Stela Kucek, 01303858

Cloud Computing: Assignment 1c

Documentation

RESTful services for airport security

Used technologies

Programming language	Python
Framework	Flask

Overview

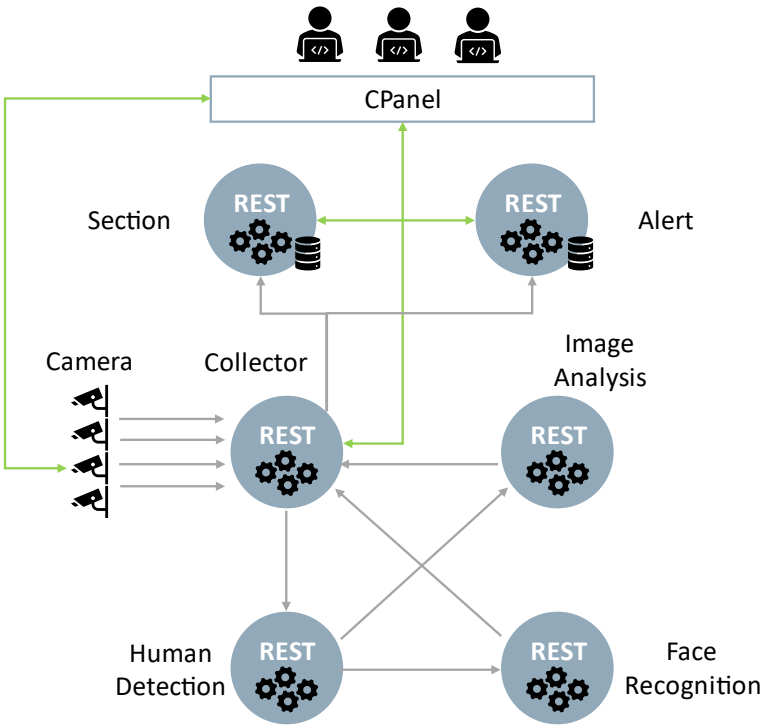

Figure 1 System overview with communication flows

Discussion

- System configuration?

- The system is configured using Docker Compose to ease and automate the building and starting processes.
- The camera service image is used to build 2 distinct instances (containers), one of which is initially configured to represent the "entry" camera, and the other the "exit" camera.
- Each of the other services is run with one instance, which should suffice for a discussion basis and initial testing such as in this task.
- The CPanel is designed and implemented as a GUI (using html) for easier and more user-friendly access and display of the functionality.

- Service discovery?

- For this task, no sophisticated service discovery technique/technology is utilized, but rather the services have preconfigured container names and are all attached to the same Docker network, therefore they communicate via addresses such as http://<container-name>:80/<...>

- Communication / Data flow?

- Using Figure 1 as the system overview, the data flow can be described as follows:
    1. The Camera starts streaming and sends a frame to the Collector
    2. The Collector automatically forwards the frame to the Human Detection service in order to determine whether there are people captured on the image
    3. The Human Detection service then, if it there are any persons detected, does the following:
        a. Takes a copy of the original frame and sets the 'destination' field to '<collector-URL>/persons' and forwards the frame to the Image Analysis service
        b. Takes a copy of the original frame and sets the 'destination' field to '<collector-URL>/known-persons' and forwards the frame to the Face Recognition service
    4. The Image Analysis and Face Recognition services fulfill their tasks and add the respective fields to the original frame and automatically forward it to the specified destination (Collector)
    5. The Collector then checks whether the new fields contain any data (i.e., persons' age and gender or persons of interest) and if so, forwards the frames to the Section service and/or Alert service, respectively

- Scalability?

- More data being streamed (more camera instances) could be handled by good scaling techniques – this may be realized through buffers/queues to respective services.

- Bottlenecks?

- In this setup, too many frames sent with e.g., many people on the image, and a short delay between the requests to the Collector may cause congestion and represent a bottleneck. Scaling up the number of e.g., Face Recognition and Image Analysis service instances would help process more images faster in such cases or implementing sophisticated load balancers.

- DNS mapping?

- Host-based DNS configuration: Docker enables the containers to do basic name resolution. Docker passes name resolution from the Docker host directly to the container instances.