Teaching IoT: Practical, Industry-Relevant, Hands-On, and Systems-Level Approaches

Nick Barendt, Case Western Reserve University
Dr. Nigamanth Sridhar, Cleveland State University

Today's Speakers

Nick Barendt, Case Western Reserve University
Executive Director, Institute for Smart, Secure, and Connected Systems

Nigamanth Sridhar, Cleveland State University
Professor, Computer Science

Overview

We will talk about an approach to designing a course that provides hands-on experiences to students

- Focus is on enabling students to translate what they have learned in other parts of degree program into practical skills
- Teaching students how to embrace and learn different technology alternatives

Introduction to Connected Devices Course Principles

Introduction to Connected Devices Course Principles

Introduction to Connected Devices Course Principles

- Hands-on, product development arc

Introduction to Connected Devices Course Principles

- Hands-on, product development arc
- Keep it real, both technologies and practices

Introduction to Connected Devices Course Principles

- Hands-on, product development arc
- Keep it real, both technologies and practices
- Don’t dumb it down

Introduction to Connected Devices Course Principles

- Hands-on, product development arc
- Keep it real, both technologies and practices
- Don’t dumb it down
- No black boxes

Introduction to Connected Devices Course Principles

Introduction to Connected Devices Course Principles

Structure lectures to explain theory and concepts, and courseware and assignments to translate that into real world, reinforcing lessons

Introduction to Connected Devices Course Principles

Structure lectures to explain theory and concepts, and courseware and assignments to translate that into real world, reinforcing lessons

Organize content into two parts

Introduction to Connected Devices Course Principles

Structure lectures to explain theory and concepts, and courseware and assignments to translate that into real world, reinforcing lessons

Organize content into two parts

1. Building the IoT Product System

Introduction to Connected Devices Course Principles

Structure lectures to explain theory and concepts, and courseware and assignments to translate that into real world, reinforcing lessons

Organize content into two parts

1. Building the IoT Product System
2. System Properties, Behaviors, and Concerns

Introduction to Connected Devices Course Principles

Structure lectures to explain theory and concepts, and courseware and assignments to translate that into real world, reinforcing lessons

Organize content into two parts

1. Building the IoT Product System
2. System Properties, Behaviors, and Concerns

Introduction to Connected Devices Course Principles

Structure lectures to explain theory and concepts, and courseware and assignments to translate that into real world, reinforcing lessons

Organize content into two parts

1. Building the IoT Product System
2. System Properties, Behaviors, and Concerns

Barendt, N., & Sridhar, N., & Loparo, K. A. (2018, June), *A New Course for Teaching Internet of Things: A Practical, Hands-on, and Systems-level Approach* Paper presented at 2018 ASEE Annual Conference & Exposition , Salt Lake City, Utah. 10.18260/1-2--29706

Introduction to Connected Devices

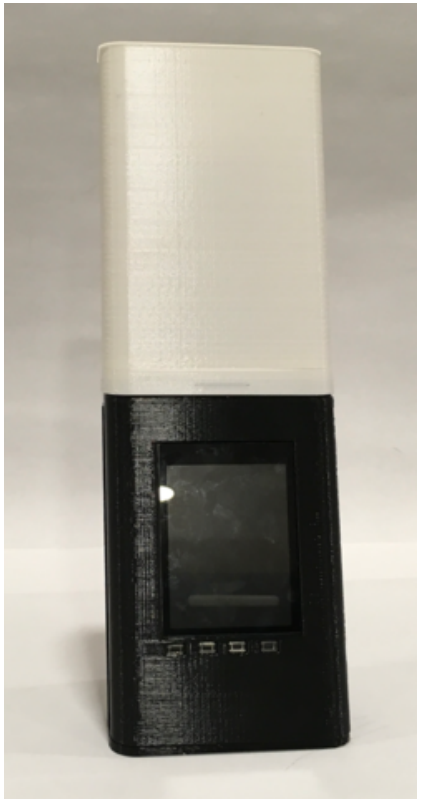
3 Credit Hour, 1 Semester

Junior/Senior EE, CE, CS Technical Elective

Hands-on, Internet-of-Things (IoT) introductory course

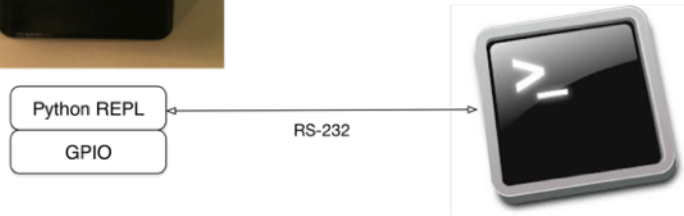
PART 1 – Building the IoT Product System

Week 01 – Assemble LAMPI & Blink LEDs

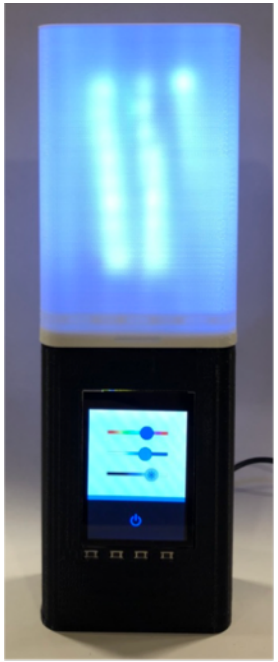


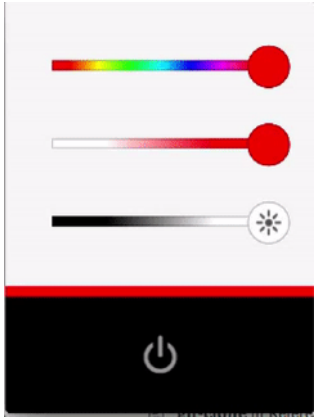
PART 1 – Building the IoT Product System

Week 01 – Assemble LAMPI & Blink LEDs



PART 1 – Building the IoT Product System



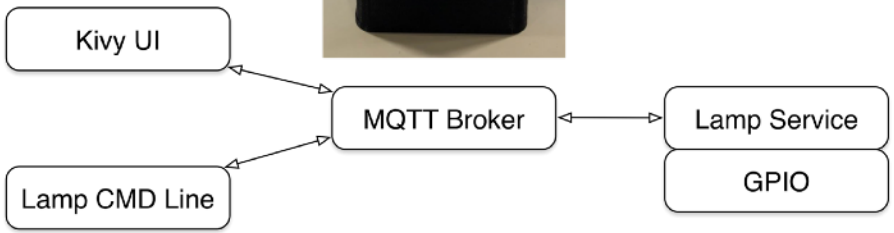


Kivy UI

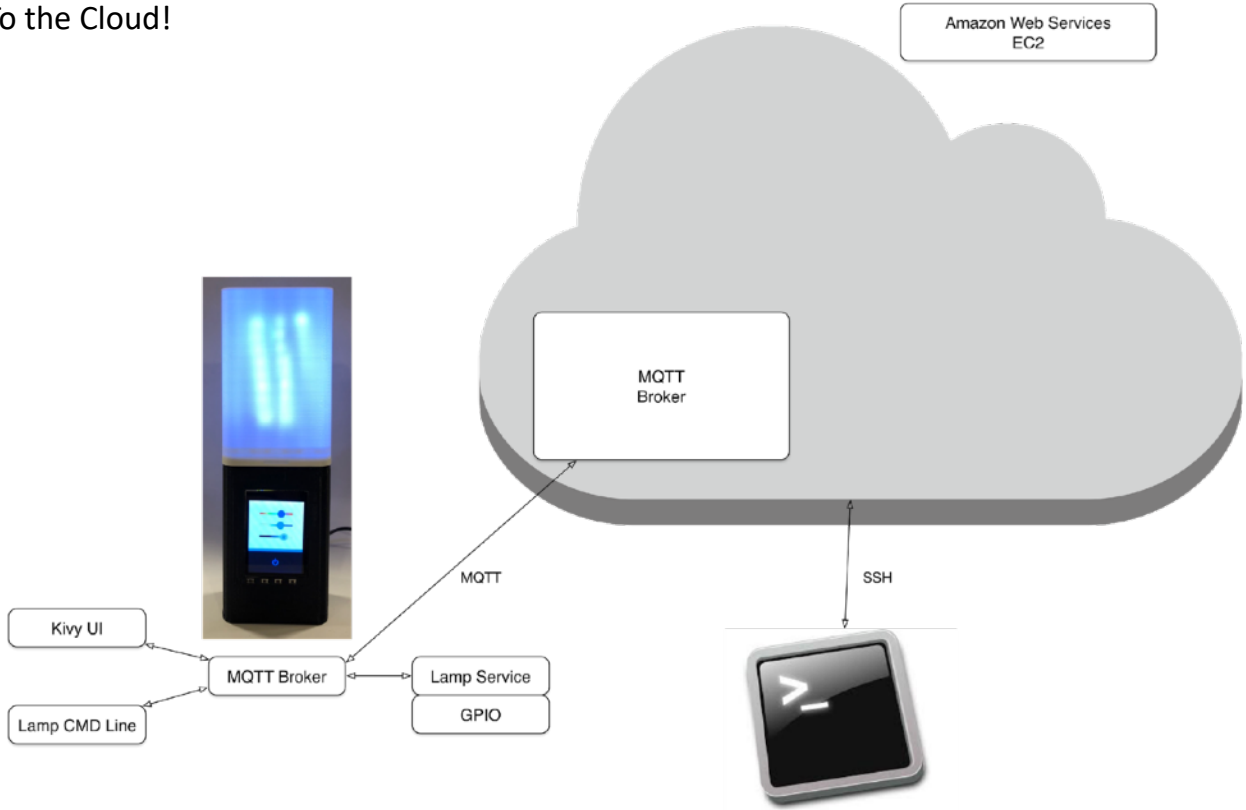
GPIO

Week 03 – Publish Subscribe

PART 1 – Building the IoT Product System

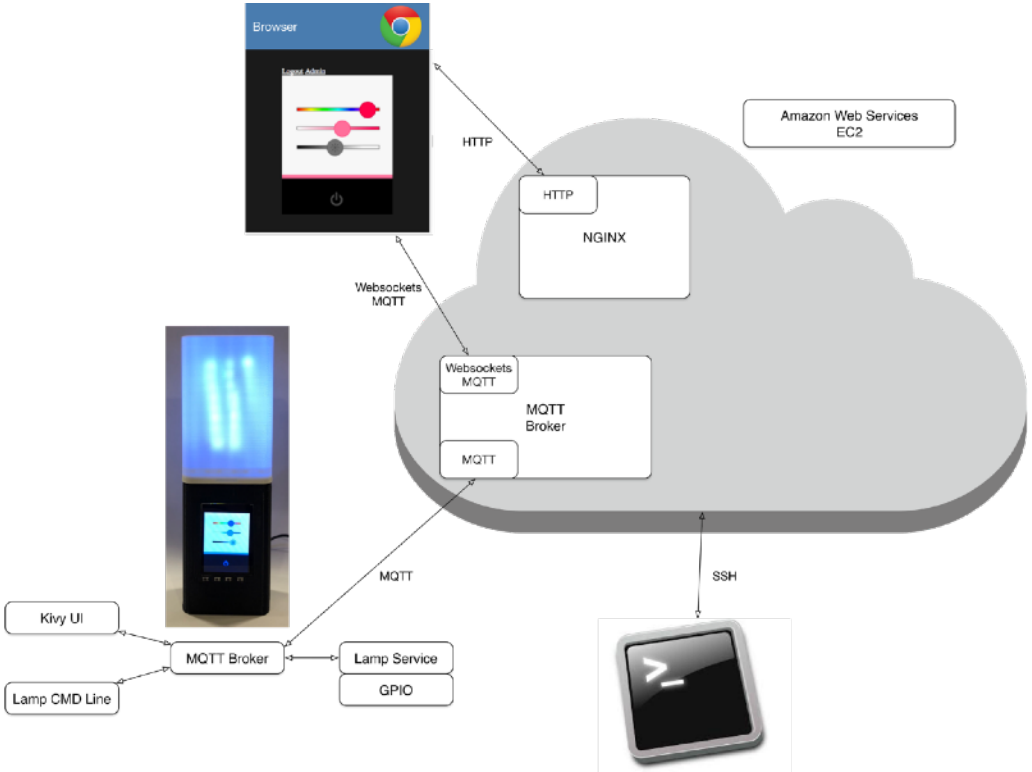

Kivy UI
MQTT Broker
Lamp Service
GPIO
Lamp CMD Line

Week 04 – To the Cloud!

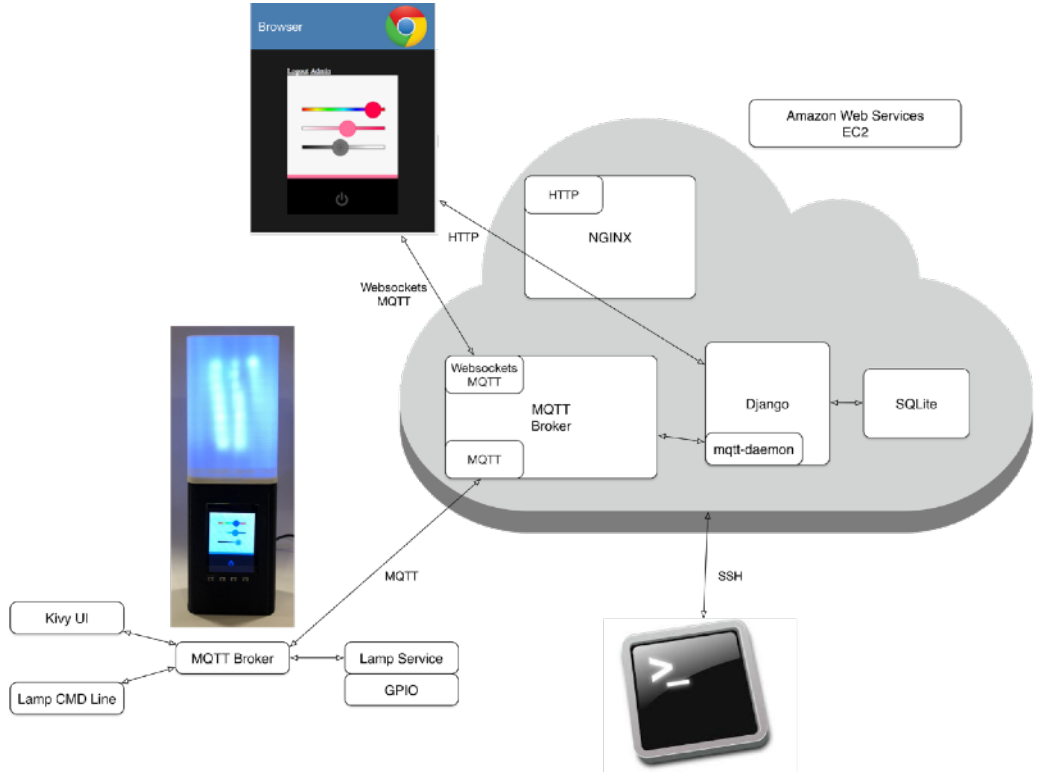


PART 1 – Building the IoT Product System

Week 05 – Web User Interfaces (Static)

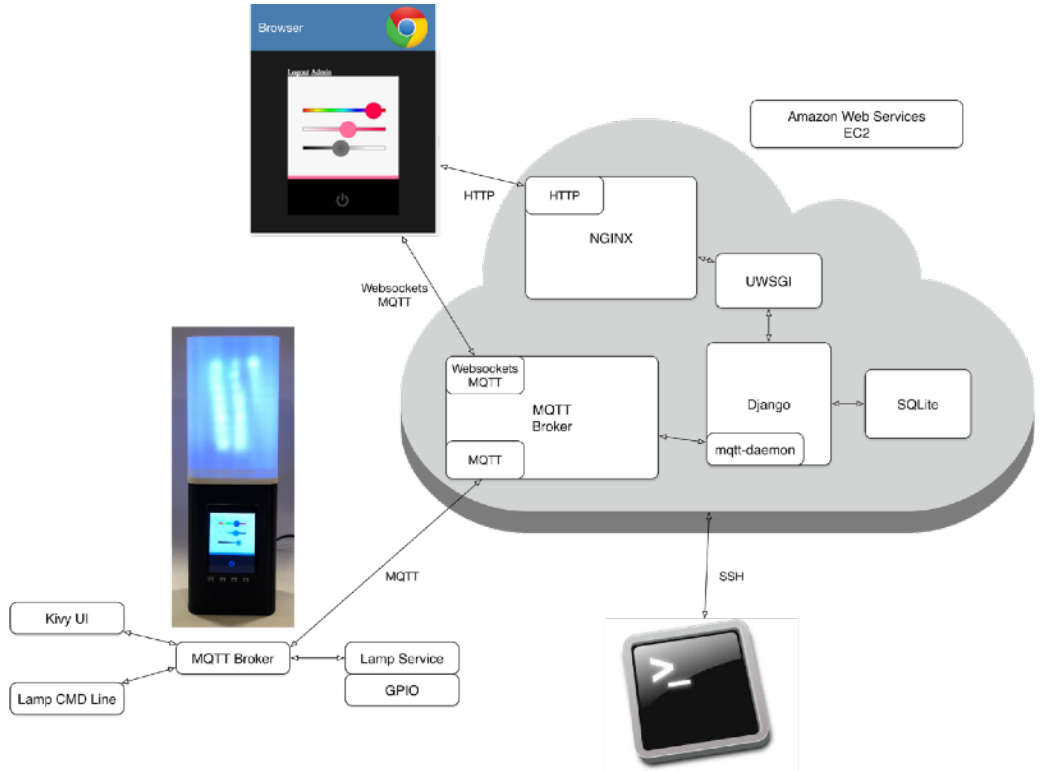


PART 1 – Building the IoT Product System

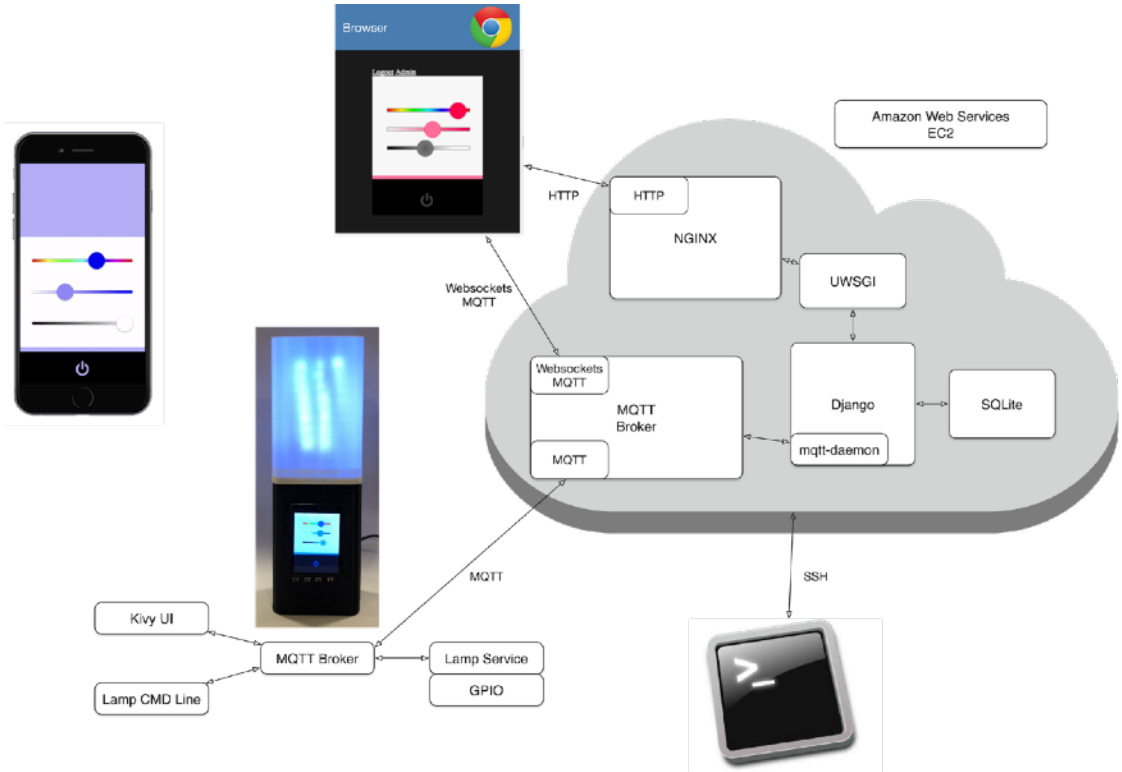


PART 1 – Building the IoT Product System

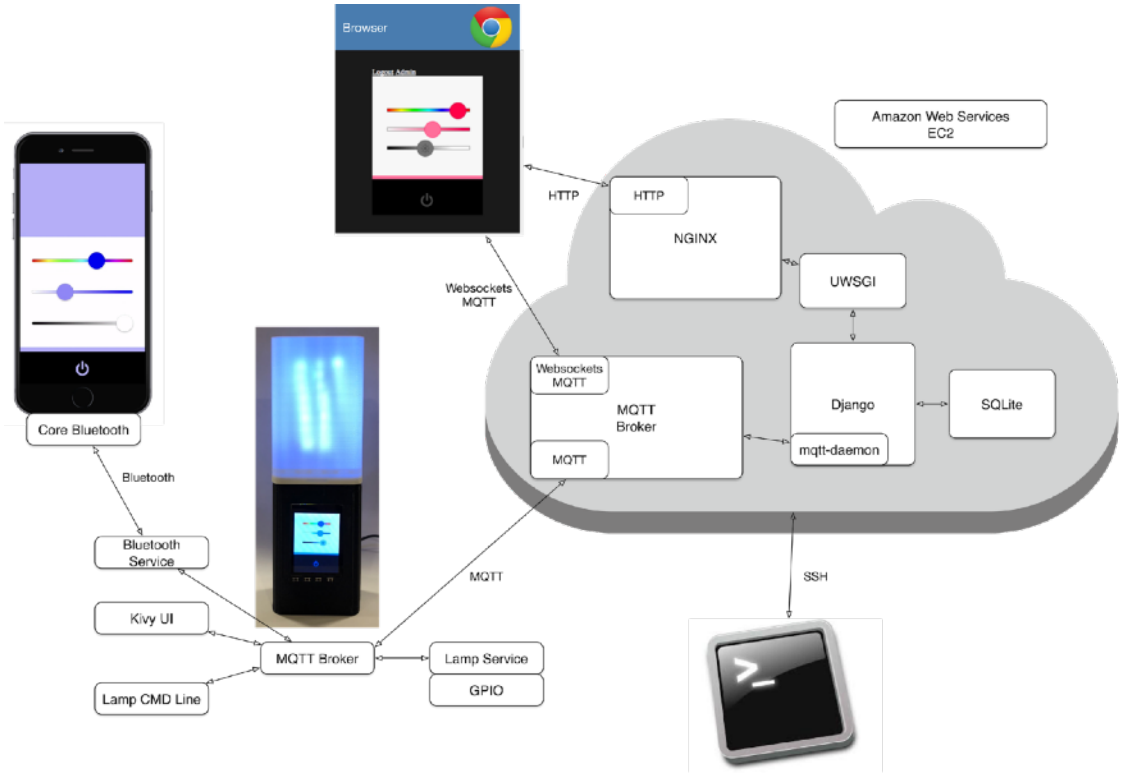
Week 07– User-Device Association & Production Deployment



PART 1 – Building the IoT Product System

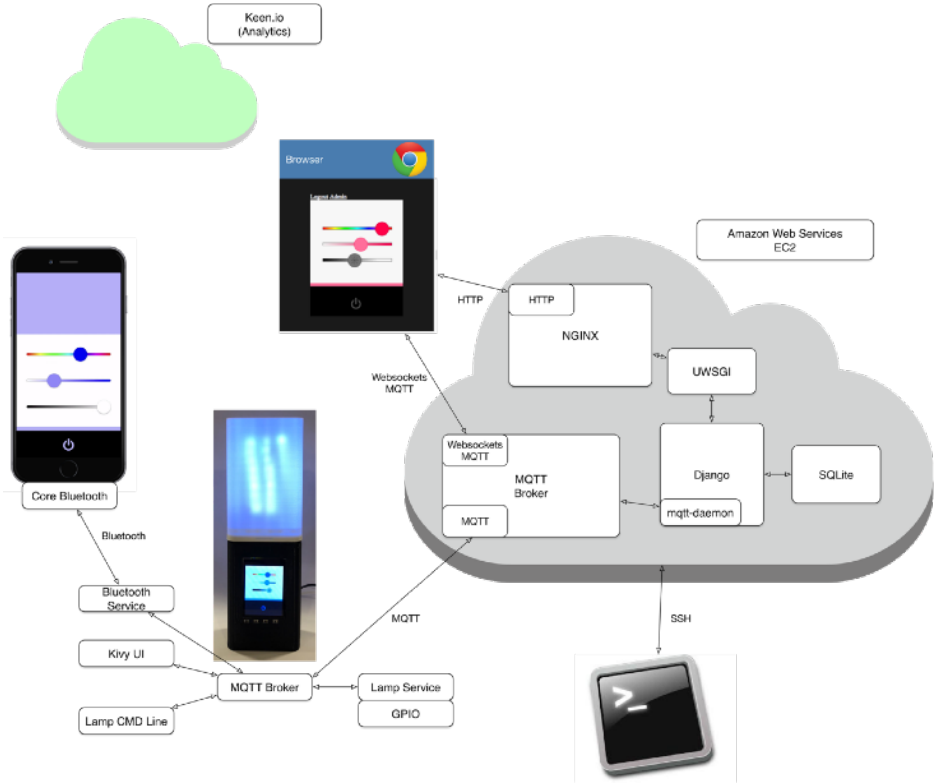


PART 1 – Building the IoT Product System

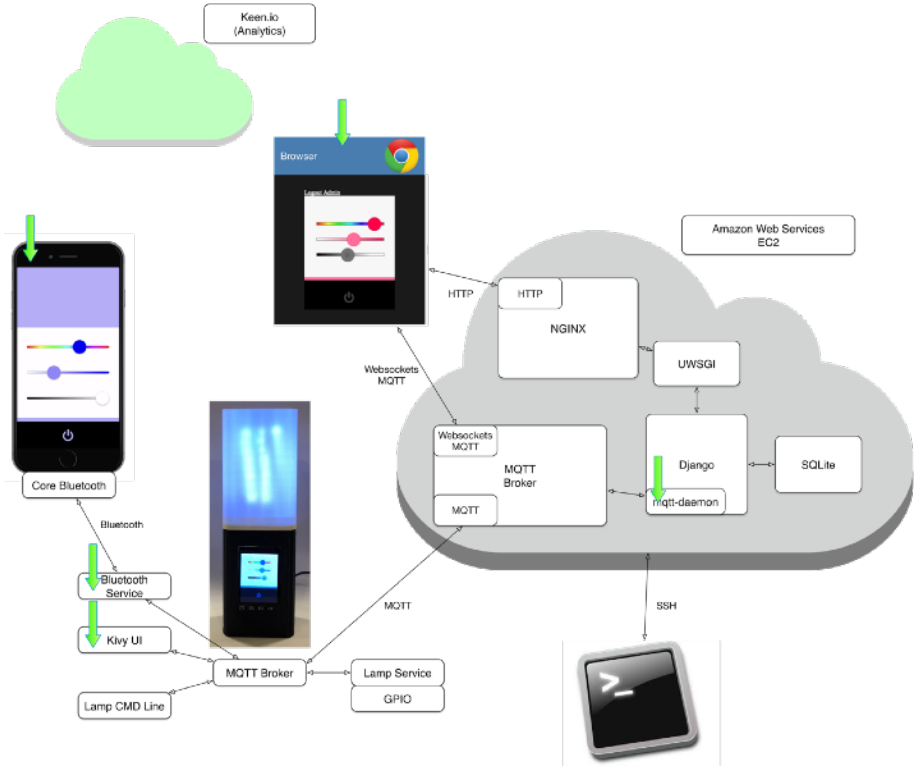


PART 1 – Building the IoT Product System

Week 10– Introduction to Analytics & Dashboards

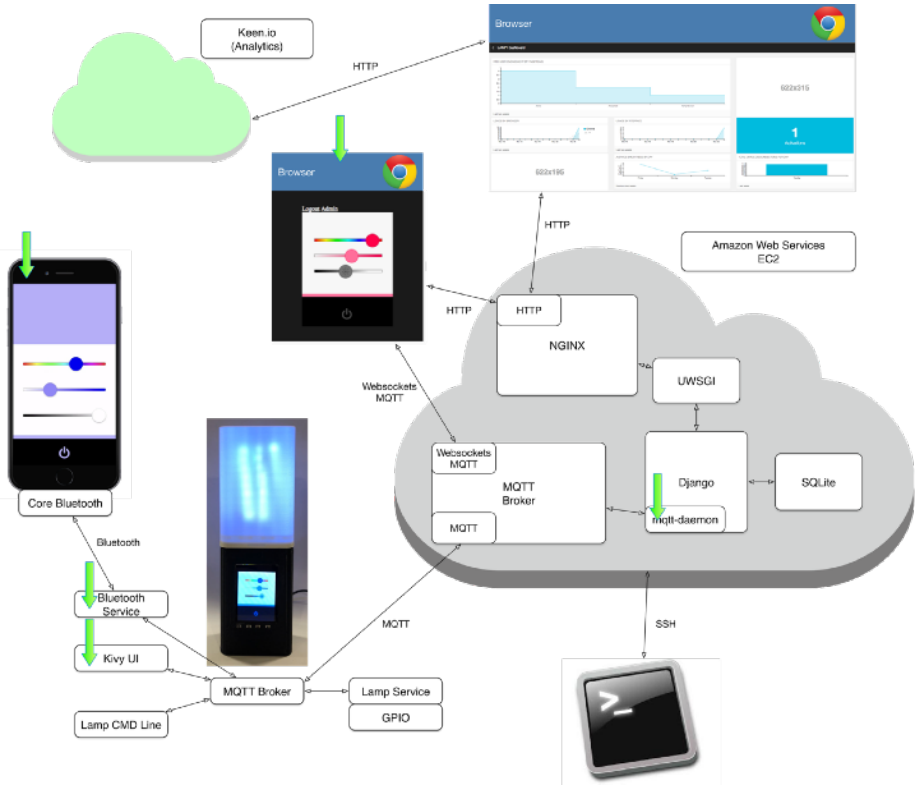


PART 2 – System Properties, Behaviors, and Concerns



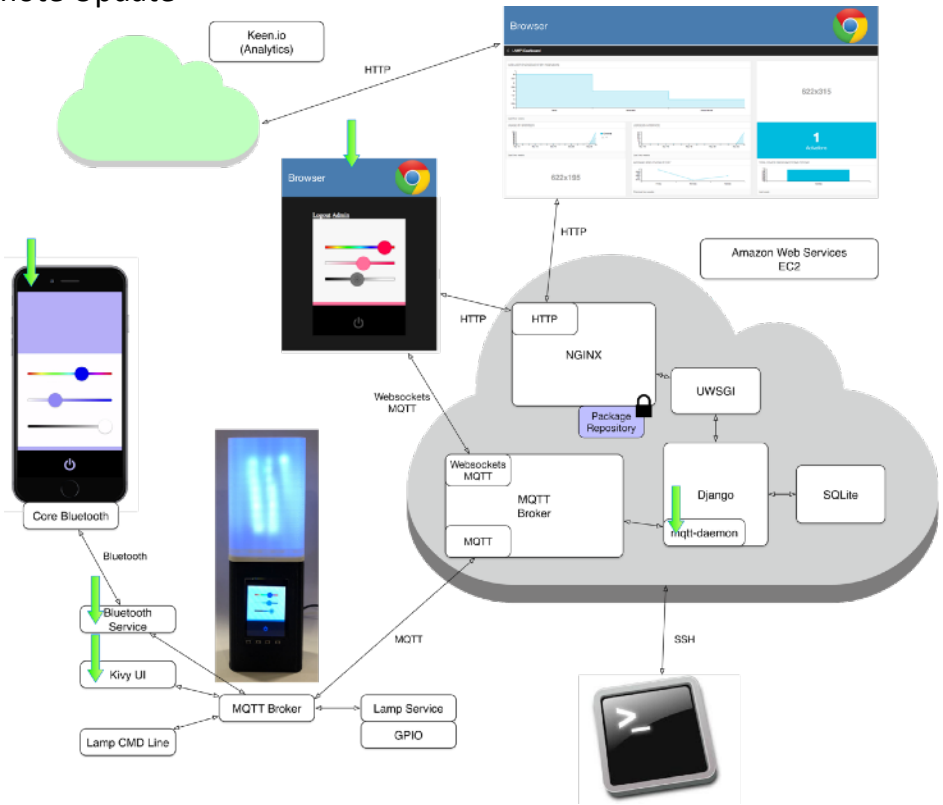
PART 2 – System Properties, Behaviors, and Concerns

Week 10– Introduction to Analytics & Dashboards



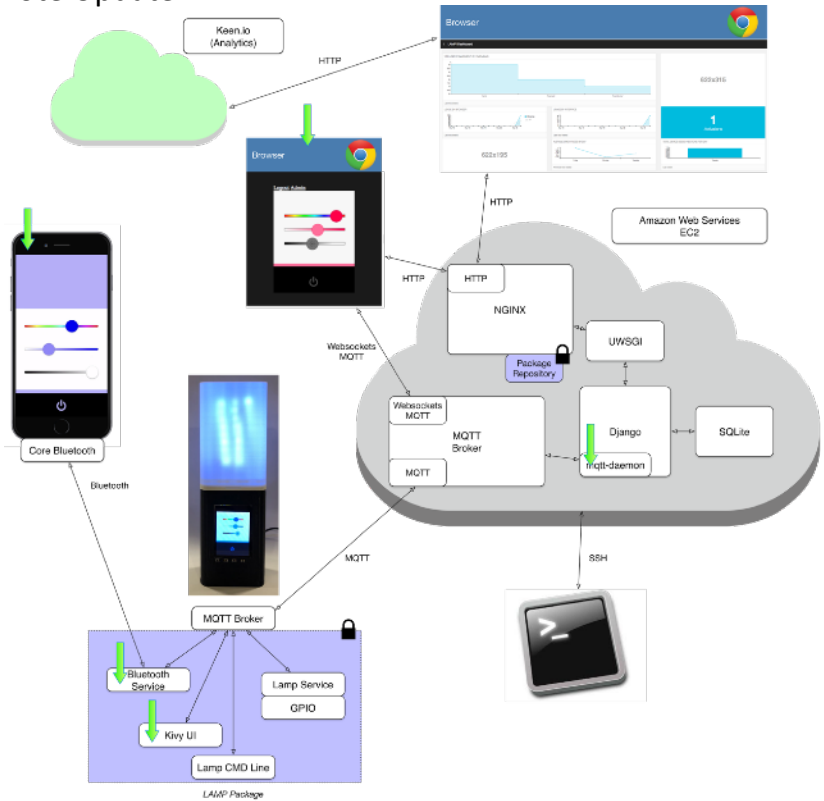
PART 2 – System Properties, Behaviors, and Concerns

Week 11– Secure Remote Update



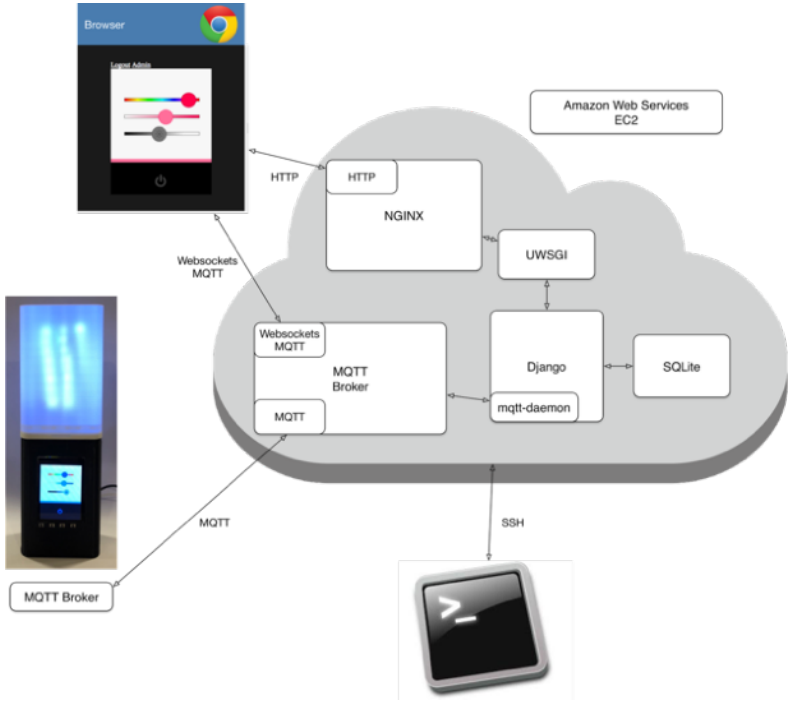
PART 2 – System Properties, Behaviors, and Concerns

Week 11– Secure Remote Update



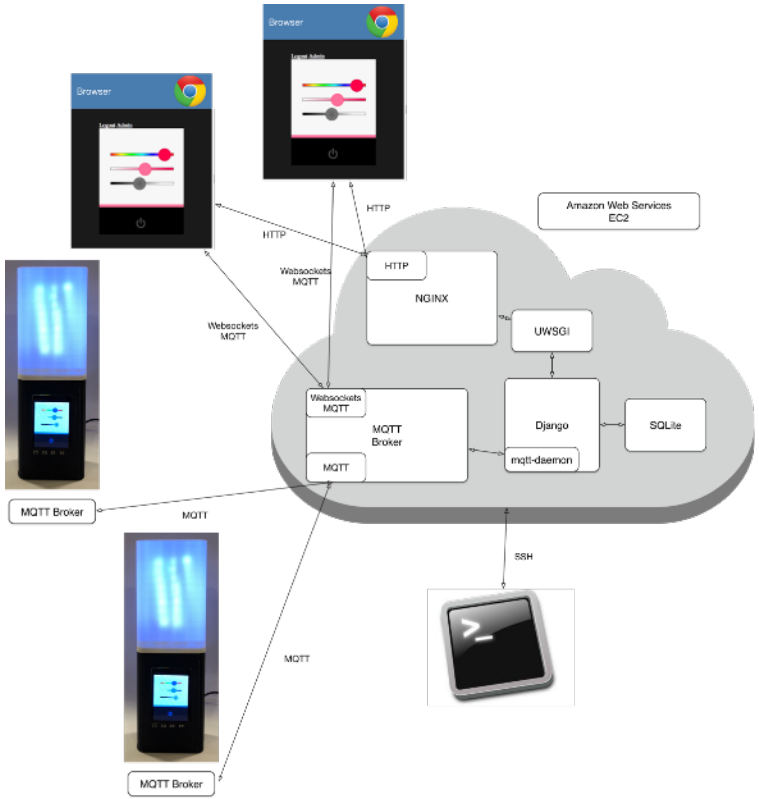
PART 2 – System Properties, Behaviors, and Concerns

Week 12– User Modelling, Load Testing, and Scaling



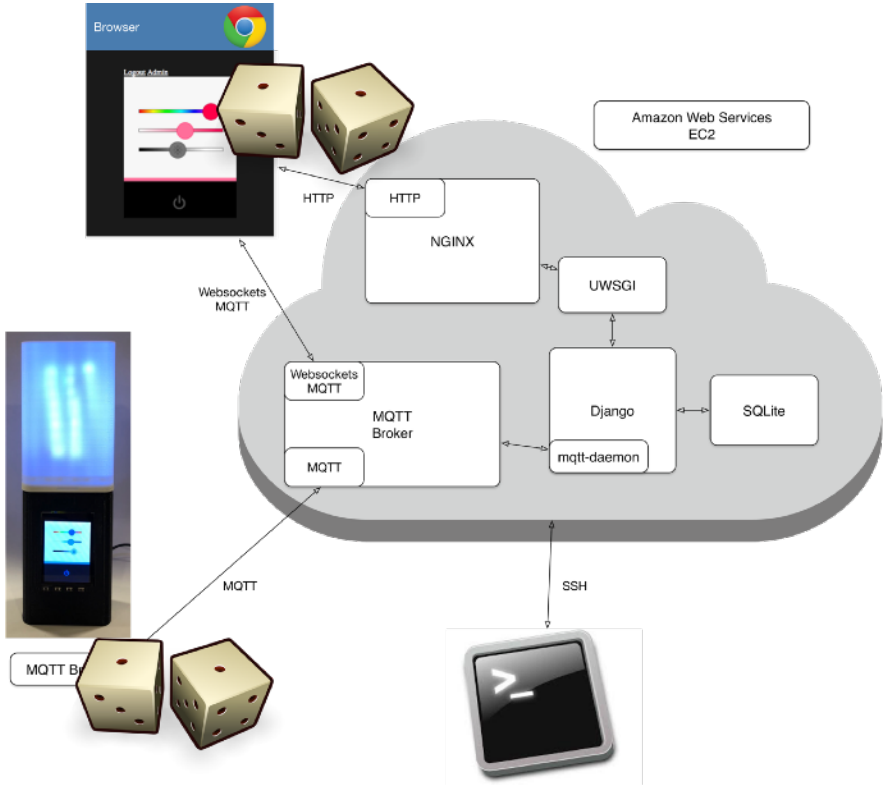
PART 2 – System Properties, Behaviors, and Concerns

Week 12– User Modelling, Load Testing, and Scaling



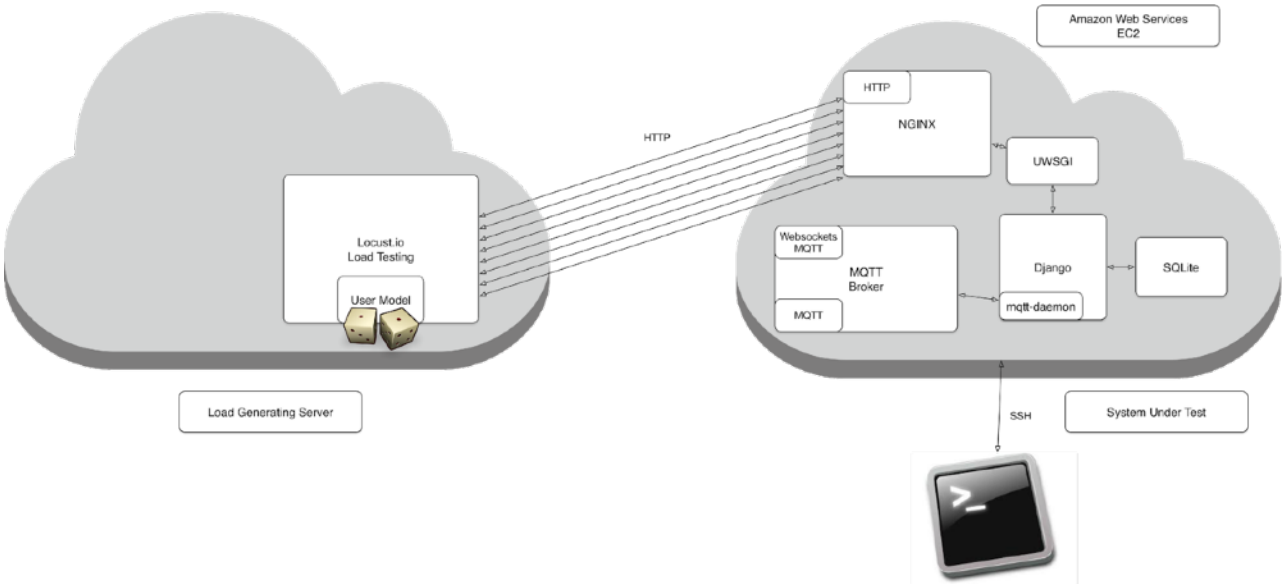
PART 2 – System Properties, Behaviors, and Concerns

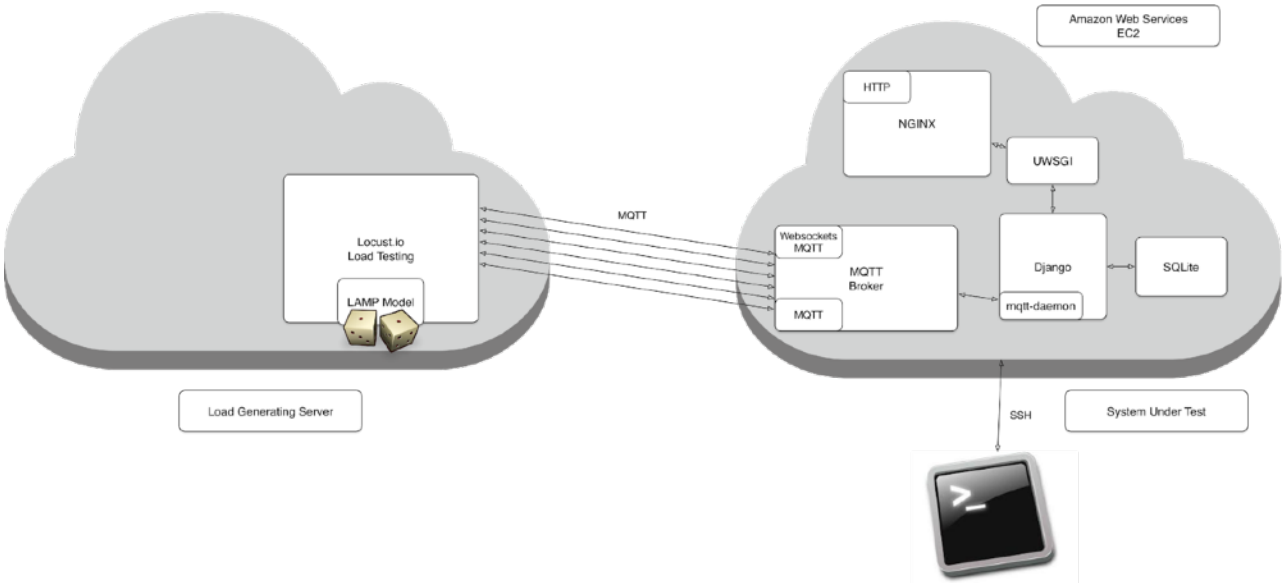
Week 12– User Modelling, Load Testing, and Scaling



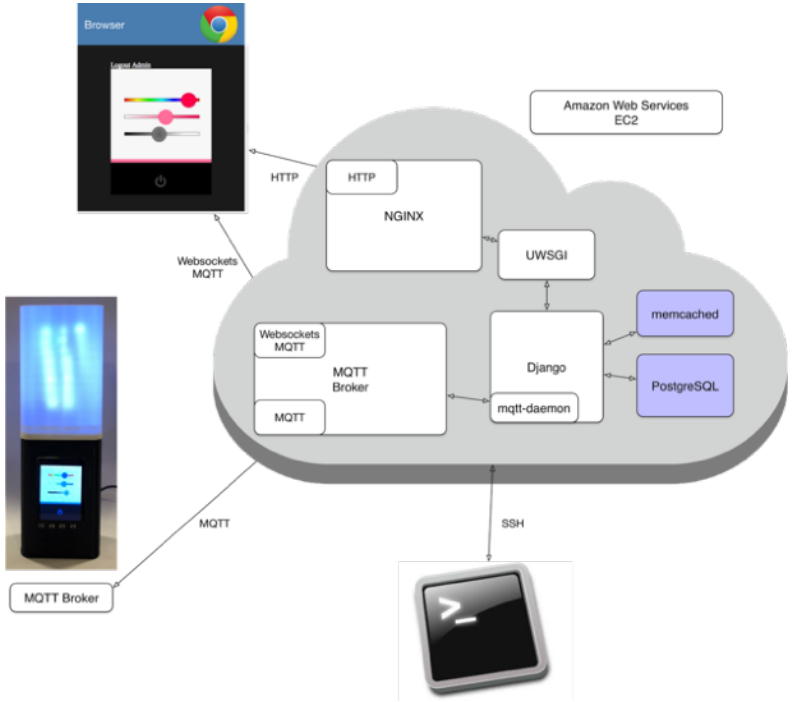
PART 2 – System Properties, Behaviors, and Concerns


Locust.io
Load Testing
Load Generating Server
Amazon Web Services
EC2
HTTP
NGINX
UWSGI
Websockets
MQTT
MQTT
Broker
MQTT
Django
mqtt-daemon
SQLite
SSH
System Under Test


Amazon Web Services
EC2
HTTP
NGINX
UWSGI
HTTP
Locust.io
Load Testing
User Model
Websockets
MQTT
MQTT
Broker
MQTT
Django
mqtt-daemon
SQLite
Load Generating Server
SSH
System Under Test

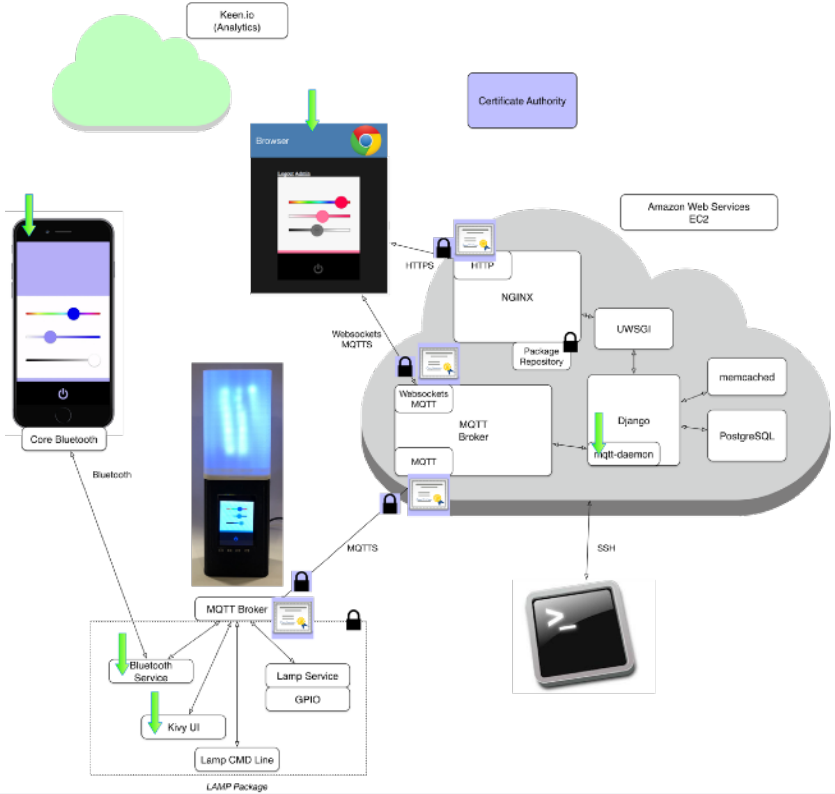

Amazon Web Services
EC2
HTTP
NGINX
UWSGI
Locust.io
Load Testing
LAMP Model
MQTT
Websockets
MQTT
MQTT
Broker
MQTT
Django
mqtt-daemon
SQLite
Load Generating Server
SSH
System Under Test

Week 12– User Modelling, Load Testing, and Scaling



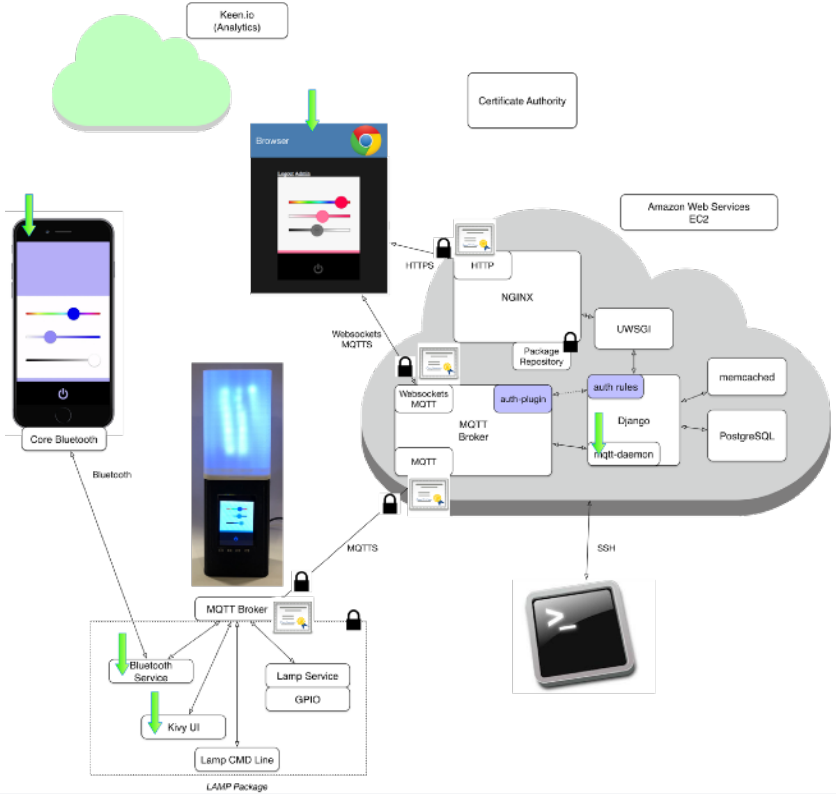
PART 2 – System Properties, Behaviors, and Concerns

Week 13– Essential Security – SSL/TLS and Certificates



PART 2 – System Properties, Behaviors, and Concerns

Week 14– Rules-Based Access Control



PART 2 – System Properties, Behaviors, and Concerns

Some examples from previous classes

- Alexa
- Google Home
- Hololens AR
- Real-Time Audio Beat Light Synchronization
- Enchanted Objects – Student Dorm Experience
- Integrating a Trusted Platform Module (TPM)
- Integrating Hyperledger Blockchain
- Bluetooth Beacon UI Behavior
- Light-based Alarm Clock
- Integrating Weather Forecast with Audio and Visual Cues
- Integrating Apple Watch

Background – Why this Course was Created

Why

- Perceived gap in undergraduate curriculum – prepare students to build, deploy, maintain Internet of Things (IoT) products and services
- Student challenges
  - IoT typically involves multiple technologies and languages
  - little experience with large software systems
- Many existing courses focus on one platform (e.g., embedded, mobile, or cloud) or are graduate level

Background –How this Course was Created

How

- Partnership between higher education and industry
  - Expert industry practitioners from multiple disciplines (UX, mobile, cloud, embedded, IoT, DevOps)
  - Solid academic foundation
- Built incrementally and delivered first as a Special Topics course in Fall 2015
- Six offerings to date

Related Courses

Related Courses

ECSE 377 - Introduction to Connected Devices (Spring ‘21)

Related Courses

ECSE 377 - Introduction to Connected Devices (Spring ‘21)

ECSE 397 - Special Topics: Designing IoT Edge Devices (Fall ‘20)

Related Courses

ECSE 377 - Introduction to Connected Devices (Spring ‘21)

ECSE 397 - Special Topics: Designing IoT Edge Devices (Fall ‘20)

ECSE 397 - Special Topics: Embedded Computing for Music! (Spring ‘21)

Courseware

- Self-Contained – Content (Markdown) Text & Images + Source Code
- Resides in a Git repository
- Two Git repositories
  - Instructor
  - Student (exists for one term)
- Instructor repository has novel branch structure:

  chapter_01_assignment
  chapter_01_solution (includes everything in *chapter_01_assignment*)
  chapter_02_assignment (includes everything in *chapter_01_solution*)
  chapter_02_solution (includes everything in *chapter_02_assignment*)
  chapter_03_assignment (includes everything in *chapter_02_solution*)
  …
- Branch structure facilitates incremental release to students and content maintenance

Challenge – Week N+1 builds on Week N

- **Challenge**
  - Week N+1 Assignment relies on correct solution to Week N
  - Student solution to Week N might work, but not be optimal for Week N+1 objectives
  - Solutions require source code + system configuration (e.g., package installation and configuration)
  - Solutions involve multiple computing environments (RPI3, AWS EC2, XCode, etc.)
  - Students that are challenged by Week N cannot be left behind

Solution – Add DevOps Solution Sets

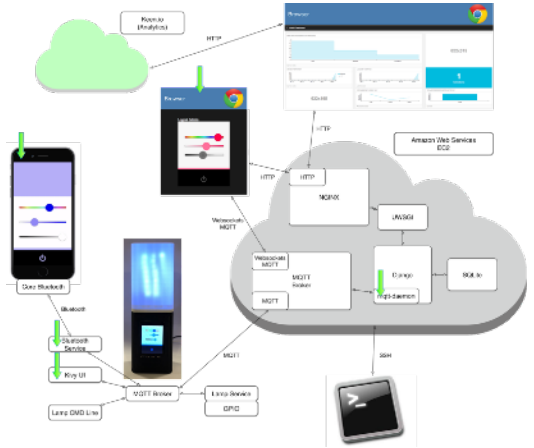
- **Solution** DevOps Solution Sets (Ansible in our case)
  - Week N solution set includes executable configuration commands
    - Installs & configures 3rd party packages for all platforms (RPI3, AWS EC2, Xcode, etc.)
    - Deploys working solution code (forcibly, when required)
- Each week, students run Ansible solution
  - Ansible manipulates all student environments to a known working state
- Bonus: also allows “fast-forwarding” environments from vanilla installs to any week in semester (point-in-time)

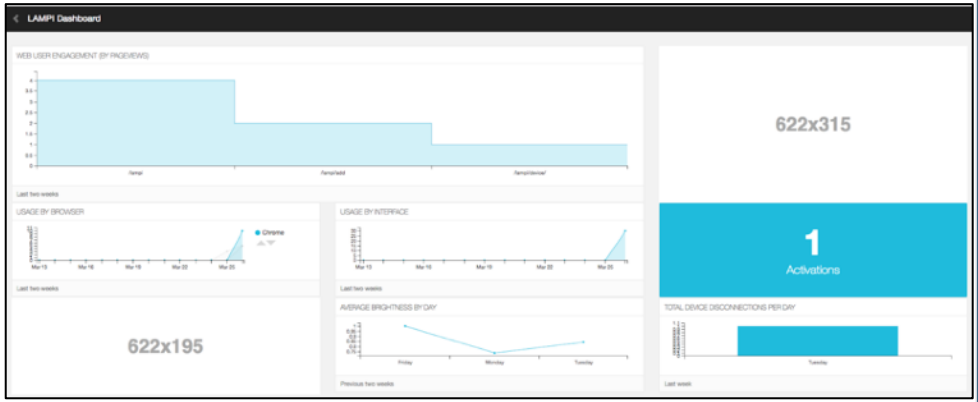
Courseware Sample
Week 10 - Analytics

Preparation

Run Ansible for Week 09 Solution (Introduction to Bluetooth Low Energy)

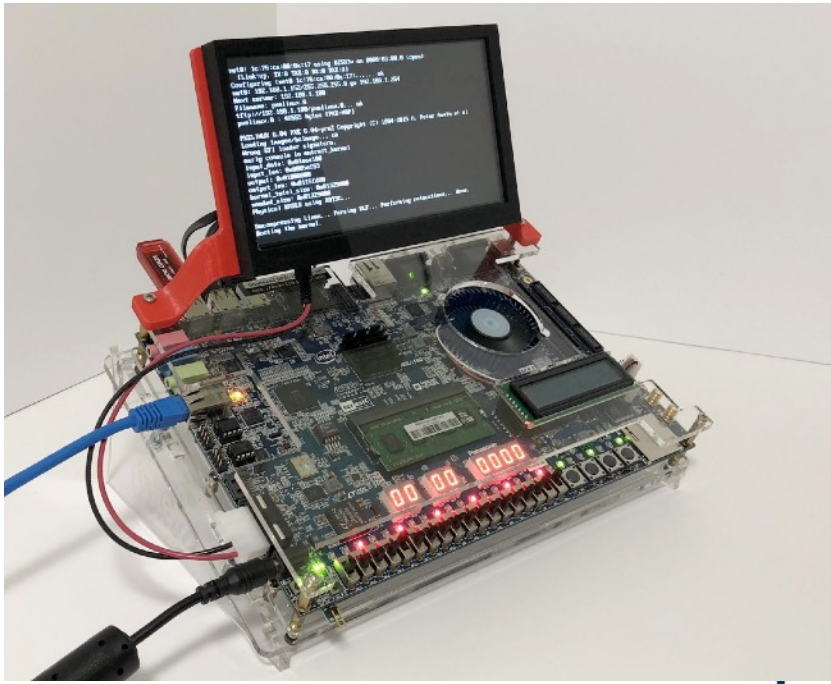
- Includes *everything* from Week 01 through Week 09
- Updates
  - RPI3B (Raspbian/Debian Linux)
  - AWS EC2 (Ubuntu/Debian Linux)
  - iOS App
- Installs all needed repositories, packages, and configurations





Designing IoT Edge Devices

- Course made possible by the generous support of Intel Corporation
- First offered Spring 2020
  - Revamped for Fall 2020 semester
  - Remote Lab (Web/Remote-Synchronous)
- Project-Based, Hands-On Laboratory
- Introduce Students to IoT Edge Device Development using Embedded Linux



Designing IoT Edge Devices

Students learn how to:

- Build custom Linux images using the Yocto Project
- Use and create Makefiles and BitBake recipes
- Configure Linux system services
- Develop and debug sensor device drivers
- Build IoT communication applications using HTTP, MQTT, WebSockets, etc
- Use AWS services (Lambda, DynamoDB, SNS, IoT, etc)
- Advanced SSH usage (SSH Keys, Port Forwarding, Proxy Connections)

Conclusions and Takeaways

- It is possible to teach a hands-on IoT course in an undergraduate academic setting
- Courseware, supported by DevOps, can be systems-level (multi-platform)
- Two-phase approach (build the system, then explore systems properties) successful
- Students are challenged and rave about the course and its impact on their development
- Connection to industry is really valuable
  - In the US, industry members participate as adjunct faculty

Questions?