Dispatcher

-- a service to connect businesses with people

**Johan Burke, Justin Gavin, Stanimir Genov,
Steve Snyder, Sean Fitzpatrick, Jake Schooley**
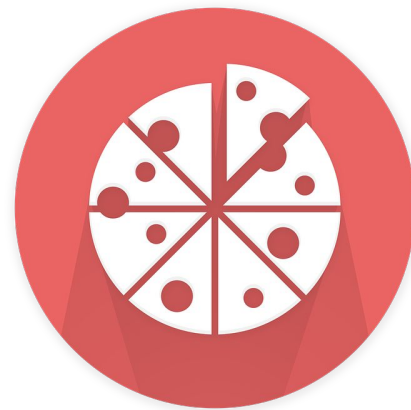
Overview

- Inspired by the **"Uber"** model
  - A pool of workers (**"Drivers"**)
  - Drivers may be associated with zero to many businesses
  - Businesses create **"Jobs"**
  - One Driver may claim and complete any given job

- Not industry specific - any delivery/courier/offsite service industry

Example Use Case -- Bob's Pizza Shop

- No delivery drivers;
  now wish to offer delivery

- Solution: subscribe to *Dispatcher*; approve a number of drivers through the service
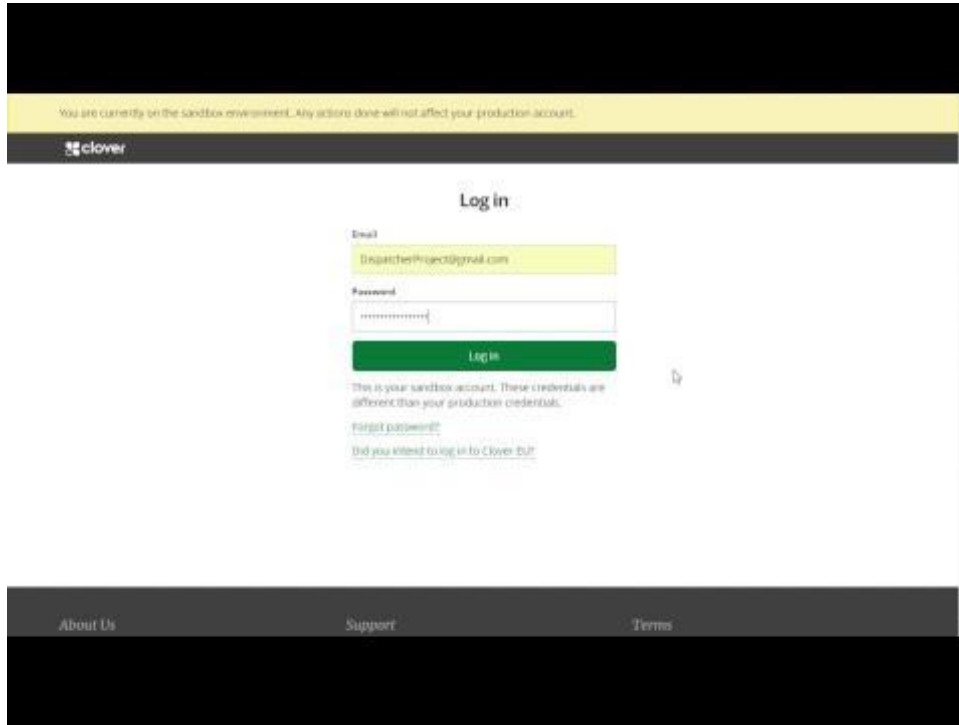
Example Use Case -- Bob's Pizza Shop

On reception of delivery order:

- Create job in *Dispatcher* app

- Notification is sent to Bob's network of drivers

- One driver claims the job, picks up the order, and performs

  the delivery

Video Demo

[Online Link](#)

[Local Link](#)

Supporting Technologies -- Backend

- AWS EC2 instance -- Ubuntu 16.04.3 LTS

- MySQL 5.7

- Apache HTTPD 2.4 & mod_wsgi module

- Flask; a python-based web development microframework

Supporting Technologies -- Frontend

- Materialize CSS

- jQuery

Supporting Technologies -- Clover

- For business authentication - oauth

- Clover REST API for business information

- *Future:* all financial transactions

Supporting Technologies - Clover OAuth
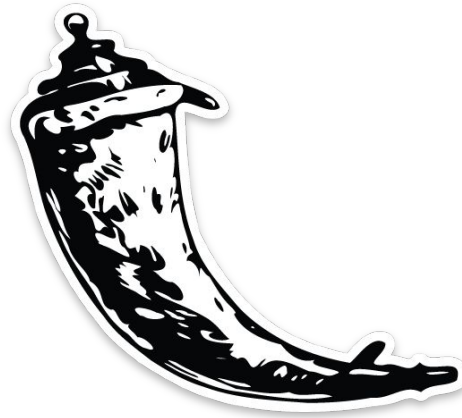
Supporting Technologies --Twilio

- Sending out notification of new job

- Sending out confirmation code when deregistering
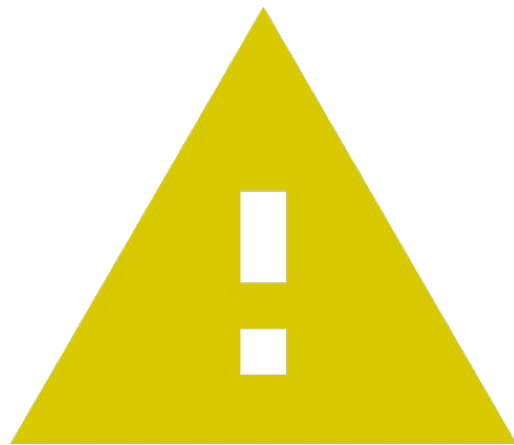
The Takeaway - **Flask**

Flask was the keystone of this project

- The combination of:
  - Stupid-easy syntax (*python*: pseudocode that runs)
  - Excellent documentation
  - Popularity - many existing packages to extend core Flask functionality
  - It "just works" -- no barriers to development

Obstacles

- New technologies

- Android app development -- dropped

- Oauth

- SMS service utilization

Acknowledgements

- Dr. Baliga

- James Strate

- Clover support

Questions