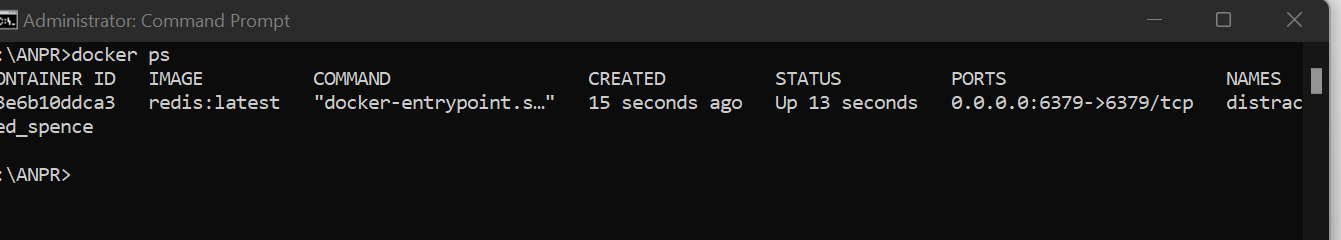
**Business Component (data flow and integration)**

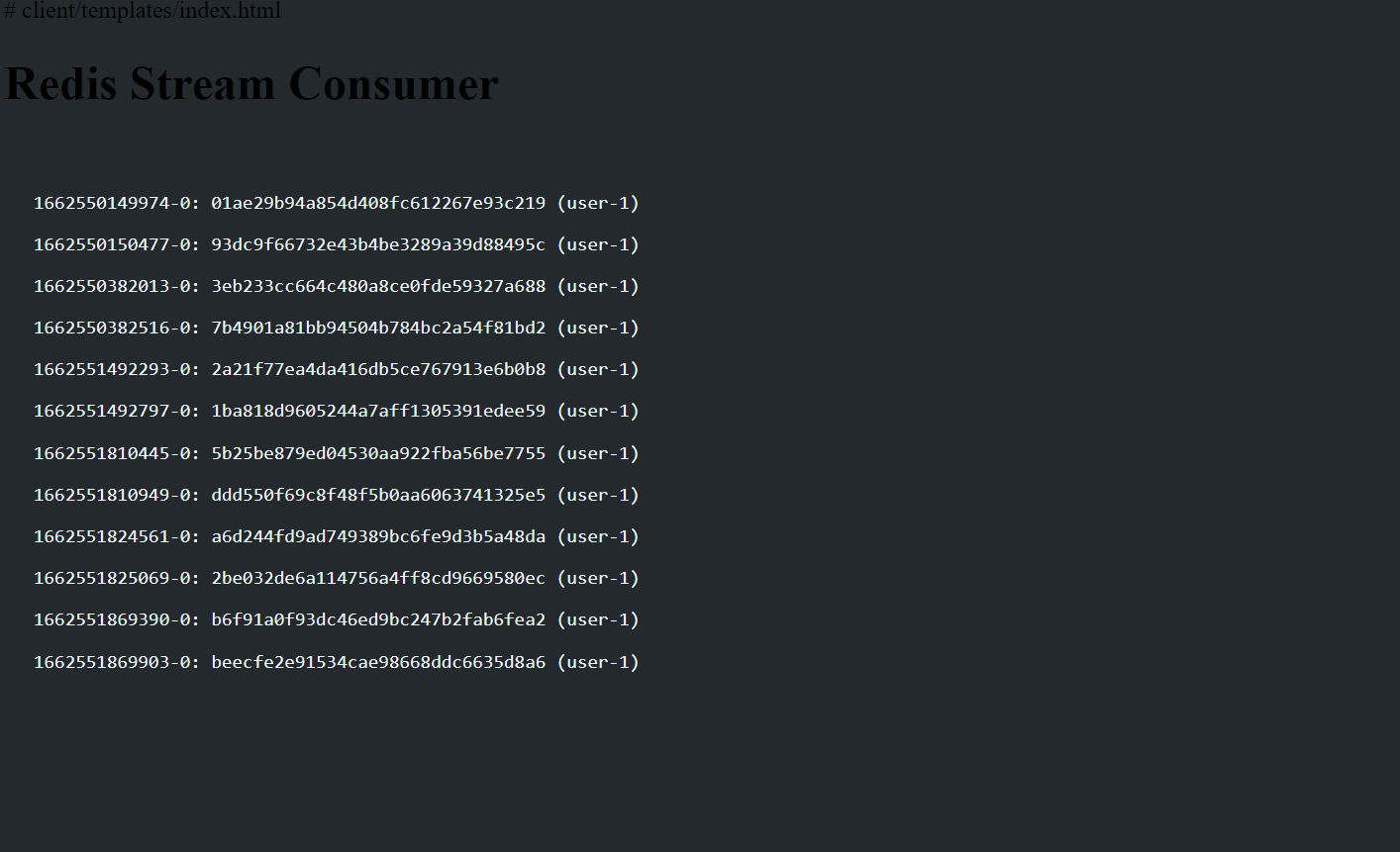
HikVision Python messaging (WIP)

Polling frequency : <to be set post discussion>

Python API to web server and web view integration

Redis on local (change for server) port 6379 mapped to host 6379





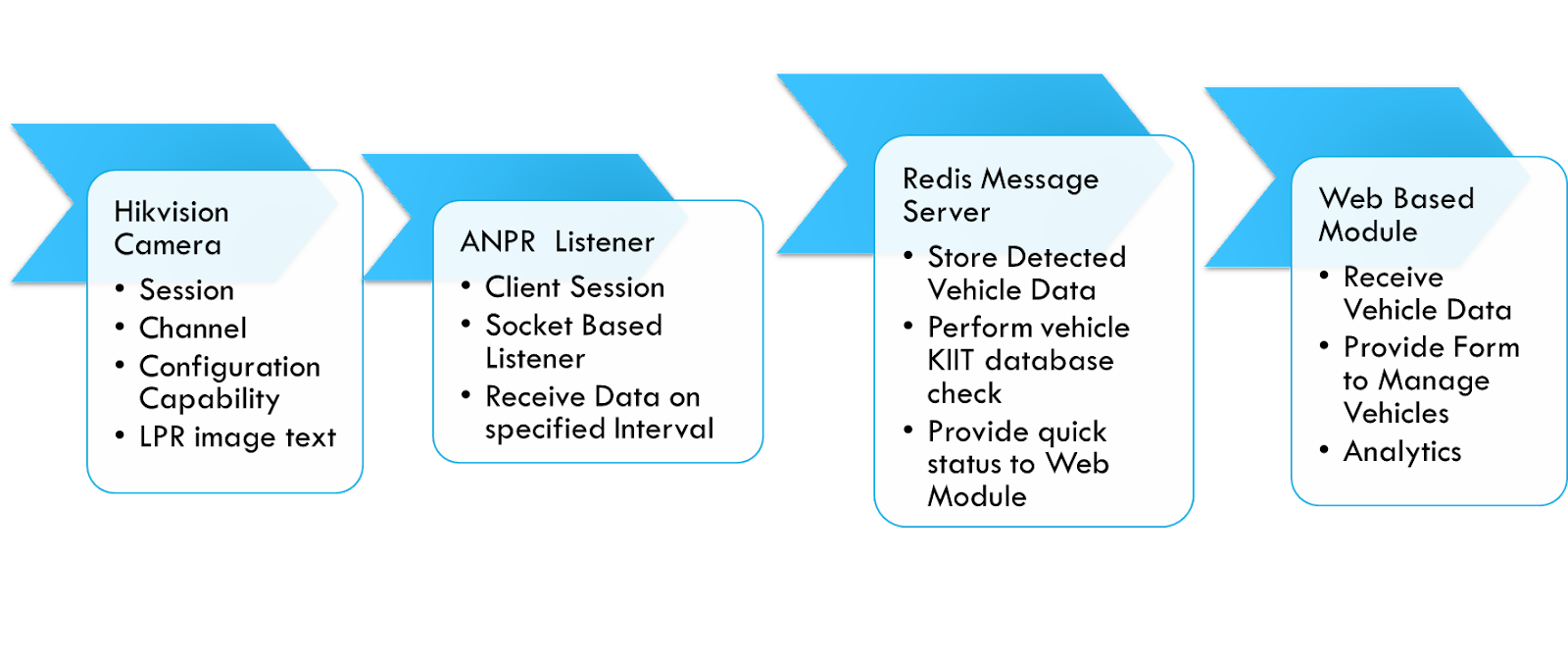
Message output on web server on an html page (can be also react js/angular, vuejs, or similar JS framework)

**Publisher/subscriber setup**



Tools: redis/nodejs socket io/web socket technology

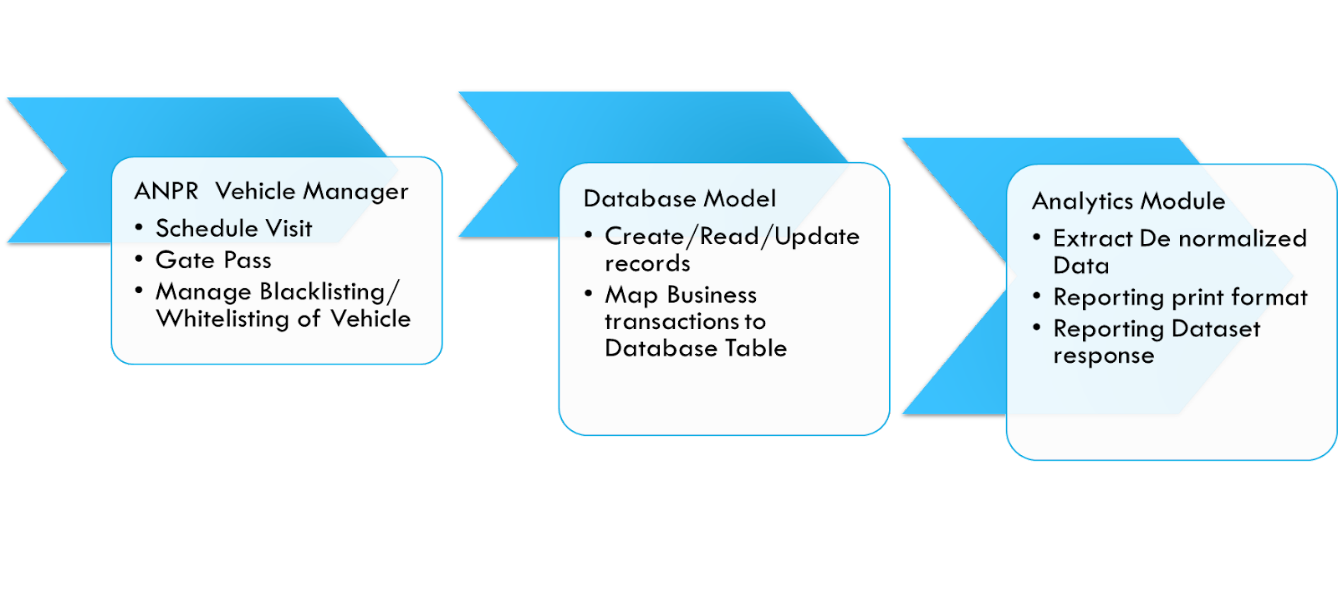
* Create a redis setup on port 6379
* Create a redis client for python/nodejs
* Create a publisher model with client side browsers as subscribers.
* When any hikvision camera capture event alarm is detected http post request is sent by a background redis cache job to node js/python module.
* Once all 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.

WIP (analytics/reporting module)



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.