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



Contents

[Middleware Structure 3](#_Toc114579718)

[Publisher/subscriber setup 3](#_Toc114579719)

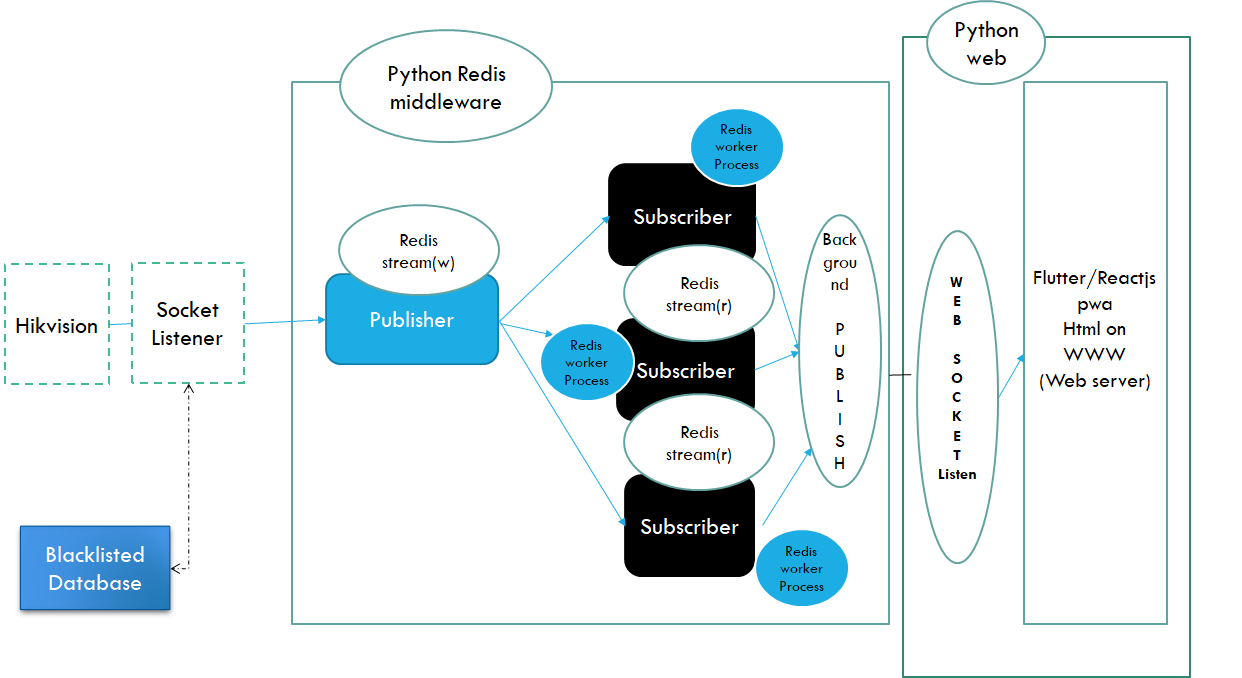
[Business Component Structure 8](#_Toc114579720)

[Business Logic/API Route 8](#_Toc114579721)

## Middleware Structure

### Publisher/subscriber setup

Each topic will correspond to a camera or gate and the system will subscribe to topic and get update for each ANPR camera.



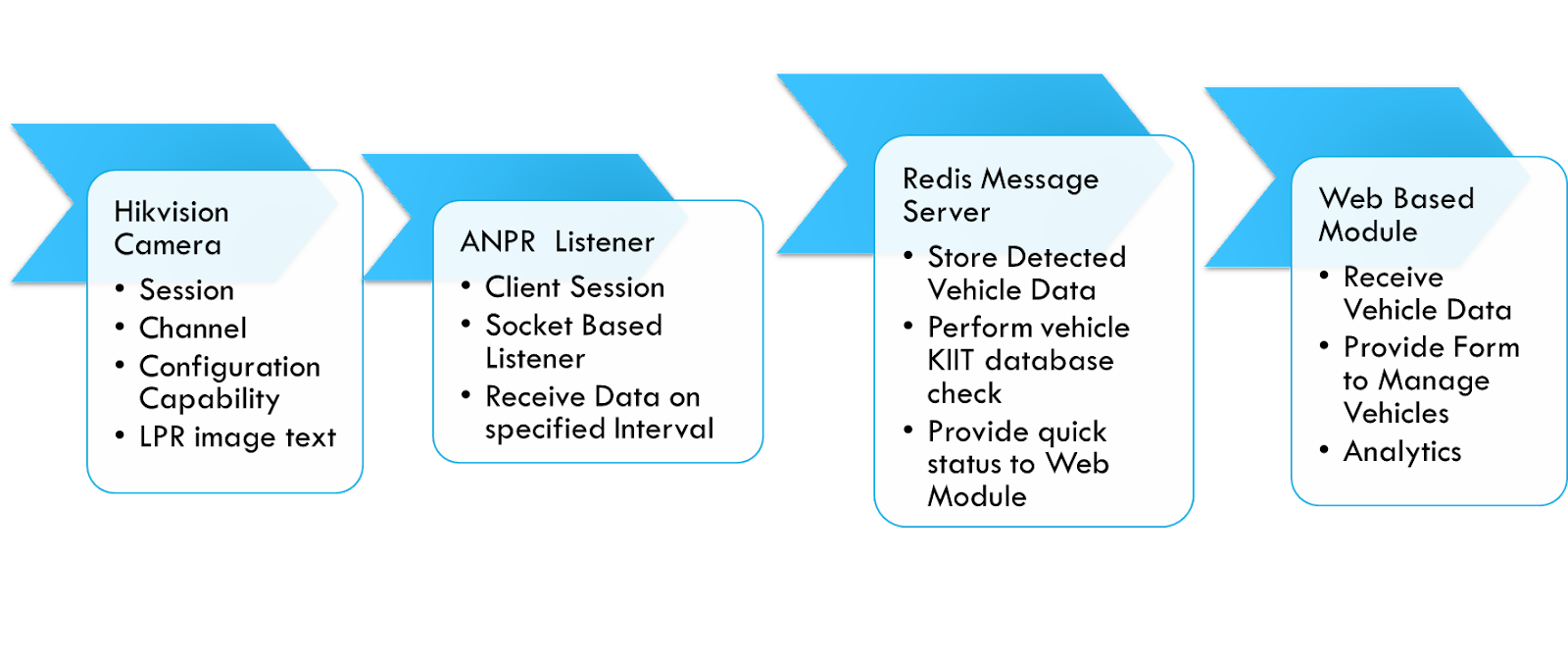
Topic

Topic

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 client, side tab browser is refreshed with the vehicle details. The page is auto refreshed every 3 seconds.



Production Requirements

* 1. Operation ability

Warm up script to check connection in specified intervals for the three 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
* 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 list of plugins**



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 or integration issues.
* The no of messages is high and server is only sending old or cached records.
* Specialized nameplates, which cannot be recognized like Defence/Ambassador corps/dignitaries etc. that may need specialized exception or manual processing.

Testing

* Socket connectivity and connection drops over a period of 1 hour.
* Memory profiling tools (psutil, memory\_profiler etc…)
* Web server memory connection and buffer testing

Controller Class: ANPR\_infra (Establish Session with Hikvision)

* Method: \_init
* Method: GetSetCamera

#Login to Hikvision Camera API and get/set session, configuration details

#Validation

Return status code

* Method: GetSetDBConn

#Login to MSSQL server and create a DB connection pool.

#Validation

Return connection instance

* Method: GetSetPuB

#Login to Redis server and create a channel for a camera/topic

#Validation

Return instance

* Method: GetSetSuB(s)

#Login to Redis server and establish connectivity between   subscriber(s) and publishers

#Validation

Return instance

* Method: Socket\_Configure

#Create a listener socket and receiver for hikvision camera

#Validation

Return status code

* Method: CameraConfigure

#Create a function to establish session with camera with values read from a DB table/config file like userid/password/url etc..

#Validation

Return status code

**Controller Class:  ANPR\_LPR**

Method \_init(get connection details)

* Input param : self-instance,tcp\_ip,port
* Processing:  listen for vehicle detect events at specified intervals.
* Exception handling : Log exception, Null Data exception,connectionissue exception
* Output :

Method: capture\_number\_plate

* Input param : self instance,tcp\_ip,port
* Processing: Get vehicle data once data is received in the socket and parse the XML file to get Vehicle information.
* Exception handling : Log exception,Null Data exception,connectionissue exception
* Output : collection (Number Plate)

Method: Blacklisting

* Input param : self instance,data
* Processing : Check vehicle number plate data in case it is blacklisted
* Exception handling : Log exception, Null Data exception,connectionissue exception
* Output : Return Boolean

Method: validate\_license\_plate

* Input param : self instance,number\_plate
* Processing : Search and return vehicle number as per format below

Odisha : OD xxxx

BH : 022BH….

Replace 0 with “O”

* Exception handling : Log exception, Null Data exception,connectionissue exception
* Output : Return Corrected License plate data.

**Common or Generic Functionality**

* User authentication with ANPR user data exported from Kiit Sap User database.
* SQL server group driven Role based authorization
* API Data interaction : XML/JSON ,Method : Get/POST
* API authentication provides user id password from a secret store instead of hardcoding user/password values in code.
* Generic CSS
* Javascript common input sanitization functions.
* Logging/Exception handling method.

## Business Component Structure

Controller Class:  ANPR\_Business\_component

* Method : RegisterVehicle
* Method : UpdateRegistration
* Method:  IssueGatePass
* Method : Authenticate
* Method : VehicleEntryApproval
* Method : MapUIDB

Controller Class:  UtilsFunction (Reusable Function)

* Method: ExceptionHandling
* Method: XMLStreamParsing (Input string,input FileData,outputArrayDict)
* Method: JSONPaser (Input string,input FileData,outputArrayDict)

## Business Logic/API Route

Registering of Vehicles (Processing)

Web pages/Route

Pages –

1. /Login

Used to authenticate users and show Profile page+menu and dashboard.

This is the init lifecycle of the software to generate sessions and provide profile specific menus as per existing role and permissions.

1. /AdminDashboard  (centralized staff)

Data to show in the page

* Counting the total number of vehicles entering the premises
* Counting the total number of unique vehicles entering the premises
* Counting the total number of vehicles scheduled to enter the premises
* Counting the total number of unscheduled vehicles that entered the premises
* Counting the total number of vehicles currently on the premises
* Counting the number of vehicles based on the type of pass issued to them
* Counting the number of vehicles with special purpose passes
* Counting the total number of vehicles that entered the premises for a day

1. /SecurityDashboard(Security Guard Dashboard)

The security staff at the gate can perform following transactions.

* Pass Issue Form
* One-Time-Entry-Exit-Pass Code verification
* Setting

1. /Masters

* Tabular records of all the entries made into and out of the premises
* A record contains plate number of a vehicle, entry and exit date and time, vehicle status based on whether the vehicle is on the premises or not, type of pass issued, pass status.

1. /Visitation\_Control

* This page is used by the admin to issue passes from themselves to vehicles and schedule visits to the premises.
* Dignitaries use this in case of visits. The page also controls the list of black listed vehicles

1. /Employee

* Contains name of the employee, employee ID, employee department ,vehicles registered against the employees name, type of pass issued to them, contact number, password.

1. /ProfilePage

Once the user is logged in he can see menus and specific navigation  for the currently logged in.

1. /Aboutus

Transactional Data entry Forms –

* Visitor scheduling form
* Login Form
* Contact us form
* Change Profile details form
* Manual Entry Form
* Advanced Search form

Security (settings)

* Configuration Data storage
* Encrypting of Data.