Team 20

Logan D Lynch – lo4592@vt.edu

Therese Ann Bernadette Hempenius - thereseh@vt.edu

Nathan Chang – nchang@vt.edu

ECE 4564 Assignment 1

This assignment is designed to consist of two independent parts, running on separate Raspberry Pi's: a client installation, and server installation. After command-line initialization with server information, the client Pi will monitor Twitter for valid posts, which contain a tag unique to this team and assignment, #ECE4564T20. Upon receiving a valid Twitter post, the client Pi will take the question from the Twitter post and encrypt it, creating a checksum and sending the encrypted payload to the server Pi, after pickling. It will then wait for a response from the server. Upon receiving an answer payload, the client will verify its checksum, decrypt the payload, and send the enclosed answer to IBM Watson. After receiving the text to speech audio of the answer from Watson, the client plays it aloud. The server installation, upon initialization, waits to receive a question payload from the client installation. Upon this event, the checksum is verified and the question decrypted. The question is sent to IBM Watson, and text to speech audio is returned and played. Additionally, the question is sent to Wolfram Alpha, and the returned answer transformed into an encrypted answer payload which is returned to the client Pi.

Regarding specific design decisions of the assignment, we chose to use the Tweepy Python library to work with the Twitter API for question monitoring. IBM Watson's Python library was also used, obviously. Text to speech was handled with a Watson-text-talker library. Tweepy code was taken from its documentation examples. The server component made use of the wolframalpha Python library, also something we had little choice in. Fortunately, this library was also reasonably documented, making development time minimal. These libraries, along with the code holding them together, resulted in the outcome of a functioning system during our testing. To summarize, a properly tagged question tweet would be correctly captured and sent to the server by the client Pi, vocalized by the server Pi, and its answer returned to and vocalized by the client Pi.