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

HikVision Python messaging (WIP)

Polling frequency : <to be set post discussion>

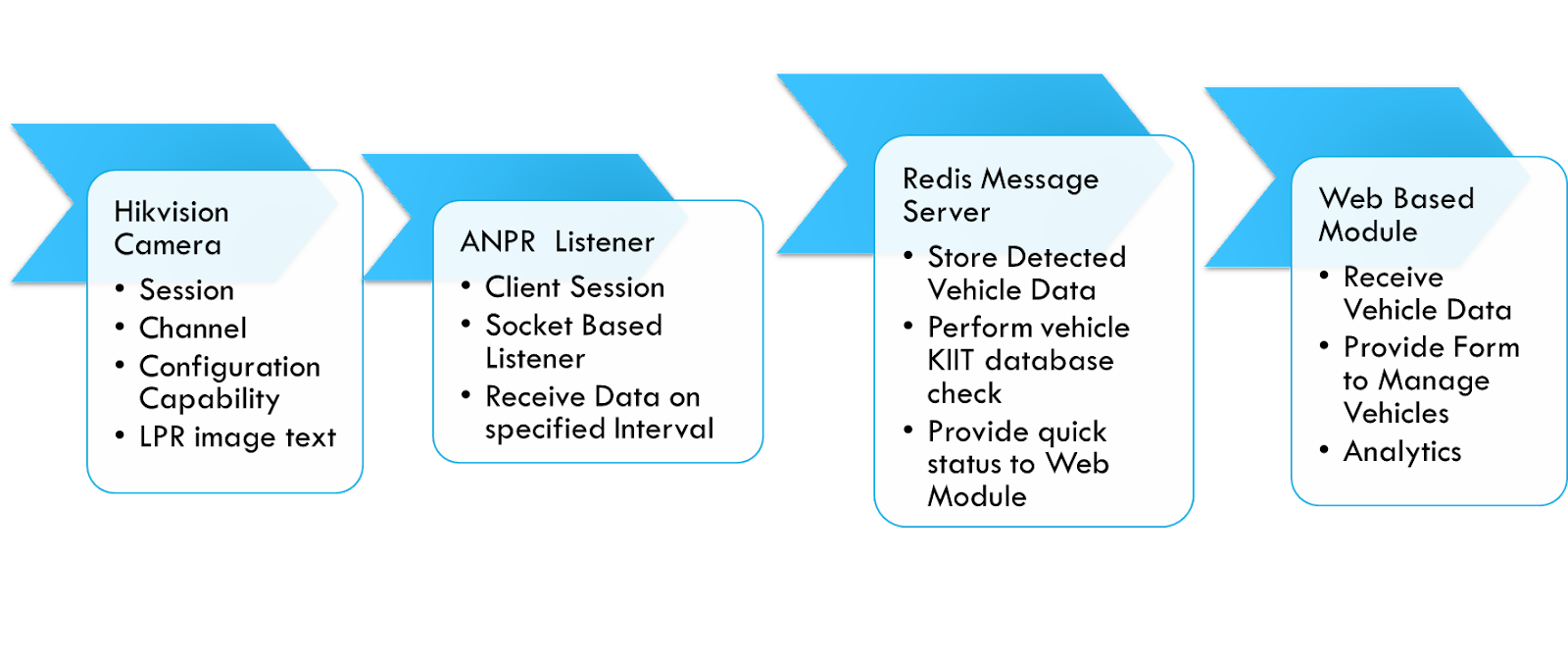
Python API to web server and web view integration

**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.



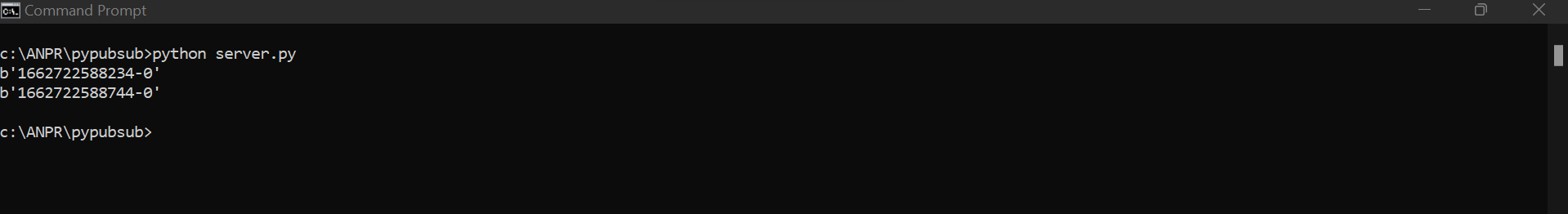
To add: background/asynchronous worker process)

Decide between web sockets versus http as messaging technology.

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.

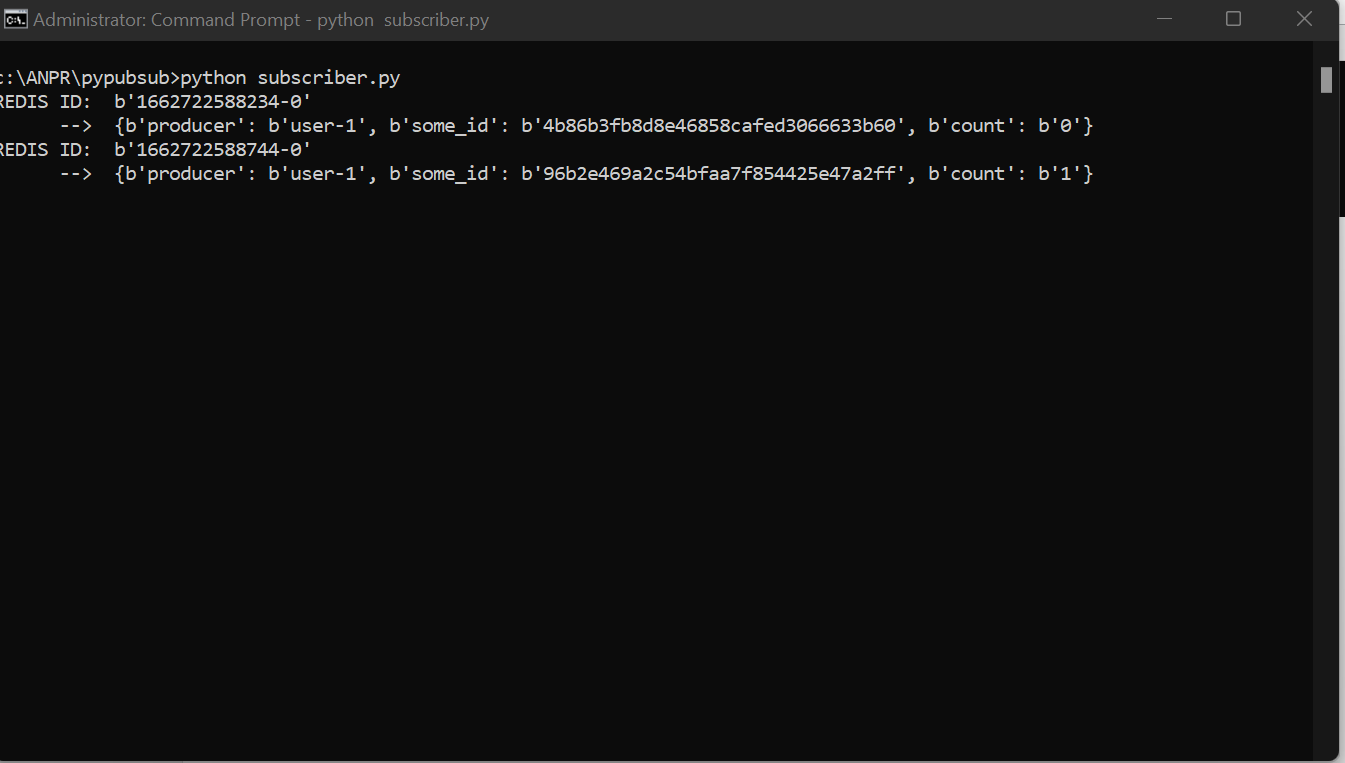
Following is the high-level process flow as below.

* 1. Switch on the publisher module

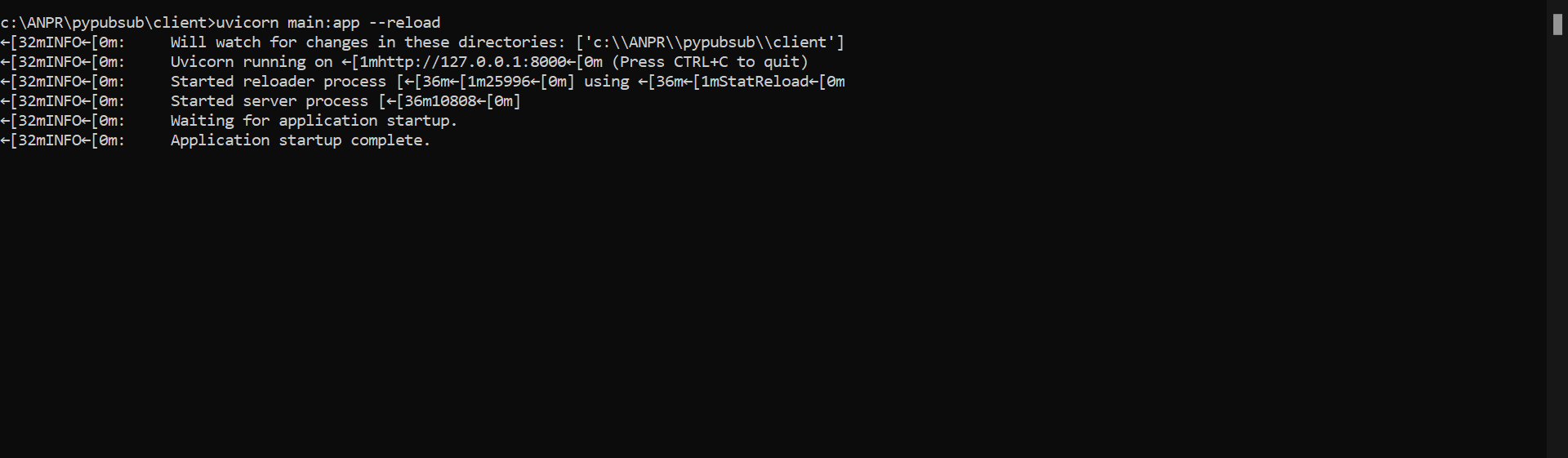


2 streams (events with messages) are added in redis queue

Switch on the client subscriber module to read the vent stream



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



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

