

# **“ The Magic Wand”**

Embedded Interface Design Super Project

By

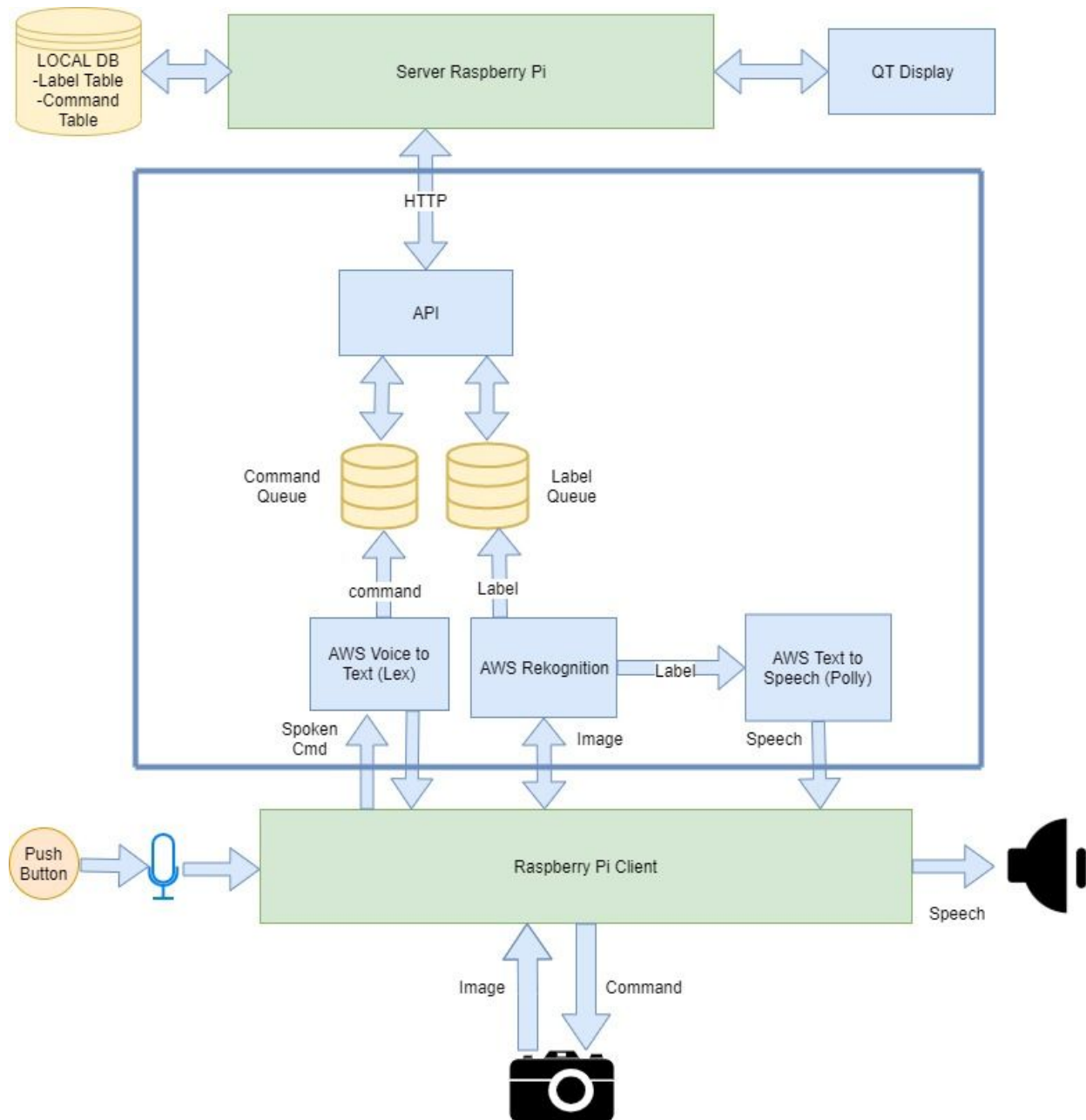
Ridhi Shah

And

Amreeta Sengupta

Date : 12/11/2019

## Final System/Architecture Diagram and Statement



The new added elements include:

- Push button which on pressing records the command from the user
- 2 separate SQS queues for labels and Commands

## Project Deviation

As Proposed in Project 4 and 5 most of the components of the projects remain the same. Some of the parts where the project has deviated from the original idea proposed is due to the addition of the following components:

- A push button has been added, which when pressed records the voice command i.e Identifio, Correcto, Wrongo from the user. The Client Application runs continuously and waits for the button to be pressed. Once the button has been pressed, the voice command is recorded and further processing takes place.
- Two separate SQS queues are created for storing the Image labels and the voice commands.
- The local Database contains 2 tables for storing the Labels and the Voice commands.

## Third Party Code Used Statement

The References listed below include all the third party sources from which code was used directly or was used as a reference for developing our code.

- The code for interfacing the push button with Raspberry Pi using Node JS was leveraged from the following resource:  
[https://www.w3schools.com/nodejs/nodejs\\_raspberrypi\\_led\\_pushbutton.asp](https://www.w3schools.com/nodejs/nodejs_raspberrypi_led_pushbutton.asp)
- The code for AWS lex was developed by referring to the following boto3 SDK link:  
<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/lex-runtime.html>
- The code for adding data and downloading files from Amazon S3 bucket was developed by referring to the following boto3 SDK link:  
<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3.html>
- The code for converting Speech to text using AWS Polly was developed by referring to the following boto3 SDK link:  
<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/polly.html>
- The code for Adding and removing data from SQS queue was developed by referring to the following boto3 SDK link :

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/sqs.html>

<https://aws.amazon.com/sdk-for-python/>

- The code developed for using various functions of the MySQL database was developed by referring to the following link:

[https://www.w3schools.com/python/python\\_mysql\\_getstarted.asp](https://www.w3schools.com/python/python_mysql_getstarted.asp)

## **Project Observation statements:**

- Finding the documentation and using the AWS Python SDK for using Amazon services like SQS, AWS S3 buckets, Amazon Lex and Amazon Polly was easier than expected.
- Running both the client and the server simultaneously in a loop for continuous reception of commands and displaying the data was more challenging than expected which is why the push button was added so that the recording of the command occurred only when the button is pressed, otherwise the application will be waiting in a loop.
- Playing the MP3 file on Raspberry Pi using node JS and also using Node JS SDK for AWS was more challenging to integrate with our application.