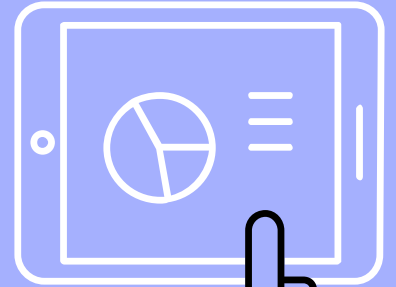
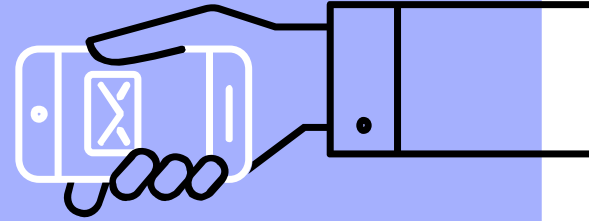
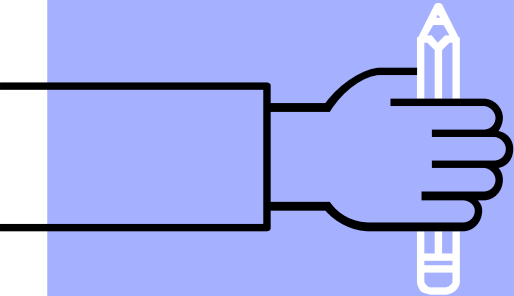
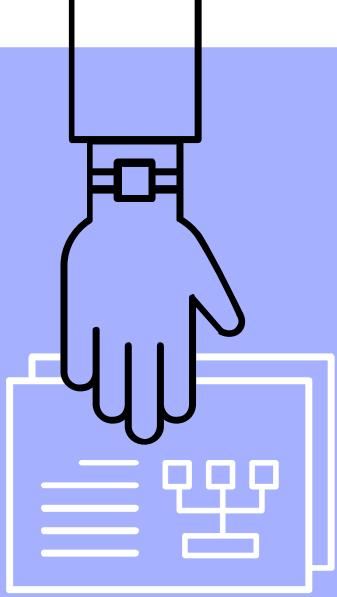


# Projet Domotique



# Membres du groupe

- ▶ Aykel Cheniour - [aykel.cheniour@cpe.fr](mailto:aykel.cheniour@cpe.fr)
- ▶ Valentin Valette - [valentin.valette@cpe.fr](mailto:valentin.valette@cpe.fr)
- ▶ Enagnon Farell Grazina Ahouandjinou - [ah.farell@gmail.com](mailto:ah.farell@gmail.com)
- ▶ Lucien Burdet - [lucien.burdet@cpe.fr](mailto:lucien.burdet@cpe.fr)

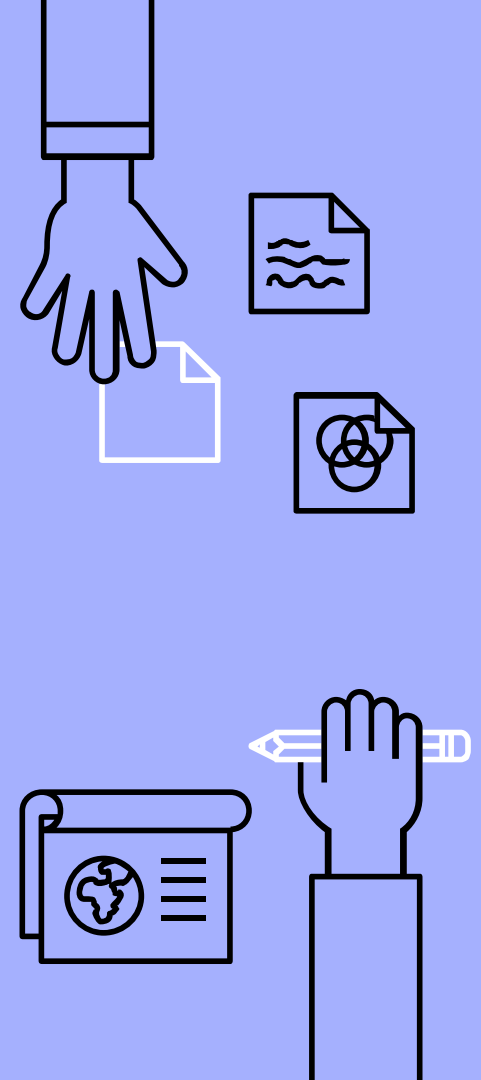
Supervisé par **Raphael Leber** et **Fabrice Jumel**.

**Lien du dashboard :**

<https://work.lucien-brd.com/>

**Lien du git :**

[https://gitlab.com/5irc\\_prj\\_21-22/Sujet\\_2\\_\\_Domotique\\_1/G2\\_Cheniour\\_Ahouandjinou\\_Burdet\\_Valette](https://gitlab.com/5irc_prj_21-22/Sujet_2__Domotique_1/G2_Cheniour_Ahouandjinou_Burdet_Valette)



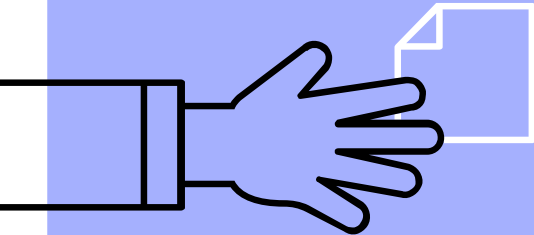
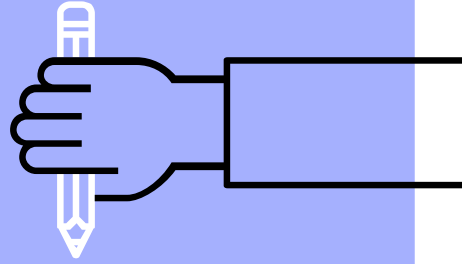
# Sommaire

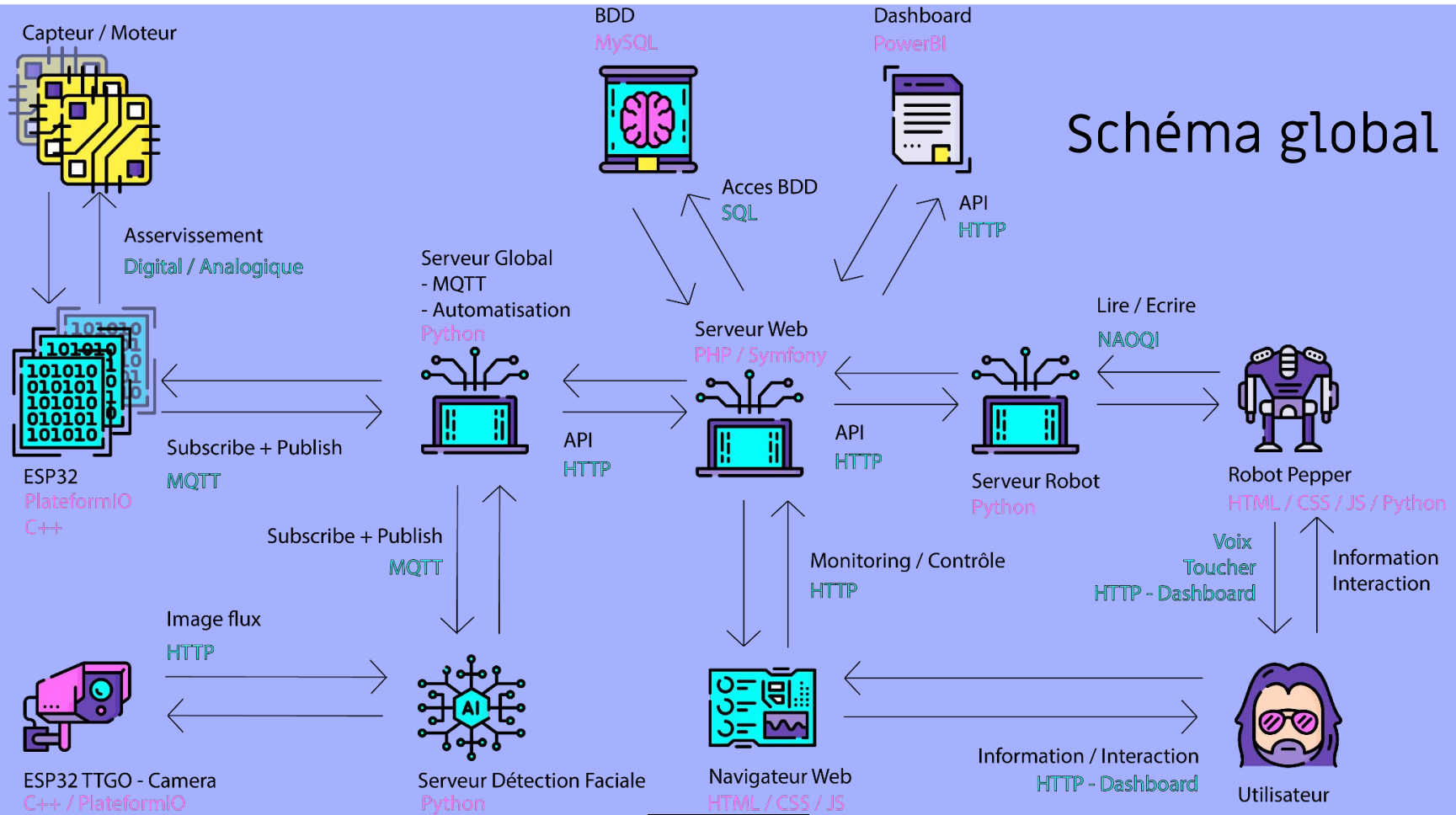
- ▷ Introduction
- ▷ Activités réalisées
  - Serveur WEB
  - Robot Pepper
  - Serveur global
  - IOT
- ▷ Démonstration
- ▷ Conclusion



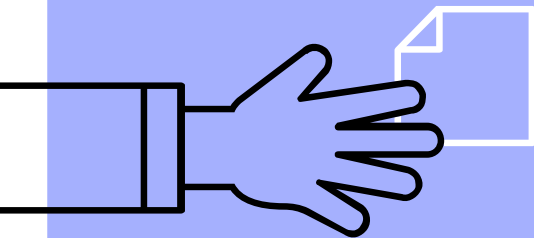
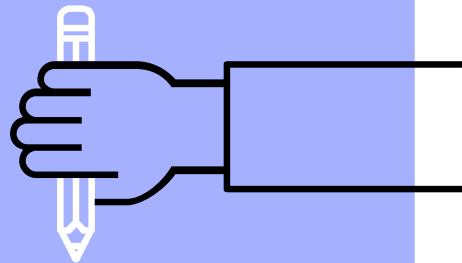
# 1. Introduction

Présentation du projet





## 2. Activités réalisées



### Serveur WEB

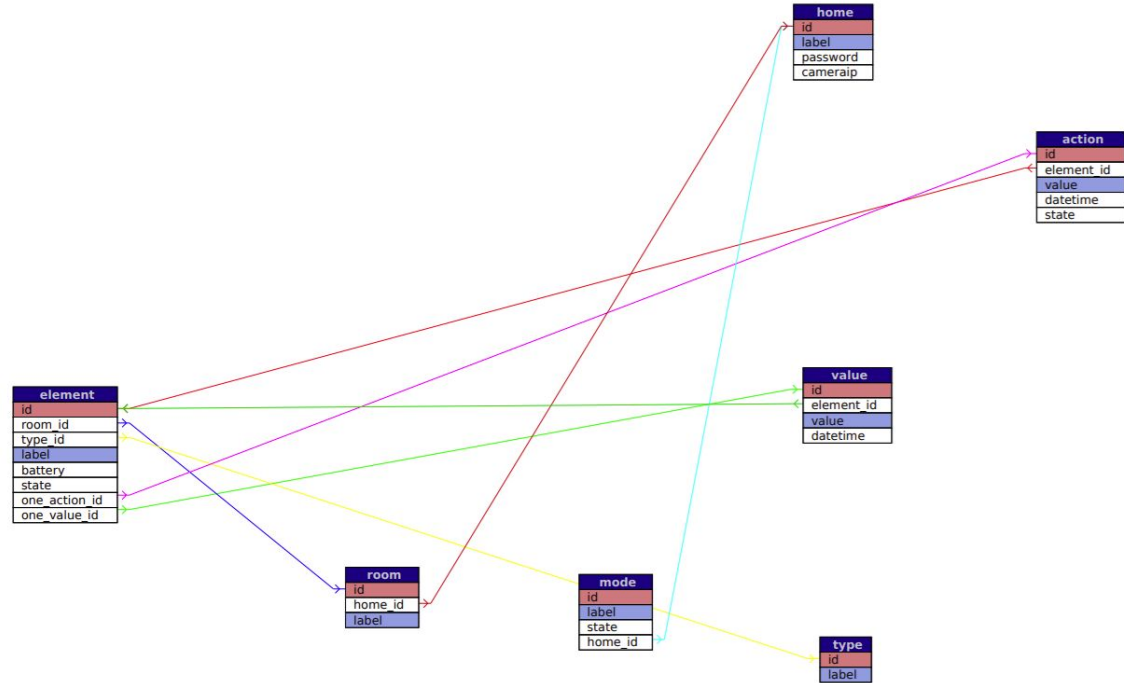
Application WEB :

- PHP - Symfony
- API
- Dashboard
- Base de données

# Serveur Web

## Base de données

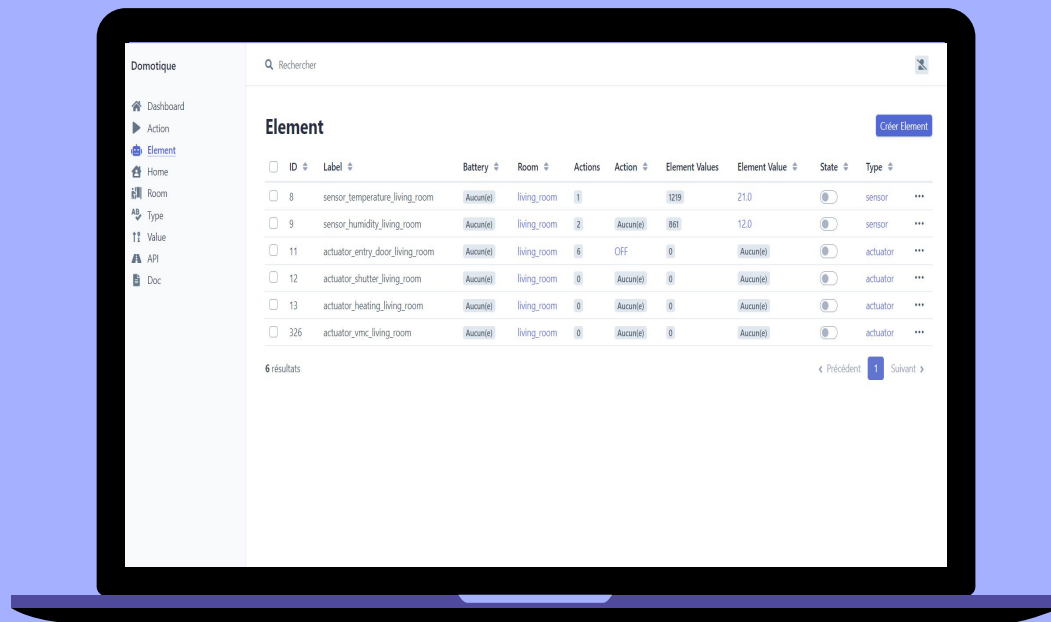
- MySQL



# Serveur Web

## Dashboard

- ▶ Ajouter
- ▶ Supprimer
- ▶ Editor

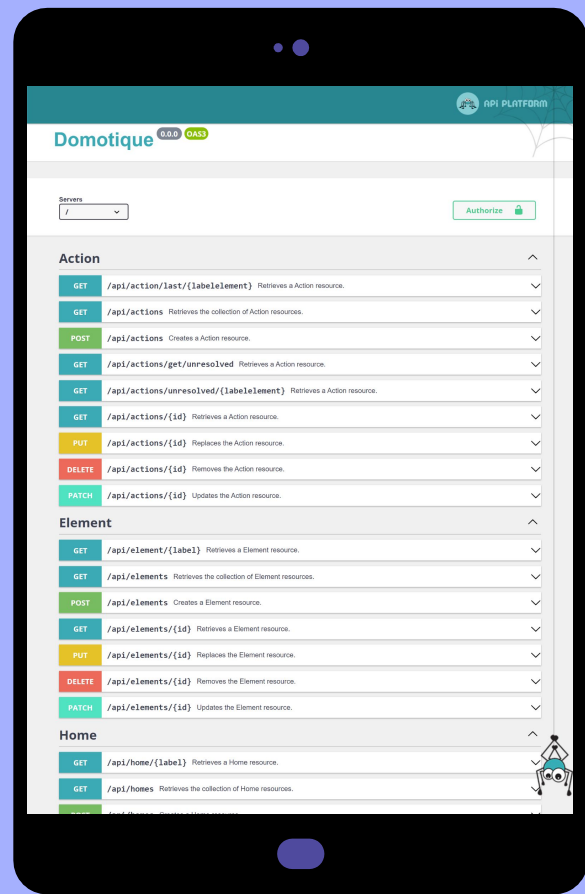
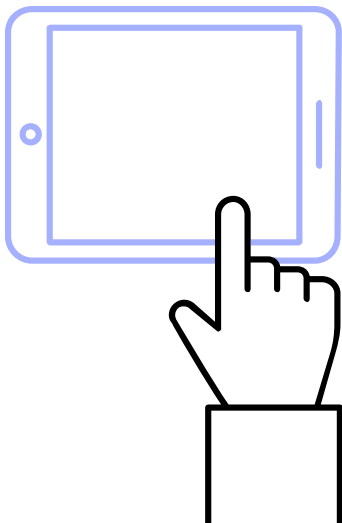




# Serveur Web

## API

- ▷ Ajouter
- ▷ Supprimer
- ▷ Editor





## Action

GET

/api/action/last/{labelement}

GET

/api/actions

POST

/api/actions

GET

/api/actions/get/unresolved

GET

/api/actions/unresolved/{labelement}

GET

/api/actions/{id}

PUT

/api/actions/{id}

DELETE

/api/actions/{id}

PATCH

/api/actions/{id}



## Element

GET

/api/element/{label}

GET

/api/elements

POST

/api/elements

GET

/api/elements/{id}

PUT

/api/elements/{id}

DELETE

/api/elements/{id}

PATCH

/api/elements/{id}



## Mode

GET

/api/get/mode/active/{labelhome}

GET

/api/mode/{labelhome}/{label}

GET

/api/modes

POST

/api/modes

GET

/api/modes/{id}

PUT

/api/modes/{id}

DELETE

/api/modes/{id}

PATCH

/api/modes/{id}



## Home

GET

/api/home/{label}

GET

/api/homes

POST

/api/homes

GET

/api/homes/{id}

PUT

/api/homes/{id}

DELETE

/api/homes/{id}

PATCH

/api/homes/{id}



## Room

GET

/api/room/{label}

GET

/api/rooms

POST

/api/rooms

GET

/api/rooms/{id}

PUT

/api/rooms/{id}

DELETE

/api/rooms/{id}

PATCH

/api/rooms/{id}



## Type

GET

/api/type/{label}

GET

/api/types

POST

/api/types

GET

/api/types/{id}

PUT

/api/types/{id}

DELETE

/api/types/{id}

PATCH

/api/types/{id}



## Value

GET

/api/value/last/{labelement}

GET

/api/values

POST

/api/values

GET

/api/values/{id}

PUT

/api/values/{id}

DELETE

/api/values/{id}

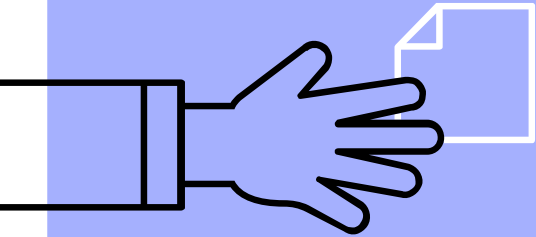
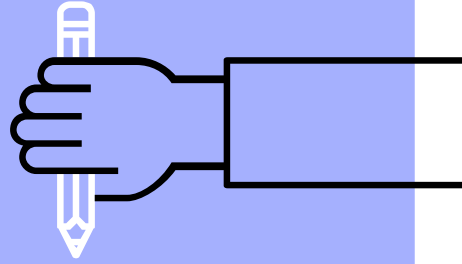
PATCH

/api/values/{id}

# Routes de l'API

## 2. Activités réalisées

Robot Pepper

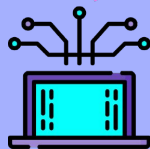


# Robot Pepper

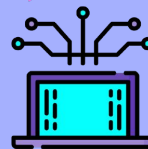
## Serveur

- Python - NAOqi
- Requêter l'API du serveur WEB
- Service visible par le Pepper

Serveur Web  
PHP / Symfony



Serveur Robot  
Python



get\_home\_by\_id

HTTP GET - /homes/<home\_id>

get\_active\_mode\_label\_by\_home\_name

HTTP GET - /get/mode/active/<home\_name>

get\_last\_value\_by\_element\_name

Temperature / Humidity

HTTP GET - /value/last/<element\_name>

get\_homes

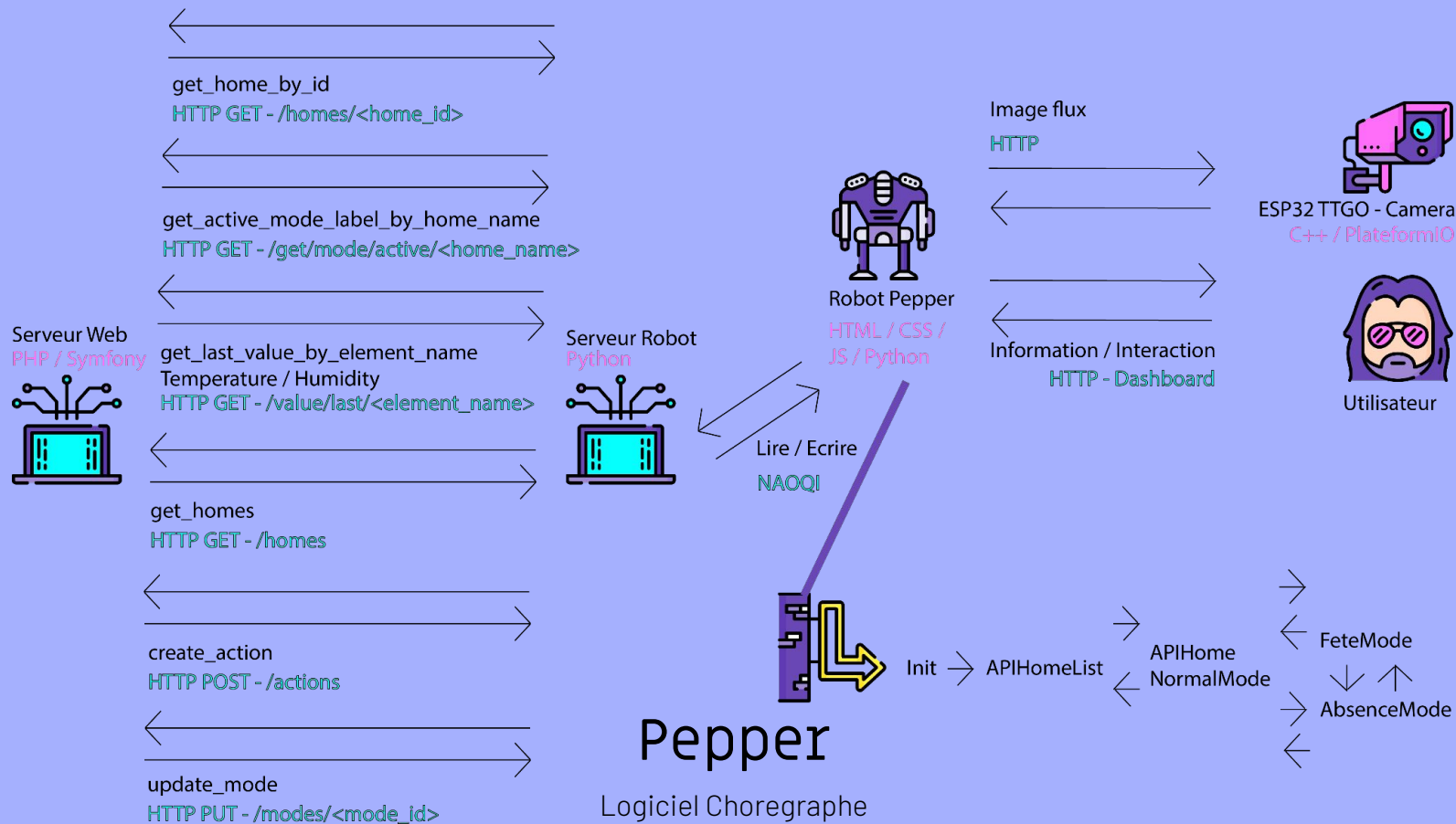
HTTP GET - /homes

create\_action

HTTP POST - /actions

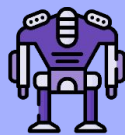
update\_mode

HTTP PUT - /modes/<mode\_id>



# Pepper

Init



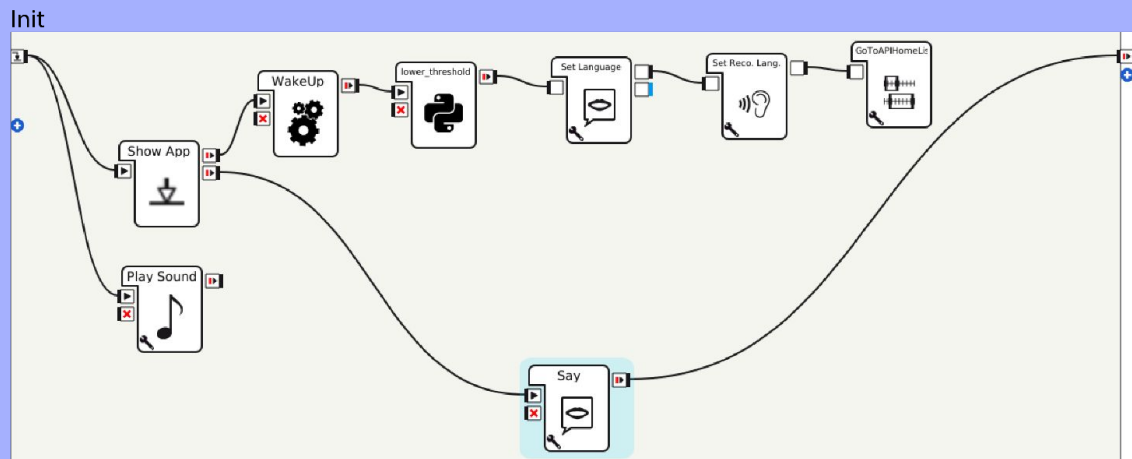
Robot Pepper

HTML / CSS / JS / Python

GoToAPIHomeList  
NAOQI



Serveur Robot  
Python



ESP32 TTGO - Camera  
C++ / PlatformIO

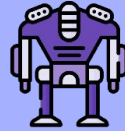


Utilisateur

# Pepper

APIHomeList

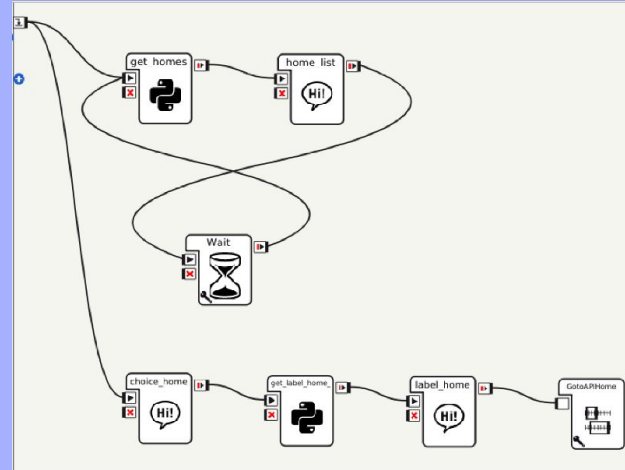
↑ Dire Choix Home  
Voix  
GoToAPIHome



Robot Pepper

HTML / CSS / JS / Python

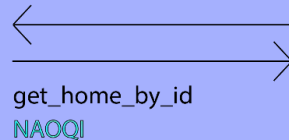
APIHomeList



Serveur Robot  
Python



get\_homes  
NAOqi

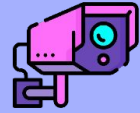


get\_home\_by\_id  
NAOqi

↑ Choix Home  
Toucher  
GoToAPIHome



Utilisateur

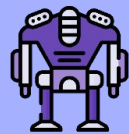


ESP32 TTGO - Camera  
C++ / PlatformIO

# Pepper

APIHome1

↑ Dire Reset Home  
Voix  
GoToAPIHomeList



Robot Pepper

HTML / CSS / JS / Python

↑ Dire Temperature / Humidity  
Voix

↑ Dire Create Action  
Voix

Image flux  
HTTP

APIHome

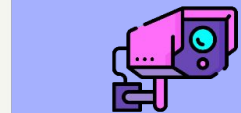
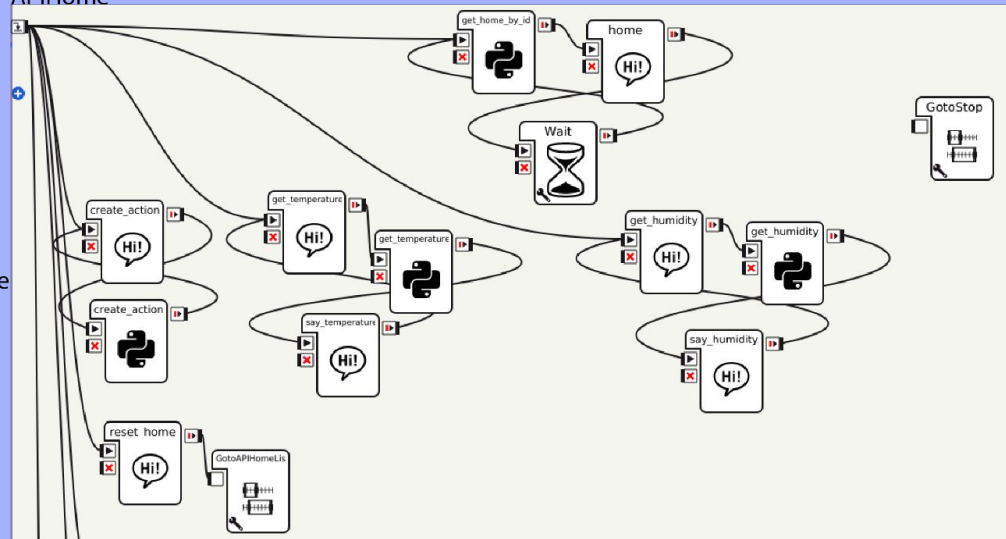
←  
get\_home\_by\_id  
NAOqi

←  
get\_last\_value\_  
by\_element\_name  
NAOqi



Serveur Robot  
Python

←  
create\_action  
NAOqi



ESP32 TTGO - Camera  
C++ / PlatformIO

↑ Reset Home  
Toucher / Voix  
GoToAPIHomeList



Utilisateur

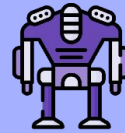
↑ Get Temperature / Humidity  
Voix

↑ Create Action  
Toucher



# Pepper

APIHome2



Robot Pepper

HTML / CSS / JS / Python

↑ Dire Update Mode  
Voix  
FeteMode / AbsenceMode

APIHome

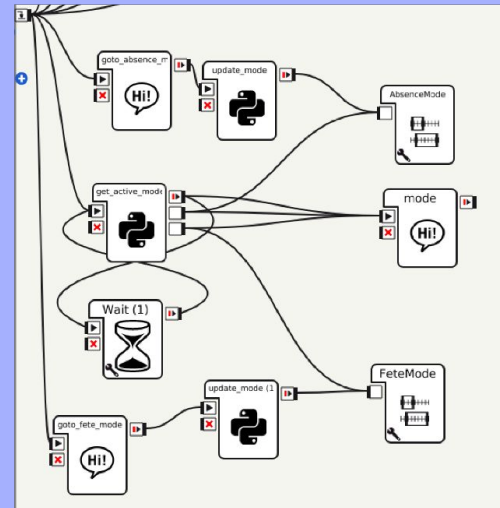
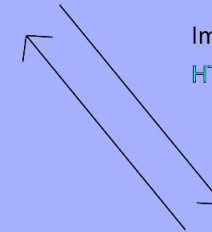


Image flux  
HTTP



ESP32 TTGO - Camera

C++ / PlatformIO



Serveur Robot

Python

← get\_active\_mode\_label\_  
by\_home\_name  
NAOQI →

← update\_mode  
NAOQI →

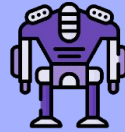


Utilisateur

↑ Update Mode  
Toucher  
FeteMode / AbsenceMode

# Pepper

ModeFete

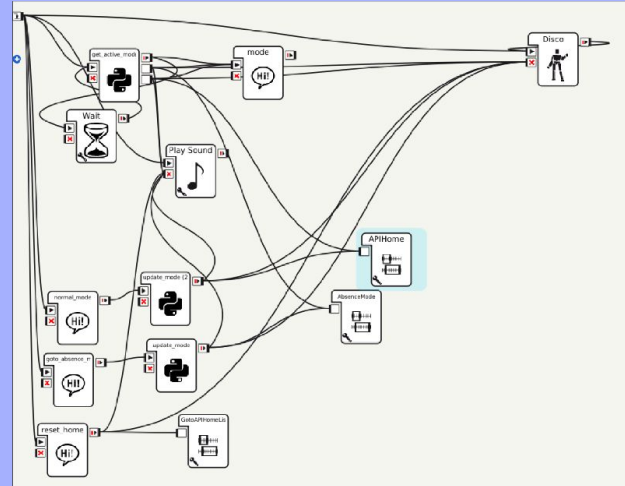


Robot Pepper

HTML / CSS / JS / Python

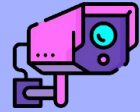
↑ Dire Update Mode  
Voix  
NormalMode / AbsenceMode

FeteMode



Serveur Robot  
Python

←  
→  
get\_active\_mode\_label\_  
by\_home\_name  
NAOQI  
←  
→  
update\_mode  
NAOQI



ESP32 TTGO - Camera  
C++ / PlatformIO



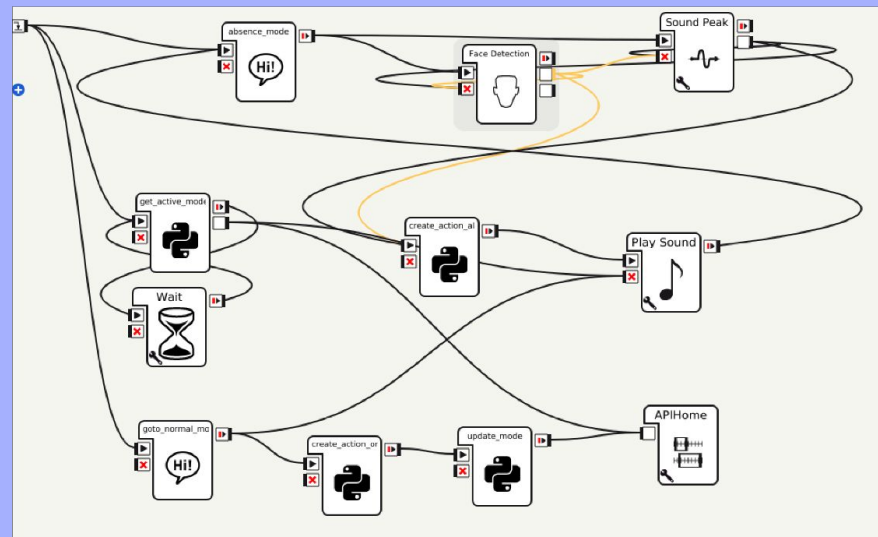
Utilisateur

↑ Update Mode  
Toucher  
NormalMode / AbsenceMode

## AbsenceMode



↑ Dire Update Mode  
Voix  
NormalMode / AbsenceMode



```
get_active_mode_label_  
by_home_name
```

NAOQI

```
update_mode
NAOOI
```

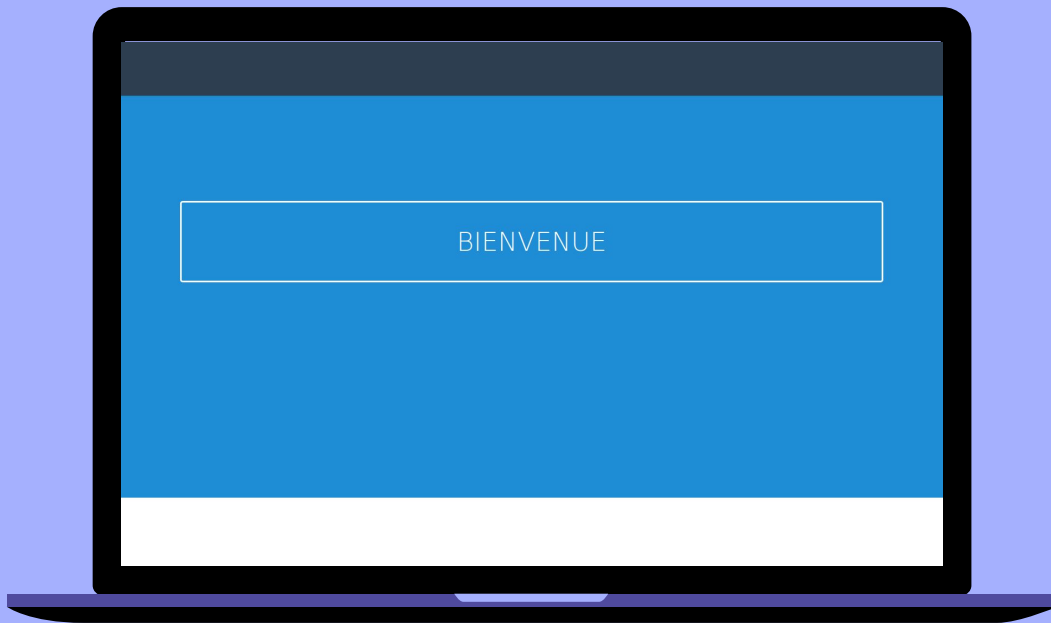
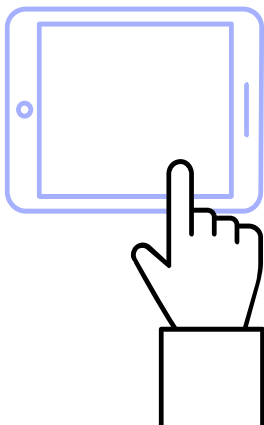


↑ Update Mode  
Toucher  
NormalMode / AbsenceMode

# Robot Pepper

## Dashboard

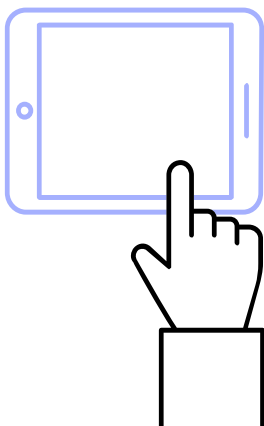
- ▷ HTML/CSS/JS
- ▷ Flux de la caméra
- ▷ Choisir Home
- ▷ Monitorer Home
- ▷ Choisir Mode



# Robot Pepper

## Dashboard

- ▷ Flux de la caméra
- ▷ Choisir Home
- ▷ Monitorer Home
- ▷ Choisir Mode



CHOIX DU HOME

HOME

MODE ABSENCE

ROBOT DE PEPPER

RETOUR MODE NORMAL

HOME: HOME

RESET

"MODE ABSENCE" "MODE FÊTE" "MODE NORMAL"

TEMPÉRATURE CLIMAT TEMPS "QUELLE TEMPS FAIT-IL"

HUMIDITÉ FRAICHEUR HYGROMÈTRE "QUELLE HUMIDITÉ FAIT-IL"

MODE ACTUEL: NORMAL

MODE NORMAL MODE FÊTE MODE ABSENCE

ROOM: LIVING\_ROOM

Element: sensor\_temperature\_living\_room

Etat Actuel	Value	Type	Action	#
OFF	0.0 le 17/12/2022 à 16:42	sensor	• Etat: Dnsat • Value: 10	<input type="text"/> Valider

Element: sensor\_humidity\_living\_room

Etat Actuel	Value	Type	Action	#
OFF	0.0 le 17/12/2022 à 16:35:4	sensor	• Etat: NA • Value: NA	<input type="text"/> Valider

Element: sensor\_luminosity\_living\_room

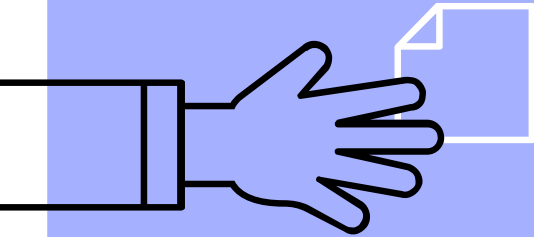
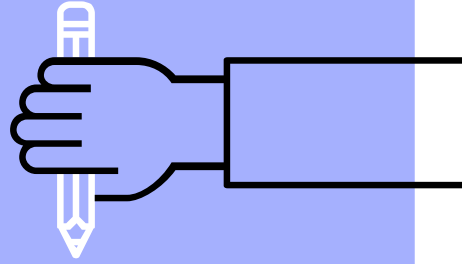
Etat Actuel	Value	Type	Action	#
OFF	0.0 le 17/12/2022 à	sensor	• Etat: NA • Value: 10000	<input type="text"/> Valider

Element: sensor\_entry\_door\_authentication\_mdq\_living\_room

Etat Actuel	Value	Type	Action	#
OFF	NA	sensor	• Etat: NA • Value: NA	<input type="text"/> Valider

## 2. Activités réalisées

Serveur Global



# Serveur Global

## Accès à l'API

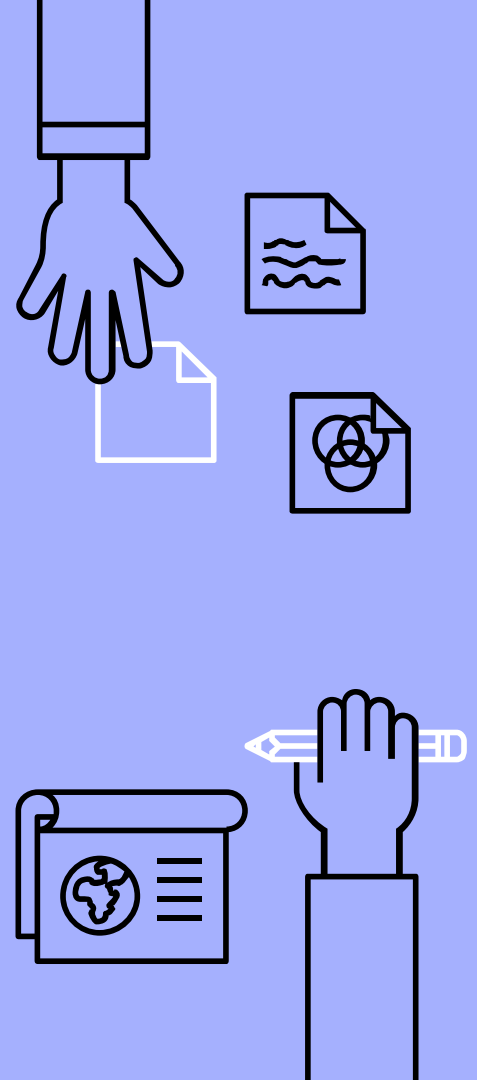
Accède à l'API via  
Python :

- ▶ Mise à jour des valeurs
- ▶ Récupération des valeurs

## Supervise les microcontrôleurs

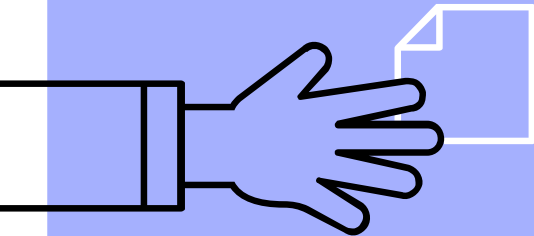
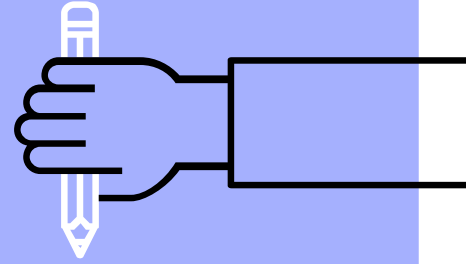
Avec MQTT, il supervise  
les microcontrôleurs :

- ▶ Récupération des valeurs (mise à jour en BDD)
- ▶ Assignment d'action (ouverture porte...)



## 2. Activités réalisées

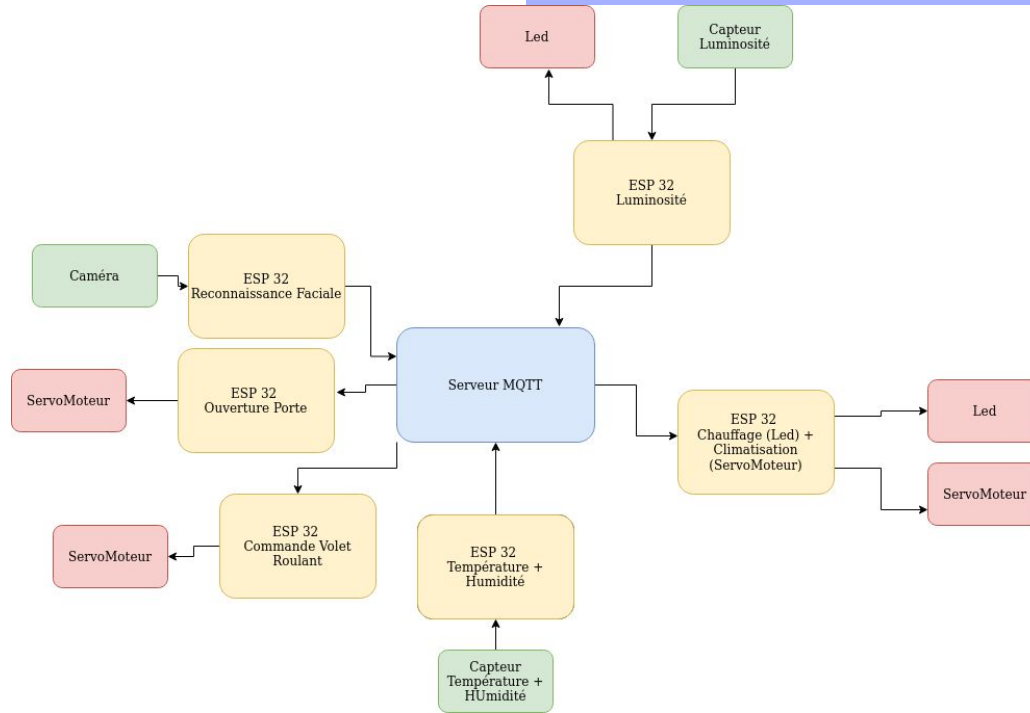
IOT





# IOT

## Schéma global



# IOT - ESP32 CAM + Python

## Serveur WEB - Wifi

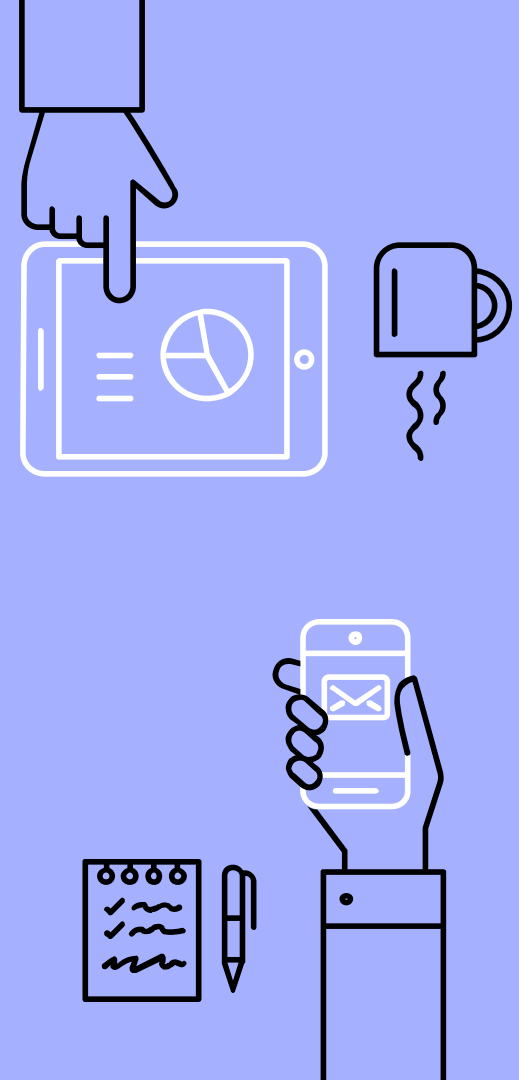
- Se connecte au Wifi.
- Serveur WEB permettant de partager le flux de la caméra.

## Ecran

Affichage sur l'écran du statut de la connection au Wifi.

## Python

- Récupère le flux de la caméra et check si il y a un visage et s'il est autorisé
- Ajouter des visages autorisés
- Notifit en MQTT



# IOT - Module Verrou

## Servomoteur

Contrôle du servomoteur grâce à un signal PWM.

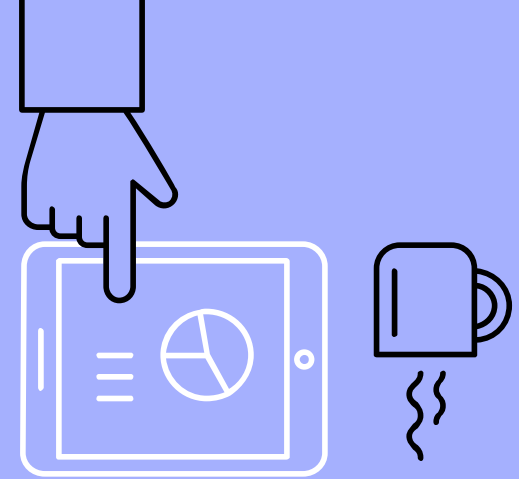
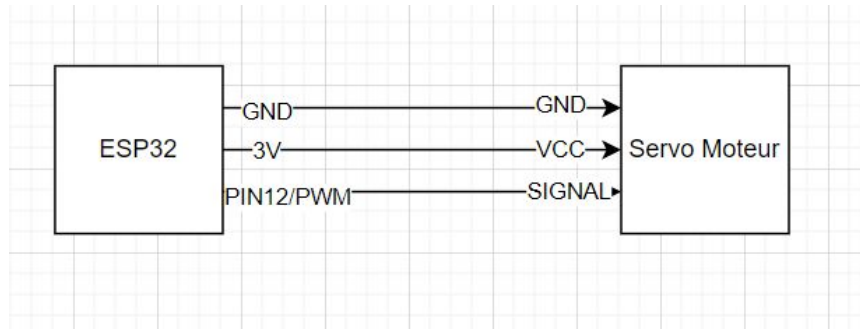
## MQTT

Publish et subscribe :

- ▶ Un topic pour acquitter
- ▶ Un topic pour écouter les ordres.

## Ecran

Affichage sur l'écran du statut de la connection au Wifi + MQTT ainsi que l'état du verrou.



# IOT - Module VMC

## Servomoteur

Contrôle du servomoteur grâce à un signal PWM.

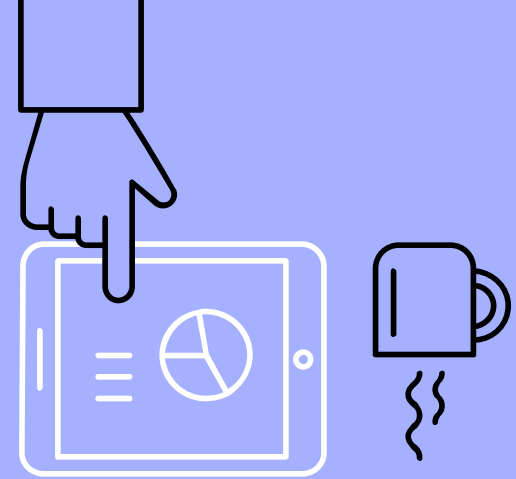
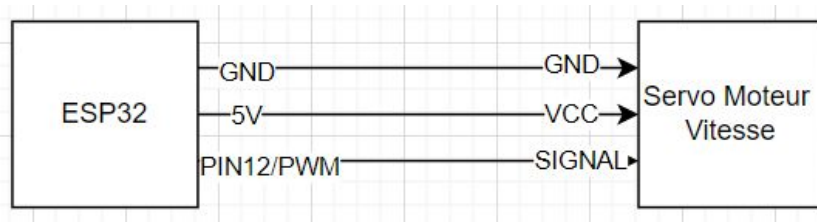
## MQTT

Publish et subscribe :

- ▶ Un topic pour acquitter
- ▶ Un topic pour écouter les ordres.

## Ecran

Affichage sur l'écran du statut de la connection au Wifi + MQTT ainsi que l'état de la VMC.

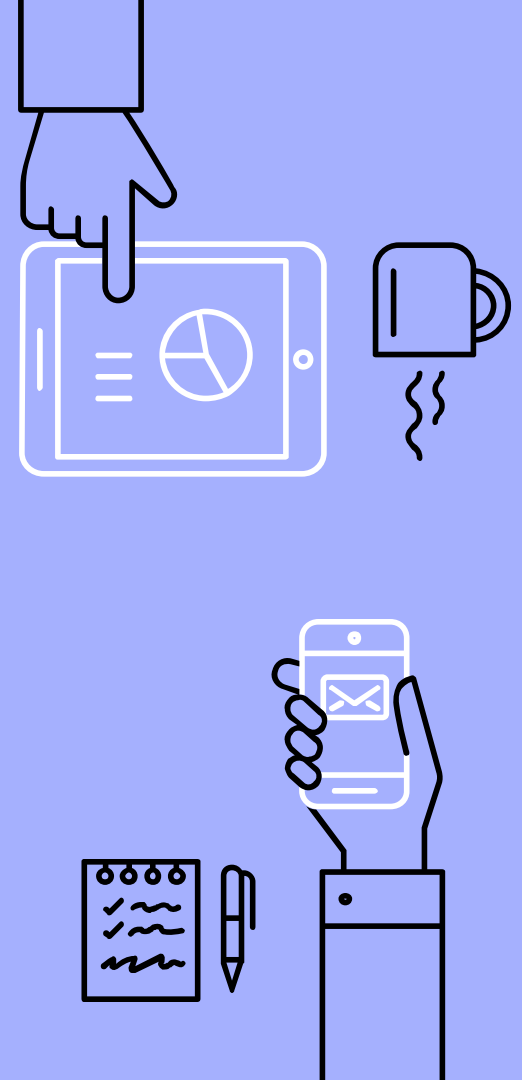


# IOT - Module Digicode

test  
test

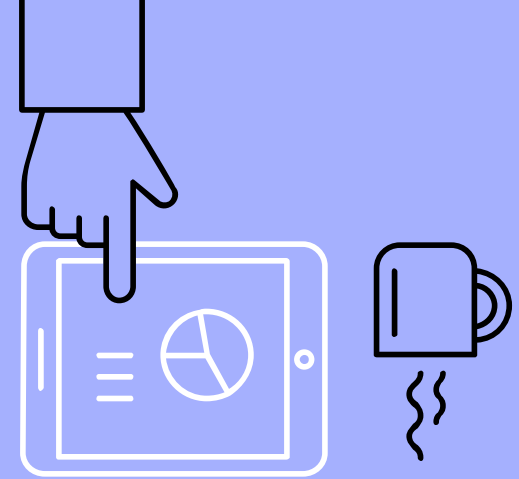
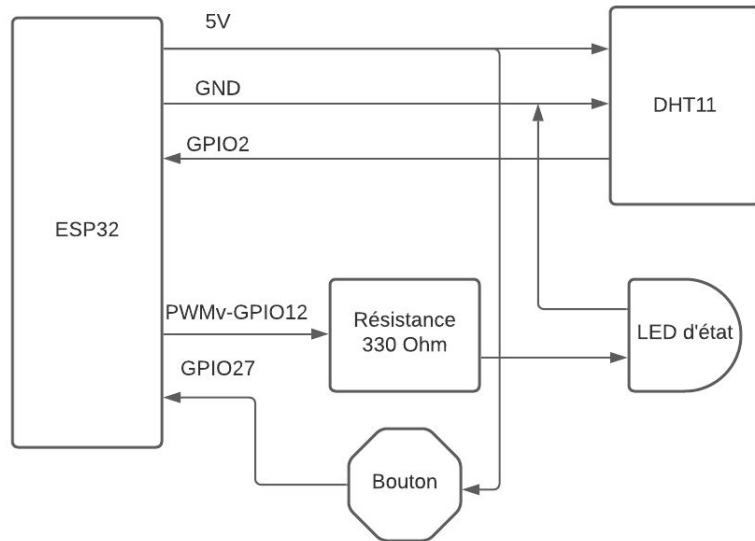
test  
test

test  
test



# IOT - Module Temp/Hum

Schema du Module Temperature/Humidite

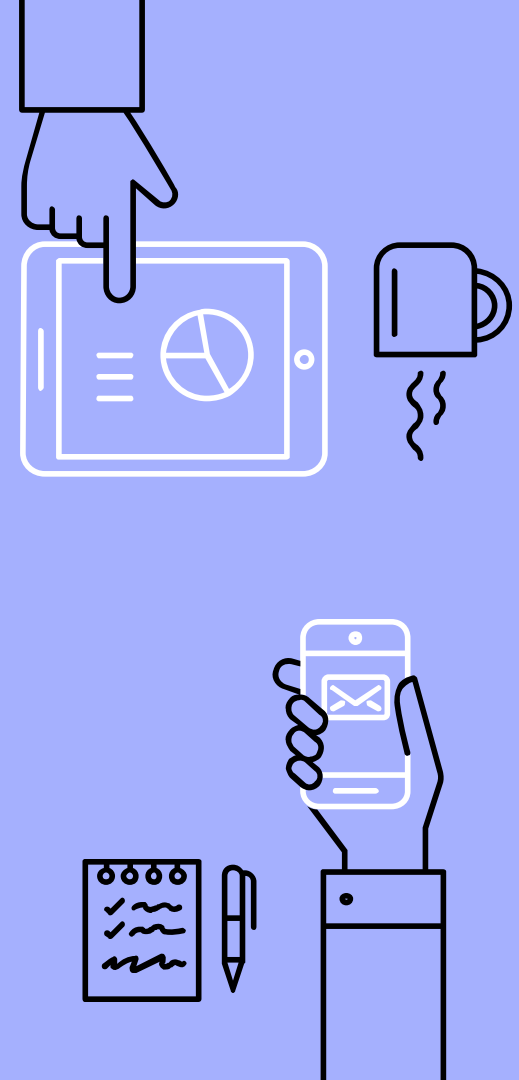


# IOT - Module Luminosité

test  
test

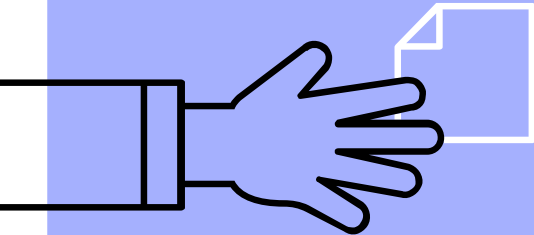
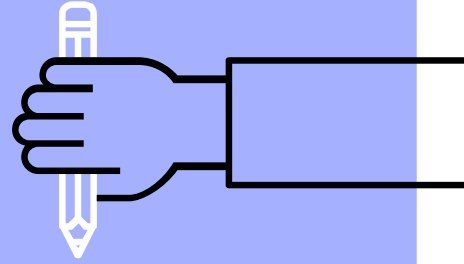
test  
test

test  
test



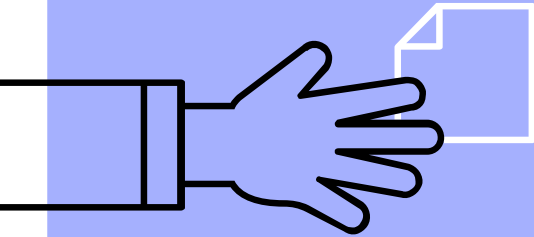
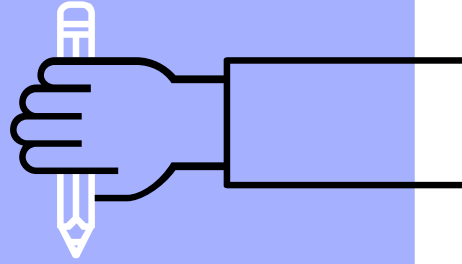
# 3. Démonstration

Démonstration du projet





# 4. Conclusion



# Fonctionnalités de base

- ▷ Gestion de la température de chaque pièce **Jalon 1**
- ▷ Gestion de la luminosité de chaque pièce **Jalon 1**
- ▷ Contrôle de l'ouverture de la porte d'entrée par détection faciale **Jalon 2**
- ▷ Consultation en temps réel de l'état des équipements connectés **Jalon 1**
- ▷ Gestion de la maison via Pepper **Jalon 1**

# Liste des éléments de base de la pièce

- ▷ VMC **Jalon 1**
- ▷ Température **Jalon 1**
- ▷ Humidité **Jalon 1**
- ▷ Volet **Évolution Future**
- ▷ Porte **Jalon 1**
- ▷ Prise **Évolution Future**
- ▷ Luminosité **Jalon 2**
- ▷ Chauffage **Jalon 2**



# Évolutions possibles

- ▶ Mode préprogrammé
  - Surveillance : détecte les mouvements dans la maison et effectue une notification **Jalon 2**
  - Fête : met de la musique et les lumières de la pièce clignotent. **Jalon 2**
- ▶ Application Android de gestion **Évolution Future**
- ▶ Dashboard de visualisation **Jalon 2**
- ▶ Intégration Google Home **Évolution Future**
- ▶ Ajout de code de vérification/confirmation pour l'ouverture de la porte d'entrée **Jalon 2**
- ▶ Cryptage des données / Sécurisation du serveur **Évolution Future**
- ▶ Contrôle des prises de la maison **Évolution Future**
- ▶ Intégration d'un capteur luminosité **Jalon 2**
- ▶ Dashboard PowerBi **Jalon 2**

# Merci

Des questions?

