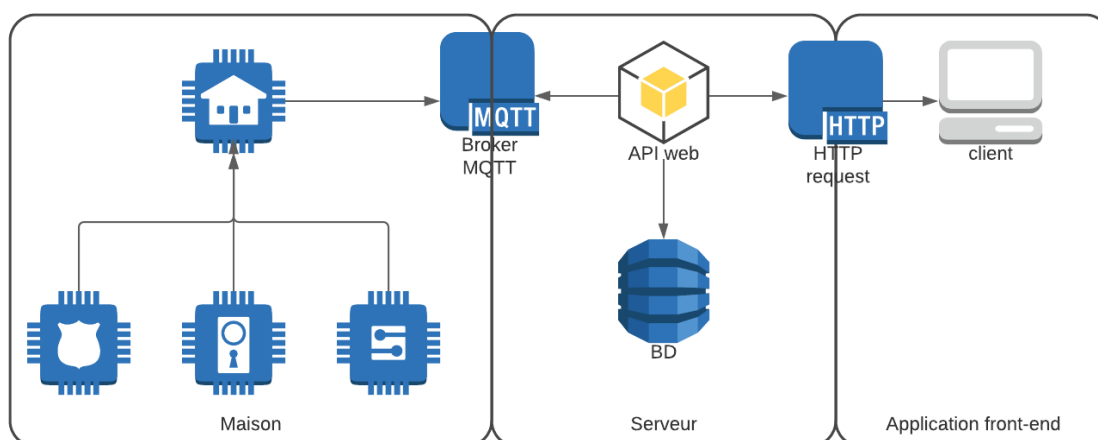


PROJET

C64 – SYSTÈME DE SÉCURITÉ INTELLIGENT

DESCRIPTION DU PROJET

La compagnie Antoine 2000 vous a engagé pour développer un système de sécurité IoT. Ce dernier sera composé de 4 éléments centraux.



1. L'APPAREIL CONNECTÉ – 1-* ÉQUIPES DE 2-3 PERSONNES

Un appareil connecté peut être n'importe quoi ! Il peut aller du simple bouton panique à la caméra qui reconnaît les visages et qui détecte si la personne est autorisée à être à cet endroit ou non. Dans le cadre du projet, l'appareil connecté enverra des informations au broker MQTT sous le format JSON suivant :

- id
 - Id unique de l'appareil
- date
 - date et heure à laquelle la mesure a été prise. Respectez le format ISO 8601. Voici un exemple : **1977-04-22T01:00:00-05:00**
- type
 - Le type de la mesure. Température? Détection d'un visage? Bouton?
- valeur
 - La valeur mesurée. Elle peut être binaire ou complexe, voir même une image
- alerte
 - 0 pour tout est ok
 - 1 pour alerte
- messageAlerte
 - Le message d'alerte. Ce champ est vide s'il n'y a pas d'alerte.

Les messages seront transmis sur le broker MQTT chaque minute en respectant la route suivante :

C64/Projet/EquipeX/capteur/...

2. BROKER MQTT – 0-1 ÉQUIPE DE 2 PERSONNES

Le broker MQTT fera exactement ce qu'on aura vu en classe. Il aura la charge de distribuer les messages reçus aux souscripteurs. Ce sera un élément central au projet. En attendant que ce dernier soit en ligne, l'équipe se servira du broker gratuit de HiveMQ.

3. API WEB – 1 ÉQUIPE DE 3-5 PERSONNES

L'api Web aura comme rôle d'entreposer les données publiées par le broker MQTT ainsi que de communiquer avec le page web via des routes d'API.

L'API devra donc **sauvegarder** les données reçues par les appareils et, au besoin, les transmettre aux équipes via des routes. Vous devez donc implémenter les routes suivantes :

- GET /sensors/ {limit, offset}
 - retourne la liste des dernières mesures de tous les senseurs confondus en commençant par la plus récente.
- GET /sensor/id {limit, offset}
 - retourne la liste des dernières mesures du senseur au id correspondant commençant par la plus récente.
- GET /alerts {limit, offset}
 - retourne la liste des dernières alertes avec leurs messages en commençant par la plus récente
- GET /sensor/id/alerts {limit, offset}
 - retourne la liste des dernières alertes du capteur avec leurs messages en commençant par la plus récente

Le serveur sera sécurisé par un système d'authentification avec jeton. Pour se connecter, les utilisateurs devront appeler la route

- POST /authenticate {username, password}
 - retourne {token}

qui permettra de se connecter. Les utilisateurs seront créés à même la base de données par l'équipe API, pas besoin d'ajouter une logique de création de comptes.

5. CLIENT WEB – 1-* ÉQUIPE DE 2-4 PERSONNES

Le client web servira à récolter les données fournies par les routes et les afficher aux usagers.

Un usager pourra donc accéder aux mesures prises par ses différents appareils ainsi que **regarder des beaux graphiques** qui montrent l'évolution des mesures prises par les appareils.

Il pourra aussi regarder s'il y a des alertes qui sont envoyées par ses appareils.

Le client web n'est pas nécessairement une page web, mais peut aussi être une application mobile.

LA COMMUNICATION ENTRE TOUS LES NŒUDS

Comme vous pouvez voir, le projet ne vous impose pas de langage ou de méthode. La structure des données est imposée ainsi que le chemin vers les routes (api, broker) mais c'est tout. Vous devez donc vous assurer de bien communiquer avec les équipes afin de standardiser les différents messages.

De plus, comme c'est la première fois que le projet est donné, il est possible qu'il y ait des oublis de ma part. S'il manque de cohérence dans ce qui est proposé ou qu'une information pertinente ne s'y trouve pas, vous devez me faire signe.

MISE EN LIGNE DU PROJET

Les équipes d'API, Broker MQTT et Client Web devront mettre leurs projets en ligne en l'hébergeant sur un serveur. Plusieurs sites offrent ces services (vous avez déjà vu Digital Ocean normalement). Il existe aussi Heroku qui permet de mettre en ligne un site web gratuitement (avec certaines limitations). Ne sous-estimez pas cette tâche, elle peut être complexe et fastidieuse, surtout les premières fois!

SUIVI DU PROJET

Le projet sera suivi par votre enseignant de deux façons.

1. Avec votre participation aux daily, qui auront lieu au début du cours à 8h et aux cours. Si vous êtes absents ou que vous ne participez pas à la totalité du cours ou du daily, vous perdrez des points attribués pour la journée.
2. En regardant la progression du projet dans la section projet sur github. En déplaçant les différentes tâches dans les colonnes appropriées, je pourrai garder un œil sur la progression du projet. Vous pourrez aussi créer des tâches dans la colonne bogues pour les autres équipes quand quelque chose ne semble pas fonctionner de leur côté.

Lors du premier cours consacré au projet, vous aurez à créer les différentes tâches que votre équipe aura à réaliser.

<https://github.com/orgs/CVM-C64-H2021/projects/1>

Afin d'éviter la confusion, toutes les tâches de votre équipe auront comme première le nom de votre équipe.

EX :

L'ÉQUIPE OVER 9000

En tant que serveur, je veux recevoir les messages MQTT

Vous devrez aussi utiliser les répertoires github présent à cette adresse :

<https://github.com/CVM-C64-H2021>

Si votre équipe vient à terminer son travail avant la fin de la session, vous pourrez ajouter des fonctionnalités à votre projet ou même aider d'autres équipes, en vous assurant d'avoir l'autorisation de votre enseignant.

REMISE DU PROJET **14 MAI**

Cette journée sera consacrée à la présentation individuelle de votre travail. Vous recevrez donc les points pour la correction fonctionnelle ainsi que vos points pour la participation au code source sur github. Ce dernier élément évaluera votre apport au code source ainsi que vos messages commits.

Assurez-vous donc d'être clairs dans vos commits!

PRÉSENTATION DU PROJET **14 MAI**

Le 18 mai, vous aurez à présenter ce que vous avez fait aux autres étudiants de la classe.

Chacun d'entre vous aura entre 4 et 6 minutes de parole. Vous ne pourrez pas dépasser le 6 minutes alloué.

La présentation du projet aura pour but de :

- Présenter ce que vous avez fait
- Montrer les technologies utilisées
- Relever les différents défis que vous aurez relevés
- Proposer des améliorations à ce que vous avez fait
- Évaluation de la communication avec les autres équipes
 - Ici, le but n'est pas de créer des situations malaisantes, mais bien de donner une note aux autres équipes sur la communication en justifiant celle-ci. Les autres équipes pourront ainsi être au courant de leurs forces et faiblesses.

Le 18 mai servira de cours tampon. Dans le cas où tout est réellement terminé, un de mes amis vous fera une présentation sur son travail qui est en lien avec le cours.

ATTRIBUTION DES POINTS

Voici la grille de correction.

En équipe	
Création des tâches dans github	5pts
Communication interéquipe (évalué par les autres)	5pts
Correction fonctionnelle	15pts
Total équipe	25pts
Individuel	
Présence et participations aux sprints	15pts
Présentation Orale	5pts
Participation au code source sur github	5pts
Total individuel	25pts

La note d'équipe d'un étudiant ne pourra pas être plus élevée que sa note individuelle.

UN DERNIER POINT

Même s'il sera difficile de plagier, vous ne devez en aucun cas laisser quelqu'un d'autre faire le projet. Je serai présent pour vous aider si vous avez de la difficulté à avancer dans le projet. Je pourrai même vous aider à régler les bogues dans votre code si vous en avez. Le but ici est d'avoir du plaisir et d'apprendre.

Dans le cas où un coéquipier ne travaille pas suffisamment ou s'il n'adopte pas une attitude professionnelle, faites-m'en part directement. Si ce dernier n'est pas en mesure de démontrer qu'il peut améliorer son apport à l'équipe, sa note sera de 55%. Prenez note que tous les membres de l'équipe n'ont pas le même niveau. Tant que chacun donne 100% de ce qu'il a à offrir!