PHASE 1
PROJECT DETECT

Overview

This project will create a framework of detecting objects in an image, which will help users in automation of the requirements. The framework will not only store the detected data over a fixed period of time but also analyse it to provide intelligent information in the data.

Various approaches has been tried to do this using video analytic algorithms relying on computer vision, but since the input of this kind system is always changing the algorithms made for a certain customer may not apply for other customers. This asks for recurring work by engineers over each project. This becomes a bottleneck in scaling the product and leads to loss in revenues in business.

But this should not discourage engineers from using video as input for such kind of automation projects. Video camera as sensors are one of the best sensors which give us detailed information and it is available to us with evidence. We need an appropriate automation system which will enable automation in the calibration process from project to project. This will help in scaling a single product over the business and will result in exponential growth of business.

In phase 1, the emphasis will be on proving the concept of automatic calibration of the system.

It is proposed to automate the calibration process using human input and develop a edge controller which will learn to work better based on the human input available over the framework.

Consistent interaction of human mind and edge controller will enable the system to behave same across all users.

Architecture

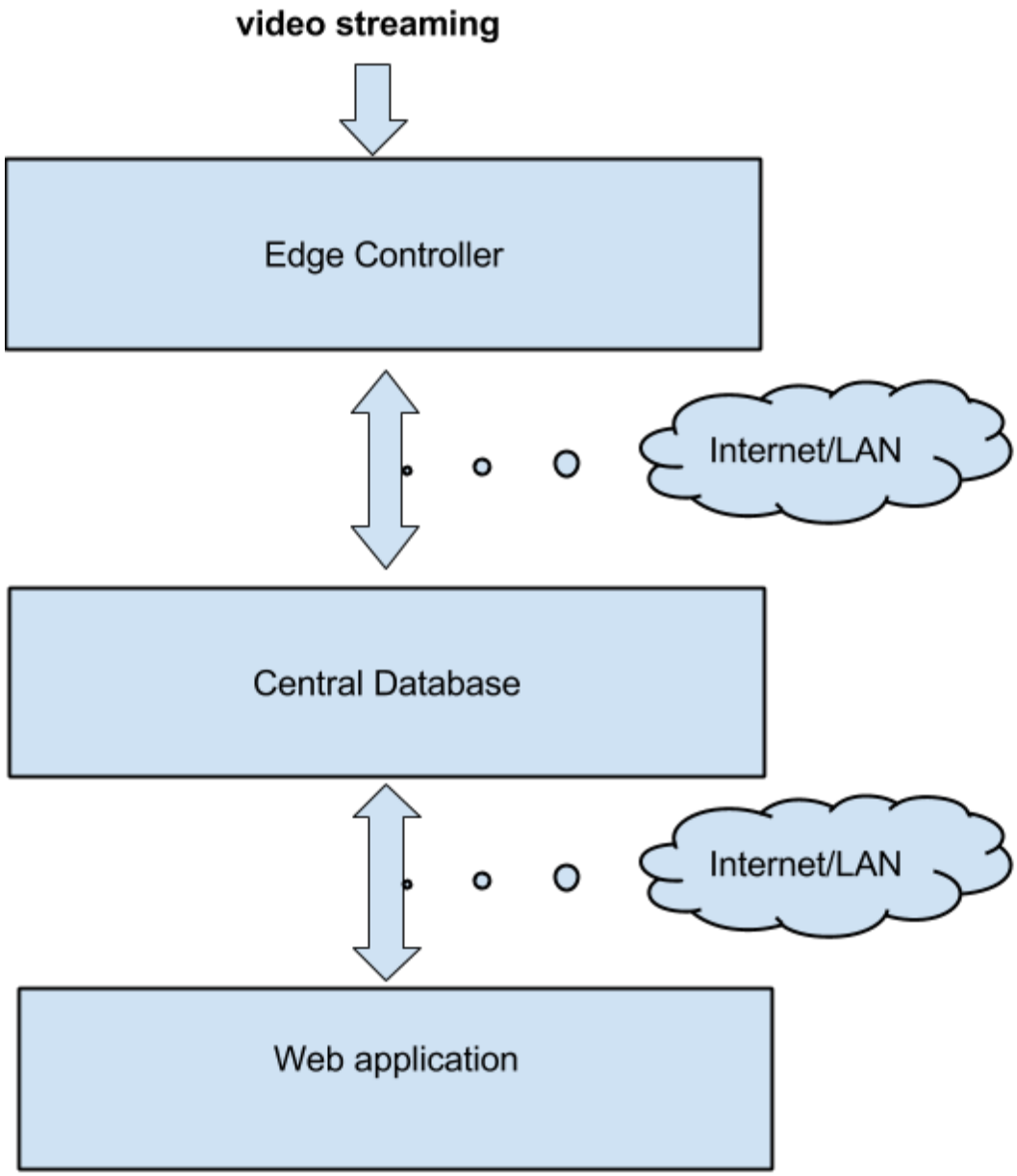

Figure 1: System Architechture

1. Edge Controller

This program will run at the client site. It can run on a user provided machine or on its own (manufacturer provided) native machine.

Following are the tasks this module will be doing:

step 1: acquire data from camera
step 2: run analytics on data
step 3: forward unchecked data to central db
step 4: receive checked data from central db
step 5: make some decisions, based on checked information
step 6: configure analytic
step 7: got back to step 1

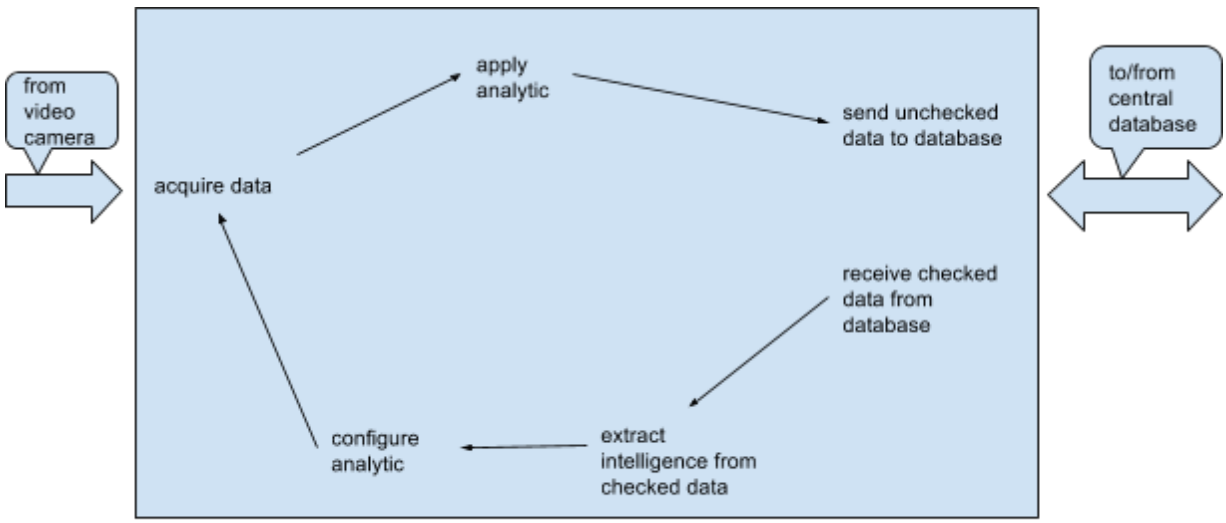

Figure 2: Edge Controller

2. Central Database

Central database will store data coming out (unchecked data) from edge controller and keep it stored for web application to work on it. It also stores the data (checked data) which is the output of web application.

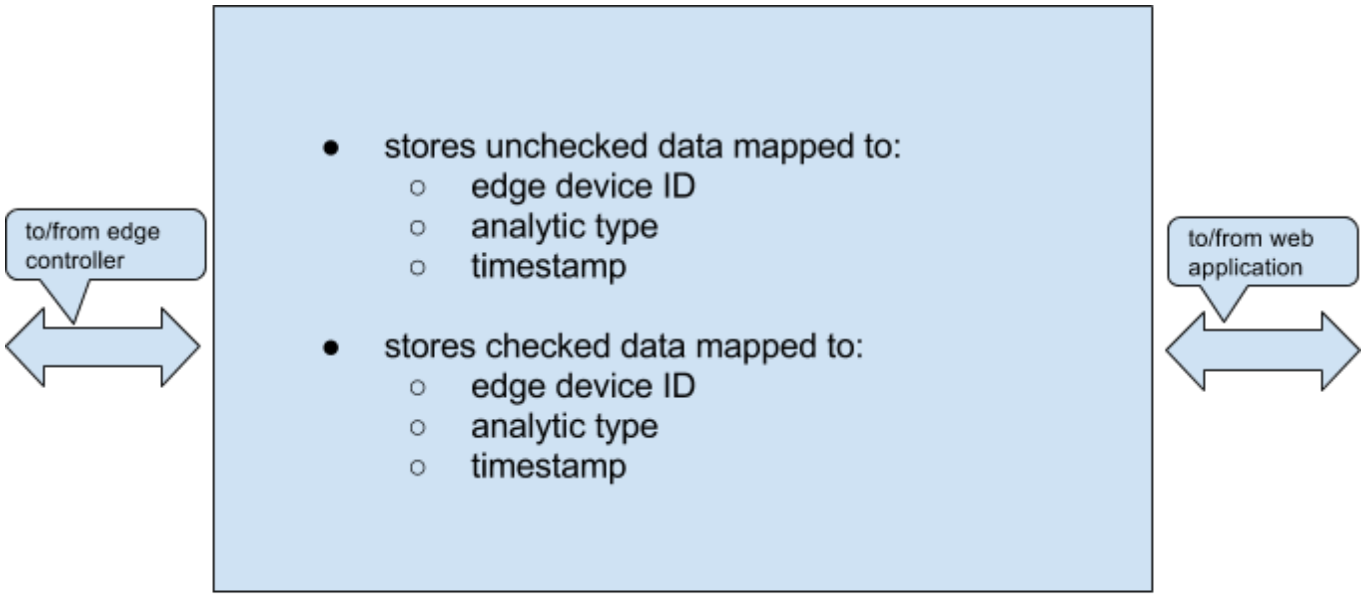

Figure 3: Central Database

3. Web Application

Web application will be a simple intuitive interface to database. It will enable user to look at the data and enter some information about the data. User will never came to know anything about the point of source of data. After user has given its input the data will be marked as checked and stored against the metadata of the same data.
Metadata constitutes the values like timestamp, edge device ID and analytic type.

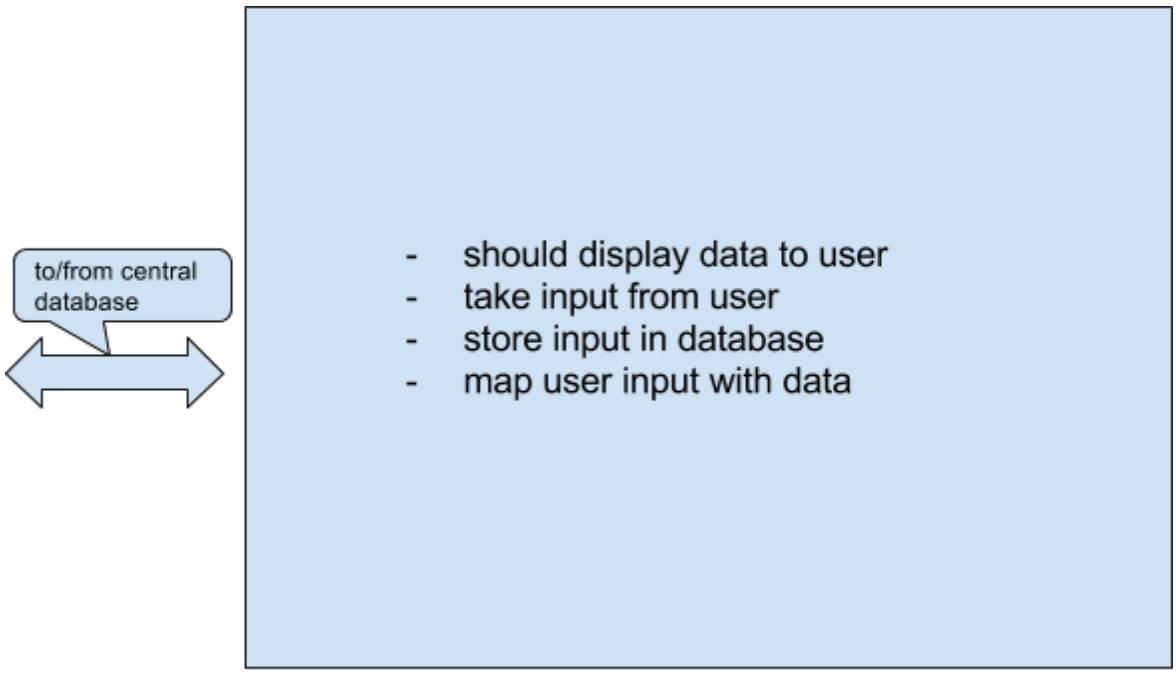

Figure 4: Web application

Architecture - Implementation details

1. Edge Controller

- programming language : c++
- access video stream using opencv/ffmpeg.
- use opencv optical flow calculation (lucas kanade algorithm)
- Interface with MQ to deliver data to Central Database.
- Interface with MQ to receive data from Central Database

2. Central Database

- postgresql
- metadata storage (timestamp, analytic type, edge device ID)
- checked and unchecked data storage

3. Web Application

- python for development.
- no disclosure of edge device ID.
- no idea of analytic.
- intuitive for user to give input about information in data.
- iterates between unchecked data, till all unchecked data is checked.