Computer Science Society

Web Backends Oct 13 2016

CSS Sign In:
https://tinyurl.com/z4xxv33

Announcements

- ACM competition November 12 2016 at Riverside Community College
- Cal Poly Pomona can send up to 4 teams of **3 members each**
- **Tryouts Oct 22 at Computer Lab (8-52) 10:00AM**
- Practice previous years' problems: http://socalcontest.org/current/index.shtml

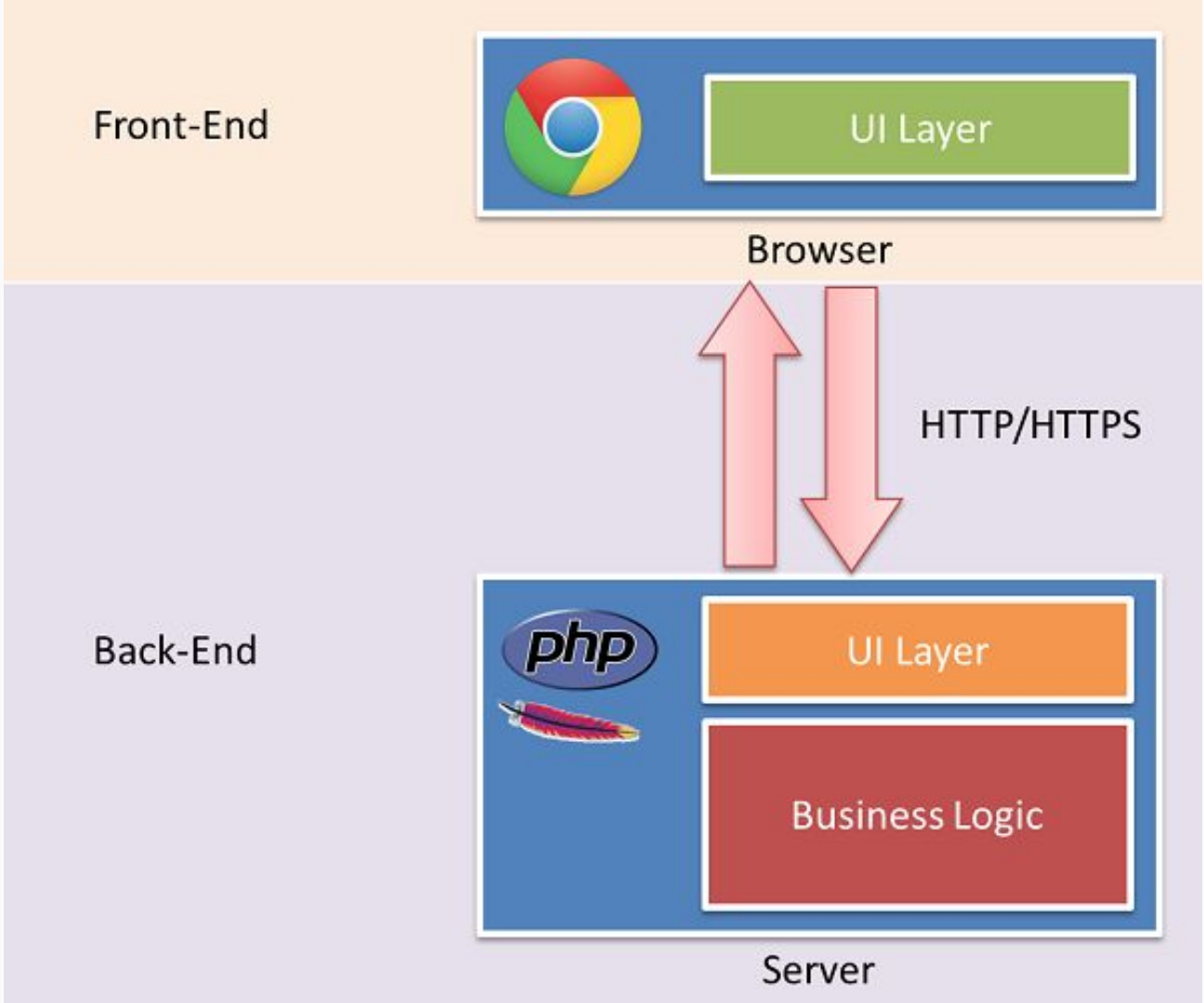
So why bother with ACM?

- Helps you practice for interview questions
- Looks good on a resume
- Companies will recruit and interview people who do well at the ACM (eg IBM)
- Meet other people and Bring some rep to Cal Poly Pomona!
- **$100 Amazon gift card to winning Cal Poly team at the tryouts!**

Announcements

- Google Tech Talks for the next 3 Tuesdays
  - No CSS Meetings for next 3 Tuesdays
- We will still meet on Thursdays
- Fall Career Fair Oct 27, 2016
  - Prepare your resumes to apply to companies!
- **Daniel Choi, currently interning at Amazon, will be presenting on how to apply big tech companies (like Facebook, Amazon, Google, Microsoft) next Thursday**

HTTP Backends


Front-End
UI Layer
Browser
HTTP/HTTPS
Back-End
php
UI Layer
Business Logic
Server

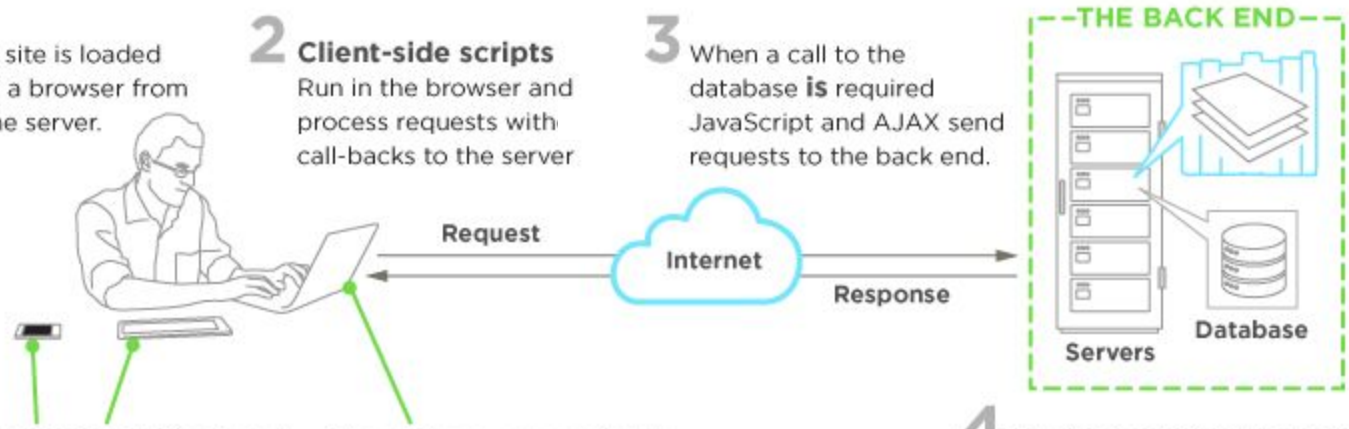
FRONT-END DEVELOPMENT

upwork

1 A site is loaded in a browser from the server.

2 **Client-side scripts** Run in the browser and process requests with call-backs to the server

3 When a call to the database **is** required JavaScript and AJAX send requests to the back end.



5 **Server-side scripts** process the data, then update the site—populating drop-down menus, loading products to a page, updating a user profile, and more.

Responsive front-end design allows a site to adapt to a user's device.

Everything a user sees in the browser is a mix of **HTML**, **CSS**, and **JavaScript**.

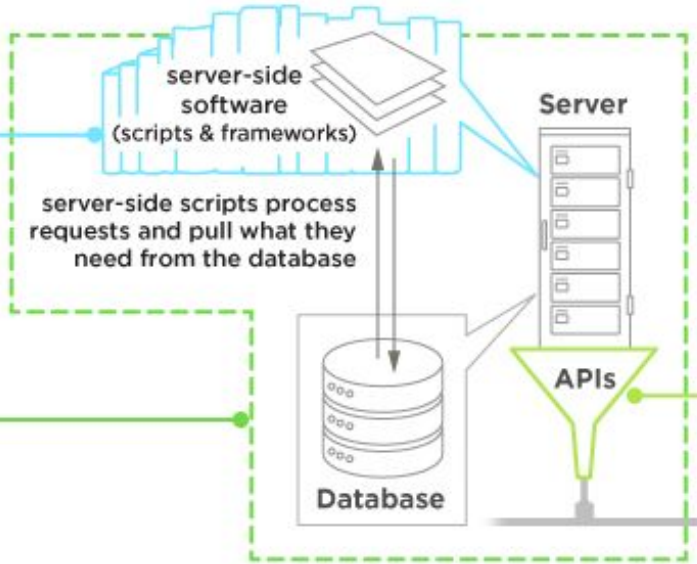
4 The **back-end server-side scripts** process the request, pull what they need from the database then send it back.

BACK-END DEVELOPMENT & FRAMEWORKS IN SERVER SIDE SOFTWARE

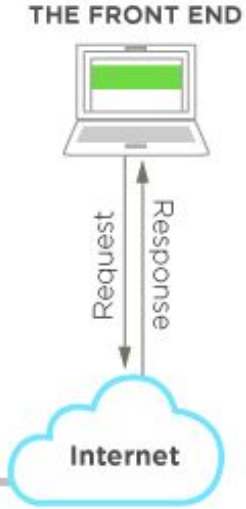
Upwork™

FRAMEWORKS are libraries of server-side programming languages that construct the back-end structure of a site.

The **"STACK"** comprises the database, server-side framework, server, and operating system (OS).



APIs structure how data is exchanged between a database and any software accessing it.



APPLICATION ARCHITECTURE

Backend	API	Frontend
Server	Cloud	Browser
I/O	Transport	GUI
Python	JSON	JavaScript
Django		ExtJS

HTTP Backend

- Just a running program “listening” on a computer at a specific port
- When the program receives an HTTP request, the program determines what action to perform based on the request’s **resource**, and **HTTP verb**
- So, you can think of an HTTP request as a way to trigger a function to run on the computer
- Normally, for static websites, function is sending back an html file located at that resource path
- But we can use it to create an interface called a **REST API**

Demo

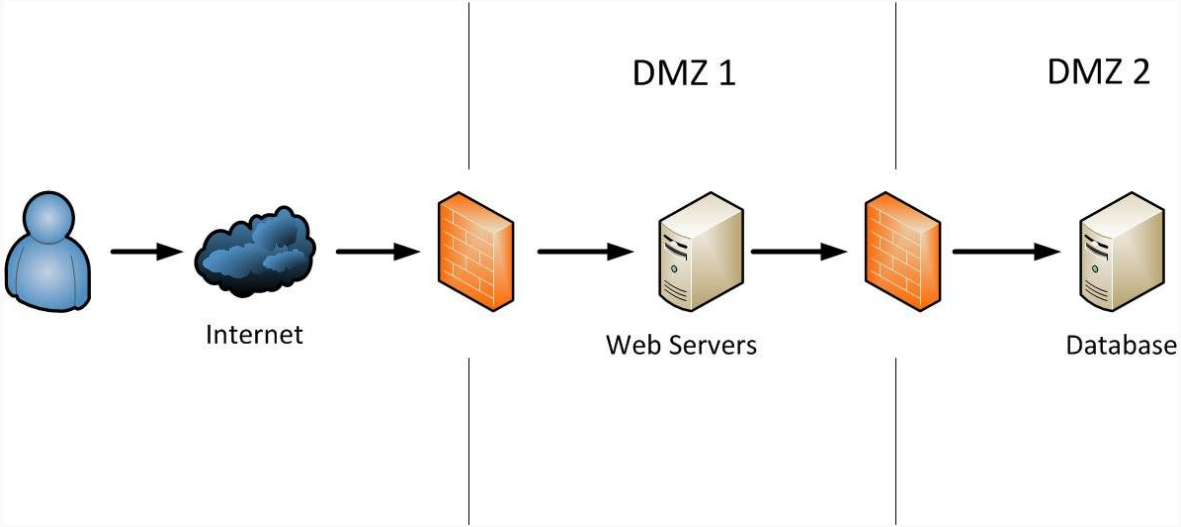
- curl www.cpp.edu = example of returning html
- https://jsonplaceholder.typicode.com/ = example of returning JSON
- postman post to http://sumyoutube.com/times/rAof9Ld5sOg

Why is this useful?

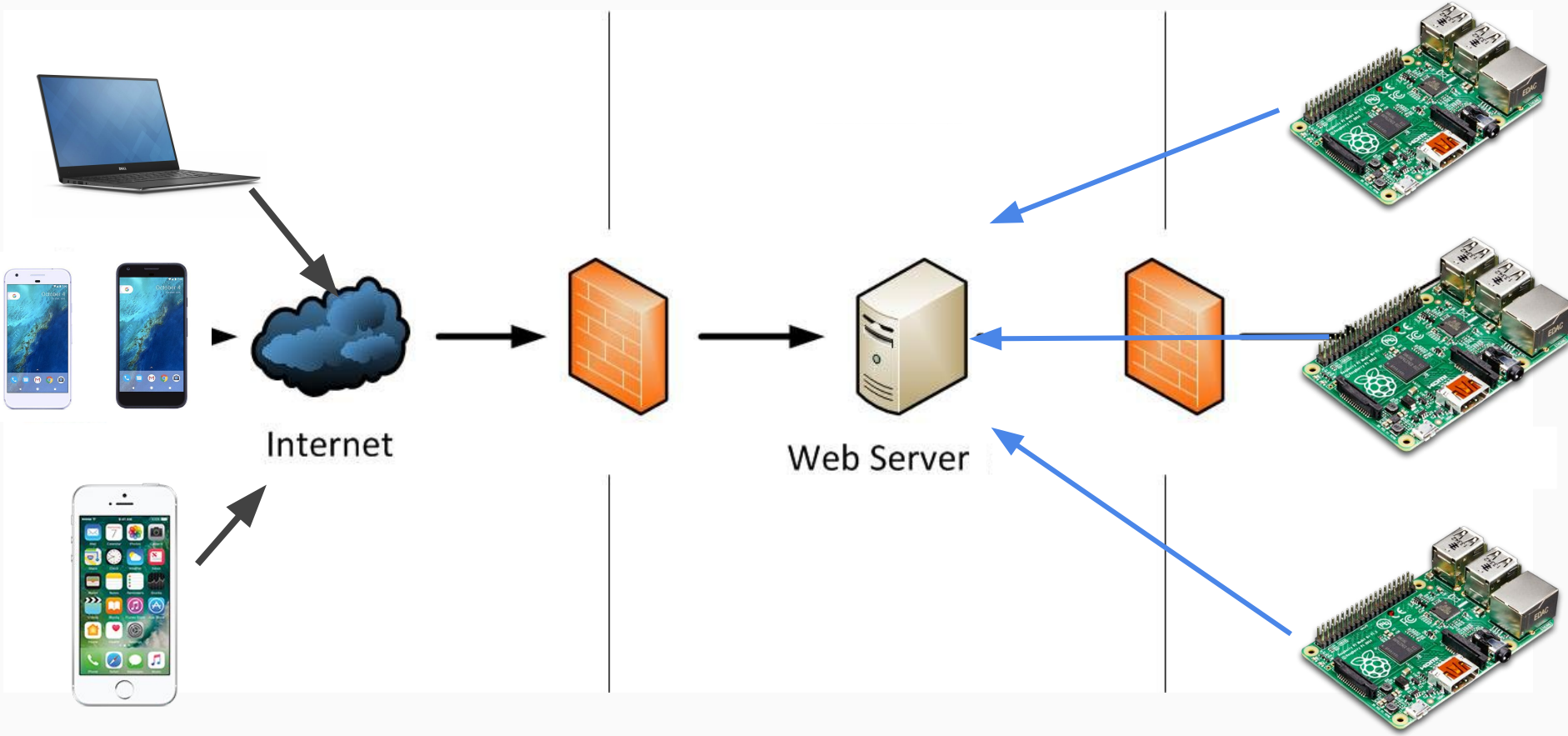
- Heavy computation can be performed on our servers, instead of on a client's computer

Why is this useful?

- Level of indirection between the client and internal services like a database
  - Restricts what the client is capable of doing

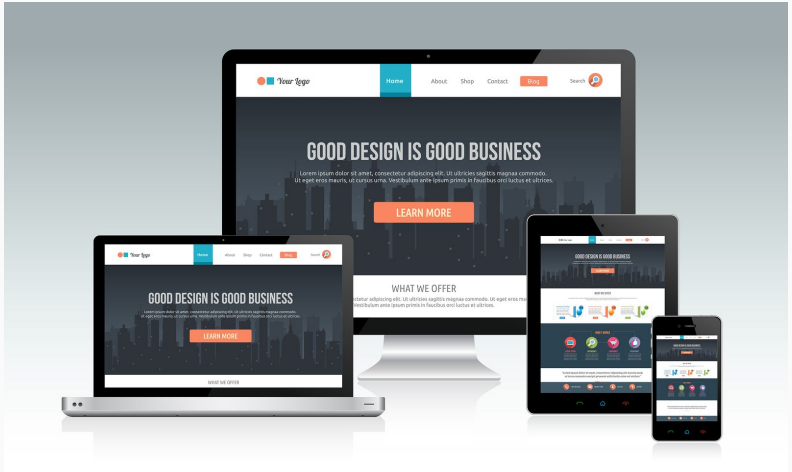


Our Parking Application’s Planned Architecture



Why is this useful?

- Provides a uniform interface that any different type of frontend can access
  - IOS, Anrdoid, Web Frontend



How to make an HTTP Backend?

- Use a framework!
  - don’t want to program complicated synchronization and networking logic
  - frameworks are stable, performant, and are regularly used by the developer community
- Examples:
  - Java: Sparkjava, Play!, Spring
  - Python: Flask, Django
  - Javascript (EW!): Node
  - Ruby: Rails
  - Swift: Vapor

Demo