**ANPR based Campus Security design document (Messaging Queue and Http integration)**



Contents

[Business Component (Camera to Tablet Data Messaging integration) 3](#_Toc114242450)

[Publisher/subscriber setup 3](#_Toc114242451)

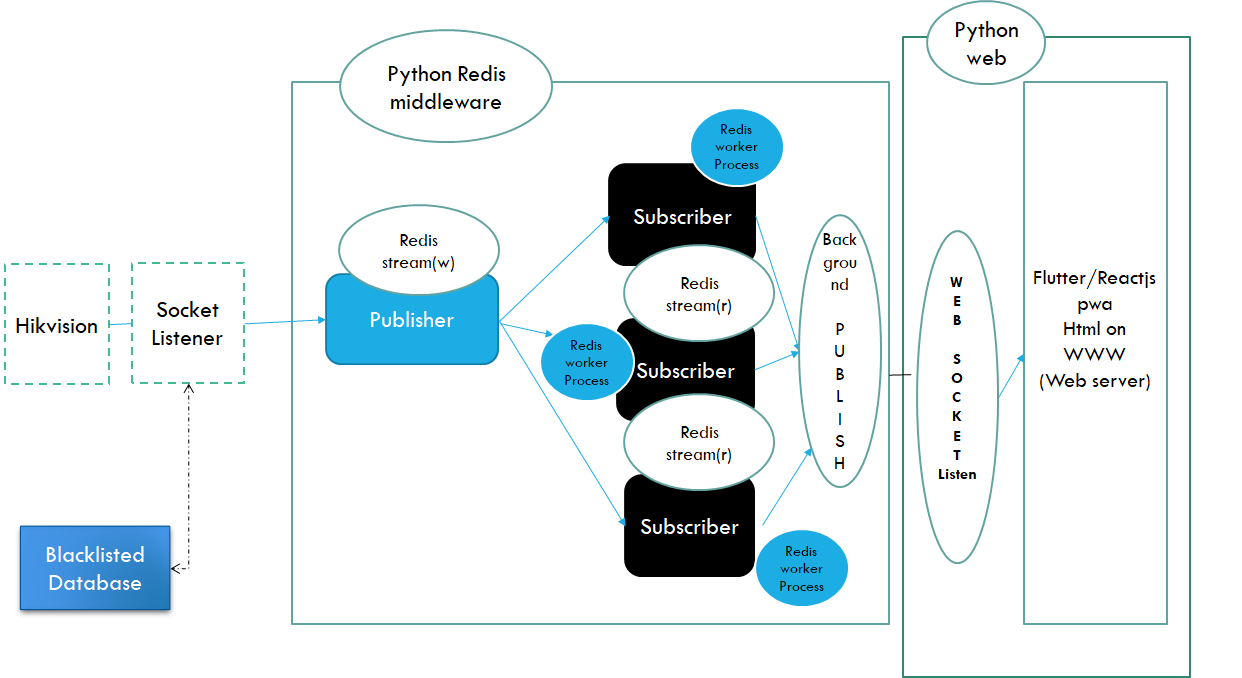
[Python loader Program plugin 6](file:///C:\ANPR\Python_messaging_component_design.docx#_Toc114242452)

# Business Component (Camera to Tablet Data Messaging integration)

* HikVision Python data integration /publishing for topic (each camera)
* Python to Nodejs subscribing (each camera specific setup)
* Nodejs to web server and web view integration (flutter/reactjs)

## Publisher/subscriber setup

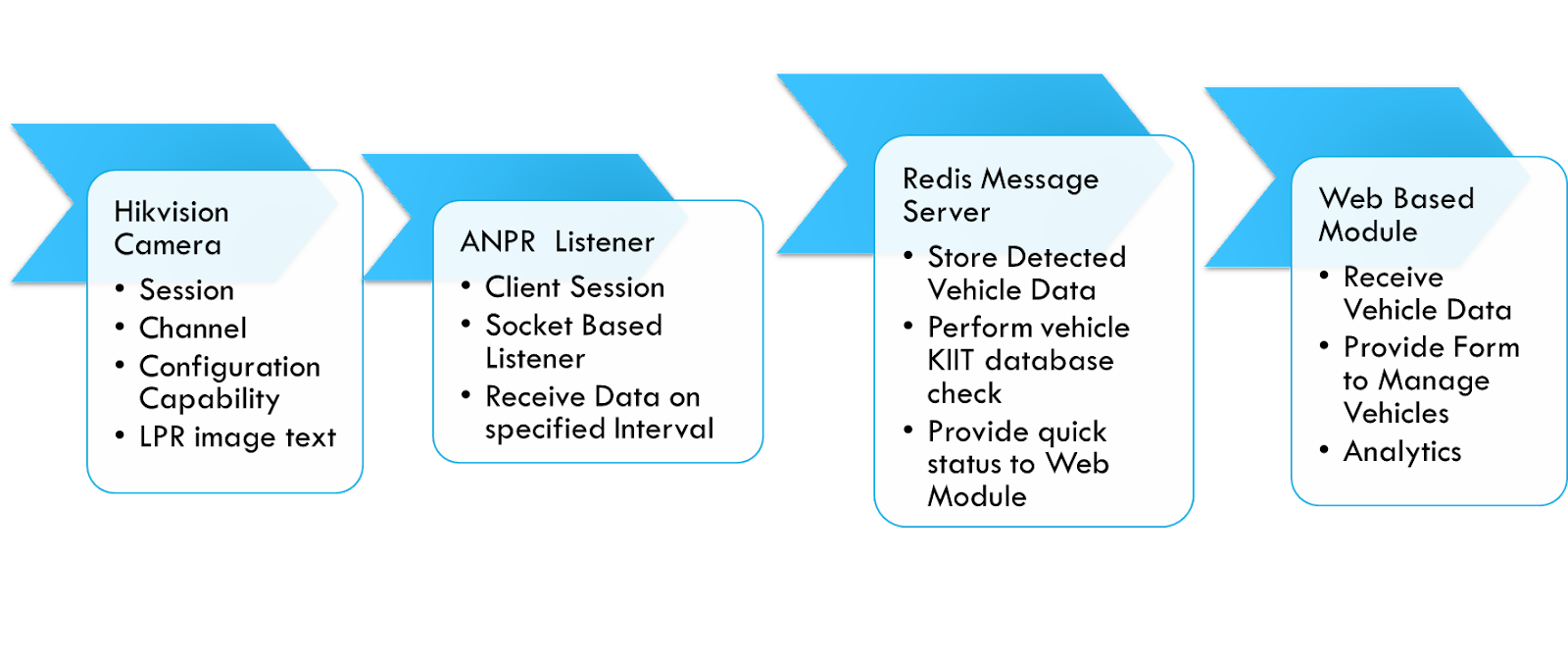
## 



Topic

Tools: redis/nodejs socket io/web socket technology

* Create a redis setup on port 6379
* Create a redis client for python/nodejs
* Create a publisher model with client side browsers as subscribers.
* When any hikvision camera capture event alarm is detected http post request is sent by a background redis cache job to node js/python module.
* Once all the data is received the react js/JavaScript/client side tab browser is refreshed with the vehicle details. The page is auto refreshed every 3 seconds.



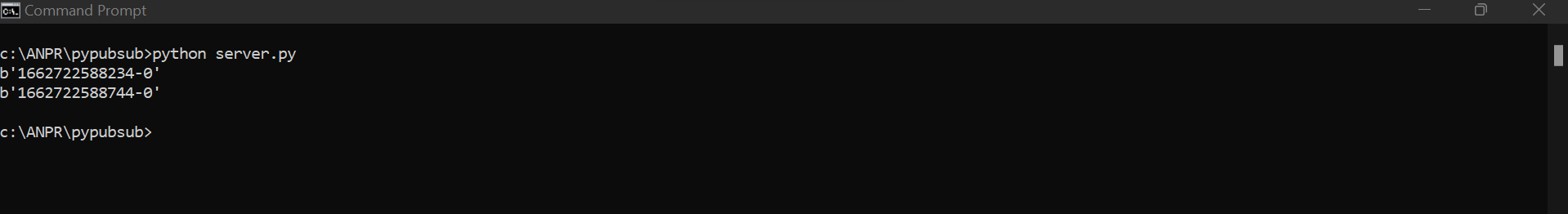
To add: background/asynchronous worker process)

Decide between web sockets versus http as messaging technology.

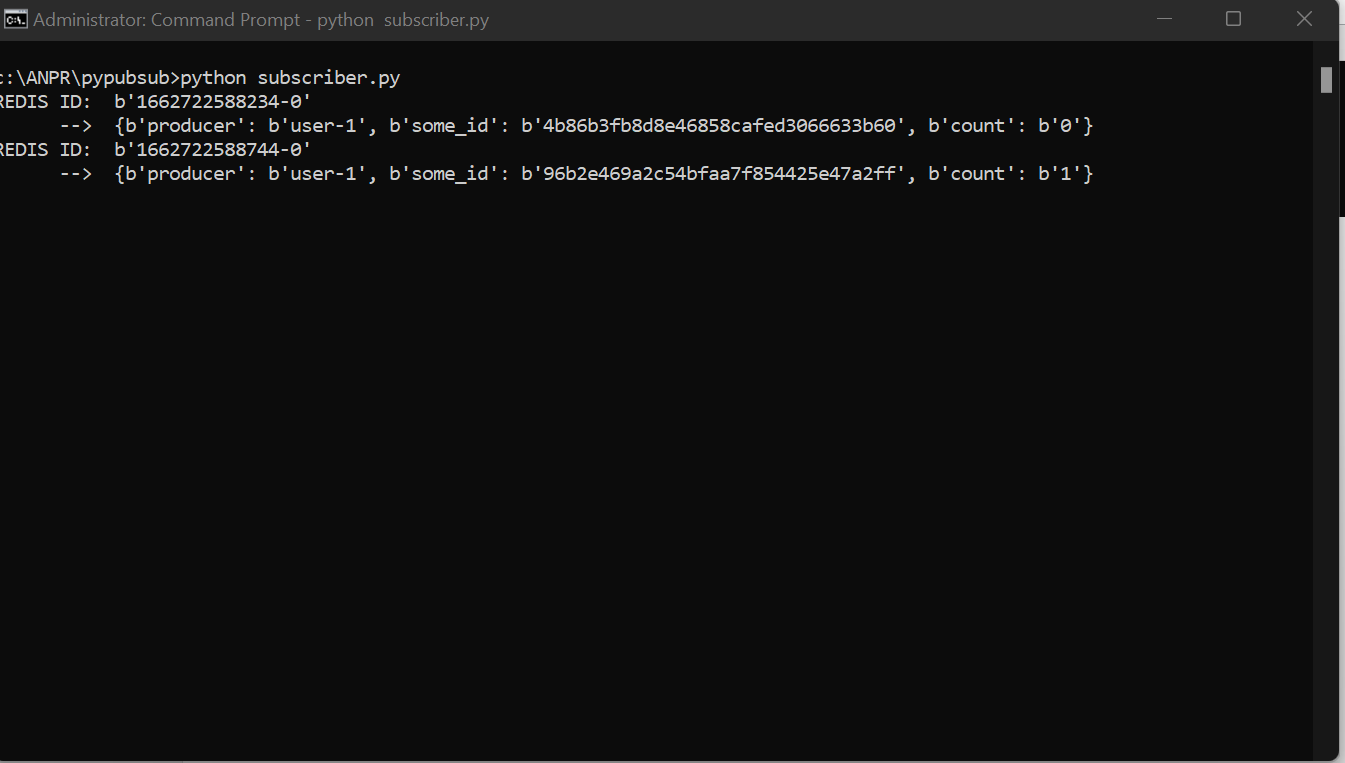
To expose the business or middleware we can expose the functionality as API so that it can be consumed by UI layer or react js.

Following is the high-level process flow as below.

* 1. Switch on the publisher module

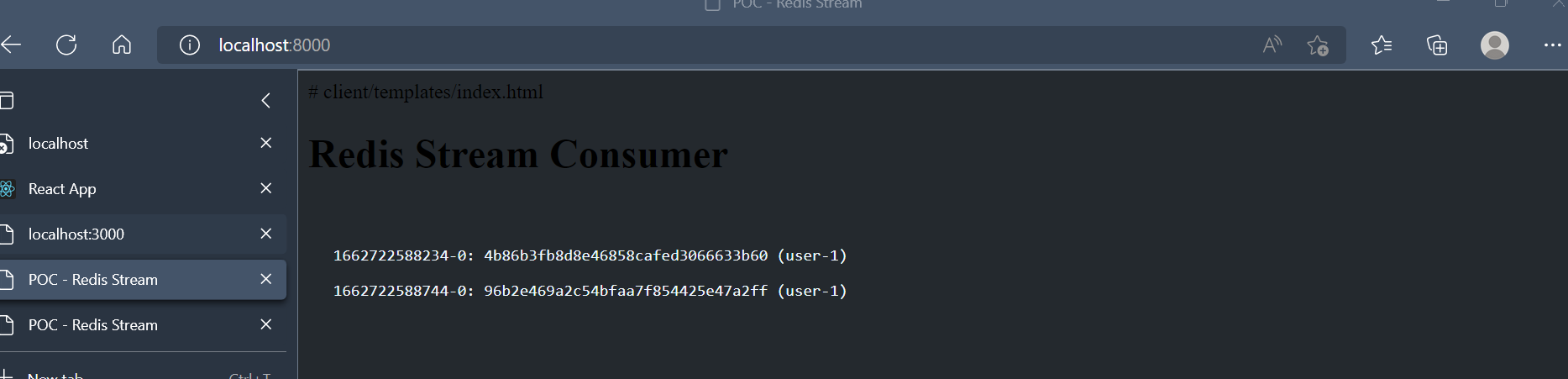


Two streams (events with messages) are added in redis queue.

Switch on the client subscriber module to read the vent stream.

* 1. Send the event details at client subscriber to web page in async mode

Display the redis ids from client receiver on a web page at port 8000 and localhost



The client side can be changed to any system, which receives an API with data from server side.

Production Requirements

* 1. Operationability

Warm up script to check connection in specified intervals for the 3 modules i.e. python on Nginx /redis/db connectivity.

* 1. Security

SSL with domain setup for transport security using TLS /SSL.

* 1. Support for Tablet as Preferred channel over mobile or desktop

Hosting the web Ui in a progressive web app with support for tablet.

Support for caching of data (optional) like list of registered vehicles and registered campus staff like faculty/KIIT employees etc.

Deployment

We are planning to run the threads in multi process mode.

* One master loader program to run all programs in a sequence or in parallel as per architecture supported by python.
* 1st python socket connector to Hikvision camera
* 2nd  python connection to redis pub sub queue
* 3rd redis sub to nodejs
* 4th nodejs to web server/http listener to display web pages .we are planning to use Flutter/React Native as the Progressive web app for tablet Form factor.
* 5th python to Database server SQL server to read ANPR profile and vehicle data with gate pass information.

We will run all the module or units independently and log exception messages in a file, In case of exception we can send emails to a support email id.

## Python loader Program plugin



## Exception

* Server crashed after certain number of transactions
* Server not able to receive any data
* Hikvision camera is not responsive.
* The plugins have some compilation errors.
* The no of messages is high and server is only sending old or cached records.