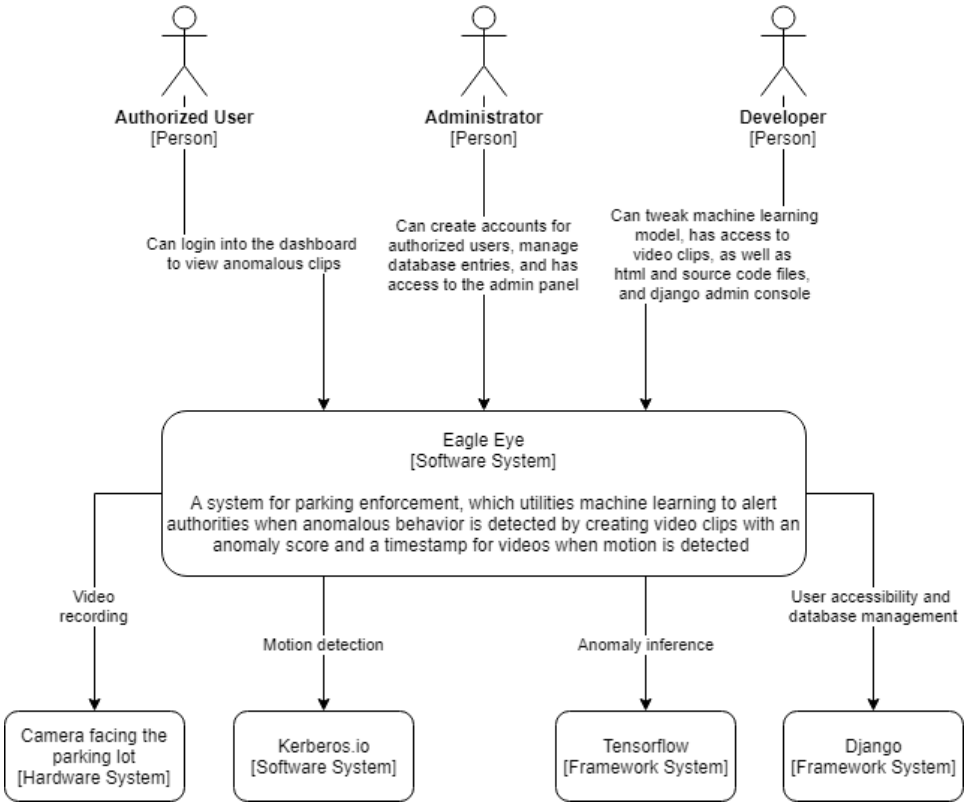
Eagle Eye: Development Guide

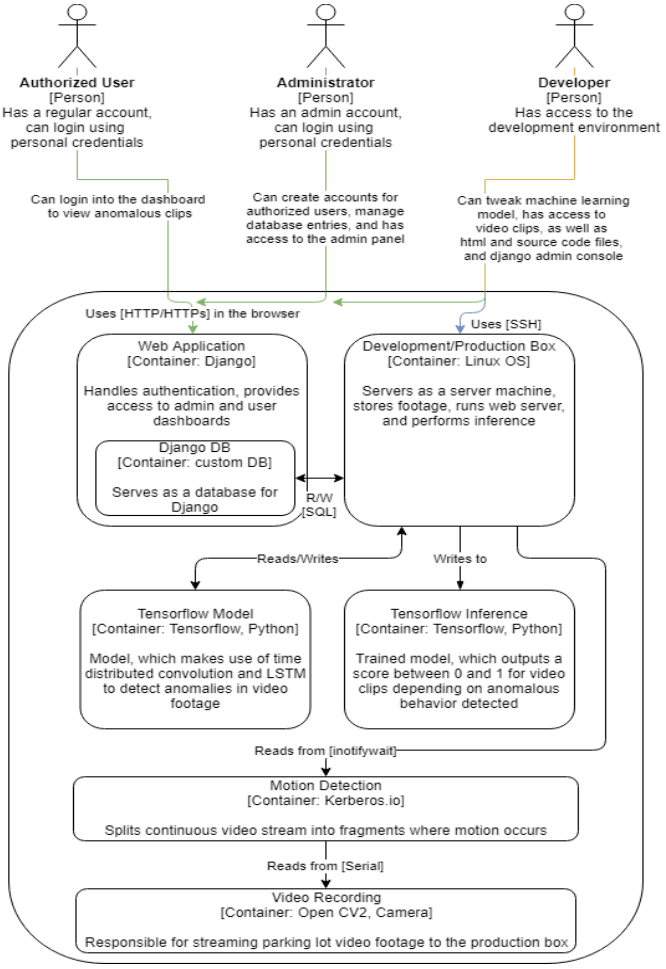
Team 10

04/30/19

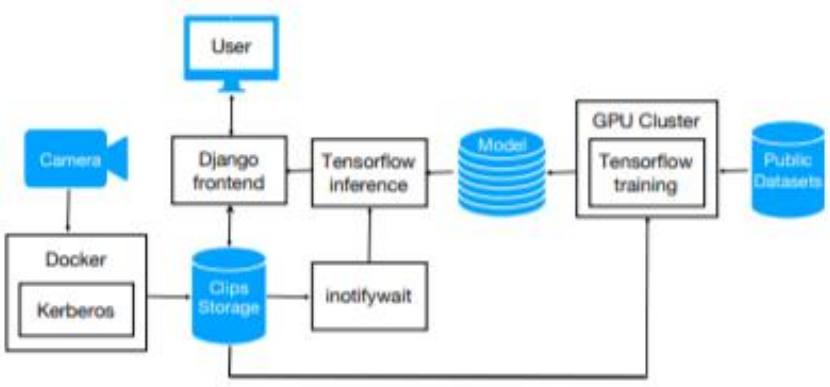
C4 Product Context Diagram



C4 Product Container Diagram



Simplified Product Pipeline Flow



Architecture Description

Architecture illustrated above utilizes a time distributed convolution and LSTM Tensorflow model trained using a public UCF “Real-World Anomaly Detection in Surveillance Videos” dataset to assign anomaly score to video clips. Said clips are produced by the Kerberos instance running in a Docker container, which saves fragments with motion present to disc, using continuous camera vide stream as input. Script “*inotifywait*” running in the background is listening for disc write events, which when triggered (meaning new motion clip was saved) will run inference using machine learning model described above and create a corresponding Django database entire with all relevant information. Finally, all vides saved in the database are accessible to the authorized users via Django frontend.

Credentials

In order to gain access to the system you will need to create a Django super user account using “*python manage.py createsuperuser*“ command. Next, you can add more authorized users via Django admin panel.