

Cahier des charges

Projet Cloud Of Things

Server Nanny

Système de surveillance de la climatisation des datacenters

Réalisé par: *Ramzi Afli & Eya Gourar*

Enseignant: *M. Mohamed Bécha Kaâniche*
Élément d'étude: Cloud Of Things
Date: 31 Octobre, 2022

Contents

| | | |
|-----------|---|-----------|
| 1 | Concept | 1 |
| 1.1 | Contexte du projet | 1 |
| 1.2 | Problématique | 1 |
| 1.3 | Objectifs | 2 |
| 2 | Clients | 2 |
| 3 | Besoins fonctionnelles | 2 |
| 3.1 | Capteurs et réseau IOT | 2 |
| 3.2 | Application mobile | 3 |
| 4 | Choix des composants | 3 |
| 4.1 | Capteurs de temperature et d’humidité | 3 |
| 4.2 | Carte électronique | 5 |
| 4.3 | Caméra | 6 |
| 4.4 | Autre compsans | 6 |
| 5 | Choix technologiques | 7 |
| 6 | Architecture | 8 |
| 7 | Diagramme de déploiement | 9 |
| 8 | Limites | 9 |
| 9 | Modèle commercial | 10 |
| 10 | Livrables | 11 |
| 11 | Contraintes | 11 |
| 11.1 | Méthodologie du travail | 11 |
| 11.1.1 | Principe | 11 |
| 11.1.2 | Techniques | 12 |
| 11.1.3 | Rôles | 12 |
| 11.2 | Échéancier et tâches | 13 |

1 Concept

1.1 Contexte du projet

Les exigences accrues en termes de traitement des données et de capacité de stockage ont poussé les grandes sociétés à investir dans de nouvelles installations capables de proposer des services Web à un nombre toujours plus élevé d'utilisateurs. Dans ces installations, il est essentiel que les conditions environnementales, telles que la température et l'humidité, soient correctes pour la conservation des équipements et la sécurisation des opérations.

Température

Dans une salle de serveurs, la température recommandée pour le fonctionnement optimal des installations varie entre 18 et 27°C. [1]

Toute variation de la température ambiante dans la salle informatique en moins ou en plus est susceptible d'occasionner des problèmes de disponibilité, de sur refroidissement ou d'inefficacité et des coûts élevés.

Humidité

Le taux d'humidité de la salle serveur est aussi un indicateur à surveiller de près. Il est le rapport entre l'humidité d'un échantillon donné et celle de la quantité maximale d'humidité qu'il peut contenir.

[1] Lorsqu'il est trop bas, il peut occasionner des décharges électrostatiques au niveau des équipements et endommager certaines de leurs composantes. Trop élevé, il crée de la condensation en eau liquide et réduit la durée de vie des serveurs.

Incendies

Malgré la rareté des incendies dans les datacenters, celui qui a touché OVHcloud en mars 2021 a marqué les esprits : en quelques heures à peine, 3,6 millions de serveurs virtuels, supportant 464 000 noms de domaines, disparaissaient d'Internet. [1]

Donc, la sécurité incendie est plus qu'une mesure de précaution ou de prévention. Un système de détection d'incendie est un investissement essentiel pour l'actif le plus précieux de toute entreprise : ses collaborateurs.

Une bonne protection contre les incendies dans les centres de données repose sur quatre éléments : l'atténuation du risque, la détection, l'extinction et la récupération.

Dans ce contexte nous cherchons à concevoir une solution de monitoring environnemental afin de surveiller les paramètres clés d'un datacenter pour qu'il fonctionne de manière optimale.

1.2 Problématique

La surveillance à temps réel des indicateurs climatiques et la prévention préalable des inconvénients sont des nécessités pour le bon fonctionnement et la durabilité des équipements des data centers.

Comment Prévenir le surrefroidissement, le sous-refroidissement et les risques liés à l'électricité statique dans le data center?

Comment prévenir les inefficacités et diminuer les coûts opérationnels et le risque d'arrêt des applications critiques ?

Comment assurer l'intervention à temps précoce d'un incendie potentiel ?

En clair, il faut mettre en place des stratégies pour réduire la probabilité d'un incendie, pour

alerter le personnel qu'un incendie se produit, et pour prévenir d'un sur-chauffage d'un sur-refroidissement

1.3 Objectifs

- Garantir la disponibilité en détectant les points chauds dans les racks.
- Économiser sur le refroidissement en augmentant de façon sûre la température du data center.
- Intervenir instantanément en cas d'incendie soudain.
- Renforcer la disponibilité du data center en recevant des alertes relatives à l'environnement ambiant.
- Prendre des décisions stratégiques concernant la conception et le confinement des équipements de refroidissement.

2 Clients

Ce projet est en faveur des data centers dans des moyennes et grandes entreprises.

3 Besoins fonctionnelles

3.1 Capteurs et réseau IOT

Ce système IOT doit impérativement satisfaire des besoins selon des aspects différents qui touchent :

- À la sécurité : Le système IOT peut détecter la présence d'un incendie potentiel ou sur-chauffage, le risque de sur-refroidissement, le risque d'inefficacité et des décharges électrostatiques et de la condensation en eau liquide, tous capable d'endommager les composants et/ou réduire leurs durées de vie.
- À l'économie d'énergie : Le système IOT a l'intérêt d'économiser sur le refroidissement des racks en augmentant de façon sûre la température du data center.

En fait Le principe de ce système IOT consiste en :

- Installation de plusieurs capteurs de température et d'humidité par rack dans le data center qui mesurent et transmettent les niveaux de température et d'humidité à une carte électronique pour traitement et calcul.
- Cette même carte envoie des paquets au serveur pour communiquer avec la base de données dans laquelle seront stockées les informations.
- Une application mobile qui affiche à temps réel des mesures de température et d'humidité dans les salles du data center.
- Une caméra installée pour surveiller les salles et détecter la fumée et les incendie à un stade très précoce.

On utilise la détection d'incendie par vidéo pour **identifier rapidement** les incendies couvants et les petits incendies directement à la source.

Cela signifie que l'alarme incendie n'a pas à attendre que la fumée atteigne physiquement ses capteurs, perdant ainsi un temps précieux avant d'alerter les équipes de sécurité.

- On définit des alertes pour être prévenu en cas de sur-chauffe, de niveau d'eau trop haut, etc ..

3.2 Application mobile

- L'application mobile peut être utilisée à partir de différentes plateformes comme Android, iOS, Web, etc.
- L'administrateur peut enregistrer un compte dans l'application avec un accès complet à tous les objets et enregistrer les surveillants pour interagir avec les différents objets. Par exemple, les surveillants de la chambre de serveurs A peuvent seulement surveiller le refroidissement de la chambre de serveurs A, etc.
- L'administrateur peut ajouter de nouvelles salles, racks et capteurs à l'application mobile en utilisant respectivement l'identifiant de la salle, du rack et des capteurs.
- L'administrateur peut obtenir l'état de tous les objets de chaque salle et chaque rack et obtenir les relevés de différents capteurs en temps réel.
- Chaque utilisateur surveillant peut afficher l'état et les mesures des capteurs de chaque rack seulement situé dans la salle qui lui est associée.
- L'utilisateur peut recevoir des notifications push via l'application pour l'alerter en cas de problème (sur-chauffage, incendie, etc)
- L'administrateur peut recevoir un message MAIL à chaque fois un inconvénient ait eu lieu.

4 Choix des composants

4.1 Capteurs de température et d'humidité

Pour pouvoir surveiller les conditions climatiques à l'intérieur d'un data center, des capteurs sont nécessaires pour transmettre les niveaux de température et d'humidité au serveur d'application. Après quelques recherches, nous avons trouvé les capteurs [2] suivants:

- **DHT22:** Ce capteur de température et d'humidité a une précision de température de +/- 0,5 C et une plage d'humidité de 0 à 100. Il est simple à connecter au Raspberry Pi..
- **DSB18B20:** Ce capteur de température possède une sortie numérique, qui fonctionne bien avec le Raspberry Pi. Il a trois fils et nécessite une planche à pain et une résistance pour la connexion.
- **BME280:** Ce capteur mesure la température, l'humidité et la pression barométrique. Il peut être utilisé à la fois en SPI et en I2C.
- **Sense HAT:** Il s'agit d'une carte d'extension pour Raspberry Pi qui comporte des LED, des capteurs et un petit joystick. Elle se connecte directement sur le GPIO du Raspberry Pi, mais l'utilisation d'un câble ruban permet d'obtenir des lectures de température plus précises.

| sensor | DHT22 | DSB18B20 | BME280 | Sense Hat |
|----------------|---------------|----------------|------------|------------------|
| Protocol | custom choice | I2C/SPI | 1-wire | I2C |
| Form Factor | standalone | breakout-board | standalone | Raspberry shield |
| Number of pins | 4 | 7 | 3 | 40 |
| temperature | yes | yes | yes | yes |
| humidity | yes | no | yes | yes |
| price(TND) | 18 | 11 | 70 | 190-210 |

Table 1: Sensors features comparisons

Après avoir examiné les caractéristiques de chaque capteur figurant dans le tableau 1, nous avons fini par choisir le DHT22 pour les raisons suivantes :

- le DHT22 ne nécessite aucun composant supplémentaire pour fonctionner, au contraire du BME280 doit être soudé dans une carte de sortie.

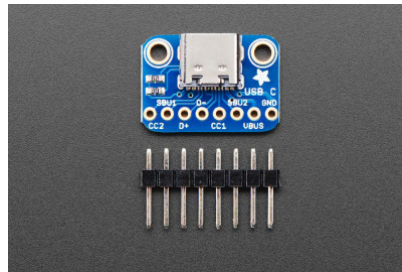


Figure 1: A Breakout-board

- Le DHT22 est relativement bon marché. Bien que le DSB18B20 soit moins cher, il ne fournit pas de relevés d'humidité, il est donc exclu de la liste. Bien que le BME280 et le Sense Hat prennent en charge les relevés de température et d'humidité comme le DHT22, ils sont beaucoup plus coûteux et plus difficiles à acquérir.
- Bonne précision et résolution de lecture (capable de détecter des variations de +/- 0,5°C et une humidité comprise entre 0% et 100%).
- Haute disponibilité sur le marché.

Décision finale: DHT22

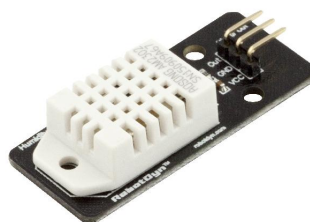


Figure 2: DHT22 sensor

4.2 Carte électronique

Pour le traitement des données et leur transmission ultérieure vers le cloud, on a eu besoin d'une carte électronique embarquée. Cette carte alimentera les capteurs et transmettra les mesures de l'environnement au premier.

- **Arduino Uno R3:** Arduino Uno est une carte microcontrôleur basée sur l'ATmega328P (datasheet). Elle contient tout ce qui est nécessaire pour supporter le microcontrôleur ; il suffit de la connecter à un ordinateur avec un câble USB ou de l'alimenter avec un adaptateur AC-to-DC ou une batterie pour commencer. Arduino peut être programmé pour exécuter des scripts afin d'acquérir les données des capteurs et de les transmettre. [2]
- **Raspberry pi4:** La Raspberry Pi 4 Modèle B est le dernier produit de la populaire gamme d'ordinateurs Raspberry Pi. Il offre des augmentations révolutionnaires de la vitesse du processeur, des performances multimédia, de la mémoire et de la connectivité par rapport au Raspberry Pi 3 Model B+ de la génération précédente, tout en conservant une compatibilité intra-modèles et une consommation d'énergie similaire. [3] [2]

| Embedded card | Arduino Uno R3 | Raspberry Pi4 |
|----------------------|------------------|---------------------|
| Type | micro-controller | mini computer |
| Architecture | AVR, ARM | ARM |
| HDMI connection | No | yes |
| Operating system | N/A | Required |
| Clock speed | 16 MHz | 1.2 GHZ |
| DRAM/SRAM | 12kB SRAM | 4GB RAM |
| Power supply | micro-USB/USB-C | USB-B |
| Programming language | C/C++ | C/C++, Java, Python |
| Price(TND) | 65 | 175 |

Table 2: Comparaison caractéristiques des Embedded cards

Comme il est nécessaire de disposer d'une puissance de calcul considérable pour effectuer des opérations de vision par ordinateur et qu'il est nécessaire de supporter le langage Python, nous allons utiliser une carte raspberry pi.

Décision finale: Raspberry pi 4 model B

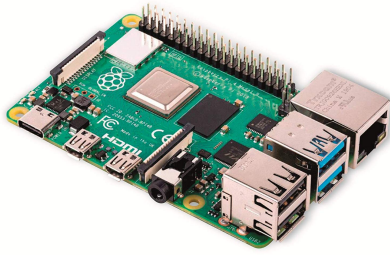


Figure 3: Raspberry PI 4 model B

4.3 Caméra

Nous allons utiliser une caméra pour ajouter une fonctionnalité de la détection d'incendie par vidéo pour identifier rapidement les incendies couvants et les petits incendies directement à la source.

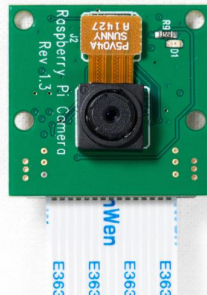


Figure 4: Raspberry PI 5MP Camera Board Module

4.4 Autre composants

Nous avons eu recours aux composants supplémentaires suivants pour compléter l'assemblage:

- des fils mâle-mâle, des fils mâle-femelle et des fils femelle-femelle.
- Deux résistances de 10K ohm qui serviront de résistances de type pull-up pour les capteurs.

ci-dessous l'assemblage des composants choisis :

On connecte deux capteurs DHT22 à la carte Raspberry pi 4 et on introduit deux résistances pull up pour chacun.

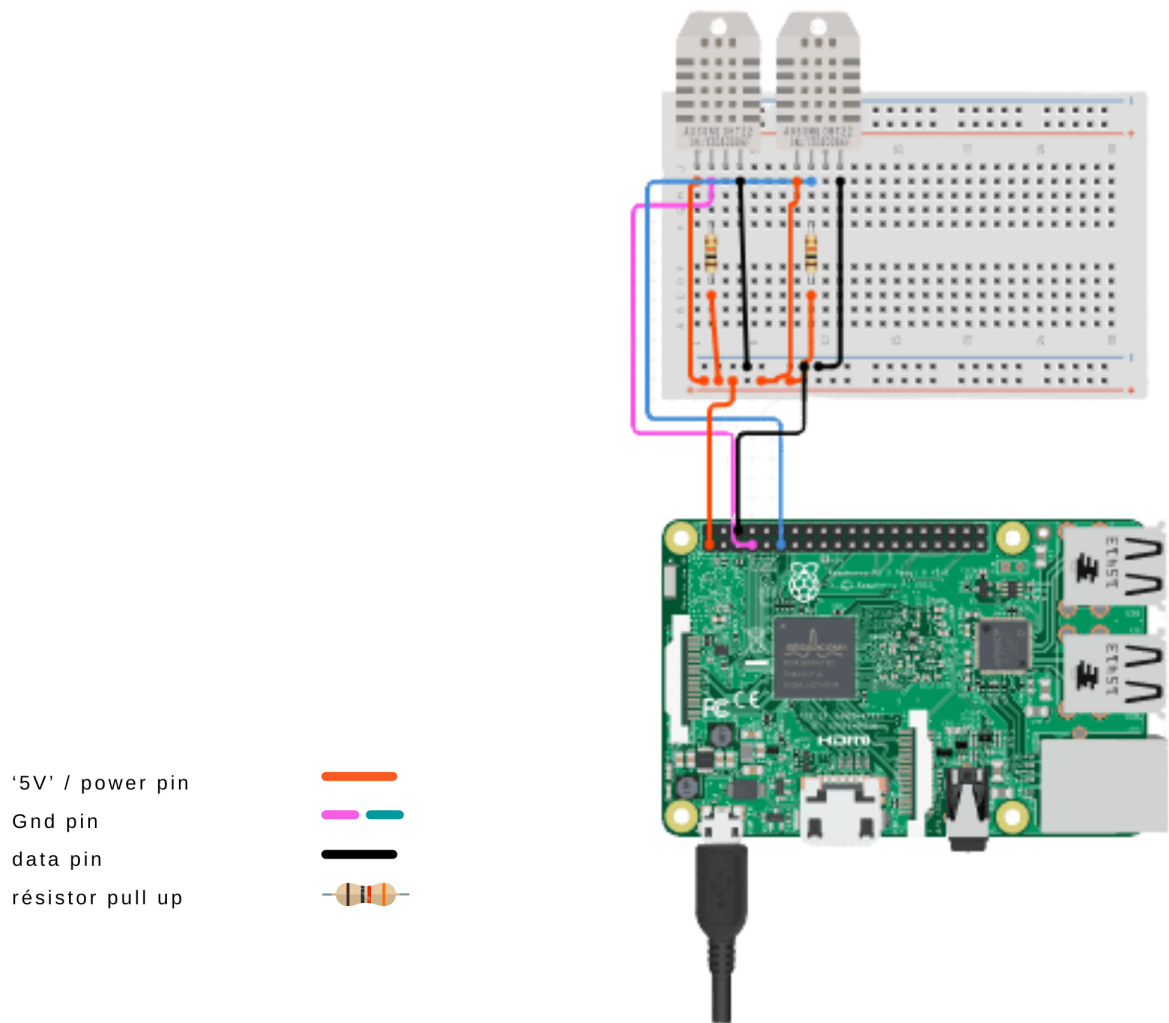


Figure 5: Schéma électronique

un DHT22 a trois pins - 5V, Gnd, et données. On Connecte le pin du voltage 5V au pin 2 (le pin supérieure à gauche, 5V) du Pi.

Le pin Gnd étiquetée '-' ou 'Gnd' est connecté au pin 6 Gnd du Pi.

Le pin restant sur le DHT22 est le pin de données et sera étiquetée 'out' ou 's' ou 'data'.

On le connecte à l'un des pin GPIO du Pi.

Une fois le câblage effectué, on allume la Pi.

5 Choix technologiques

Afin d'implémenter les différentes fonctionnalités de surveillance dans l'application mobile, différentes technologies seront utilisées :

– Back-end :

- MongoDB : Une base de données NoSQL orientée document. MongoDB est utilisé pour stocker les données des utilisateurs. MongoDB est pratique et facile à utiliser avec Jakarta EE.

- MQTT : un protocole de réseau léger de publication et d'abonnement utilisé pour communiquer les données collectées par les capteurs à un courtier MQTT en cloud.
- Middleware :
- JAX-RS : Java API for RESTful Web Services est une interface de programmation Java permettant de créer des services Web avec une architecture REST. JAX-RS 1.0 est défini par la JSR 311 et JAX-RS 2.0 est défini par la JSR 339.
- WildFly : WildFly, anciennement JBoss Application Server ou JBoss, est un serveur d'applications Java EE Libre écrit en Java, publié sous licence GNU LGPL. Étant écrit en Java, WildFly peut être utilisé sur tout système d'exploitation fournissant une machine virtuelle Java.
- Front-end :
- Flutter : Flutter est un SDK multiplateforme qui donne la possibilité de développer des applications pour de multiples plateformes à partir d'un code unique. Flutter permet d'accéder à des fonctionnalités natives et à un aspect natif sans utiliser de frameworks comme Apache Cordova ou Capacitor. Les applications Flutter sont faciles et rapides à faire et rapide à coder et ne nécessitent pas de connaissances préalables en développement web.

6 Architecture

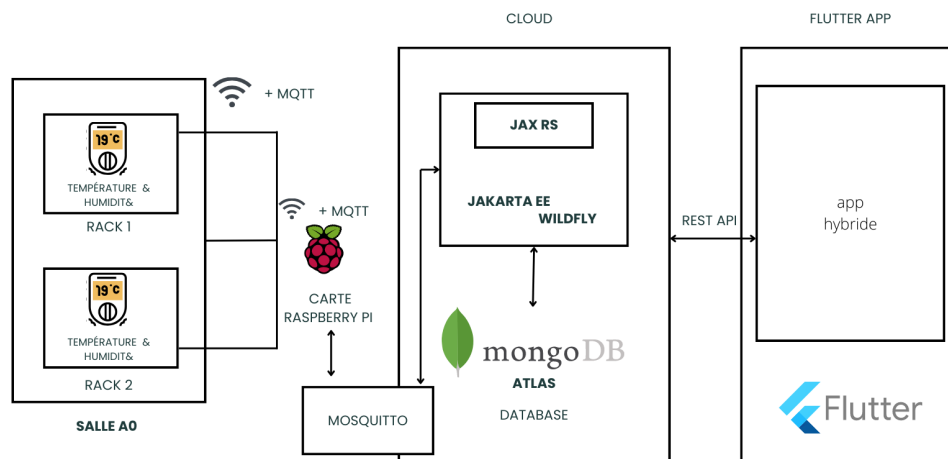


Figure 6: Architecture du projet

7 Diagramme de déploiement

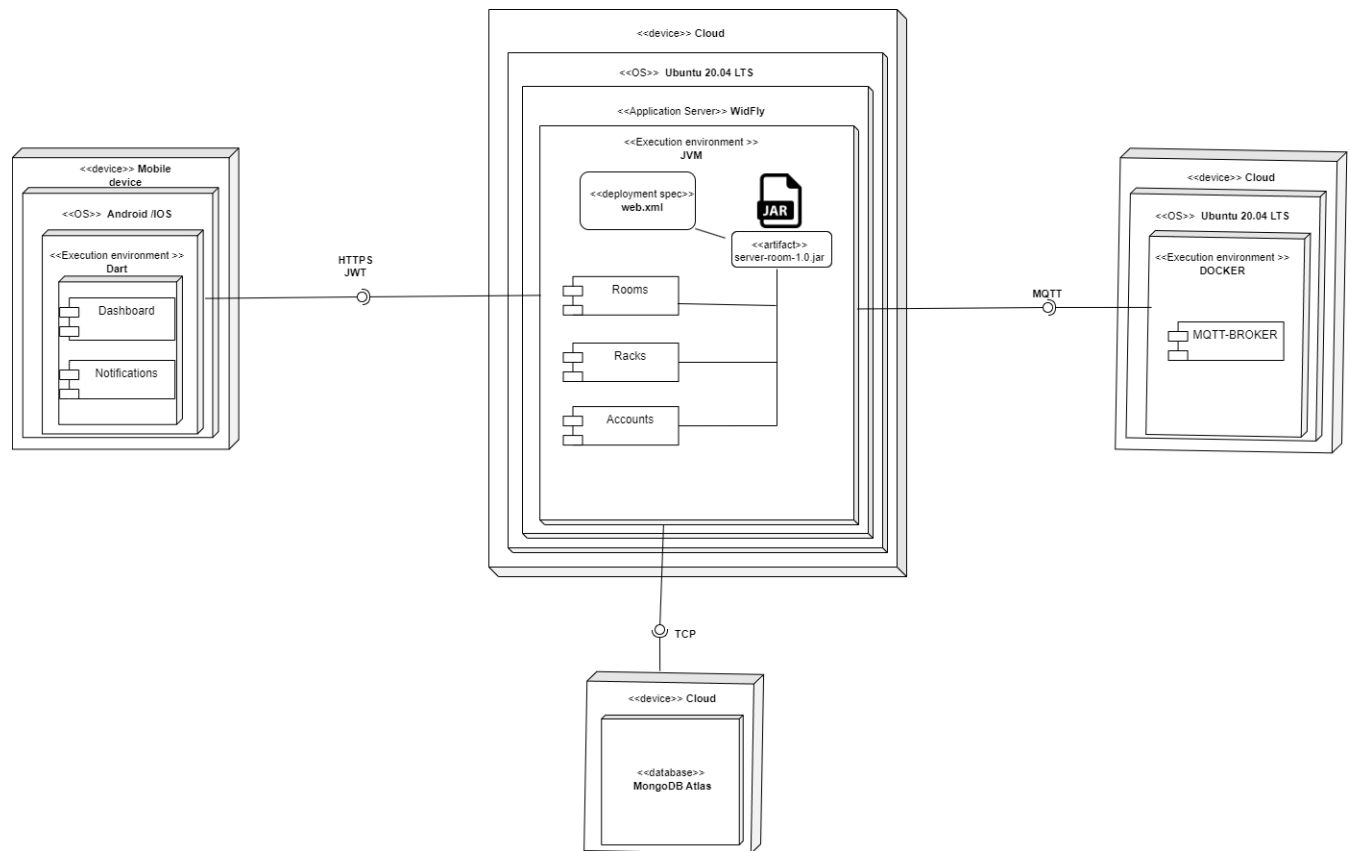


Figure 7: Diagramme de déploiement

8 Limites

D'une part, il existe le problème de la panne en connexion Internet. Si il y'a un problème au niveau de la connexion à Internet la suivie à temps réel des changements et l'alarme à un inconvénient passé sera inachevable par l'utilisateur surveillant.

D'une autre part, il existe le cas de fausses alertes des capteurs et de l'algorithme de détection de fumée.

Il y'a encore le problème d'identification correcte des points chauds dans les racks atteints par l'inconvénient. En fait, on associe à chaque rack un nombre défini de capteurs (en haut, en arrière, en bas, etc..). Si les racks seraient placés très proches, les capteurs peuvent detecter les points chauds ou un changement significatif de l'état du rack à côté. Il faut justement savoir bien placer les capteurs et de fixer des seuils bien étudiés des métriques mesurés depuis lesquels il y'aura déclenche de l'alerte.

9 Modèle commercial

Avoir un produit qui corresponde en tout point à la réalité du marché est un exercice complexe. Les habitudes des consommateurs évoluent et il est parfois compliqué de suivre le rythme du changement au sein d'une entreprise.

Le marketing-mix ou les 4P est la recette pour atteindre les objectifs fixés:

Politique Produit : Notre Produit sera le centre de notre stratégie Marketing.

Marque : Solution "Server Nanny" qui combine deux parties hardware et software qui se résume en une application hybride.

Services liés au produit :

- Service Après-Vente.
- Des mises à jour, des nouvelles fonctionnalités et des améliorations quotidiennes.
- Des garanties allant jusqu'à 5 ans.

Caractéristiques: Diversification et variété en termes des options et du design.

Politique Prix :

Le prix est au cœur du positionnement et l'élément clé de la communication. La politique suivie est la politique d'écroumage.

– Stratégie d'écroumage:

Les entreprises ont souvent recours à l'écroumage des prix lorsqu'elles lancent de nouveaux produits novateurs qui n'ont que peu de concurrence. Cette stratégie permet également de viser des segments de clientèle spécifiques avec un fort pouvoir d'achat.

Suite à l'absence d'une concurrence concrète à la Tunisie. Le prix de la solution dépend des fonctionnalités que le client souhaite implémenter dans sa serre, telles que l'introduction de l'alarme de l'incendie ou le dévrouillage des racks à proximité.

La politique de réduction commerciale :

- Réductions exceptionnelles accordées aux clients pour cause de problème de qualité ou de non conformité.

Les modalités de paiement varient de virement bancaire ou paiement online.

Politique Distribution: La distribution et le choix de son canal conditionne la visibilité et l'accessibilité du produit.

– Circuits de distribution :

- Direct : Notre produit est vendu directement, sans intermédiaire, aux clients : Il s'agit de la solution logicielle complète.
- Court : On va se fournir directement aux équipements IOT (capteurs, cartes électroniques,...) et revend ensuite au client.

– Canaux de distribution:

- La livraison directement chez nos clients pour la solution logicielle.
- Le commerce de détail : Achat des équipements pour les revendre aux clients.

– Les intermédiaires :

Le commerce indépendant : les détaillants et grossistes vendant de manière isolée les équipements IOT (capteurs, caméra de surveillance, ..)

Politique Communication : Nous proposons toutes les techniques publi-promotionnelles destinées à soutenir un produit.

- Mesures publicitaires en ligne: Display Advertising, Vidéos (sur YouTube, Facebook, etc.), publi- cité sur les réseaux sociaux, etc.
- Communication personnelle : Échanges avec le client.
- Communication internet : E-mailings, Newsletters, Marketing des médias sociaux.
- Les Ventes : Démarchage, Démonstration, Participation à des salons et foires...
- La Publicité : Affichage, brochures, plaquettes.
- Les Relations Publiques : communiqués et dossiers de presse, opérations de sponsoring, parrainage, tenue d'événements.

10 Livrables

À la fin du projet, les éléments suivants seront livrés :

- Application mobile de surveillance.
- Code source des différents composants du projet sur GitHub.
- Prototypage/simulation de datacenter intelligent.

11 Contraintes

11.1 Méthodologie du travail

On va travailler avec Extreme Programming (XP) qui est un cadre de développement logiciel agile qui vise à produire des logiciels de meilleure qualité, XP est le plus spécifique des cadres agiles concernant les pratiques d'ingénierie appropriées pour le développement de logiciels. [4]

XP tente de stimuler la créativité des développeurs et accueille les erreurs comme un facteur évident dans le processus de travail.

11.1.1 Principe

Les principes de la méthode XP ne sont pas nouveaux puisqu'il s'agit de ceux des méthodes agiles. La différence et l'originalité résident dans le fait qu'elles sont poussées à l'extrême. La méthode XP s'appuie sur [4] :

- Une forte réactivité au changement des besoins du client
- Un travail d'équipe : dans notre contexte il s'agit du **Pair Programming**
- La qualité du travail fourni
- La qualité des tests effectués au plus tôt

11.1.2 Techniques

Les pratiques de XP sont des consignes et des méthodes de travail très concrètes. Alors que les valeurs et principes présentés sont également appliqués dans d'autres méthodes de travail agiles, les techniques concrètes de l'extreme programming sont des caractéristiques uniques. Elles ont aussi légèrement évolué au fil du temps et sont différentes d'une source à une autre. De manière générale, les techniques se divisent en quatre domaines différents. [5]

Feedback détaillé:

Dans le cadre de l'extreme programming, les équipes de développeurs travaillent dans des cycles extrêmement courts. Cela permet de tester encore et encore le code écrit. Le test unitaire ou Test-Driven Development va même jusqu'à écrire un environnement de test avant la création du code source à proprement parler. Le code ne passant pas ce test ne peut pas continuer à être développé. Le feedback vient donc ici du même système. [5]

Processus continu

Les équipes XP révisent leur code en permanence. Ce remaniement du code ou refactoring doit veiller à améliorer le code source ainsi qu'à éliminer des répétitions et des composants inutiles. Un code optimisé de la sorte est plus compréhensible, même pour des lecteurs externes, et donc moins source d'erreur. Dans le cadre de l'extreme programming et d'autres méthodes de travail agiles, l'intégration continue permet aux équipes d'intégrer en permanence le nouveau code dans l'ensemble du projet.

Plusieurs fois par jour, un développeur intègre son travail dans le projet. Les différentes contributions sont ainsi contrôlées en continu et tous les intervenants travaillent avec le dernier état. [5]

Compréhension commune

Avec une conception simple (Simple Design), le code est compréhensible pour tous les intervenants. Tout ce qui complique inutilement le code source doit donc être supprimé. Pour les développeurs qui travaillent suivant l'extrême programming, le but est d'éviter tous les doublons. L'objectif du programmeur concerné devrait par ailleurs ressortir clairement du code source.

Toute l'équipe en porte la responsabilité tant pour les erreurs que pour les réussites. Cette technique invite en outre à perfectionner le code d'un autre et à apporter ses idées. [5]

11.1.3 Rôles

Dans l'extrême programming, les rôles servent à répartir les tâches et les compétences entre tous les intervenants, tant les développeurs que les clients. [4]

Client

L'extreme programming agit de manière très orientée client, ce qui peut aller jusqu'à intégrer le client comme un membre de l'équipe et à avoir au moins un représentant sur site (On-Site Customer). Le client pose ses exigences pour le produit, mais n'indique que partiellement la façon d'atteindre les objectifs. [4]

Développeurs

L'équipe de développeurs n'est pas sous-divisée. Autrement dit : toute personne créant activement le produit prend la casquette de développeur. L'équipe regroupe donc non seulement les programmeurs, mais aussi d'autres personnes participant à la création, en fonction des exigences du projet.

Outre le travail de développement effectif, la mission des développeurs est également de réagir aux désirs du client : évaluer la charge de travail, établir un calendrier, planifier la mise en œuvre. [4]

Manager

Le rôle du manager consiste à faire le lien entre les développeurs et les clients. Le manager veille ici à ce que les règles définies au préalable ainsi que les conventions générales d'une discussion constructive soient respectées. Le manager endosse donc si nécessaire un rôle de médiateur. Son rôle est parfois également qualifié de tracker (traqueur). [4]

Coach

Toute l'équipe (y compris le client) doit savoir gérer l'extrême programming et pouvoir mettre en pratique cette méthode de travail de manière cohérente. Le coach accompagne l'équipe idéalement pendant toute la phase de développement, est disponible pour répondre aux questions et aide à clarifier des points obscurs. [4]

11.2 Échéancier et tâches

Le développement du projet passera par différentes étapes sur 9 semaines :

- Planification de l'architecture et connexion des objets.
- Traitement des données collectées et connexion au broker MQTT du cloud.
- Mise en œuvre de la logique de la base de données.
- Développement de l'application, création des API nécessaires et connexion à la base de données.
- Hébergement de Wildfly sur un serveur en cloud.
- Développement de l'application mobile Flutter et connexion au point de terminaison du serveur web.
- Créer un prototype/simulation de data center.
- Organiser et mettre à jour le dépôt du projet avec un livre de conception, le code source complet, la documentation technique et une vidéo de démonstration.

References

- [1] J. Niemann, K. Brown, and V. Avelar, “Impact of hot and cold aisle containment on data center temperature and efficiency,” *Schneider Electric Data Center Science Center, White Paper*, vol. 135, pp. 1–14, 2011.
- [2] “How to Build a Raspberry Pi Temperature Monitor,” accessed: 2022-10-15.
- [3] “Raspberry pi foundation,” accessed: 2022-10-15.
- [4] J. T. Bell, “Extreme programming,” 2001.
- [5] K. Beck and M. Fowler, *Planning extreme programming*. Addison-Wesley Professional, 2001.