

ÉCOLE CENTRALE DE LILLE

PROJET D'INTÉGRATION

---

# Suivi d'un grimpeur par caméra dome

---

RAPPORT DE PDI  
18 mars 2025



CHEVALIER Romain

*Tuteur* : BOURDEAUD'HUY T.



## Table des matières

<b>1</b>	<b>Contexte</b>	<b>2</b>
1.1	Objectifs . . . . .	2
<b>2</b>	<b>Architecture du système</b>	<b>2</b>
2.1	Architecture matérielle . . . . .	2
2.2	Architecture logicielle . . . . .	3
<b>3</b>	<b>Développement de l'application</b>	<b>4</b>
3.1	Développement du serveur web . . . . .	4
3.1.1	Page web Index . . . . .	4
<b>4</b>	<b>Tests et résultats</b>	<b>5</b>
<b>5</b>	<b>Perspectives d'amélioration</b>	<b>5</b>



## 1 Contexte

Afin d'améliorer les services offerts aux utilisateurs de la salle d'escalade *Sac à Pof* situé à Mons-en-Barœul (59), on envisage de mettre en place un système de suivi automatisé des grimpeurs à l'aide de caméras IP dômes motorisées. Ces caméras permettront de suivre les mouvements des grimpeurs sur les différentes voies en ajustant automatiquement le cadrage et le zoom en fonction de leur position. Le service sera accessible via une application web hébergée sur un serveur local. Cette solution vise à offrir un support d'analyse post-session, afin de permettre aux grimpeurs de comprendre leurs forces et faiblesses pour s'améliorer.

Afin de réaliser ce projet, nous sommes en possession de deux caméras IP dômes motorisées de la marque *AXIS* modèle *Q6304-E*<sup>1</sup>, d'un routeur, d'une tablette Android et d'un raspberry pi 4.

### 1.1 Objectifs

Les objectifs de ce projet sont multiples et sont résumés dans la liste suivante :

- Prendre en main les caméras IP dômes motorisées et les intégrer dans un réseau local.
- Développer une interface web permettant de piloter les caméras, l'enregistrement de vidéo et leur visualisation.
- Avoir un retour visuel de la caméra en temps réel sur l'interface web.
- Développer une méthode de communication entre l'application web et les programmes de contrôle des caméras.
- Développer un algorithme de suivi de grimpeur en temps réel.
- Développer une méthode permettant d'enregistrer sur un stockage une vidéo.
- Avoir une application modulable permettant de modifier les caméras et les paramètres de détection.

## 2 Architecture du système

### 2.1 Architecture matérielle

Le système est composé de plusieurs éléments qui interagissent entre eux, la figure 1 montre l'architecture du système. Le système repose sur un **Raspberry Pi 4**, qui joue un rôle central en assurant la communication entre les différentes composantes. Il est connecté au réseau local via un **routeur**, qui agit comme un serveur DHCP, attribuant dynamiquement les adresses IP aux appareils du système.

Les **caméras Dome IP** sont accessibles via HTTP et envoient leur flux vidéo vers le Raspberry Pi lorsqu'elles sont activées. Ces caméras disposent de positions prédéfinies correspondant aux différentes voies d'escalade, mais peuvent également être orientées dynamiquement par l'utilisateur via la **tablette** fournie ou tout autre appareil connecté au réseau local.

Le Raspberry Pi exécute un serveur d'application web, ce serveur permet :

- de piloter les caméras (changement de position, orientation manuelle),
- de gérer les enregistrements vidéo,

---

1. Voir [fiche technique](#) pour plus d'informations



- de stocker dans une base de données les vidéos enregistrées ainsi que les positions des caméras définies par l'utilisateur.

L'enregistrement des vidéos est déclenché manuellement depuis l'interface de la tablette et les fichiers sont conservés jusqu'à suppression par l'utilisateur.

L'architecture ainsi mise en place assure une gestion centralisée et accessible du système, tout en permettant un contrôle précis et flexible des caméras et des enregistrements.

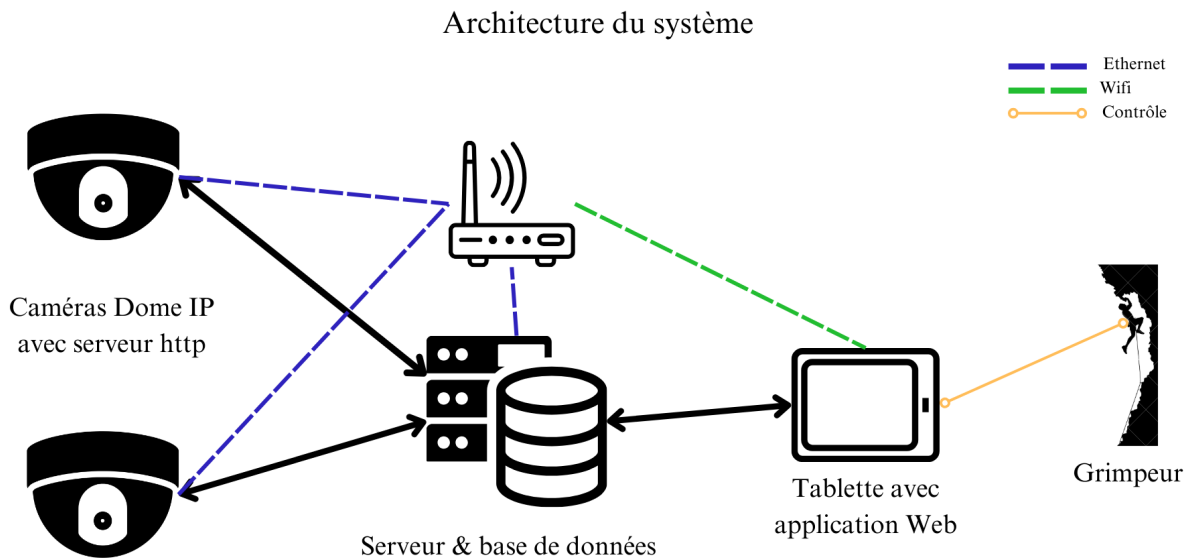


FIGURE 1 – Architecture du système

## 2.2 Architecture logicielle

L'architecture logicielle du système est représentée sur la figure 2. Le raspberry Pi divise ses tâches en deux parties distinctes : le serveur web et la partie gestion de l'application développée. En fonction des actions effectuées sur la page web, le serveur apache exécuté deux scripts `cgi` différents, le premier script, nommé **video.cgi** codé en `c++`, démarre au chargement de la page permet d'afficher le flux vidéo de la caméra sur la page web en lisant les images brutes dans la mémoire partagée et en les encodant en JPEG. Le second (**action.cgi** développé en `c`) permet de transmettre les ordres de déplacement de la caméra ou lancer un enregistrement vidéo via l'écriture de cet ordre dans un segment de mémoire partagé puis de réveiller le programme principal en lui envoyant un signal.

De l'autre côté, le programme principal (**main** programmé en `c`) est en charge de la gestion des caméras, de la détection de mouvement et de l'enregistrement des vidéos. Au démarrage, il crée un processus fils (**écriture mémoire**) qui est en charge de récupérer le flux vidéo du serveur HTTP de la caméra et de le stocker en image brute dans une mémoire partagée. Le programme principal est ensuite en attente d'un signal pour effectuer une action, comme le déplacement de la caméra, l'enregistrement de position ou encore le lancement d'un enregistrement qui va créer un processus fils (**enregistrement vidéo**) qui encode le flux vidéo en MP4 afin de le stocker sur le disque dur et lance la détection du grimpeur pour suivre ces mouvements à l'aide d'un second fils (**détection**). Ces deux



processus récupèrent les images en lisant la mémoire partagée et les traitent pour effectuer leur tâche. Enfin, le programme de détection peut directement agir sur la position de la caméra en lui envoyant une requête HTTP.

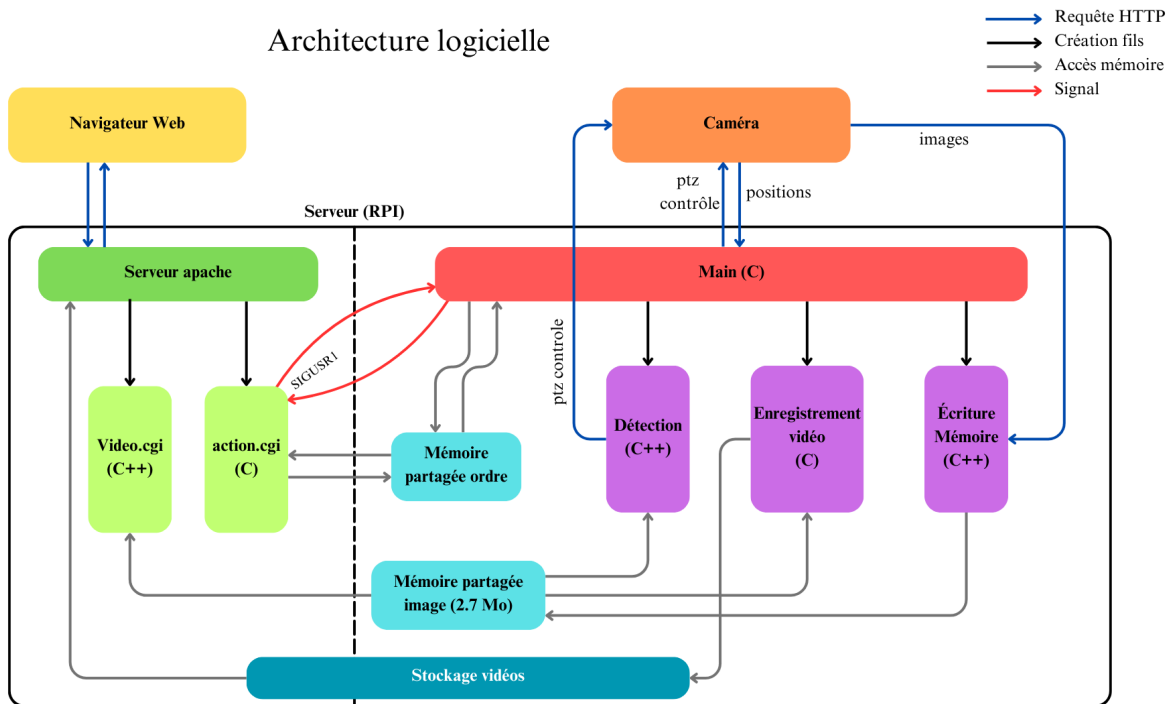


FIGURE 2 – Architecture logicielle

### 3 Développement de l'application

Dans cette partie, nous allons détailler les différentes étapes de développement de l'application pour chaque brique du système présent dans l'architecture logicielle en figure 2.

#### 3.1 Développement du serveur web

Le serveur web est composée de deux pages web, la première permet de visualiser le flux vidéo de la caméra, de la piloter et de piloter l'application. La seconde permet de visualiser les vidéos enregistrées et de les télécharger, le code de ces pages est écrit en HTML, CSS et JAVASCRIPT peut-être consulté sur la page [GitHub](#).

##### 3.1.1 Page web Index

Au chargement de cette page, une fonction **javascript** permet de charger les voies disponibles dans la base de données et de les ajouter dans une liste déroulante pour permettre à l'utilisateur de sélectionner la voie qu'il souhaite visualiser. Une seconde fonction **js** charge les caméras disponibles

(a) Index de l'application

(b) Page web de visualisation des vidéos

FIGURE 3 – Pages web de l'application



et les ajoute dans une seconde liste déroulante. Puis le script `video.cgi` est exécuté pour afficher le flux vidéo de la caméra sélectionnée.

Le processus lit dans la mémoire partagée les images brutes à l'aide d'OPENCV pour récupérer une matrice de pixel, puis le profil colorimétrique de l'image est changé car OPENCV travaille en *BGR* (*blue green red*) et le JPEG utilise un profil (*RGB* (*red green blue*)) puis est encodé en JPEG à l'aide de la librairie `libjpeg`. Enfin, on ajoute les entêtes HTTP pour que le navigateur puisse interpréter l'image et on écrit l'image encodée dans la sortie standard.

Sur la gauche de cette page, on retrouve tout les boutons de contrôle de la caméra (haut, bas, gauche et droite) associé à un curseur pour la précision de l'angle afin de permettre un déplacement plus précis. On retrouve un second curseur pour gérer le zoom de la caméra entre 1x et 18x. La pression de l'un de ses boutons créent une requête AJAX pour exécuter le script `action.cgi` qui va écrire dans la mémoire partagée l'action à effectuer puis réveiller le programme principal pour qu'il effectue l'action et attendre le feedback du main ou un timeout de 5 secondes avec l'instruction `SIGALRM` (il récupère le pid du main dans un fichier que ce dernier crée au démarrage). Finalement, il envoie une réponse HTTP pour indiquer que l'action a bien été effectuée.

En dessous de ces curseurs, il y a une liste déroulante pour choisir la voie que l'on souhaite observer. À la selection d'une voie on vient changer de caméra si besoin et la positionner au bon endroit.

En dessous de la video, on retrouve un bouton pour enregistrer une vidéo avec deux champ pour entre le nom et prenom du grimpeur, un exemple de requête AJAX est présenté ci-contre : `http://localhost/cgi-bin/action.cgi?enrg=on&nom=Chevalier&prenom=Romain&voie=0`. On récupère le nom et la voie afin de produire un nom de fichier unique et pouvoir le récupérer facilement dans la page de visualisation des vidéos.

**Paramètres avancés** En bas de page, on trouve une liste déroulante pour choisir la

## 4 Tests et résultats

## 5 Perspectives d'amélioration

- Amélioration de l'interface Web, avoir une interface plus intuitive et plus ergonomique.
- Reformater le code et tout traduire en anglais
- Afficher sur l'écran du Raspberry Pi, un meilleur status de l'exécution de l'application.
- Utiliser les boutons du Raspberry Pi pour relancer l'application.
- Ajouter un système de notification pour les erreurs.
- Mieux gérer le stockage des vidéos.
- Ajouter un système de détection de chute.