

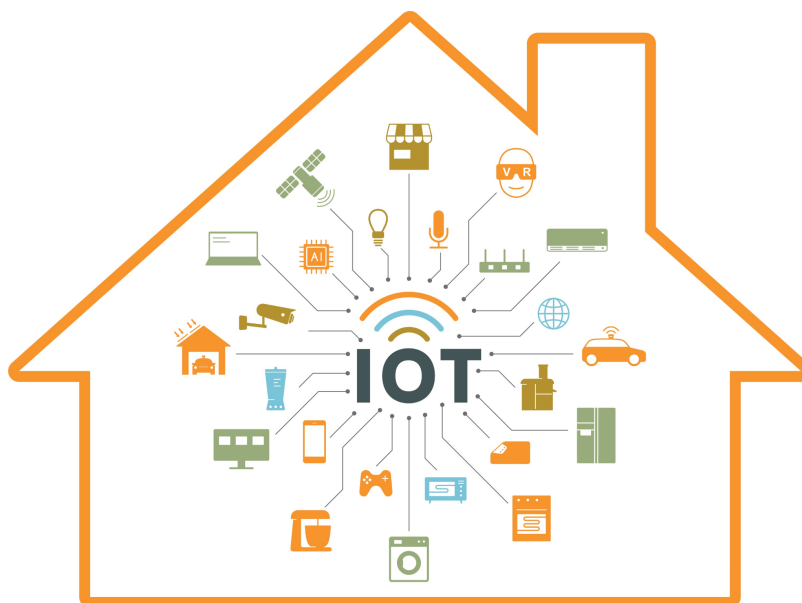
Master of Science HES-SO in Engineering

IoT

2021 – 2022

Internet of Things

Lab 1 Smart Building



**Anuraag POTHULA**anuraag.pothula@master.hes-so.ch



Google Cloud

Gaëtan PASSERIgaetan.passeri@master.hes-so.ch

**Glenn MULLER ARDAYA**

glenn.mullerardaya@master.hes-so.ch

Introduction

Dans les sections suivantes, nous allons décrire et expliquer de l'architecture qui a été imaginée pour le labo 1 Smart Building utilisant les trois plateformes cloud: AWS, Google Cloud et Microsoft Azure.

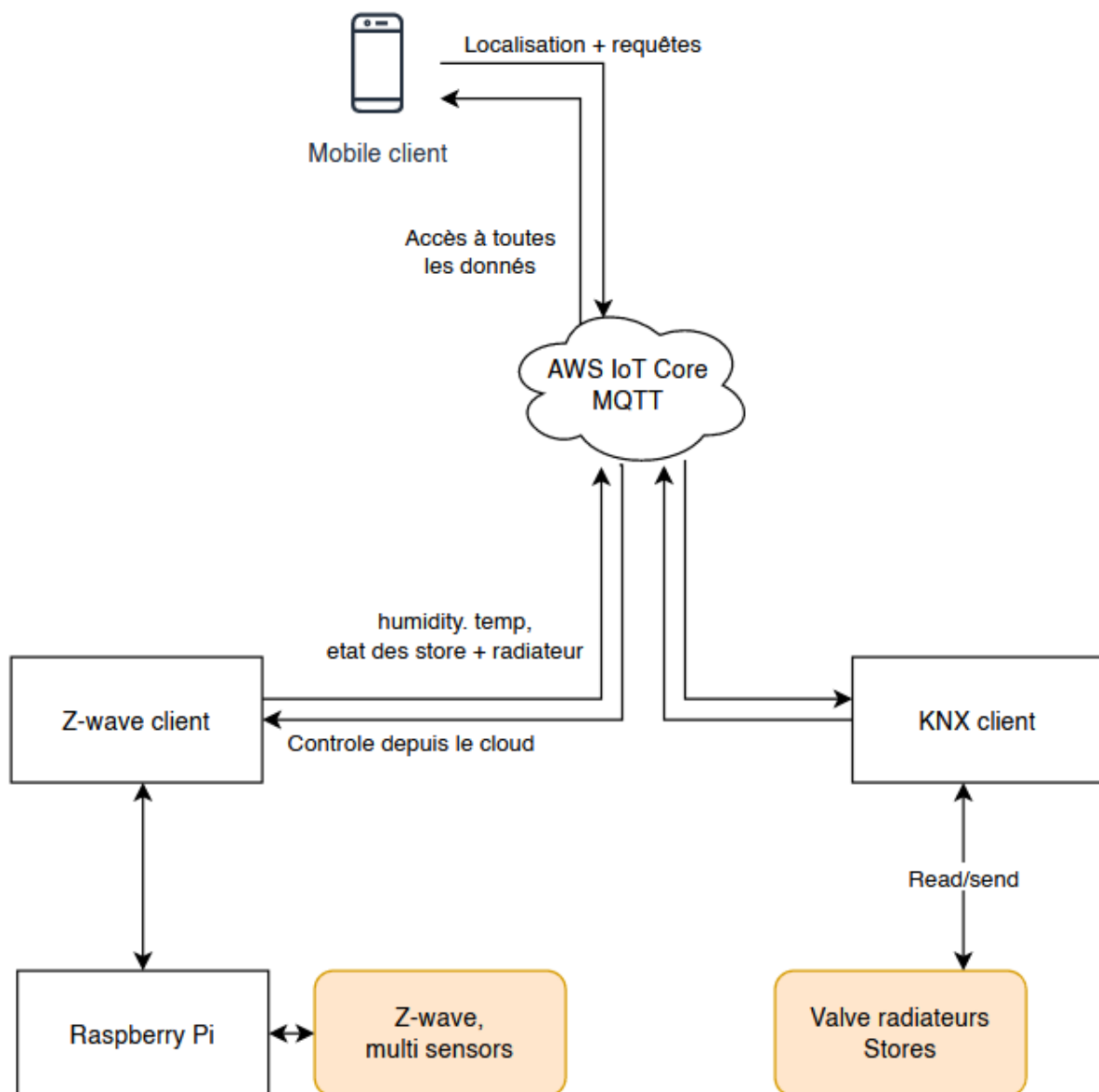
Ensuite pour comparer les différents services, nous nous basons sur des critères de comparaison tels que, leur compatibilité et gestion avec les devices de lumière, radiateur, store, etc., les fonctions possibles dans le cloud, les protocoles de communication utilisés, et enfin les coûts de chaque plateforme.

AMAZON WEB SERVICES

Sur l'illustration ci-dessous nous pouvons voir comment l'architecture imaginé pour la mise en place de notre application. Cette architecture utilise AWS IOT Core, basé sur l'envoi de message MQTT.

Avec AWS IOT Core, des devices virtuels sont créés puis liés à une clé publique et privée. Puis ces clés peuvent être utilisées pour communiquer avec un MQTT Broker. Dans notre architecture, nous avons un device, le Raspberry Pi, et un client KNX.

Pour l'utilisation du MQTT nous avons pour l'instant effectué des tests publish/subscribe avec broker publique et gratuit *mqtt.eclipseprojects.io*. À l'avenir il serait bien d'implémenter cela directement avec AWS via leur service Elastic Cloud Computing (EC2). Ceci permettra de faire le pont entre le client KNX et Z-Wave. Cette couche permet également d'automatiser des tâches telles que le mouvement des stores en fonction de l'humidité/lumière ou en fonction de la présence d'une personne dans la pièce. Il permet également de gérer le radiateur en fonction de la présence et de la température.



KNX client

Le client KNX devra récupérer et garder les valeurs de chaque pièce. Quand une de ces valeurs change, le client l'envoie à la couche de support (AWS EC2) via des messages MQTT. Pendant qu'il continue à surveiller la variation des valeurs, il reste prêt à recevoir des commandes pour gérer les stores et les valves. Étant subscribed à des topics MQTT , il s'attend à recevoir des commandes automatiques ou manuelles.

Le client KNX peut être placé sur un PC, un serveur ou sur une instance EC2. Dès que le client peut être connecté à l'internet pour accéder au point d'accès IoT Core, il fonctionne

Ce client est indépendant du reste de l'infrastructure.

ZWave client

Le client ZWave fonctionne de manière similaire au client KNX. La principale différence est qu'un client est nécessaire par pièce car il doit être placé sur chaque Raspberry PI utilisé. Chaque client récupère toutes les valeurs du capteur et les envoie à la couche de support via des messages MQTT lorsqu'une valeur change.

Ce client attend des commandes pour allumer/éteindre la lumière liée au Raspberry PI.

PRIX DE LA PLATEFORME

AWS possède un calculateur de prix dépendant qui varie selon différents facteurs tels que les services que l'on souhaite avoir, le nombre de devices connectés, le temps des connexions, le nombre de messages etc. Avec 10 devices connectés 24h/24 et 10k messages traités par device par mois, on arrive à 0.16\$/mois. On imagine bien qu'à plus grande échelle, cela peut vite devenir coûteux.

GOOGLE CLOUD

Google Cloud met à disposition le service IOT core. Celui-ci permet de gérer des infrastructures IOT. Il prend en charge MQTT. Pour ce faire, il faut utiliser soit MQTT bridge, soit HTTP bridge.

Dans ce travail, nous utiliserons HTTP Bridge, car nous pensons que les requêtes seront plus faciles à prendre en main.

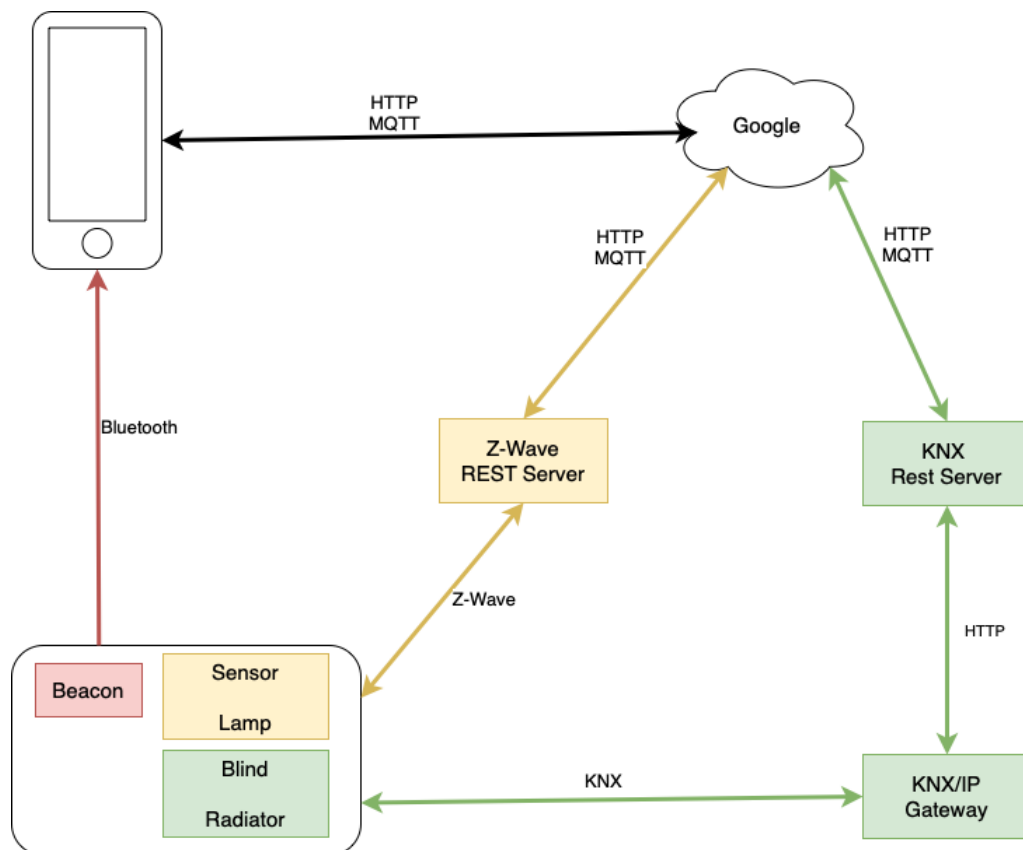
Pour ce qui est de l'architecture globale, la figure ci-dessus nous montre comment les différents dispositifs communiquent entre eux.

Tout d'abord l'application communique avec les beacons via Bluetooth, ensuite elle se connecte en tant que client au cloud Google en utilisant MQTT. Le Cloud communique avec les deux serveurs REST également par MQTT.

Le serveur KNX communique avec la Gateway par HTTP. Cette dernière envoie et reçoit des informations avec les radiateurs et les stores par KNX.

Le serveur Z-Wave communique directement par Z-Wave avec les capteurs et les lampes.

En utilisant cette infrastructure, notre application Smart Building met en œuvre un écosystème formé par :



FONCTIONS GOOGLE

Les fonctions Google sont déclenchées par les messages publiés par les dispositifs et sont utilisées pour appliquer la logique de commande des actionneurs (automatisés et contrôlés par l'utilisateur) et pour enregistrer les données de statistiques dans la base de données. L'avantage est qu'ils sont intégrés dans la plateforme cloud et interfacés avec le système de messages Pub/Sub et l'accès à la base de données.

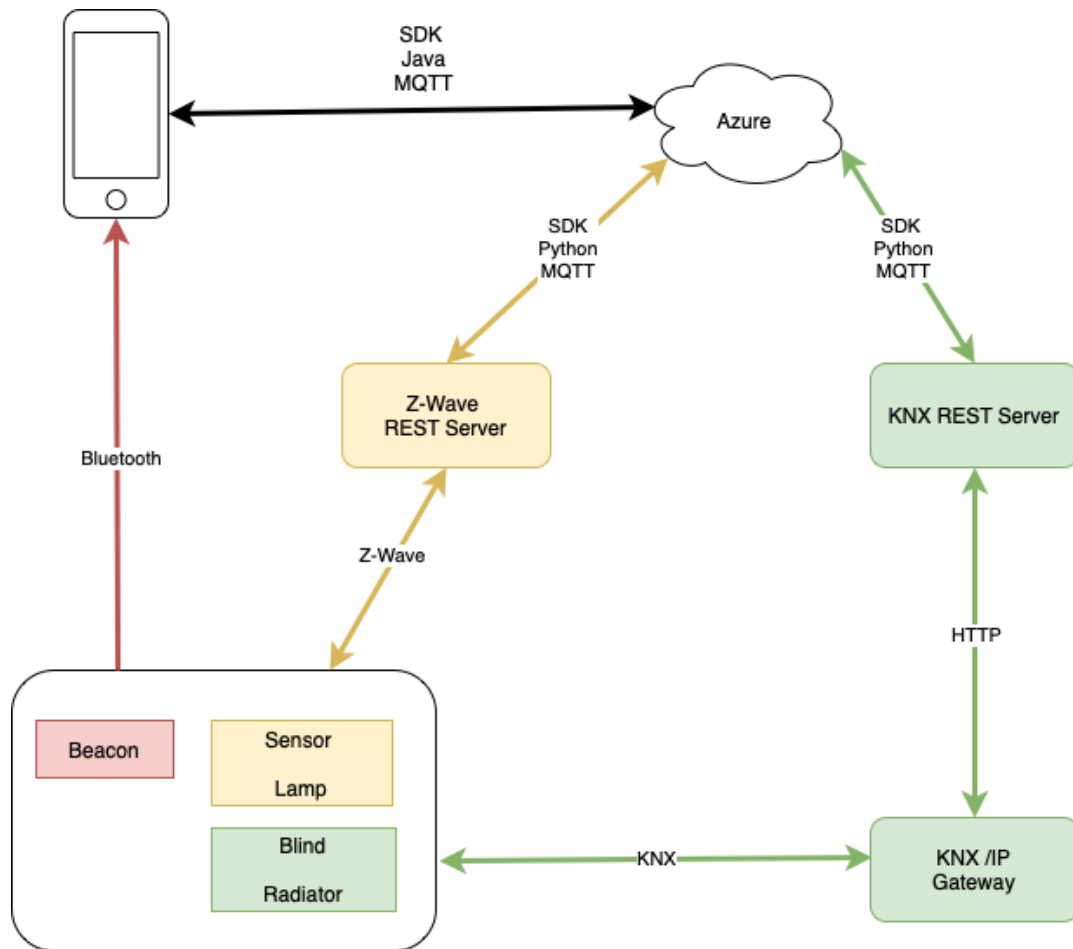
DATA STORAGE AND PERSISTENCE

Google Cloud Platform offre un service de base de données Firebase pour une mise en œuvre facile via l'authentification Firebase pour communiquer avec les applications mobiles.

PRIX DE LA PLATEFORME

L'utilisation des services Google nécessite un compte de facturation valide, dont la création est gratuite et qui nous accorde 300\$ de crédit plus l'accès à plus d'une vingtaine de services. Chaque service est facturé en fonction des données utilisées/stockées et lues/écrites. Les prix proposés incluent des quotas gratuits pour certaines limites d'utilisation, par exemple pour le service de IoT core, jusqu'à un volume de données de 250 MB/mois et pour Firebase jusqu'à 50kB lectures/j, 20kB écritures/j et 20kB suppressions/j.

MICROSOFT AZURE



La solution IOT de Azure (IOT Hub) permet de se connecter avec un client MQTT via un SDK disponible sous plusieurs langages. Dans le cadre de notre application, les SDK python et Java sont ceux qui nous intéressent. Ils permettent une connexion facile vers le cloud.

Le SDK Java permet de connecter l'application mobile vers le cloud, alors que le SDK python permet de connecter les deux serveurs REST sur le cloud.

L'image ci-dessus illustre les différentes connexions entre les éléments. L'application reçoit la distance entre l'appareil mobile et les plus proches Beacon via Bluetooth et envoie ou reçoit des données sur le cloud via MQTT.

Le cloud, quant à lui, reçoit et envoie des données vers l'application mobile, les serveurs REST KNX et Z-Wave.

Les deux serveurs communiquent avec les objets qui leur sont attribués. HTTP pour KNX et Z-Wave pour le serveur Z-Wave.

COMPARAISON DES 3 OPTIONS

Le tableau suivant reprend les différentes approches testées, classées en fonction des critères de comparaison définis dans la première section de ce rapport.

Critère	AWS	Google Cloud	Azure
Gestion de device	Enregistrer des devices qui utilisent le protocole MQTT à l'aide de scripts fournis par AWS.	Les devices capables de communiquer en MQTT peuvent subscribe/publish aux topics implémentés.	Ajouter et gérer des devices (avec ou sans MQTT) via le portail Azure.
Fonctions Cloud (Faas)	Oui, AWS Lambda offre la possibilité de travailler avec des fonctions dans de nombreux langages (C#, python, Node.js...).	Google functions fournit également des fonctions cloud. Ceux-ci peuvent être déclenchés par des communications MQTT.	Oui, Azure permet aussi d'implémenter des fonctions dans plusieurs langages.
Processing + database	AWS Lambda gère les calculs mais il faut une autre solution de stockage	Les fonctions sont gérées par Google Cloud Function ensuite les données peuvent être stockées dans Firebase par exemple	Le processing se fait dans le Cloud Azure functions puis il y a la possibilité d'utiliser MySQL intégré dans Azure
Protocol de communication	Device communication sont possible via MQTT et HTTPS	MQTT bridge + HTTP bridge	MQTT, AMQP et HTTPS pour le device-side communication
Pricing	Cela varie énormément selon notre usage. Pas de version gratuite. Mais c'est peu cher quand on utilise des ressources limitées.	Un plan gratuit est possible avec 300\$ de crédit et une limite sur la quantité de données. Ensuite cela devient payant.	Un service gratuit IoT Hub avec un nombre de device limité et des messages par jour limité est possible. Ensuite cela devient payant de 25\$ - 2500\$ selon le standard tier choisi