

Assignment 3

My API

Due Date: Sunday, April 21, 2019 @ 23:55

ECE 4564 - Network Application Design

Learning Objectives

RESTful API

- Python Requests Library
- Flask Microframework

Service Advertisement

- Zeroconf

cURL

Raspberry Pi GPIO

noSQL

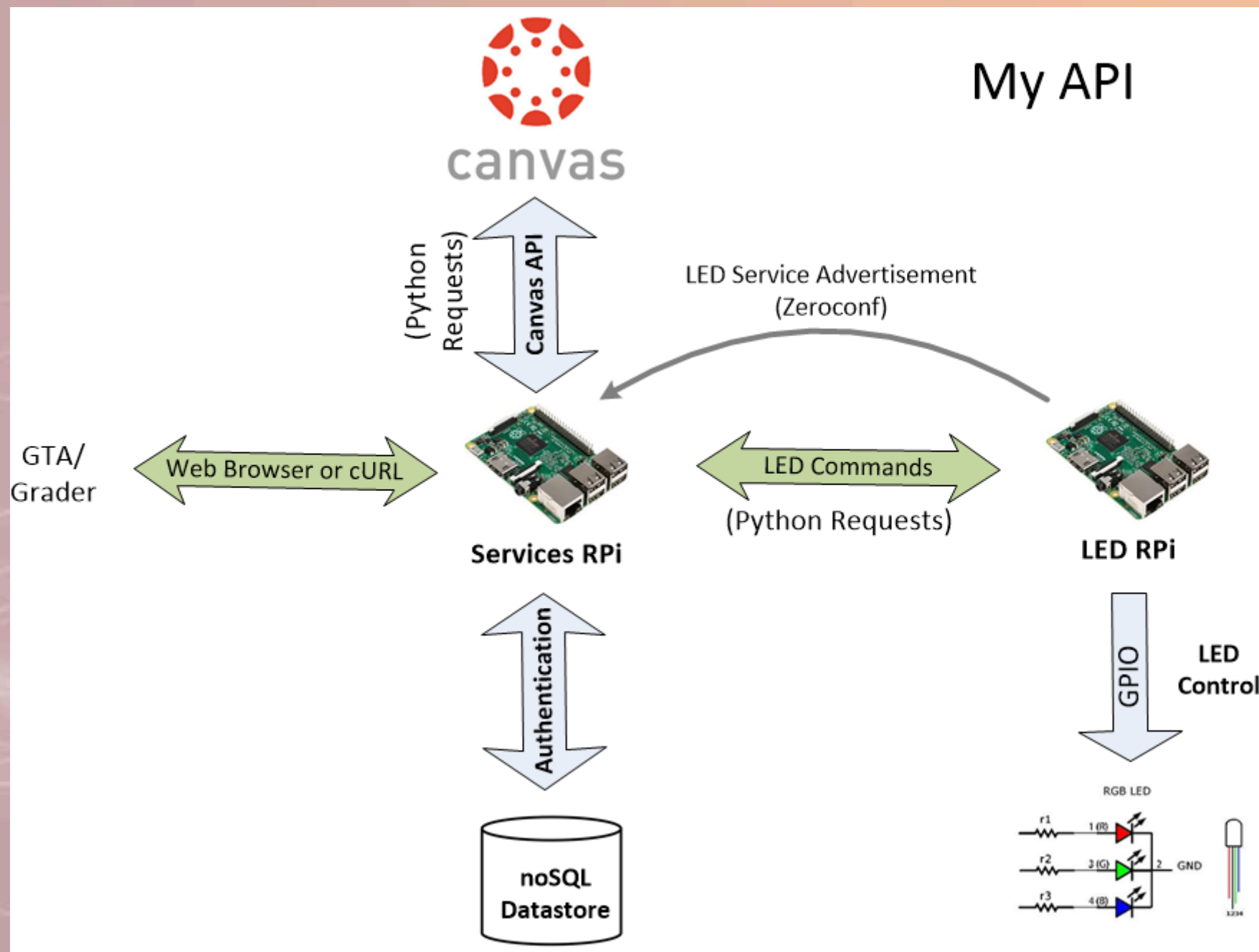
- MongoDB

Overview

Demonstrate web service interactions using REST

- Build Requirements
 - Flask Microframework
 - Python Requests Library
 - HTTP Basic Authentication
 - Auth data maintained in MongoDB datastore
 - Service Advertisement
 - Zeroconf
- Supported Services
 - RGB LED Controller
 - Canvas Interaction

System Overview



Interactions

Web service interactions via REST API

```
curl -u bad_user:bad_pass http://0.0.0.0:8081/goodbye
```

>Could not verify your access level for that URL. You have to login with proper credentials

```
0.0.0.0 - - [08/Nov/2017 14:43:59] "GET /goodbye HTTP/1.1" 401 -
```

```
=====
```

```
curl -u admin:secret http://0.0.0.0:8081/goodbye
```

>Goodbye World

```
0.0.0.0 - - [08/Nov/2017 14:38:15] "GET /goodbye HTTP/1.1" 200 -
```

Interactions

Web service interactions via REST API
GPIO

```
curl -u admin:secret "http://0.0.0.0:8081/LED?status=on&color=red&intensity=50"
```



Burning half as
bright ...

Interactions

Canvas

To services Rpi:

```
curl -u admin:secret "http://0.0.0.0:8081/Canvas?file=<file_name>"
```

From services Rpi to Canvas:

```
"https://vt.instructure.com/api/v1/courses/<course_id>/files/<file_id>?access_token=<token>" |  
python -mjson.tool
```

```
{  
  "content-type": "image/jpeg",  
  "created_at": "2017-09-19T22:55:03Z",  
  "display_name": "RGBLED-cathode.jpg",  
  "filename": "RGBLED-cathode.jpg",  
  "folder_id": 879378,  
  "hidden": false,  
  "hidden_for_user": false,  
  "id": 4896186,  
  "lock_at": null,  
  "locked": false,  
  "locked_for_user": false,  
  "media_entry_id": null,  
  "mime_class": "image",  
  "modified_at": "2017-09-19T22:55:03Z",  
  "size": 82276,  
  "thumbnail_url": "...",  
  "unlock_at": null,  
  "updated_at": "2017-09-20T20:01:13Z",  
  "url": "..."  
}
```

Service Authentication

- Use HTTP Basic Authentication
- Store user name and password in MongoDB datastore
- Authenticate all services

Warehouse : ECE4564_Assignment_3

Collection : service_auth

MongoDB Document

```
{
  "username": "Bud_Barclay",
  "password": "Polar-Ray_Dynasphere"
}
```


Zeroconf

- "The goal of the Zero Configuration Networking (Zeroconf) is to enable networking in the absence of configuration and administration.
- Zero configuration networking is required for environments where administration is impractical or impossible, such as in the home or small office, embedded systems 'plugged together' as in an automobile, or to allow impromptu networks as between the devices of strangers on a train.

Essentially, to reduce network configuration to zero (or near zero) in Internet Protocol (IP) networks, it is necessary, inter alia, to:

- Distribute IP addresses (without a Dynamic Host Configuration Protocol [DHCP] server)
- Provide name resolution (without a Domain Name System [DNS] server)
- Find and list services (without a directory service), and
- Distribute multicast IP addresses, if necessary (without a multicast server)."

Zeroconf Working Group of the Internet Engineering Task Force (IETF)

Assignment References

REST and Flask

[Serving Raspberry Pi with Flask](#)

[Designing a RESTful API with Python and Flask](#)

Flask and MongoDB

[Flask Rest API with MongoDB](#)

Python Requests

[Requests:HTTP for Humans](#)

cURL

[Conquering the Command Line](#)

Zeroconf (... start here ...)

<https://www.pronoy.in/2015/03/07/experiments-with-avahi-and-python-zeroconf/>

Assignment References

Authorization

[HTTP Basic Auth](#)

Canvas API

[Canvas LMS API Documentation](#)

[Obtaining a Token](#)

GPIO PWM

<https://sourceforge.net/p/raspberry-gpio-python/wiki/PWM/>

Grading

GTA will provide grading rubric

Python Style

Follow style guide PEP0008 when writing and commenting
your code

<https://www.python.org/dev/peps/pep-0008/>

What You Turn In

All assignments must be submitted through Canvas, no later than the due date of Sunday, April 21, 2019 @ 11:55pm

Note: Teams will receive a 10 point deduction per day past the due date

Your assignment should be a single tar gz (tgz extension) which contains the following:

- All source code you wrote for this assignment
 - Python code running on Capture and Repository Rpi's
 - Client code identified as "services.py"
 - Server code identified as "led.py"
 - api/dev key file as "servicesKeys.py"
- Report (PDF file)

Be sure to name the tar gz file as follows: HW3_TeamXX.tgz

Academic Integrity

- For this assignment, it is expected that a team's work is their own.
- The code you turn in must be your own (i.e. you need to have written your assignment).
- You are allowed to copy and paste example code from other websites, but you must include a comment in your code that attributes the website you copied the code from (i.e. original author's name and URL to the original code).
- You can discuss the assignment with other teams.
- However, you cannot just tell another team the answer to a particular problem.

Final Thoughts

In many cases, engineers are expected to just make things work given a particular design constraint (e.g. software package to use or are limited to a particular hardware platform).

You will likely run into similar situations in this class while designing and implementing your assignments and project.

.

When you're stuck, try searching online for a solution. Many times others have tried something similar and documented their experiences for others to learn and benefit from

If you find a neat way of doing something on your Raspberry Pi, please share your findings in a discussion post on Canvas.

Do not publically post answers to assignments, or example code until after the assignment due date.

Contact your instructor or GTA as soon as you encounter a problem you're unable to solve.

Don't wait until right before the assignment is due.