
Architecture Document

Ethan GAUTHIER - Sterenn LE HIR
Alexandre LECOMTE - Axel MARTIN
Kechiedou MEDA - Ivann VYSLANKO

Sommaire

1	Principe de mise en œuvre de la solution	2
2	Règles d'architecture	2
3	Modèle statique	5
3.1	Cas d'utilisation	5
3.2	Descriptions des classes principales et de leurs responsabilités	6
3.3	Diagramme de composants	7
4	Modèle dynamique	8
4.1	Diagrammes de séquence	8
4.2	Diagramme d'état	11
5	Explication de la prise en compte des contraintes d'analyse	11
6	Cadre de production	11
6.1	Outils de développement	11
6.2	Outils de configuration	12

1 Principe de mise en œuvre de la solution

Nous avons plusieurs fonctionnalités à intégrer. Pour l'allumage physique du chauffage, nous allons intégrer un bouton ON/OFF permettant d'allumer/éteindre le chauffage, et un écran LCD et bouton + et - pour gérer la température cible qu'on souhaite avoir dans la pièce. Pour l'allumage à distance, nous aurons une application mobile, liée à des API. Une API concentrera les calculs de prédiction d'énergie tandis qu'une autre, le contrôle à distance du chauffage. Une base de donnée sera également mise en place dans le cloud.

2 Règles d'architecture

Nous allons échanger plusieurs données entre le chauffage, la base de données, l'API de prédiction et celle de contrôle. Pour la température nous utiliserons l'unité degré Celsius. Pour la régression linéaire de l'API de prédiction, les paramètres à prendre en compte seront normalisés afin de mieux analyser ces derniers.

Au niveau des règles de qualité de code, nous souhaitons atteindre plusieurs critères. Nous voulons que notre code suivent les standards de PMD ou de Checkstyle. Nous avons également des critères sur nos tests, avec une couverture de minimum 90% et un score de mutation de minimum 85%.

Voici la liste des topics que nous utilisons pour nos échanges mqtt :

CONTROL

/ONOFF

Rôle : Pour activer/désactiver le chauffage

Messages : “0” (éteindre) et “1” (allumer)

Subscriber(s) : ESP

Publisher(s) : APP MOBILE

/TEMP

Rôle : Pour indiquer la température cible d’allumage du chauffage

Messages : température au format : XX.X

Subscriber(s) : ESP

Publisher(s) : APP MOBILE

/DOOR

Rôle : Pour indiquer l’intervalle de temps durant laquelle les trappes sont fermées

Messages : intervalle au format : HH :MM ;HH :MM

Subscriber(s) : ESP

Publisher(s) : APIControlleur

/ACK

Rôle : Pour activer/désactiver le chauffage

Messages : “0” (éteindre) et “1” (allumé)

Subscriber(s) : APIControlleur, BDD

Publisher(s) : ESP

DATA

/TEMP-IN

Rôle : Pour récupérer la valeur de la température intérieure de la pièce (capteur en bas)

Messages : Température au format : XX.X

Subscriber(s) :

Publisher(s) : ESP

/TEMP-OUT

Rôle : Pour récupérer la valeur de la température extérieure de la pièce (capteur en bas)

Messages : Température au format : XX.X

Subscriber(s) :

Publisher(s) : ESP

/TEMP-TARGET

Rôle : Pour envoyer la température cible que l'on veut

Messages : Température au format : XX.X

Subscriber(s) :

Publisher(s) : ESP

/ENERGY/ASK

Rôle : Pour demander d'envoyer l'énergie produite

Messages : 1

Subscriber(s) : ESP

Publisher(s) : API Prédiction

/ENERGY/RESP

Rôle : Pour envoyer l'énergie produite à 10h - 13h - 16h - 20h

Messages : Énergie au format : XX.XX

Subscriber(s) : API Prédiction

Publisher(s) : ESP

/PREDICTION

Rôle : Pour envoyer la daily prediction à l'app mobile

Messages : "'matin' : XX, 'midi' : XX, 'soir' : XX"

Subscriber(s) : APP MOBILE

Publisher(s) : API Prédiction

3 Modèle statique

3.1 Cas d'utilisation



FIGURE 1 – Diagramme de cas d'utilisation

3.2 Descriptions des classes principales et de leurs responsabilités

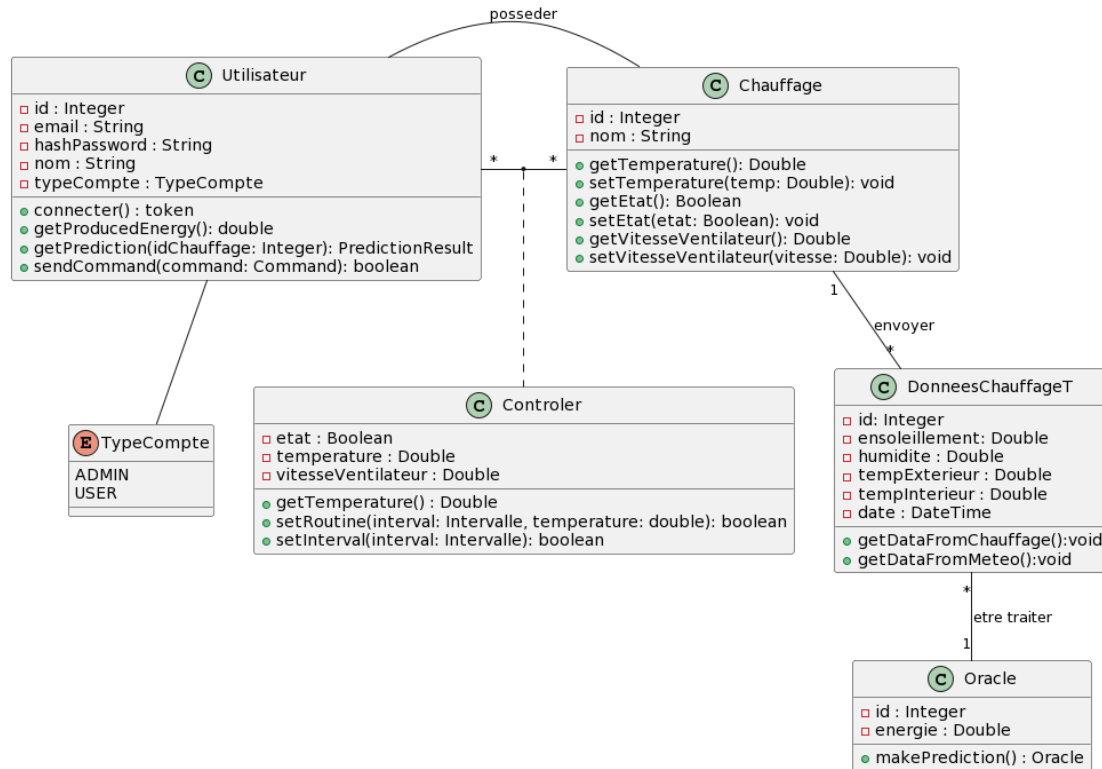


FIGURE 2 – Diagramme de classes

3.3 Diagramme de composants

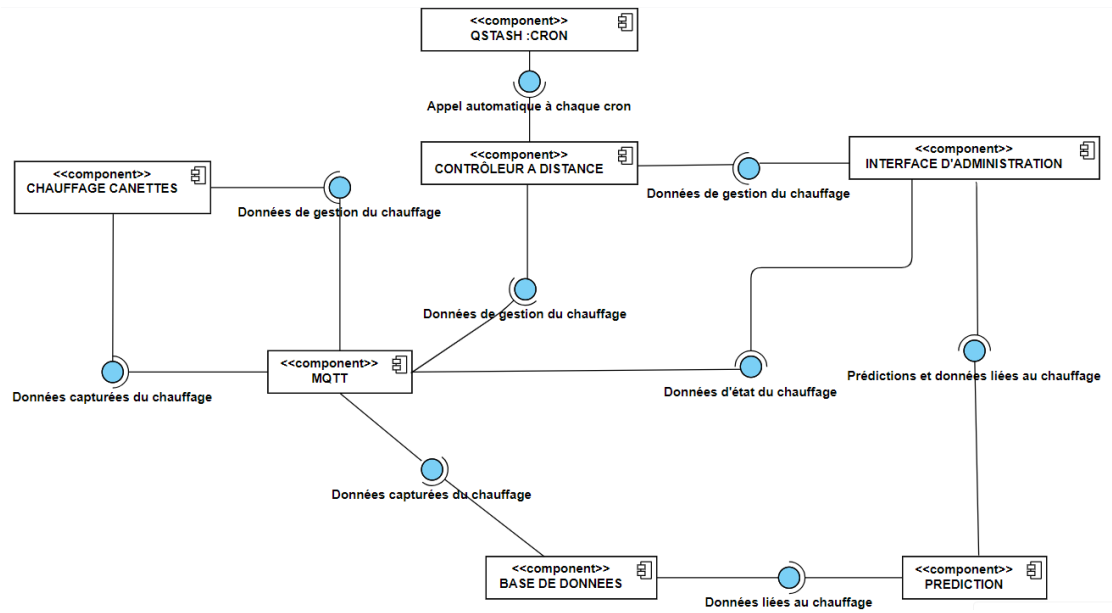


FIGURE 3 – Diagramme de composants

4 Modèle dynamique

4.1 Diagrammes de séquence

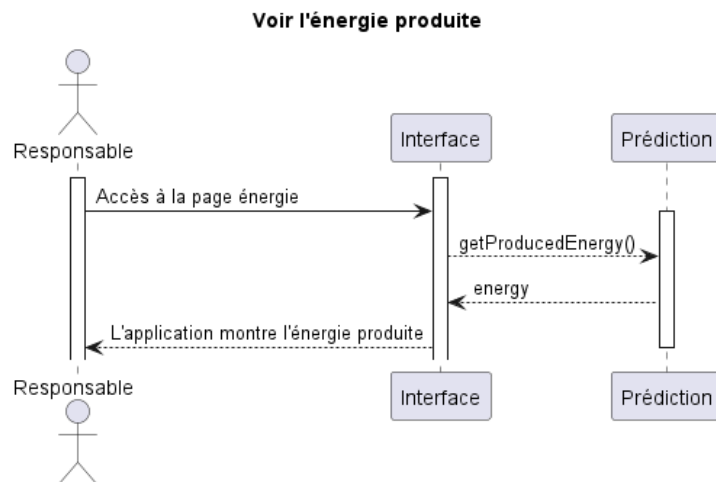


FIGURE 4 – Observer l'énergie produite

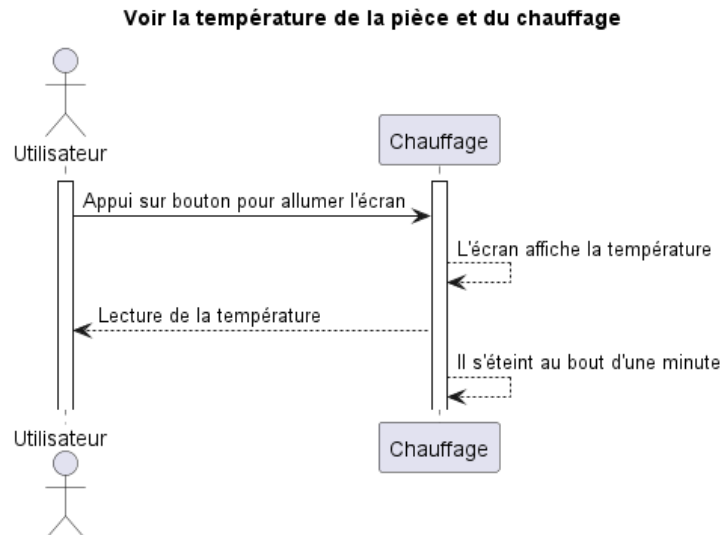


FIGURE 5 – Observer la température

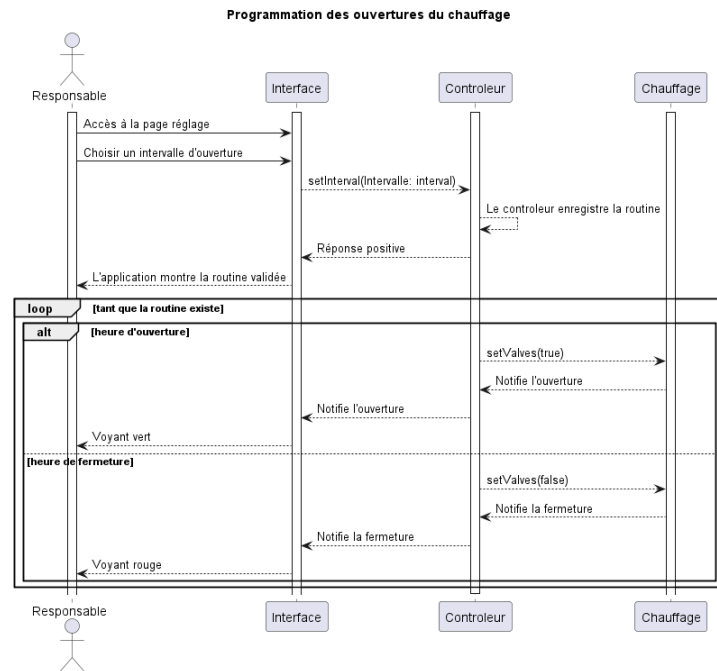


FIGURE 6 – Régler l'ouvertures des clapets

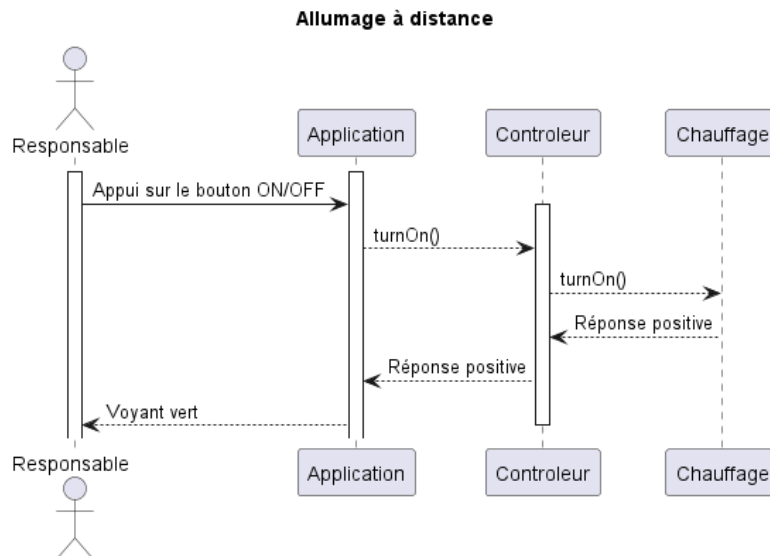


FIGURE 7 – Allumage à distance

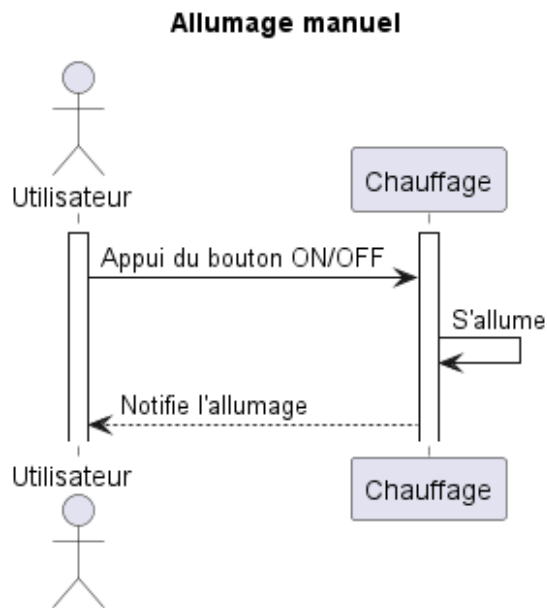


FIGURE 8 – Allumage à proximité

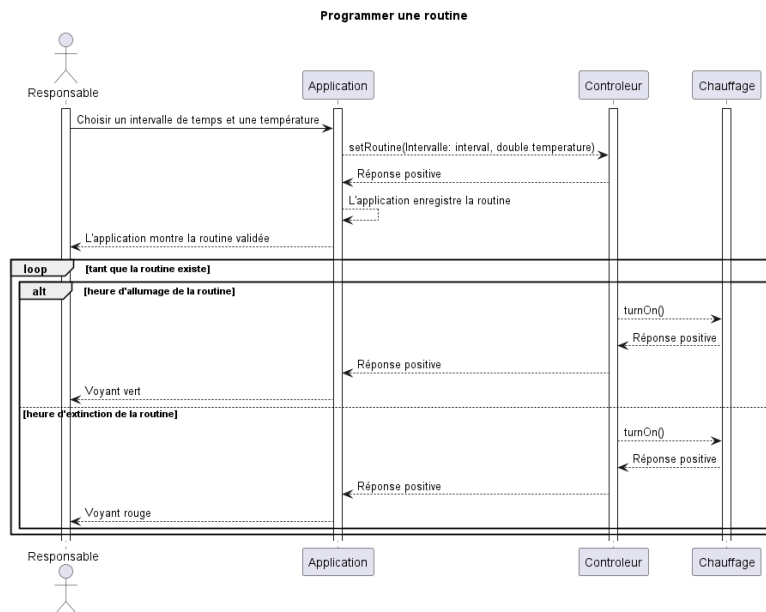


FIGURE 9 – Allumage par routine

4.2 Diagramme d'état

Nous avons réalisé le diagramme d'état du chauffage. Ce dernier montre les transitions du dispositif, comment il est allumé ou éteint.

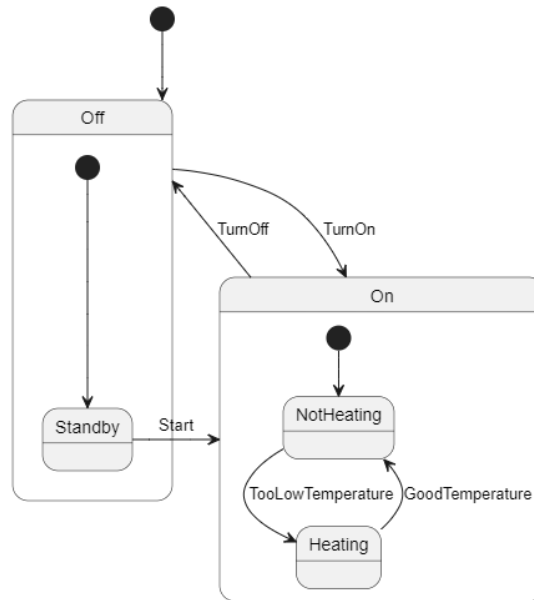


FIGURE 10 – Diagramme d'états pour le chauffage

5 Explication de la prise en compte des contraintes d'analyse

Le projet consistant en un chauffage le plus écologique possible, nous cherchons à limiter sa consommation d'énergie. Les prédictions sont donc réalisées en dehors du chauffage par souci de consommation, et également de performance. Les données sont également stockées séparément du service de prédictions, et la communication entre les deux est assurée par un communicateur MQTT.

6 Cadre de production

6.1 Outils de développement

Comme IDE, nous utiliserons IntelliJ IDEA de la suite JetBrains en utilisant le plugin Flutter pour l'application. Il nous servira également pour le développement des deux API. Pour le côté électronique, nous coderons sur Arduino IDE.

6.2 Outils de configuration

Pour gérer la planification du projet, nous utiliserons l'outil ClickUp, permettant d'assigner des tâches aux différents membres de l'équipe et de les visualiser sous forme de Kanban ou de diagramme de Gantt. Nous avons également un serveur Discord pour communiquer ainsi qu'un dossier partagé Google Drive concentrant divers documents tels que les livrables ou le cahier des charges.