

Cloud Computing Project Assignment

2023-2024

Jens Nicolay (jens.nicolay@vub.be), Angel Luis Scull Pupo (ascullpu@vub.be)

Scenario

Crowdy is a startup aiming to help citizen science analysts by providing them with next generation software to analyze crowded areas in European cities. In its constant expansion, engineers at Crowdy want to offer a real time “privacy aware” head counting software. The engineers expect a system where cameras are deployed in specific areas within cities to take pictures of the crowd. Images taken by the cameras are then upload to a server for further processing and analysis, i.e., head counting. For further and future analyses, pictures, together with their creation time and head count should be stored.

To help citizen science analysts with crowd management, Crowdy will provide a monitoring dashboard with aggregated results, originated from picture processing. The results include, a global head count chart updated periodically, and a head count per camera.

The image head counting software uses object detection techniques to then identify and count figures (i.e., humans). The object detection and counting features have been outsourced to an US company. This implies that pictures are sent outside European borders for processing, which raises sensitive privacy concerns.

Crowdy wants your help to build such a feature for its next gen software. Concretely, you are kindly asked to design and develop the following requirements.

Functional requirements

- 1- **Image upload.** The software should provide means to upload photos taken from the city cameras. You can assume that a fixed set off cameras is used. However, cameras are smart, they only take and send photos for processing when the objective is “sufficiently different” from the last photo uploaded. This implies that cameras in quiet places will upload less pictures than cameras in busy areas.
- 2- **Global head count.** The software should provide a real-time web dashboard where citizen analyst can monitor the evolution of total head count. Such a dashboard should be implemented as a 2-dimensional chart, where each bar represents the head count of all pictures taken in the last 30 seconds.
- 3- **Camera-specific head count.** The dashboard should provide miniature view component for all cameras connected to the system. Each camera view in such a component must show the latest photo uploaded by the camera, together with the latest head count for that camera.

- 4- **Global head count.** The dashboard should contain a component displaying real-time total head count of the system.

Non-functional requirements

- 1- **Fault tolerance.** The system services may fail for many different reasons. For example, a corrupt image may cause the image processing unit to crash. Therefore, you must ensure that crashes are isolated, and that the system can recover from such an eventual crash in a specific component. Moreover, you must ensure that such isolated/unrelated services keep functional in case of a crash of another service. For example, the failure of the image processing functionality cannot prevent the system from receiving new photos.
- 2- **Effective resources usage.** Although, the number of cameras to be used is predictable, processing times, for example, edge detection is CPU intensive and can vary from image to image. Therefore, you must pay careful attention to the resource allocation for your services.
- 3- **Laws and regulations ***.** Any data processing service that manipulates personal information of european citizens is subject to the GDPR regulations. In our case, private information such as pictures of people faces cannot be processed outside the eurozone without proper anonymization. Therefore, before sending images to the outsourced head counting service located in the US, you must ensure that only the necessary information required by the service is provided.

Practical Information

- The implementation of this assignment is individual. You cannot collaborate with other students.
- You are required to use JavaScript for implementing your assignment.
- You should deliver a zip file containing the application's code and all deployment archives.
- Together with the source code of the project you should submit a project report describing your solution to the assignment. The document should also discuss the choices you made for implementing the assignment's requirements. Moreover, the document must describe the steps required to deploy the application. The project report should not exceed 5 pages.
- The student should prepare a presentation (e.g., PowerPoint, Keynote, etc.) to discuss his/her solution to the assignment during the oral exam. In addition to the presentation, the student could provide a video showcasing the system's behavior, elasticity and robustness.
- Plagiarism will not be tolerated.
- Questions about the assignment can be sent to Angel Luis.