ECOLE SUPERIEURE
D'INGENIEURS DE RENNES

UNIVERSITÉ DE
RENNES 1

Chenillard

Projet RESIoT

ESIR2 | IoT

Mathis Certenais | Matéo Fontanel

01 **Objectifs**

Connexion serveur
Connexion client

02 **Architecture**

03 **Client**

Frontend
Backend
Application mobile

04 **Serveur**

05 **Démonstration**

06 **Conclusion**

01

Objectifs

Connexion serveur
Connexion client

01 Objectifs

Principe :

Commander une maquette KNX disposant de 4 LED depuis la maquette avec 4 boutons, le site internet et l'application web

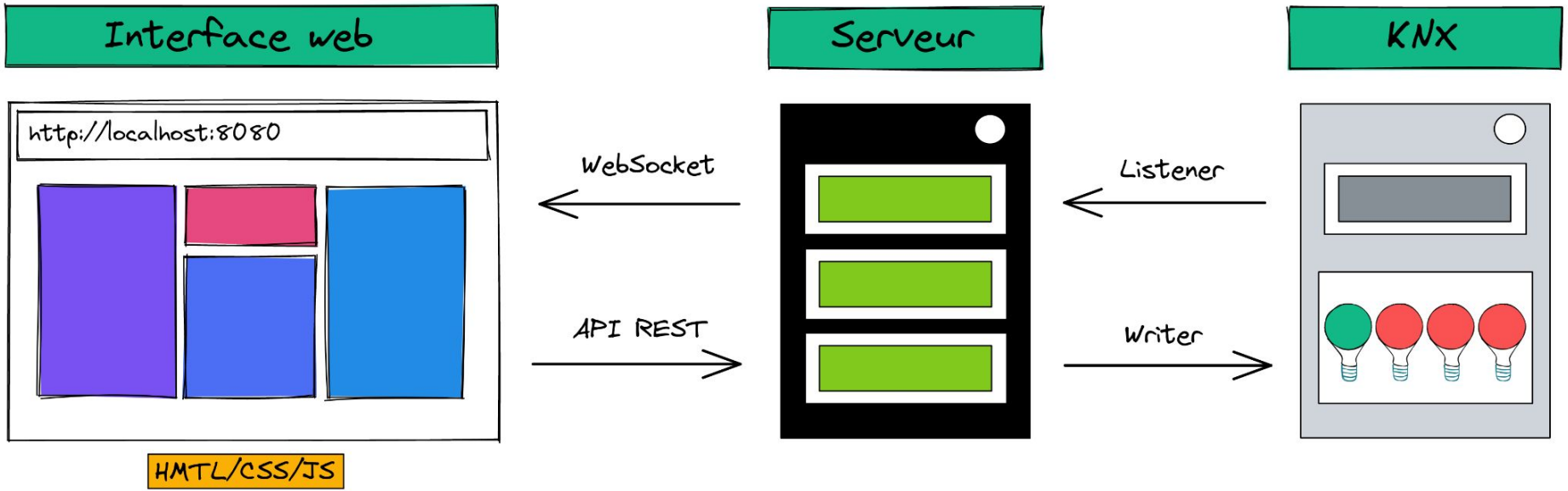
Etapes du projet :

- Réalisation de l'architecture
- Choix des langages de programmation
- Création d'un serveur
- Création d'une interface web pour commander la maquette KNX
- Communication client - serveur
- Communication serveur - maquette KNX

02

Architecture

02 Architecture - Vue d'ensemble



03

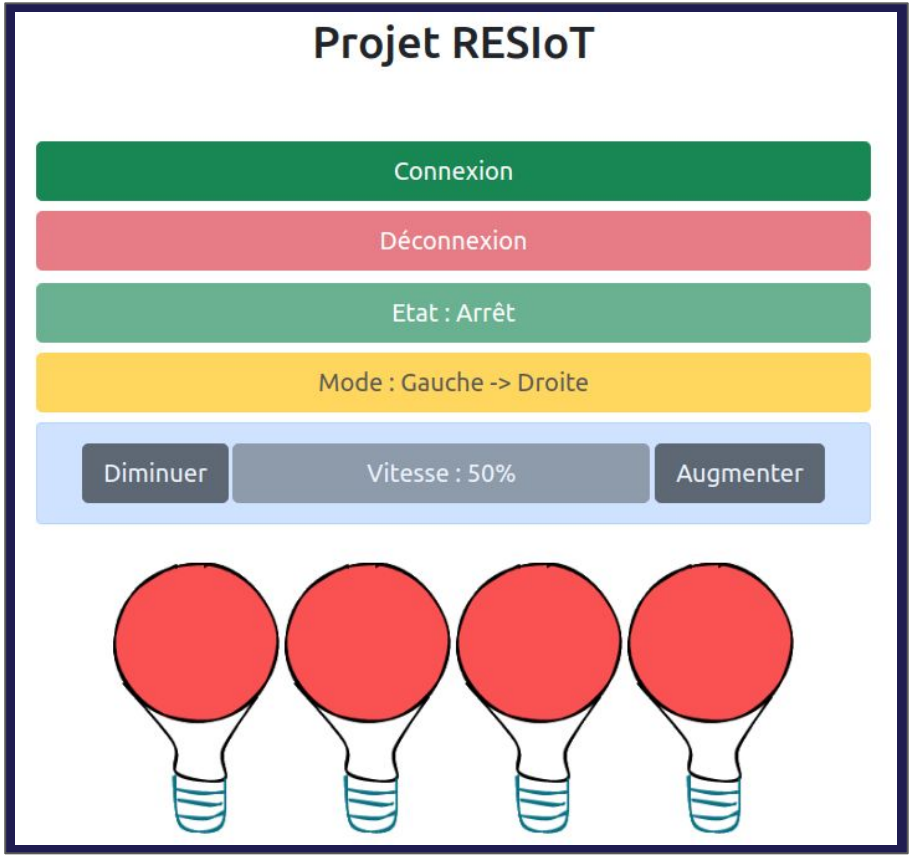
Client

Frontend
Backend

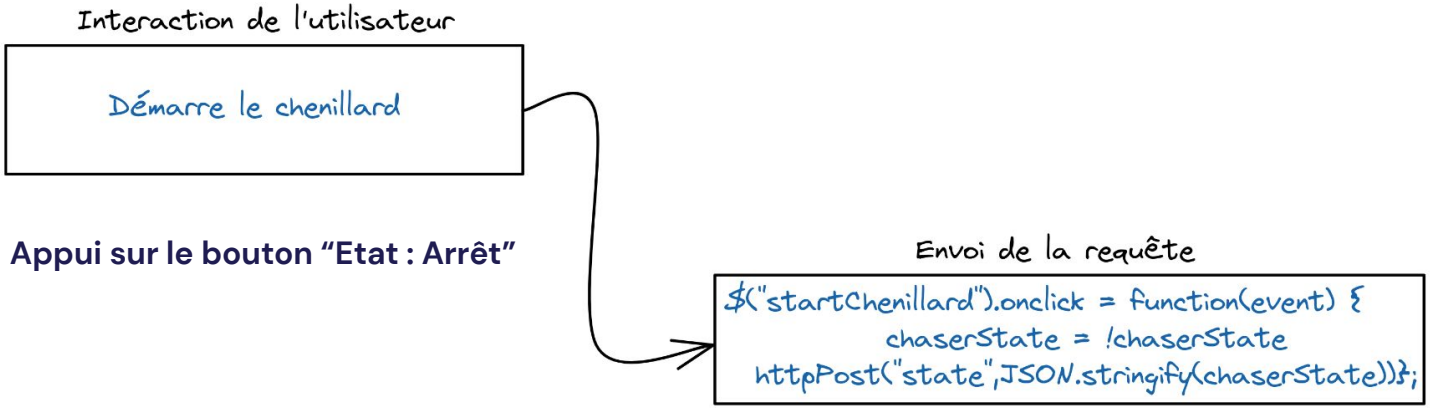
03 Client - Frontend

Fonctionnalités principales :

- Connexion/Déconnexion de la maquette
- Allumer/Éteindre les LED
- Changer le mode de fonctionnement (chenillard de gauche vers la droite et inversement, mode flipper et random)
- Augmenter/Diminuer la vitesse



03 Client - Requêtes



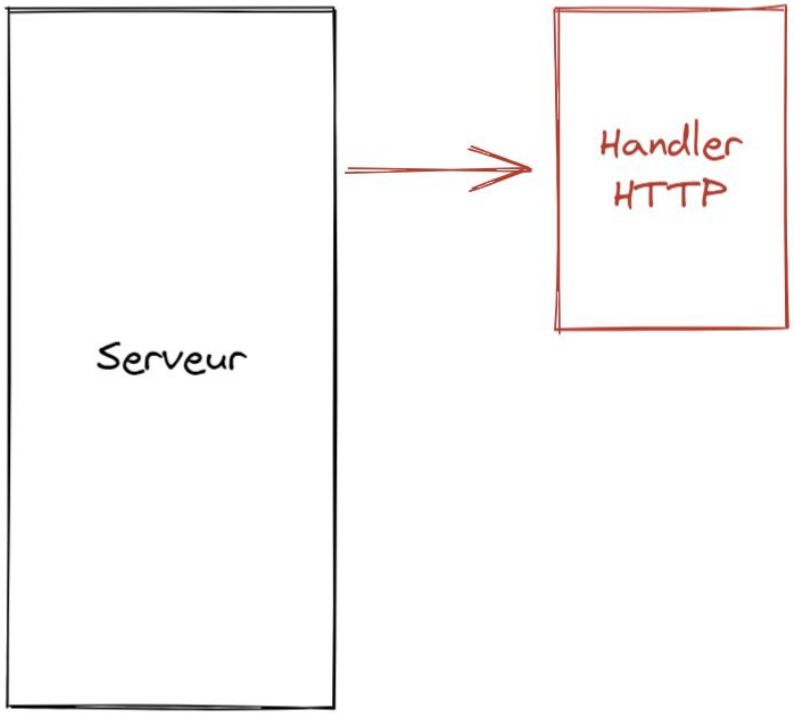
04

Serveur

Communication avec le client
Gestion du chenillard
Communication avec le KNX

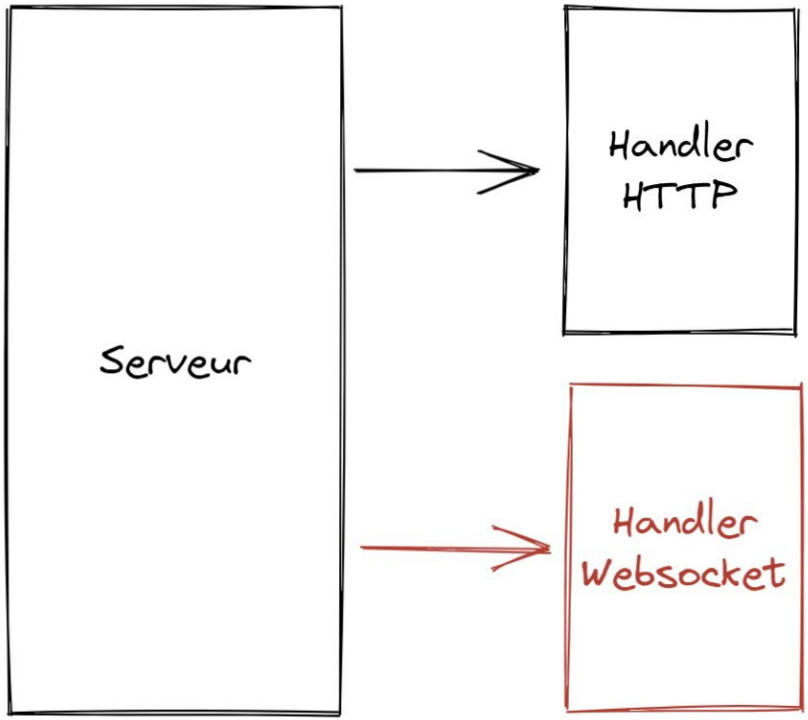
Serveur – Communication avec le client

Serveur – Communication avec le client



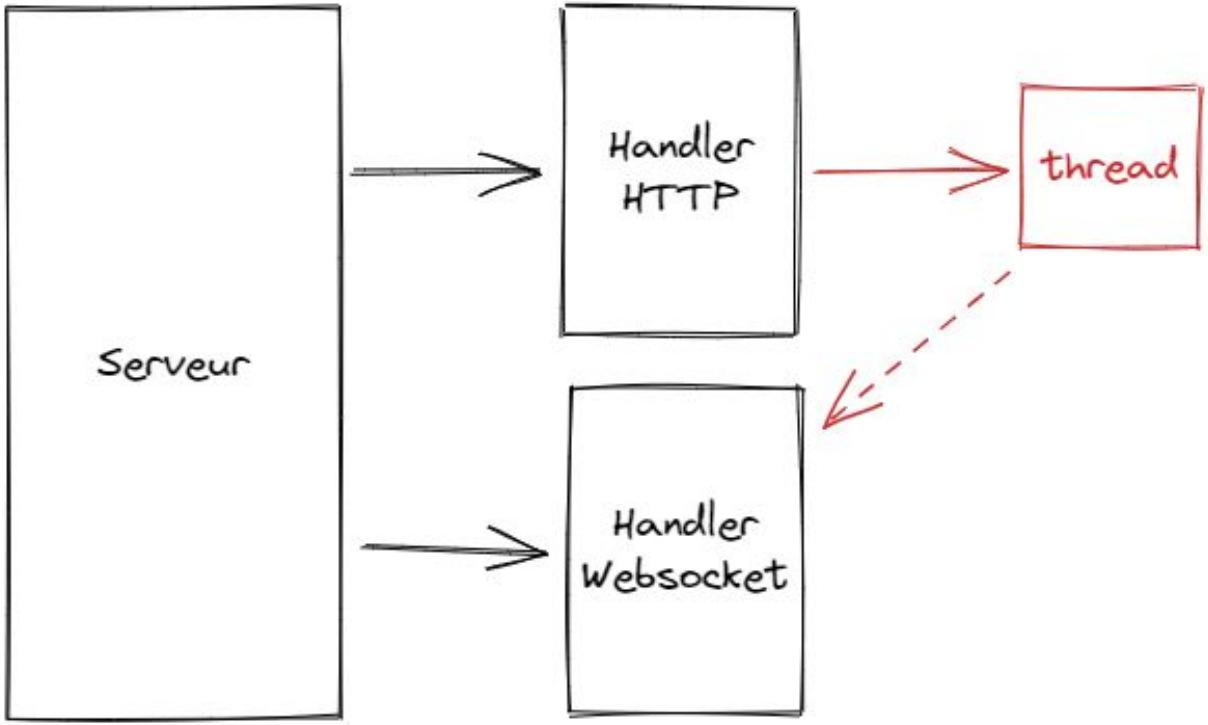
- Implémentation du handler de Jetty
- Appelé à chaque requête POST
- Routes /etat, /sens et /vitesse
    - Actions sur le chenillard selon la route et le corps de la requête

Serveur – Communication avec le client



- Implémentation du handler de Java
- Connexion persistante
- Utilisation unidirectionnelle (serveur → client)

Serveur – Gestion du chenillard



Serveur – Gestion du chenillard

- Boucle infinie du thread où chaque itération change l'état des LEDs
- Mémorise les paramètres du chenillard :
  - Etat (marche/arrêt)
  - Sens (gauche → droite, droite → gauche, gauche ↔ droite, aléatoire)
  - Vitesse (250 ms de délai à 100%)

1) Attente ou action (selon l'**état** du chenillard)
2) Récupération des LEDs allumées à l'itération précédente
3) Détermination des LEDs à éteindre ou allumer (selon le **sens** du chenillard)
4) Envoi des informations au KNX
5) Attente avant l'itération suivante (selon la **vitesse** du chenillard)

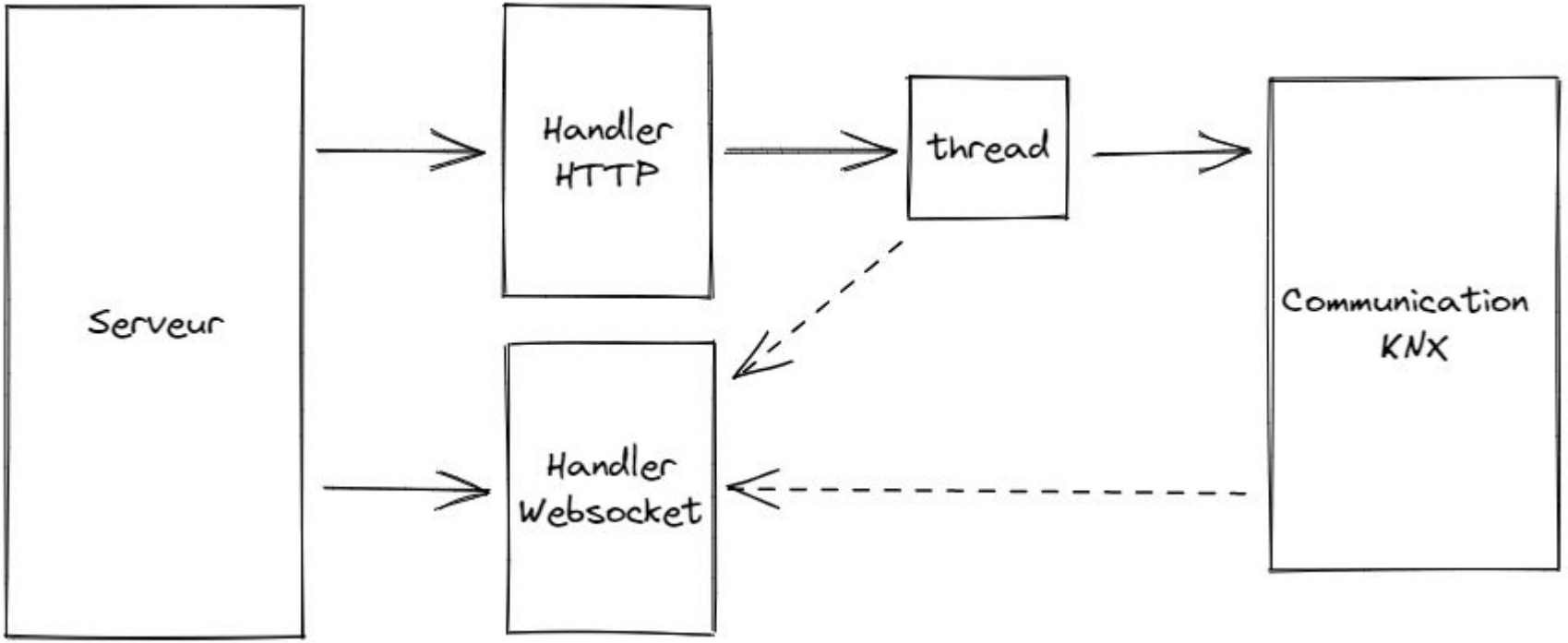
Serveur – Communication avec le KNX

- Bibliothèques Calimero

- Connexion à la maquette via IP + port

- Communication bidirectionnelle
  - Depuis le serveur
    - Méthode ProcessCommunicator.write(groupe, valeur)
  - Depuis la maquette
    - Appui sur un interrupteur : action sur les paramètres du chenillard
    - Changement d'état d'une LED : action sur l'affichage côté client

Serveur – Architecture



05

Démonstration

06

Conclusion