

EMBEDDED INTERFACE DESIGN

# PROJECT - 06

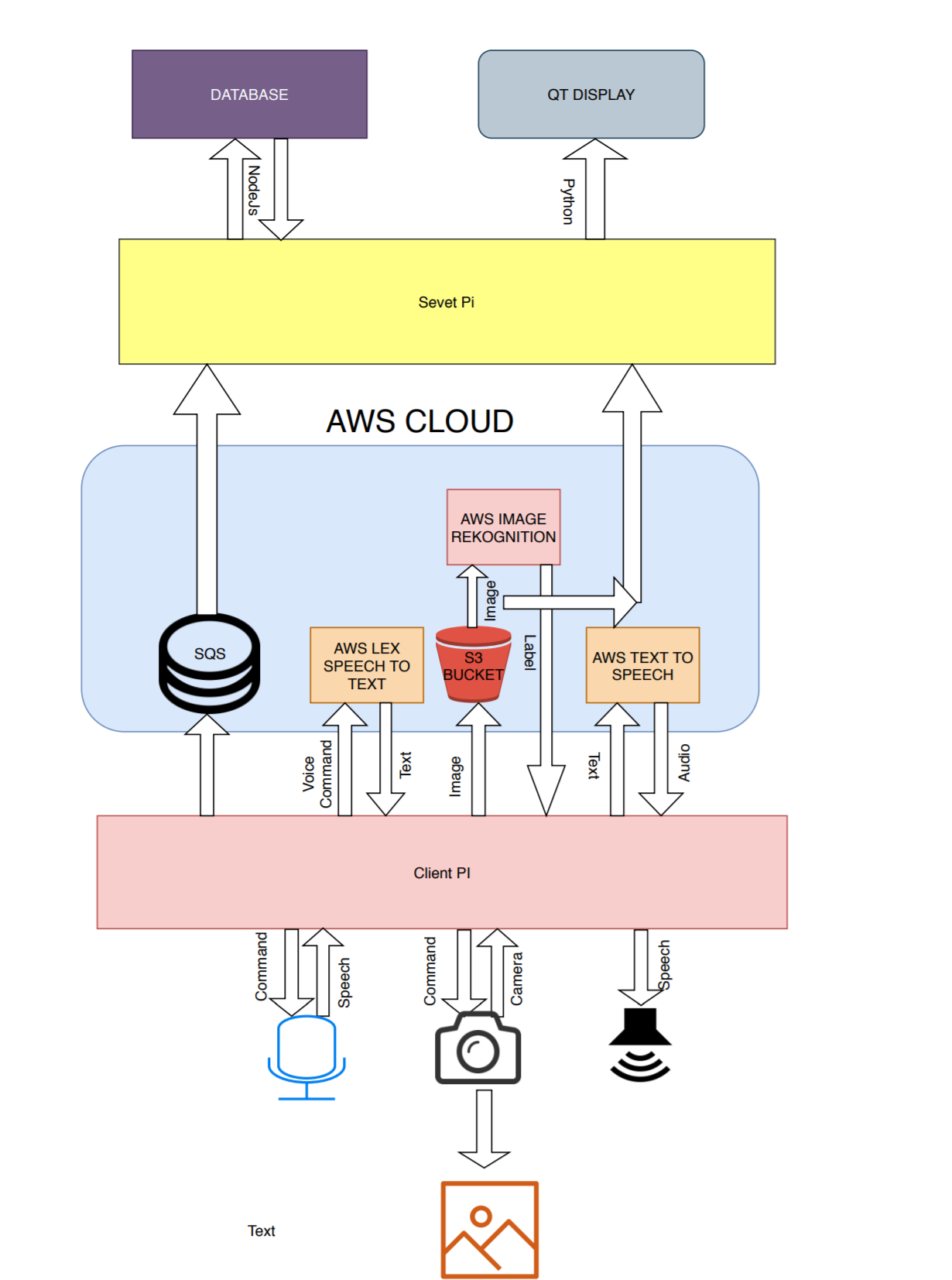
THE MAGIC WAND

**Under the guidance of Professor Bruce Montgomery**

Deepesh Sonigra

Madhumitha Tolakanahalli Pradeep

## ARCHITECTURE DIAGRAM



The Magic Wand performs as outlined below –

* Client PI
  + The User presses the mic button to enable the mic and says “Identify” using AWS Lex
  + An Image is captured from the RPi Camera, sent to AWS Rekognition to recognize it and output on the Speaker(using AWS Polly)
  + The wand waits for the user to confirm if what has been recognized was right / wrong.
  + The recognized label is sent to AWS Cloud (SQS)
* Server Pi
  + The Server Pi functions mainly as the PyQt UI for statistics and a database that is implemented using Node.js
  + The user can obtain information about the wand like – last clicked image, accuracy of commands and image recognition

# PROJECT DEVIATION STATEMENT

The design of the Project remains the same for the most part. Below are some of the deviations -

* One of the major deviations from that original proposed was to use PyQt for UI on the Client instead of HTML Webpage. This decision was made to keep the UI consistent across the Client and Server. Also, accessing the database was relatively easier from Python than HTML Javascript.
* We had planned to activate the wand using “Identify”. However, we changed the design to include a mic button that would start listening for Identify and then perform the rest of the operations.
* We included two SQL Tables – one for tracking if commands were recognized correctly and one for tracking of the image was recognized correctly.
* The image that is displayed on the Server UI is downloaded from Buckets, deviating from our original plan to fetch it from AWS SQS.

# THIRD-PARTY CODE USED STATEMENT

* <https://www.youtube.com/watch?v=T8T6S5eFpqE>
* <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/4>
* <https://picamera.readthedocs.io/en/release-1.10/recipes1.html>
* <https://iotbytes.wordpress.com/connect-configure-and-test-usb-microphone-and-speaker-with-raspberry-pi/>
* <https://stackoverflow.com/questions/48135955/by-installing-pyaudio-python3-on-my-raspberry-pi-3-noobs-i-get-an-error-how>
* <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/s3-example-download-file.html>

# PROJECT OBSERVATION STATEMENT

* Getting the camera, microphone and speaker up and running was easier than expected. The extensive list of resources for the RPi came really handy.
* We struggled a bit with getting Speech to Text work on AWS. Initially, we experimented with AWS Transribe, which was not a real time tool. We had to switch to AWS Lex for allowing for real time speech recognition. But that did require some tinkering with sampling rate to get it up and running.
* Displaying tables on PyQt5 GUI was considerably easier than doing the same on HTML. Manipulating json data from the SQL Database was easier with Python than on Javascript.