Team 14
Will Campbell – wtc@vt.edu
Jaime Otwell- jmotwel@vt.edu
Tori Guy- gtori6@vt.edu
Extra Libraries:
Hashlib, Time, flask, requests, zroconf, RPi.GPIO, Json, wget, avahi_utils

The server pi creates a service browser object that scans the local network for broadcasting services. When a the LED service is found a global variable is set so that any curl requests don't go through until the LED Pi is up and running. A flask service is created that has three routes. The first is get canvas. Canvas api is used to access the files. Then wget is used to download the file based on the url outputted by the json return statement sent from the request.get call. The second route uses the marvel api to search for stories based on a story id. Hashlib is used because the public key, private key, and timestamp must be hashed together in the url. Since stories cannot be downloaded without paying money, our team saved the json output of the get request to a text file called MarvelStory.txt. The last route is called getColor. When a correctly formatted request is sent as well as the led global variable being set to true(when the LED Pi broadcast is found) a get request is sent to the led raspberry pi to with 3 variables to set status, color, and intensity.

Client ends to Services Rpi
curl –u admin:secret http://192.168.43.38:8081/LED?command=on-red-50

Services RPi send to LED RPi
requests.get(http://pumpkin-pi.local:8081/LED?status=on&color=red&intensity=50)