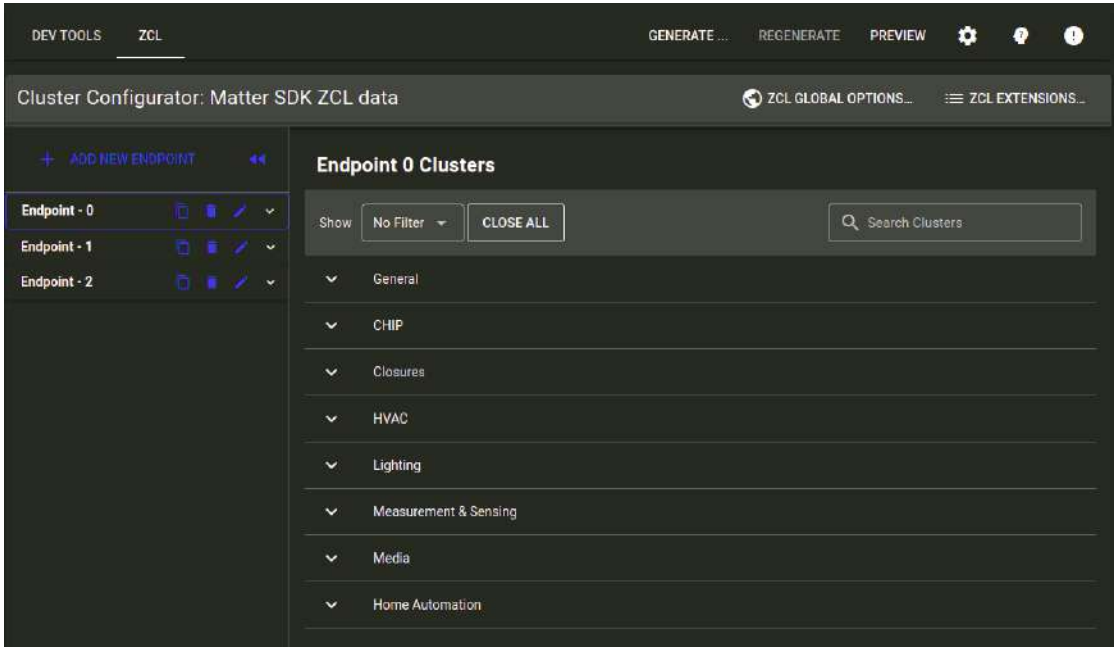
ID	ClusterName	Client/Server	Quality	Conformance
0x0028	Basic Information	Server	I	M
0x001F	Access Control	Server	I	M
0x002E	Power Source Configuration	Server	I	O
0x0038	Time Synchronization	Server	I	P, O
0x003F	Group Key Management	Server	I	M
0x0030	General Commissioning	Server	I	M
0x0031	Network Commissioning	Server		!CustomNetworkConfig
0x003C	Administrator Commissioning	Server	I	M
0x003E	Node Operational Credentials	Server	I	M
0x002B	Localization Configuration	Server	I	LanguageLocale
0x002C	Time Format Localization	Server	I	TimeLocale
0x002D	Unit Localization	Server	I	UnitLocale
0x0033	General Diagnostics	Server	I	M
0x0032	Diagnostic Logs	Server	I	O

ID	ClusterName	Client/Server	Quality	Conformance
0x0034	Software Diagnostics	Server	I	O
0x0037	Ethernet Network Diagnostics	Server	I	[Ethernet]
0x0036	Wi-Fi Network Diagnostics	Server	I	[Wi-Fi]
0x0035	Thread Network Diagnostics	Server	I	[Thread]

Cluster requirement for root node

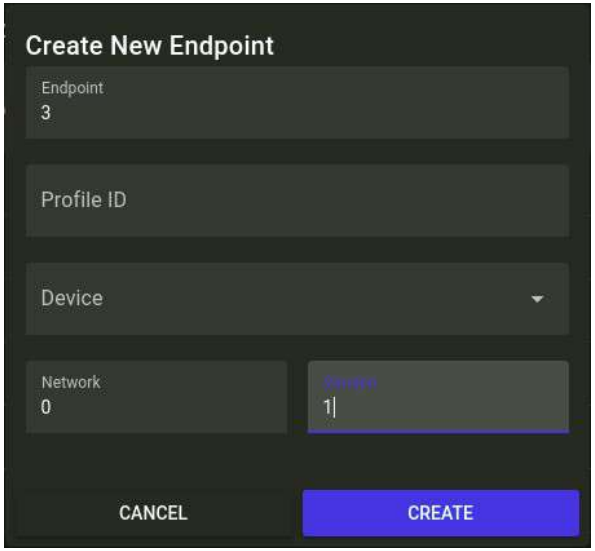
Outils de génération

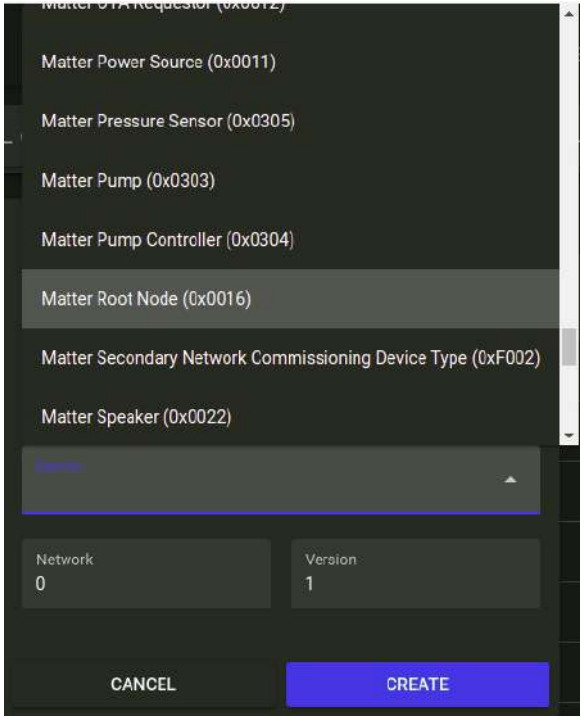
- Utilitaire ZAP
  - Outil graphique basé sur la librairie de cluster Zigbee
  - Génère le modèle de données (data model)

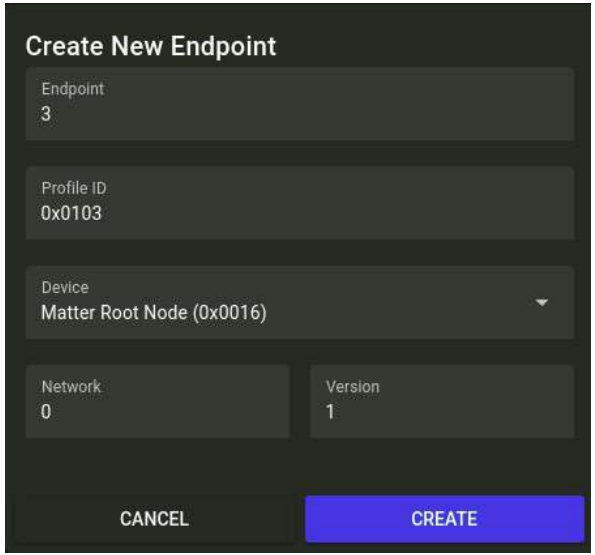


Outils de génération

- Utilitaire ZAP : endpoint

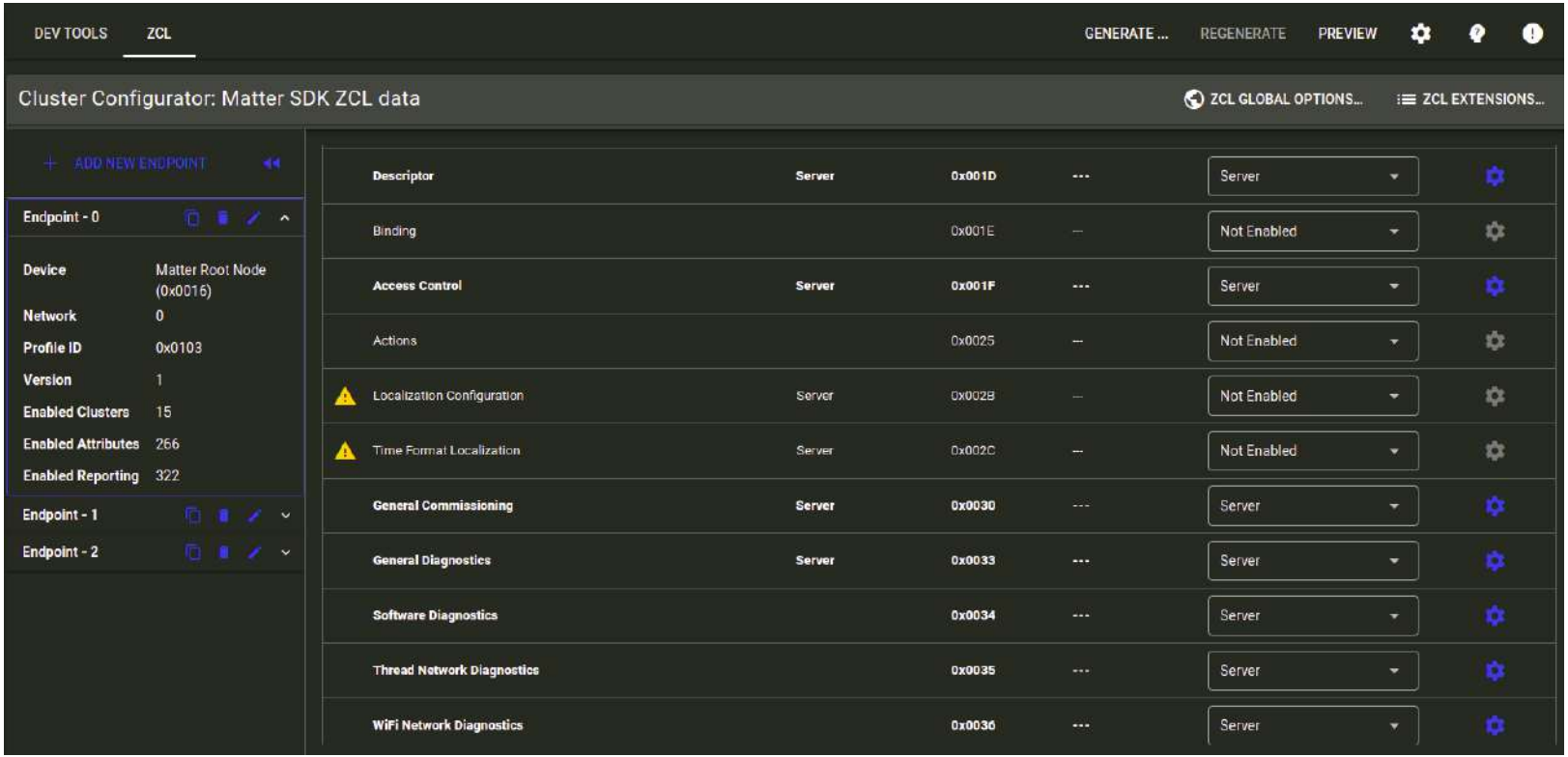






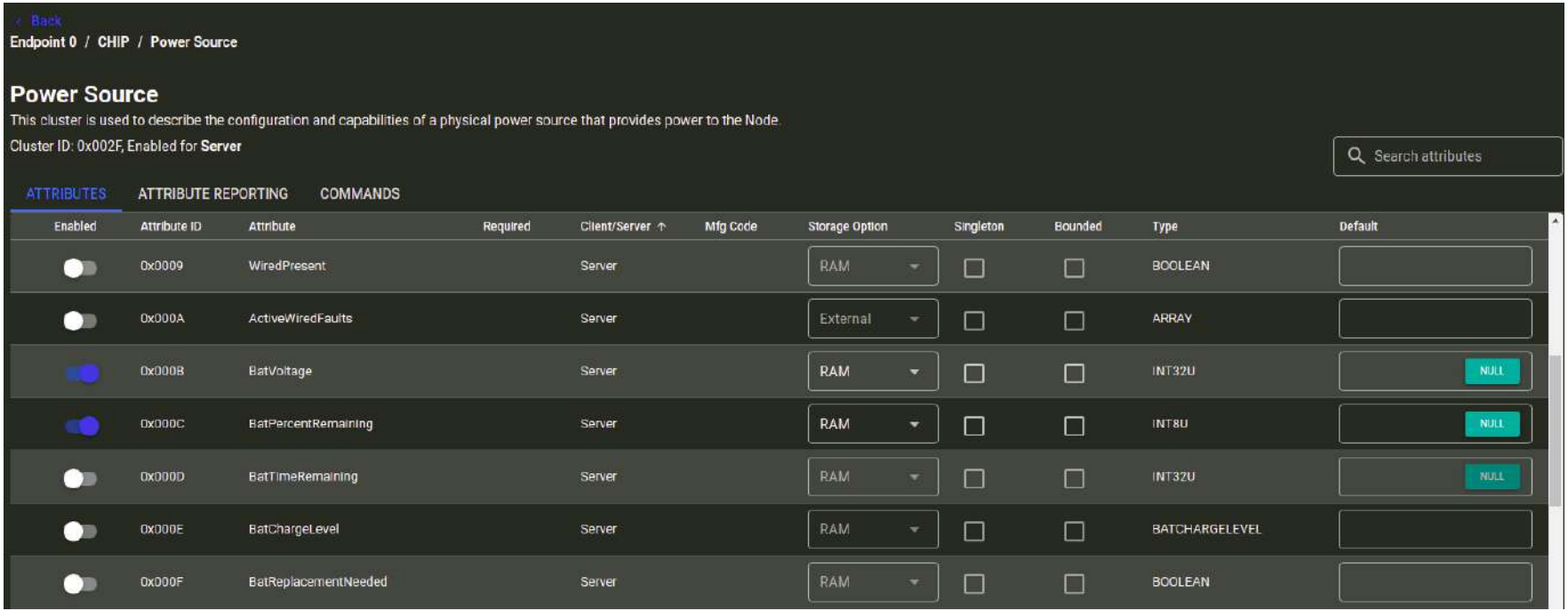
Outils de génération

- Utilitaire ZAP : cluster



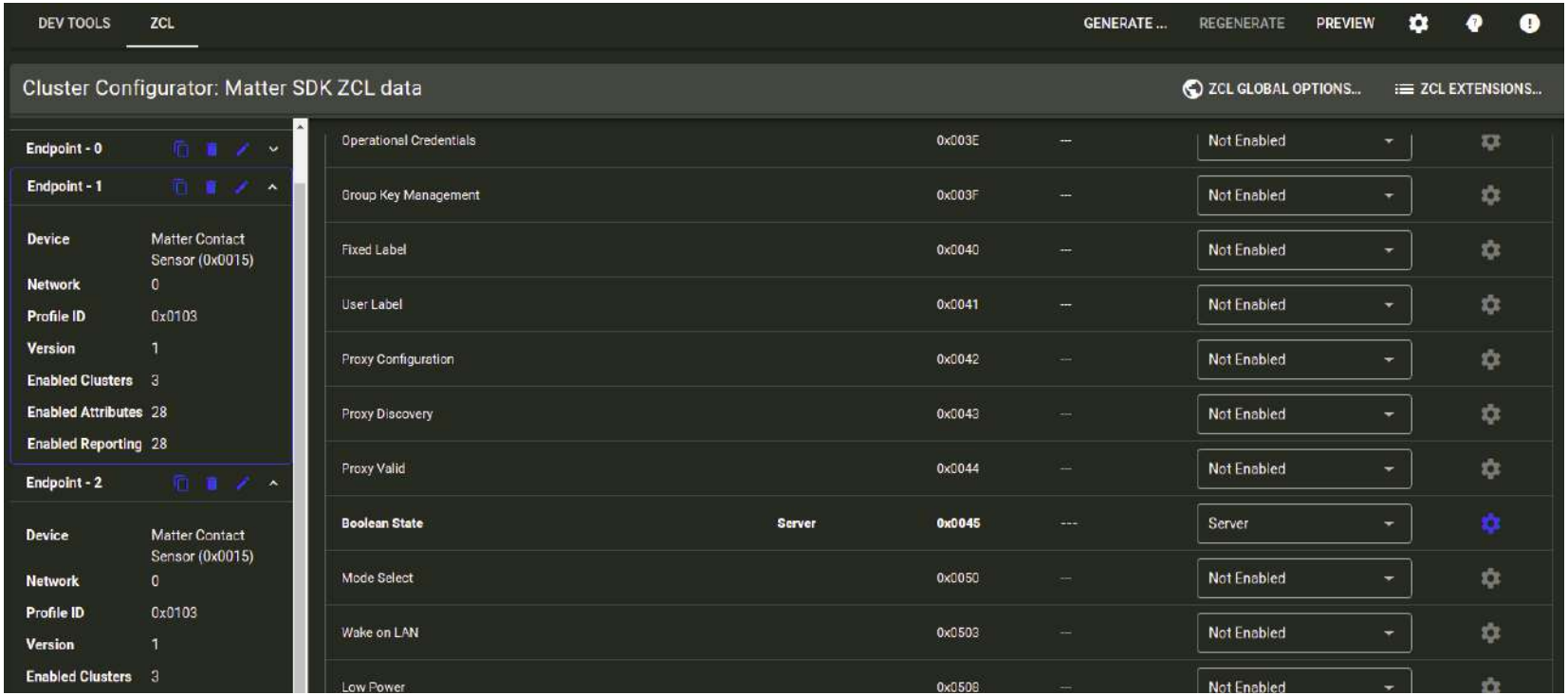
Outils de génération

- Utilitaire ZAP : cluster - attributs



Outils de génération

- Utilitaire ZAP :



Outils de génération

- Utilitaire ZAP : génération des librairies

access.h

callback-stub.cpp

CHIPClient Callbacks.h

CHIPCluste rs.h

Clusters. matter

endpoint_ config.h

gen_config. h

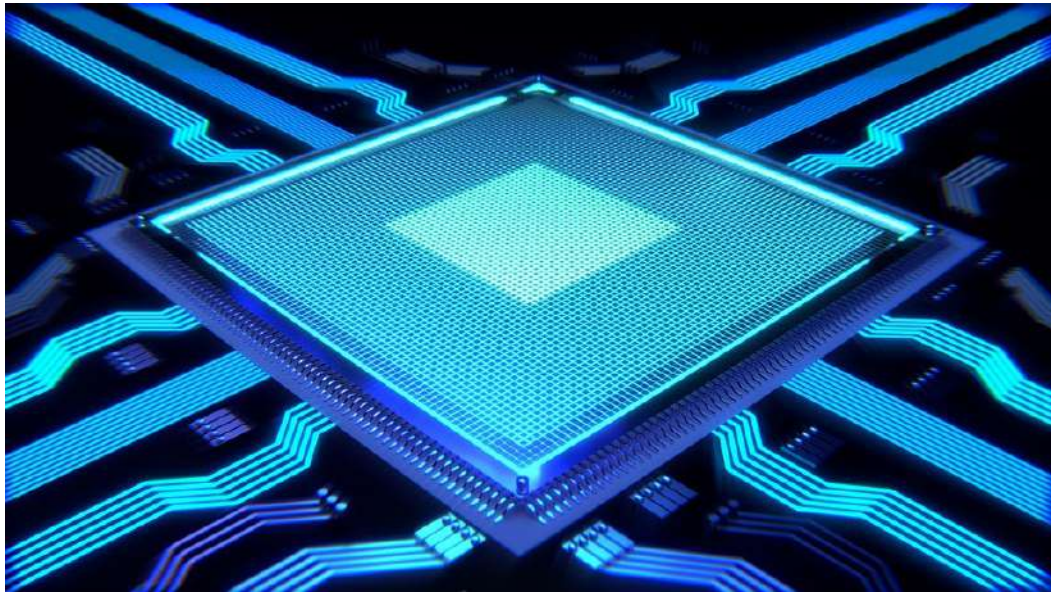
IMClusterC ommandHa ndler.cpp

PluginAppli cationCallb acks.h

```
#define ZAP_ATTRIBUTE_MASK(mask) ATTRIBUTE_MASK_ ## mask
// This is an array of EmberAfAttributeMetadata structures.
#define GENERATED_ATTRIBUTE_COUNT 183
#define GENERATED_ATTRIBUTES { \
\
  /* Endpoint: 0, Cluster: Descriptor (server) */ \
  { 0x00000000, ZAP_TYPE(ARRAY), 2, 0, ZAP_SIMPLE_DEFAULT(0) }, /* DeviceTypeList */  \
  { 0x00000001, ZAP_TYPE(ARRAY), 0, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE), ZAP_EMPTY_DEFAULT() }, /* ServerList */  \
  { 0x00000002, ZAP_TYPE(ARRAY), 0, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE), ZAP_EMPTY_DEFAULT() }, /* ClientList */  \
  { 0x00000003, ZAP_TYPE(ARRAY), 0, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE), ZAP_EMPTY_DEFAULT() }, /* PartsList */  \
  { 0x0000FFFC, ZAP_TYPE(BITMAP32), 4, 0, ZAP_SIMPLE_DEFAULT(0) }, /* FeatureMap */  \
  { 0x0000FFFD, ZAP_TYPE(INT16U), 2, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE), ZAP_EMPTY_DEFAULT() }, /* ClusterRevision */  \
\
  /* Endpoint: 0, Cluster: Access Control (server) */ \
  { 0x00000000, ZAP_TYPE(ARRAY), 0, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(WRITABLE), ZAP_EMPTY_DEFAULT() }, /* ACL */  \
  { 0x00000001, ZAP_TYPE(ARRAY), 0, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(WRITABLE), ZAP_EMPTY_DEFAULT() }, /* Extension */  \
  { 0x00000002, ZAP_TYPE(INT16U), 2, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE), ZAP_EMPTY_DEFAULT() }, /* SubjectsPerAccessControlEntry */  \
  { 0x00000003, ZAP_TYPE(INT16U), 2, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE), ZAP_EMPTY_DEFAULT() }, /* TargetsPerAccessControlEntry */  \
  { 0x00000004, ZAP_TYPE(INT16U), 2, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE), ZAP_EMPTY_DEFAULT() }, /* AccessControlEntriesPerFabric */  \
  { 0x0000FFFC, ZAP_TYPE(BITMAP32), 4, 0, ZAP_SIMPLE_DEFAULT(0) }, /* FeatureMap */  \
  { 0x0000FFFD, ZAP_TYPE(INT16U), 2, 0, ZAP_SIMPLE_DEFAULT(1) }, /* ClusterRevision */  \
\
  /* Endpoint: 0, Cluster: Basic (server) */ \
  { 0x00000000, ZAP_TYPE(INT16U), 2, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* DataModelRevision */  \
  { 0x00000001, ZAP_TYPE(CHAR_STRING), 33, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* VendorName */  \
  { 0x00000002, ZAP_TYPE(VENDOR_ID), 2, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* VendorID */  \
  { 0x00000003, ZAP_TYPE(CHAR_STRING), 33, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* ProductName */  \
  { 0x00000004, ZAP_TYPE(INT16U), 2, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* ProductID */  \
  { 0x00000005, ZAP_TYPE(CHAR_STRING), 33, ZAP_ATTRIBUTE_MASK(TOKENIZE) | ZAP_ATTRIBUTE_MASK(SINGLETON) | ZAP_ATTRIBUTE_MASK(WRITABLE), ZAP_EMPTY_DEFAULT() }, /*
 NodeLabel */  \
  { 0x0000000e, ZAP_TYPE(CHAR_STRING), 3, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON) | ZAP_ATTRIBUTE_MASK(WRITABLE), ZAP_EMPTY_DEFAULT() }, /
 * Location */  \
  { 0x00000007, ZAP_TYPE(INT16U), 2, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* HardwareVersion */  \
  { 0x00000008, ZAP_TYPE(CHAR_STRING), 65, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* HardwareVersionString */  \
  { 0x00000009, ZAP_TYPE(INT32U), 4, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* SoftwareVersion */  \
  { 0x0000000A, ZAP_TYPE(CHAR_STRING), 65, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* SoftwareVersionString */  \
  { 0x0000000B, ZAP_TYPE(CHAR_STRING), 17, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* ManufacturingDate */  \
  { 0x0000000F, ZAP_TYPE(CHAR_STRING), 33, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* SerialNumber */  \
  { 0x00000012, ZAP_TYPE(CHAR_STRING), 33, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_EMPTY_DEFAULT() }, /* UniqueID */  \
  { 0x00000013, ZAP_TYPE(STRUCT), 0, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE), ZAP_EMPTY_DEFAULT() }, /* CapabilityMinima */  \
  { 0x0000FFFC, ZAP_TYPE(BITMAP32), 4, 0, ZAP_SIMPLE_DEFAULT(0) }, /* FeatureMap */  \
  { 0x0000FFFD, ZAP_TYPE(INT16U), 2, ZAP_ATTRIBUTE_MASK(SINGLETON), ZAP_SIMPLE_DEFAULT(1) }, /* ClusterRevision */  \
\
  /* Endpoint: 0, Cluster: OTA Software Update Requestor (server) */ \
  { 0x00000000, ZAP_TYPE(ARRAY), 0, ZAP_ATTRIBUTE_MASK(EXTERNAL_STORAGE) | ZAP_ATTRIBUTE_MASK(WRITABLE), ZAP_EMPTY_DEFAULT() }, /* DefaultOtaProviders */  \
  { 0x00000001, ZAP_TYPE(BOOLEAN), 1, 0, ZAP_SIMPLE_DEFAULT(1) }, /* UpdatePossible */  \
```

```
bool __attribute__((weak)) emberAfAttributeWriteAccessCallback(
    EndpointId endpoint, ClusterId clusterId, AttributeId attributeId)
{
    return true;
}

bool __attribute__((weak)) emberAfDefaultResponseCallback(
    ClusterId clusterId, CommandId commandId, EmberAfStatus status)
{
    return false;
}

bool __attribute__((weak)) emberAfPreMessageSendCallback(
    EmberAfMessageStruct * messageStruct, EmberStatus * status)
{
    return false;
}

bool __attribute__((weak)) emberAfMessageSentCallback(
    const MessageSendDestination & destination,
    EmberApsFrame * apsFrame, uint16_t msgLen, uint8_t * message,
    EmberStatus status)
{
    return false;
}

EmberAfStatus __attribute__((weak)) emberAfExternalAttributeReadCallback(
    EndpointId endpoint, ClusterId clusterId,
                                                const EmberAfAttributeMetadata * attributeMetadata,
                                                uint8_t * buffer, uint16_t maxReadLength)
{
    return EMBER_ZCL_STATUS_FAILURE;
```

Choix du microcontrôleur



Matériel

- Contraintes :

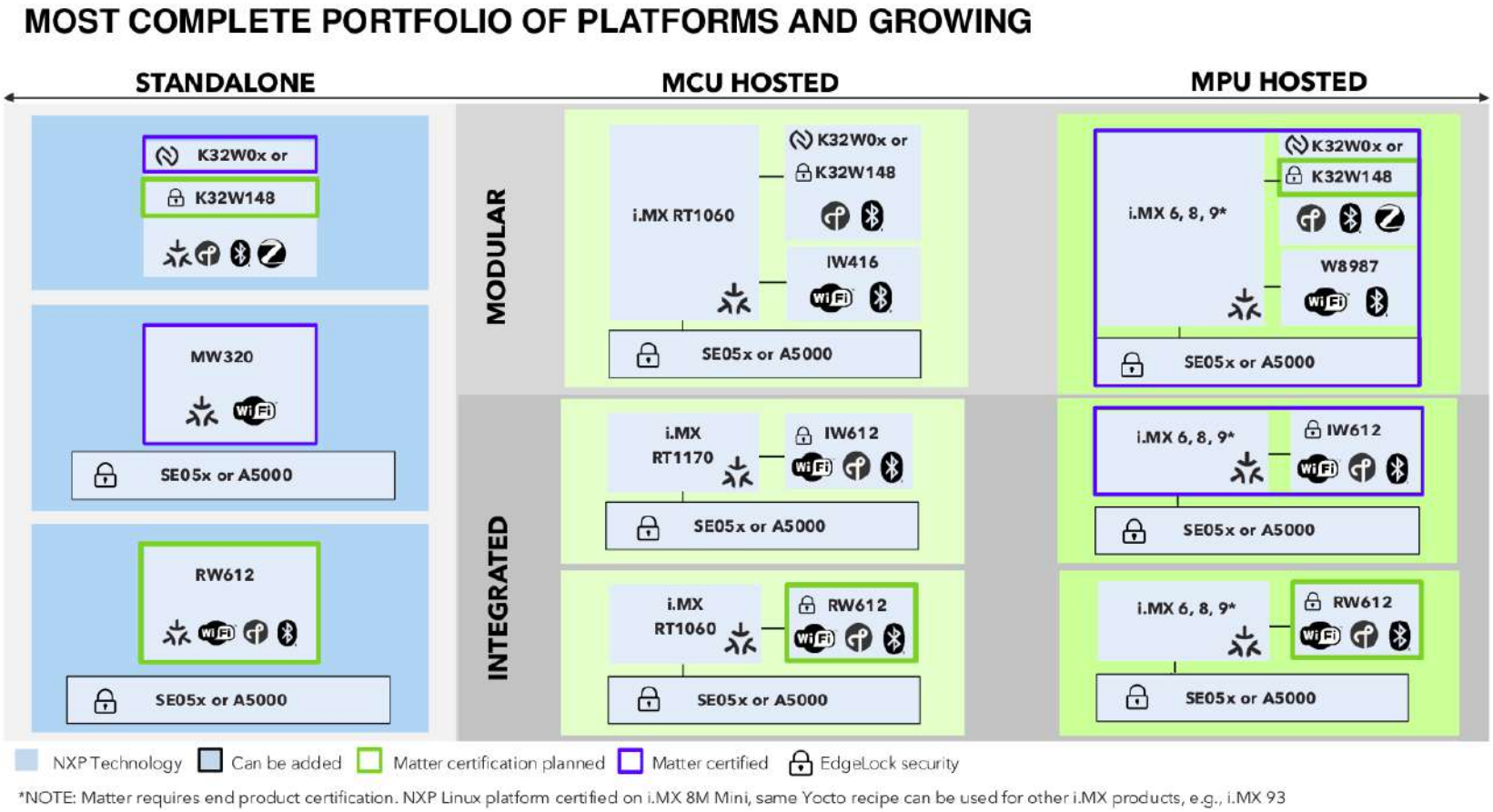
- Pile Matter 1.0
- Thread
- Interface capteurs
- Peu encombrant (SoC)
- Maîtriser

Matériel

- Possibilités:

- NXP
- STMicroelectronics
- Nordic Semicondutor
- Silicon Labs
- Texas Instrument
- Espressif
- ...

Solutions Matter NXP



Matter 1.1

Solutions Matter STMicroelectronics

STM32WB	• **Dual core** & security (Arm® Cortex®-M4 /-M0+) • Up to **1 Mbyte flash- memory/ 256 Kbytes RAM** • Bluetooth® Low Energy 5.4, Zigbee R22 & Thread, proprietary, Matter
Matter Over Thread End Device	• **STM32WB55: Dual core** & security (Arm® Cortex®-M4 /-M0+), **Ultra-low-power**; Up to **1 Mbyte flash- memory/ 256 Kbytes RAM** • **Matter ready: Matter SDK integrated, Matter Pre-certified, Thread 1.3 certified, BLE 5.4 certified** • **Optimized for Matter**: **Optimized OTA, Easy Matter provisioning APIs, Easy Device Programming** • **Wide peripheral coverage**: USB, QuadSPI, LCD, ADC, Temperature sensor, capacitive touch, ULP UART and many others! • **Package Flexibility:** Module, WLCSP, UFBGA, VQFN, UQFN
Thread Border Router	• **Complete Thread Border Router Solution**: **STM32MP1** for traffic processing & **STM32WB35** as Radio Co-Processor for Thread connectivity • Ideal to **setup your own matter ecosystem**, **reduce dependency on existing controllers** and **provide device to cloud connectivity** • **Pre-certified solution**
Matter to non-Matter Bridge	• **Complete Matter to non-Matter Solution**: **STM32H7** for Matter Bridged node & your preferred wireless solution for the non-Matter network (**STM32WL, STM32WB, STM32WBA**…) • **Dedicated team to support** your Matter to non-Matter bridge project: define virtual device list, certification strategy, integration….

1.1

Solutions Matter STMicroelectronics

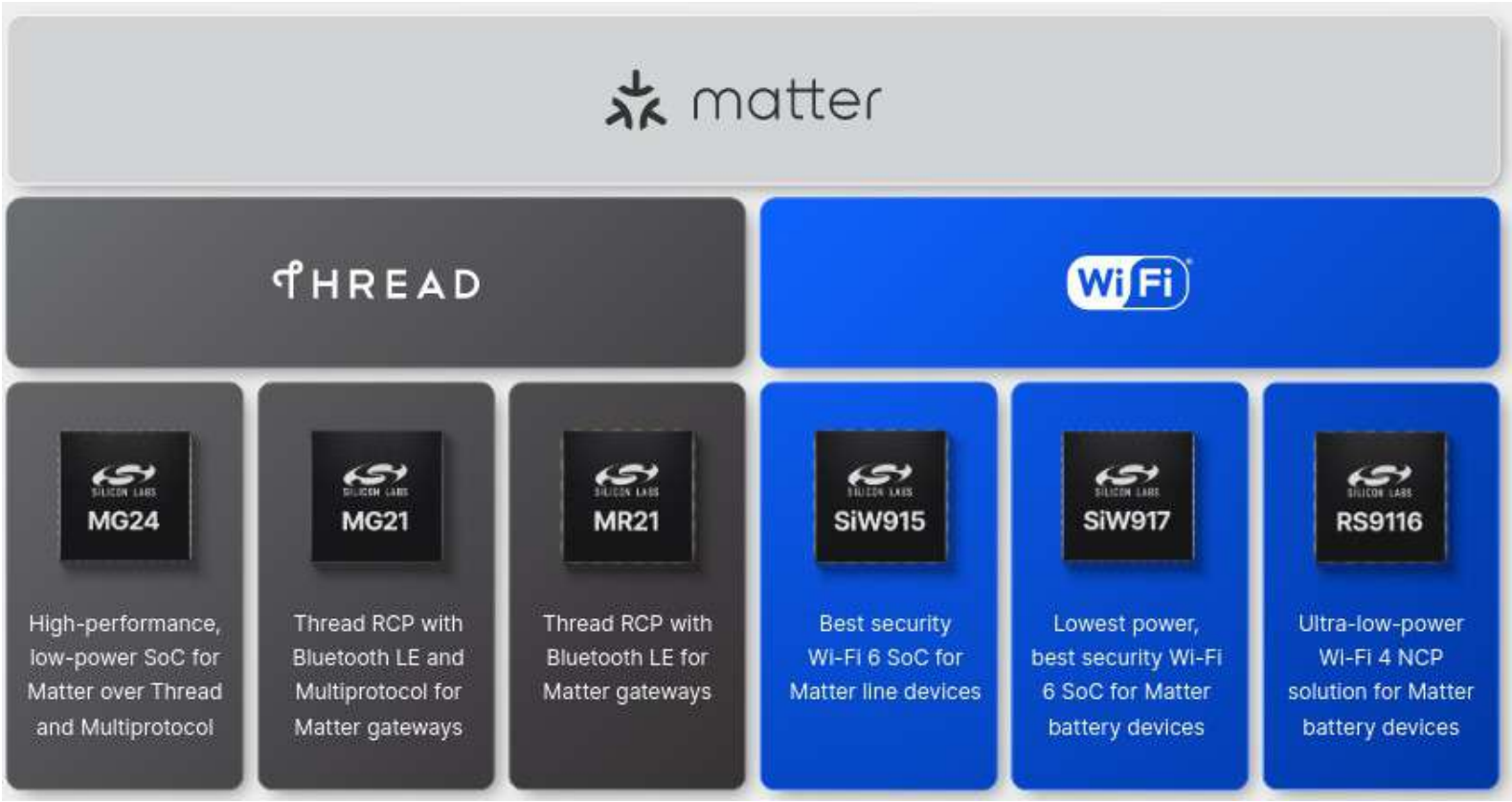
Device type	Connectivity	Wireless MCUs in standalone	Hosted architecture (MCU/MPU + radio companion)
Matter Gateway (Border router)	Thread RCP Ethernet / WiFi		RCP model STM32H7 + STM32WB STM32MP1 + STM32WB
Matter over Thread End Device	Concurrent Dynamic Thread - Bluetooth LE	STM32WB*	NCP model STM32H7 + STM32WB STM32U5 + STM32WB STM32MP1 + STM32WB
Matter bridge to no-Matter technologies	Any**		NCP STM32H7 + STM32WB STM32MP1 + STM32WB STM32H7 + STM32WL STM32MP1 + STM32WL

*OTA on external Flash
**data model bridging Matter to any other technology

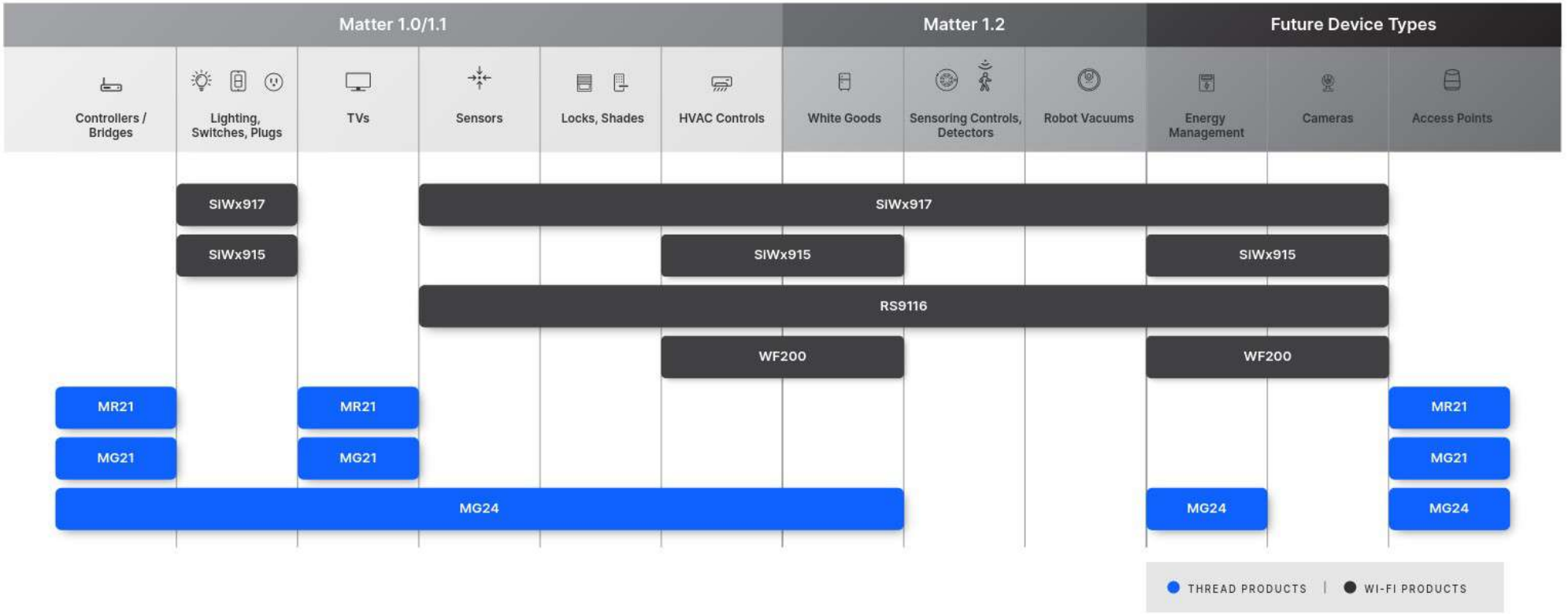
Solutions Matter Nordic Semiconductor

- nRF7002
- nRF5340
- nRF52840
- *nRF54H20, nRF54L15* à venir

Solutions Matter Silicon Labs



Solutions Matter Silicon Labs



Solutions Matter Texas Instrument

Bluetooth®

THREAD

	CC2674x10	CC2652x7	CC2652x
Description	SimpleLink Arm® Cortex®-M33 TrustZone® multiprotocol 2.4-GHz wireless MCU	SimpleLink Arm Cortex-M4F multiprotocol 2.4-GHz wireless MCU with 704-kB Flash	SimpleLink 32-bit Arm Cortex-M4F multiprotocol 2.4 GHz wireless MCU with 352-kB Flash
Protocols	Bluetooth 5.3 Low Energy Thread Zigbee 3.0 Matter	Bluetooth 5.2 Low Energy Thread Zigbee 3.0 Matter	Bluetooth 5.2 Low Energy Thread Zigbee 3.0
Matter use-case	Matter end device Matter gateway (Thread RCP)	Matter end device Matter gateway (Thread RCP)	Matter gateway (Thread RCP)

Matter 1.0

	CC3230SF	CC3235SF	CC3235MODSF	CC3235MODASF
Description	SimpleLink 32-bit Arm Cortex-M4 2.4-GHz Wi-Fi CERTIFIED wireless MCU	SimpleLink 32-bit Arm Cortex-M4 2.4 & 5-GHz Wi-Fi CERTIFIED wireless MCU	SimpleLink 32-bit Arm Cortex-M4 2.4 & 5-GHz Wi-Fi CERTIFIED™ module	SimpleLink 32-bit Arm Cortex-M4 2.4 & 5-GHz Wi-Fi CERTIFIED module with integrated antenna
Protocols	Wi-Fi Matter	Wi-Fi Matter	Wi-Fi Matter	Wi-Fi Matter
Device feature	Matter SoC 2.4-GHz Wi-Fi SoC	Matter SoC 2.4-GHz & 5-GHz Wi-Fi SoC	Matter SoC module 2.4-GHz & 5-GHz Wi-Fi SoC module	Matter SoC module 2.4-GHz & 5-GHz Wi-Fi SoC module Integrated antenna

Wi Fi

Solutions Matter Espressif

Propose solutions 0 code

Matter 1.2 (11/23)

Environnement de développement

Software Development Kit (SDK)

Ensemble des outils existant :

- SDK du fabriquant de microcontrôleur retenu
- Interface de développement intégrée (IDE)
- API/stack Matter
  - Projet chip (Zigbee): *https://github.com/project-chip/connectedhomeip*
  - Inclus dans le sdk avec le compilateur
  - Inclus dans l'exemple

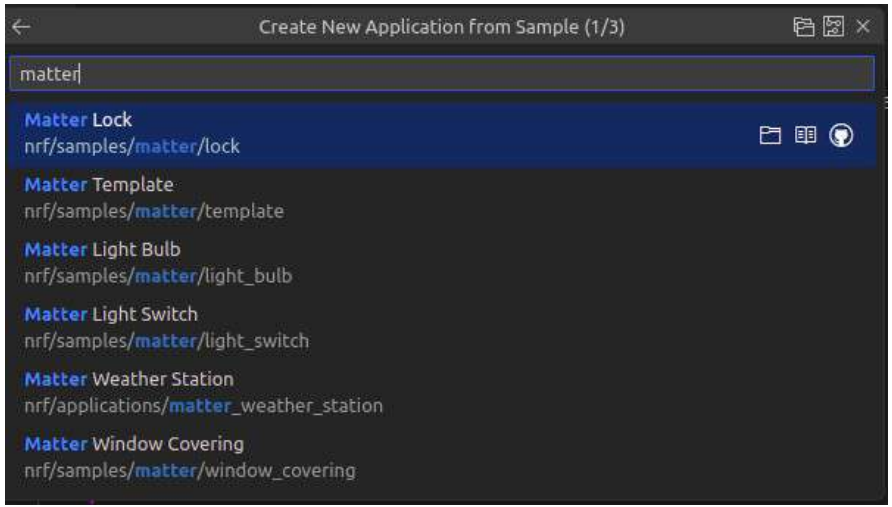
Software Development Kit (SDK)

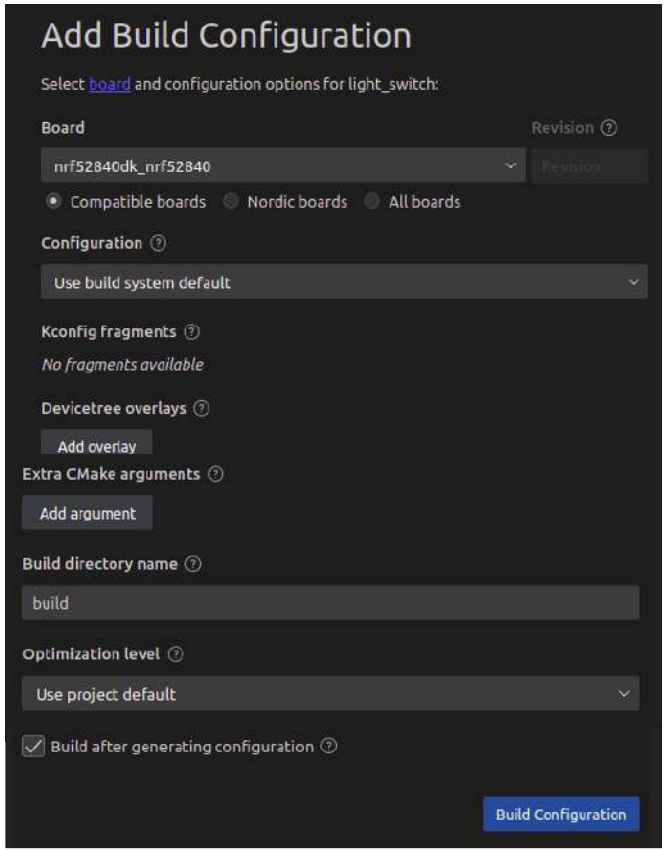
- Exemple : Nordic Semiconductor

nRF Connect SDK version	Matter specification version	Matter SDK version
v2.5.99 (latest)	1.2.0	1.2.0.1
v2.5.0	1.1.0	1.1.0.1
v2.4.2		
v2.4.1		
v2.4.0		
v2.3.0	1.0.0	1.0.0.2
v2.2.0		1.0.0.0
v2.1.4		1.0.0.0
v2.1.3		1.0.0.0
v2.1.2		1.0.0.0

Software Development Kit (SDK)

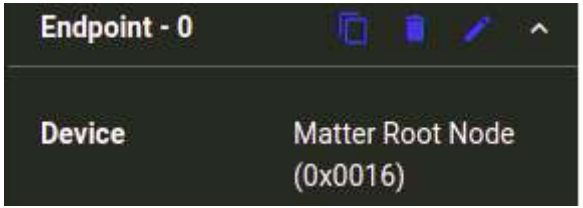
- Nordic Semiconductor Visual studio + nrf connect sdk v2.2





Software Development Kit (SDK)

- Nordic Semiconductor
  - Exemple de gestion des valeurs d'attributs de cluster sur un device matter




```
chip::app::Clusters::PowerSource::Attributes::BatVoltage::Set(/* endpoint ID */ 0, /* bat voltage */ batteryVoltage);
chip::app::Clusters::PowerSource::Attributes::BatPercentRemaining::Set(/* endpoint ID */ 0, /* bat percent */ batteryLevelInPercent);
```

```
chip::app::Clusters::BooleanState::Attributes::StateValue::Set(/* endpoint ID */ 1, /* boolean state */ false);
chip::app::Clusters::BooleanState::Attributes::StateValue::Set(/* endpoint ID */ 2, /* boolean state */ true);
```

Software QR code

- Génération du QR code
  - Utilisation de chip-tool

```
$ ./chip-tool-debug payload generate-qrcode --vendor-id 0xFFF1 --product-id 0x8000  --rendezvous 2 --discriminator 0x0F04 --setup-pin-code 25147864
[1700654700.221505][51803:51803] CHIP:TOO: QR Code: MT:Y.K90-C714-U6O37K00
```

--vendor-id : 0xFFF1 = utilisé pour tous les exemples
--product-id : 0x8000 = produit non répertorié
--rendezvous 2 = commissioning via ble

https://github.com/project-chip/connectedhomeip/blob/482e6fd03196a6de45465a90003947ef4b86e0b1/docs/examples/discussion/PID_allocation_for_example_apps.md

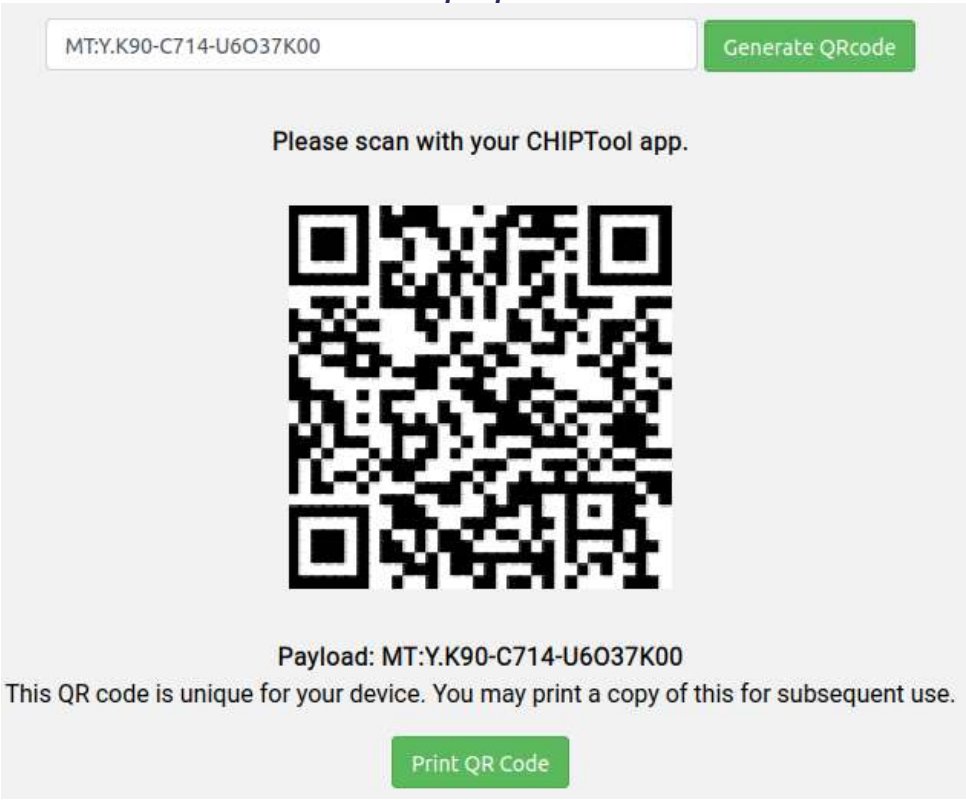
App	PID
All Clusters	0x8001
Bridge	0x8002
Door Lock	0x8003
Light switch	0x8004
Lighting	0x8005
Lock	0x8006

OTA provider	0x8007
OTA requestor	0x8008
Persistent Storage	0x8009
Pigweed	0x800B
Pump	0x800A
Pump Controller	0x8011

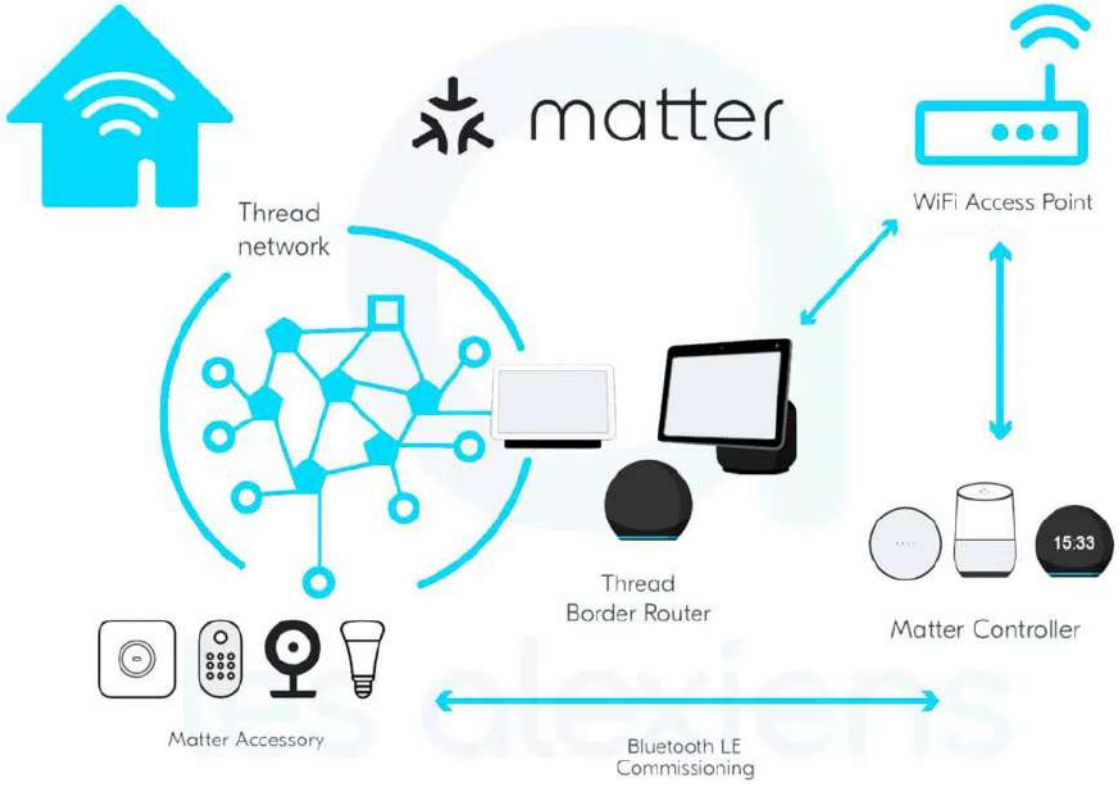
Shell	0x8012
Temperature measurement	0x800D
Thermostat	0x800E
TV	0x800F
Window	0x8010

Software QR code

- Génération du QR code : Utilisation du projet chip connectedhomeip
  https://project-chip.github.io/connectedhomeip/qrcode.html

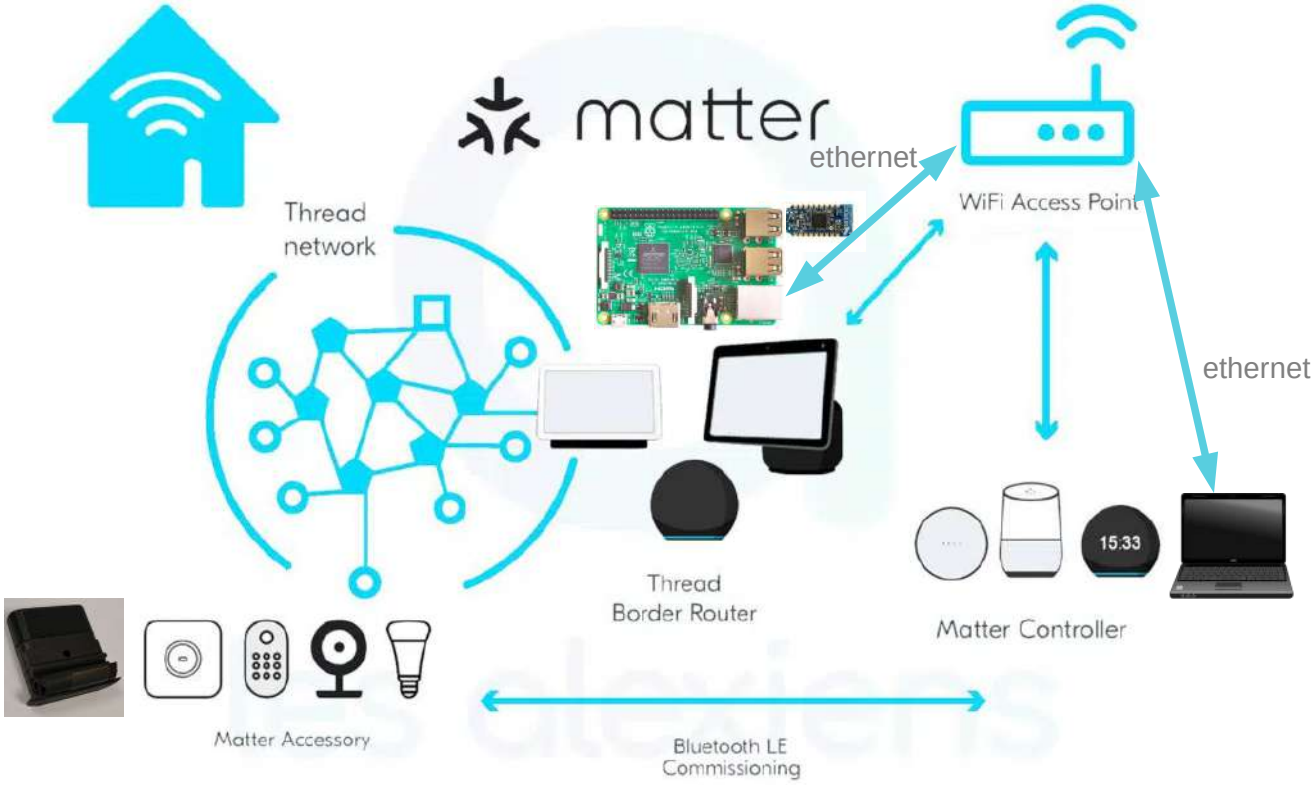


Ecosystème Matter



Src :https://www.lesalexiens.fr/

Ecosystème Matter

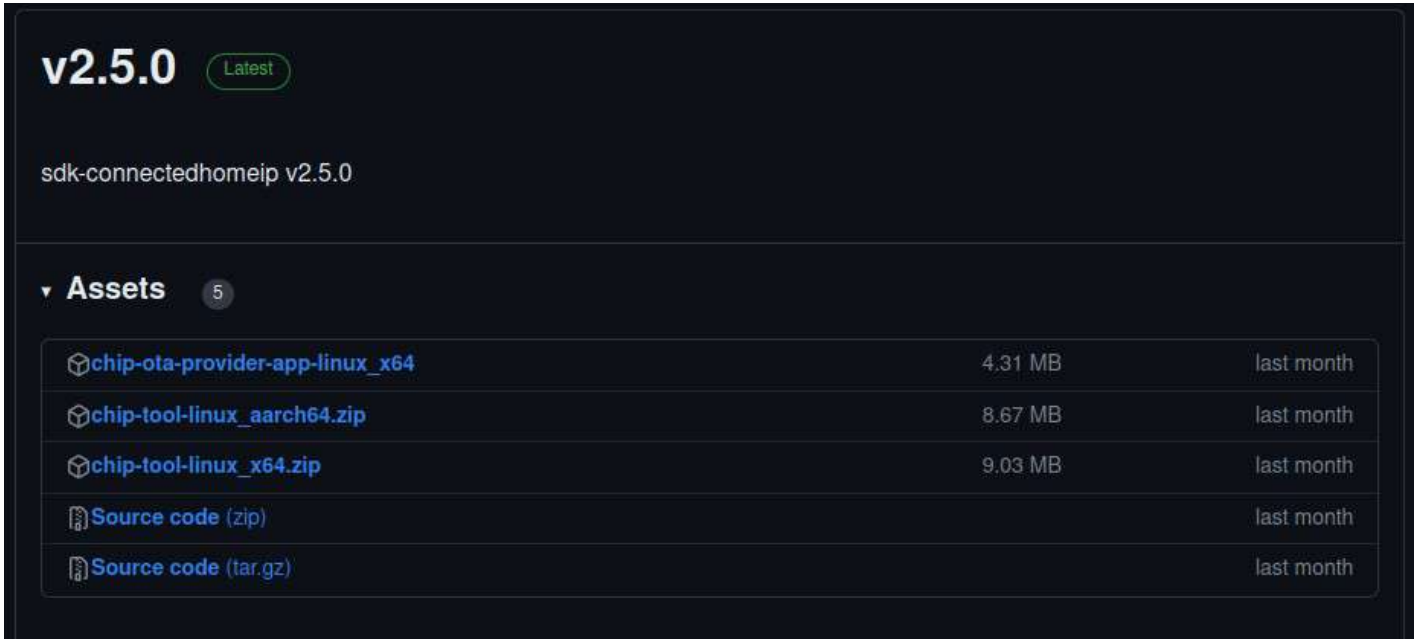


Src :https://www.lesalexiens.fr/

Software Chip-tool

- Outil de test : chip-tool
  https://github.com/nrfconnect/sdk-connectedhomeip/releases version identique à la version, du sdk utilisé

Contrôleur Matter



Software Chip-tool

- chip-tool / chip-tool-debug / chip-tool-release :
  - implémentation d'un contrôleur Matter
  - permet le commissioning (enregistrement sur le réseau)
  - fournit toutes les commandes d'interaction avec un end point matter

Contrôleur Matter

```
Usage:
  ./chip-tool-debug cluster_name command_name [param1 param2 ...]

  +-------------------------------------------------------------------------------+
  | Clusters:                                                                     |
  +-------------------------------------------------------------------------------+
  | * accesscontrol                                                               |
  | * accountlogin                                                                |
  | * actions                                                                     |
  | * administratorcommissioning                                                 |
  | * any                                                                         |
  | * applicationbasic                                                            |
  | * applicationlauncher                                                         |
  | * audiooutput                                                                 |
  | * ballastconfiguration                                                        |
  | * barriercontrol                                                              |
  | * basic                                                                       |
  | * binaryinputbasic                                                            |
  | * binding                                                                     |
  | * booleanstate                                                                |
  | * bridgeddevicebasic                                                          |
  | * channel                                                                     |
  | * colorcontrol                                                                |
  | * contentlauncher                                                             |
  | * descriptor                                                                  |
  | * diagnosticlogs                                                              |
  | * discover                                                                    |
```

Software Chip-tool

- chip-tool / chip-tool-debug / chip-tool-release :
  - Commissioning :

Contrôleur Matter

./chip-tool-debug pairing ble-thread 1
hex:0e080000000000100000030000d35060004001fffe0020865cd2860
b51779f30708fd52b0791db02673051021aaca7fd95b9623db32b6ae818d43
6e030f4f70656e5468726561642d3038313201020812041019490af9429a76
a91754e60392a6f8110c0402a0f7f8 25147804 3804

Support de com (ble-wi-fi, ble-thread, ethernet..)

node id

setup-pin-code

discriminator

Clef (operationaldataset fourni par le Thread border router)

Border Router Thread

- Raspberry pi 3B
  configuration : *https://openthread.io/codelabs/openthread-border-router?hl=fr#1*

- nrf52840 dongle usb
  https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/tools.html#ug-thread-tools-tbr
  firmware : co-processor

  - A chaque démarrage du border router il faut récupérer la valeur de operationaldataset qui va servir au commissioning

    $ sudo ot-ctl dataset active -x
    0e080000000000010000000300000d35060004001fffe0020865cd2860b517
    79f30708fd52b0791db02673051021aaca7fd95b9623db32b6ae818d436e03
    0f4f70656e5468726561642d303831320102081204101949 0af9429a76a917
    54e60392a6f8110c0402a0f7f8

Software Chip-tool

- chip-tool / chip-tool-debug / chip-tool-release :
  - Interrogation :

Contrôleur Matter

./chip-tool-debug doorlock read lock-state 3 1
./chip-tool-debug levelcontrol read current-level 3 2
./chip-tool-debug booleanstate read state-value 2 1
./chip-tool-debug booleanstate read state-value 2 2
./chip-tool-debug powersource read bat-voltage 2 0
./chip-tool-debug powersource read bat-percent-remaining 22 0
./chip-tool-debug generaldiagnostics read active-hardware-faults 22 0
Periodic : ./chip-tool-debug powersource subscribe bat-voltage 10 20 2 2

  - *Commande*

./chip-tool-debug doorlock unlock-door 3 1 --timedInteractionTimeoutMs 3000
./chip-tool-debug doorlock lock-door 3 1 --timedInteractionTimeoutMs 3000

Matter

Smart Building Gasket

Maison verrouillée

Maison non verrouillée

SBG

SBG

Fenêtre PVC

Baie ALU

Fenêtre Racine MGL

ARTIBAT

LE SALON DE LA CONSTRUCTION

18 | 19 | 20 OCTOBRE 2023

S=100m²

BAR

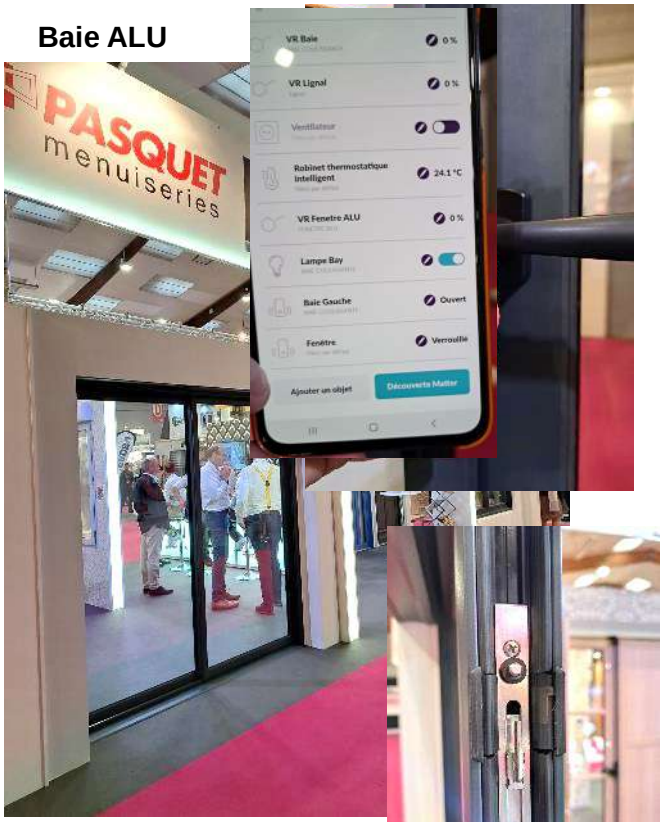
Lignal CVI

SBG

Fenêtre ALU

Smart Building Gasket

Baie ALU



ouvert/fermé/verrouillé/invalide

ARTIBAT
LE SALON DE LA CONSTRUCTION
18 | 19 | 20 OCTOBRE 2023

Fenêtre PVC

ouvert/verrouillé/invalide

Fenêtre ALU

ouvert/oscillo-battant/verrouillé/invalide

Merci / Questions