**Simulator** :

Rôle : créer des incendies, gérer leur propagation en fonction des secours présents sur place

Actions : créer un incendie, diminuer ou augmenter l’intensité, détecter la présence des secours, persister les incendies simulés dans sa base de données

Langage : Java (obligé LOL)

Communique avec sa base de données directement (JDBC)

**Simulator web server**

Rôle : Expose la base de données du Simulator à d’autres programmes (Sensors) et propose éventuellement une interface web permettant l’interaction avec le simulator.

Actions : récupérer les informations d’incendies et la position des camions depuis la base de simulation. Servir une page web

Langage : PHP, JavaScript

Communique directement avec BD du simulateur via PDO

**Sensors**

Rôle : Récupérer les informations des incendies via le simulator web server et les envoyer par ondes radio au bloc data collect

Actions : demander à une fréquence définie l’état des différents capteurs simulés, envoi des informations en cas de changement d’état des capteurs

Langage : Python

**Data collect**

Rôle : réception des données transmises par sensor pour publication sur un serveur MQTT (souscription des blocs Dashboard et Emergency web server)

Actions : réception des données, envoi des données sur serveur MQTT

Langage : Python

**Emergency web server|**

Rôle : enregistrer les données de data collect dans la base d’emergency manager, présenter la carte aux pompiers, récupérer les informations de emergency manager et les retourner à data collect

Langages : PHP, JavaScript

Connecté directement à la BDD de emergency manager via PDO

**Emergency manager**

Rôle : Alloue les ressources nécessaires en fonction des données récupérées dans sa base de données remplie par son web server. Sauvegarde des ressources allouées dans la base.

Langage : Java

**Dashboard**

Rôle : collecter et afficher des statistiques sur les incendies ayant eu lieu

Est abonné au sujet MQTT de data collect, et enregistre les données récupérées dans sa base de données.

Langage : Python

**Communication de retour d’emergency à simulator**

Emergency -> son web server –> data collect -> sensors –> simulator web server -> simulator