

Internet and Web Application Concepts

2018 L.D. Nel

Internet and Web Application Reference Concepts

This presentation contains some reference pictures that we will refer to throughout the course.

We will likely add to this presentation as the course progresses.

Check for the latest version of this presentation as the course progresses

The Big Picture



COMP 2406

COMP 1406

Desktop
(Java, GUI apps)

Java 1.8
JavaFX,
Webkit support

Web
client-server
client: browser rendering
HTML, JSON, javascript,
client HTTP requests

AJAX, HTML5
Websockets

COMP 1601,2601

Mobile
client-server
client: Apps vs browser,
notifications vs requests

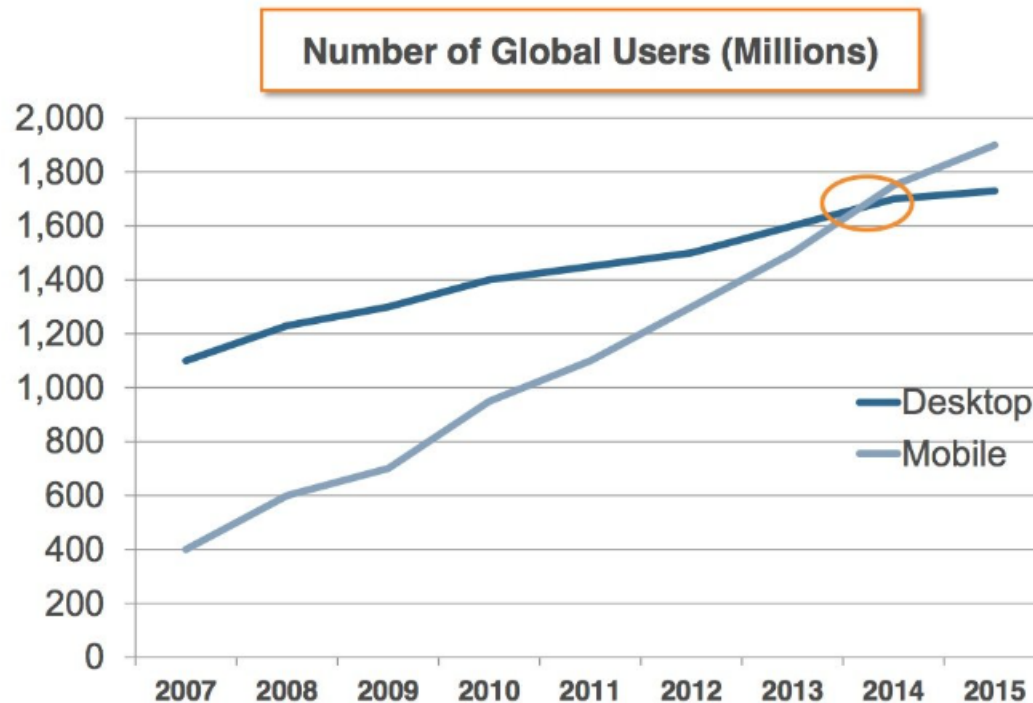
Andriod, IOS app
frameworks
HTTP and
Mobile protocols



Is everyone trying to get to the same place?

Mobile Web Access vs. Desktop Web Access

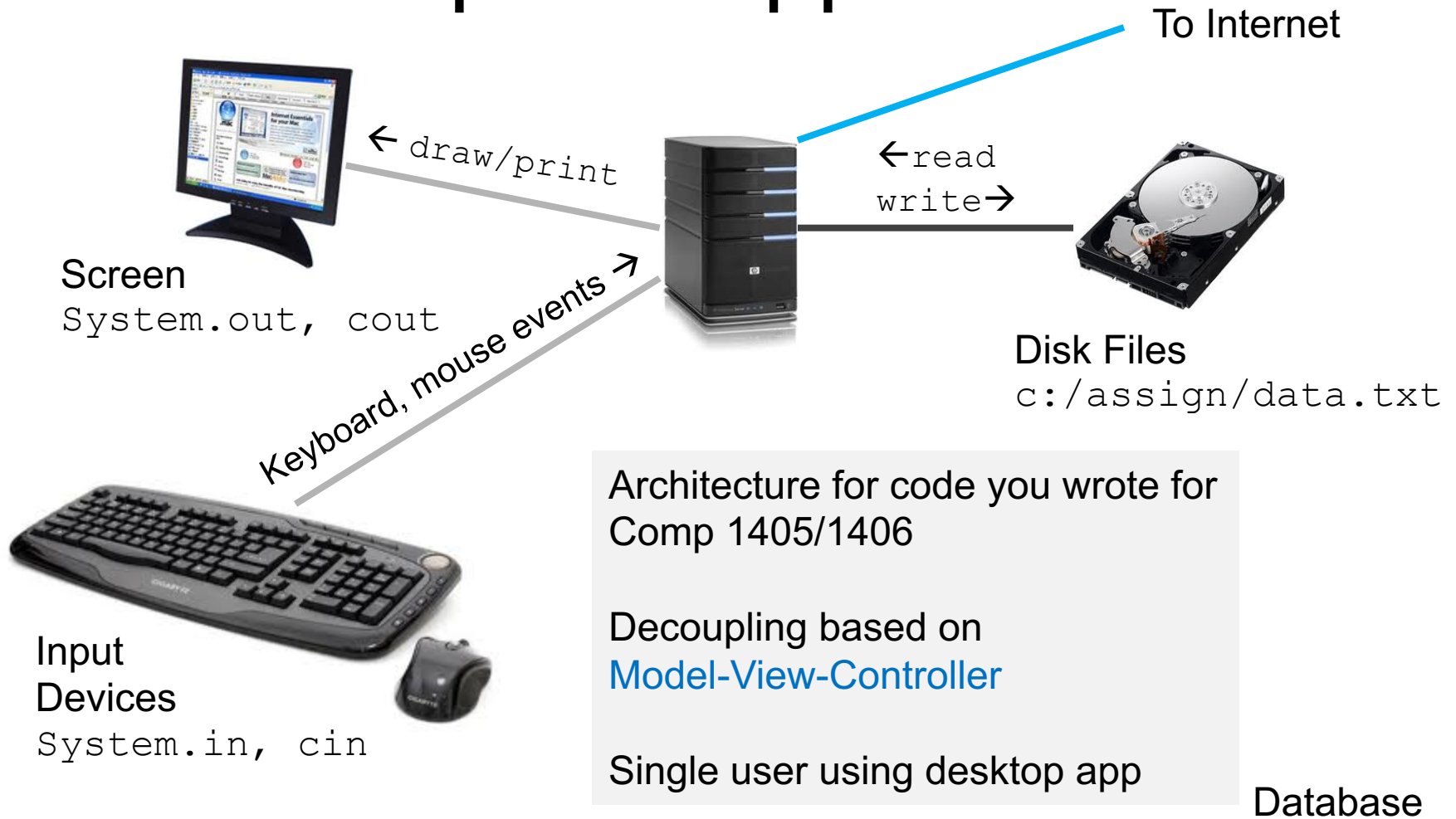
<http://www.businessinsider.com/mobile-web-vs-app-usage-statistics-2014-4>



Architecture Concepts



Reference Picture of Desktop Computer Application



Reference Picture of Web Application



Browser

Application

Client
126.24.17.01



http
get, post
← http
response

Internet/Web

Server
128.64.08.08:8080

http
get, post
← http
response



Database

Client
194.123.67.01



Browser

Use Case: Basic Static Server (e.g. course website)

Client

126.24.17.01

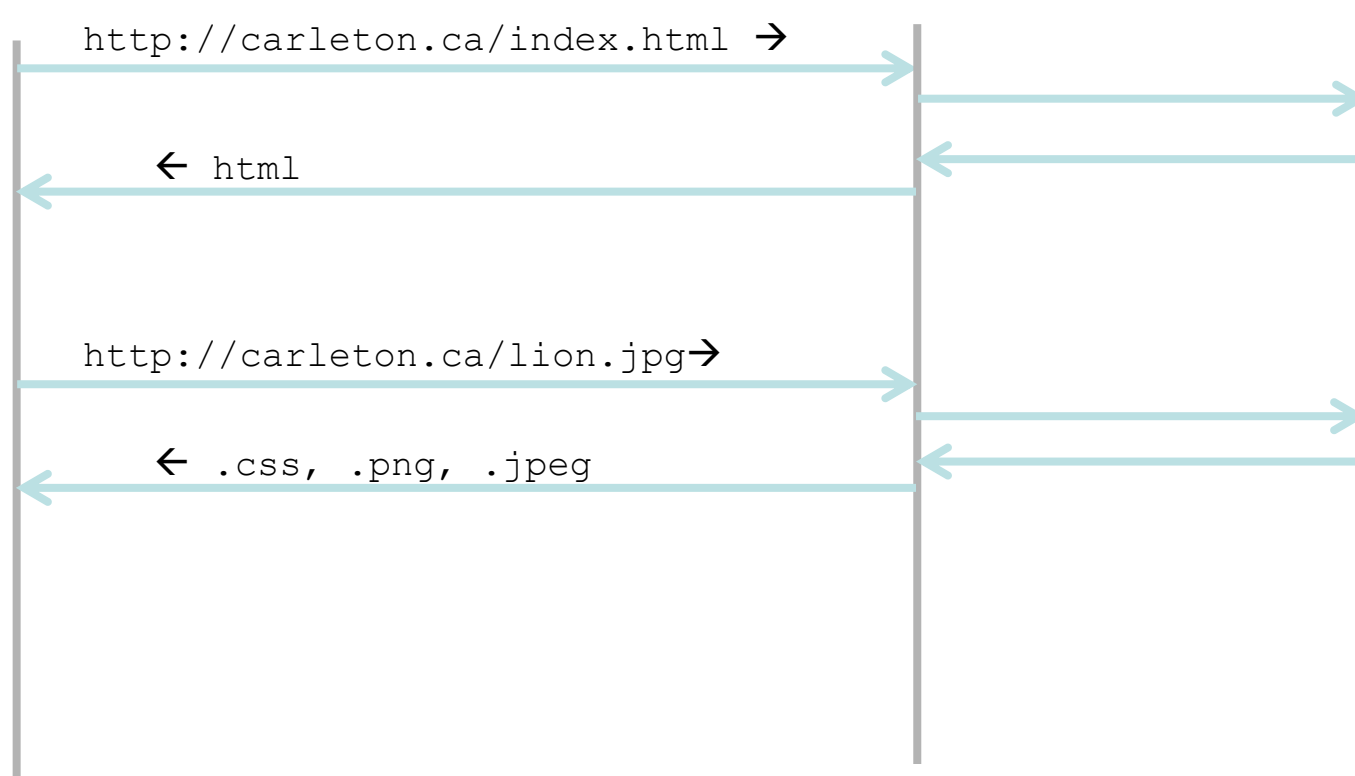


Server

128.64.08.08:8080



/public



Essentially client is just asking for html web pages stored on the server

Use Case: Simple Web App (e.g. 3005 Normalizer)

Client

126.24.17.01



Browser



Server



/public

128.64.08.08:8080

http://carleton.ca/index.html →

← html

http://carleton.ca/script.js →

← .js

mouse clicks

...

execute
javascript

Server provides client-side javascript to run client-side app

Use Case: Dynamic Web Page (e.g. Template Rendering)

Client

126.24.17.01



Browser



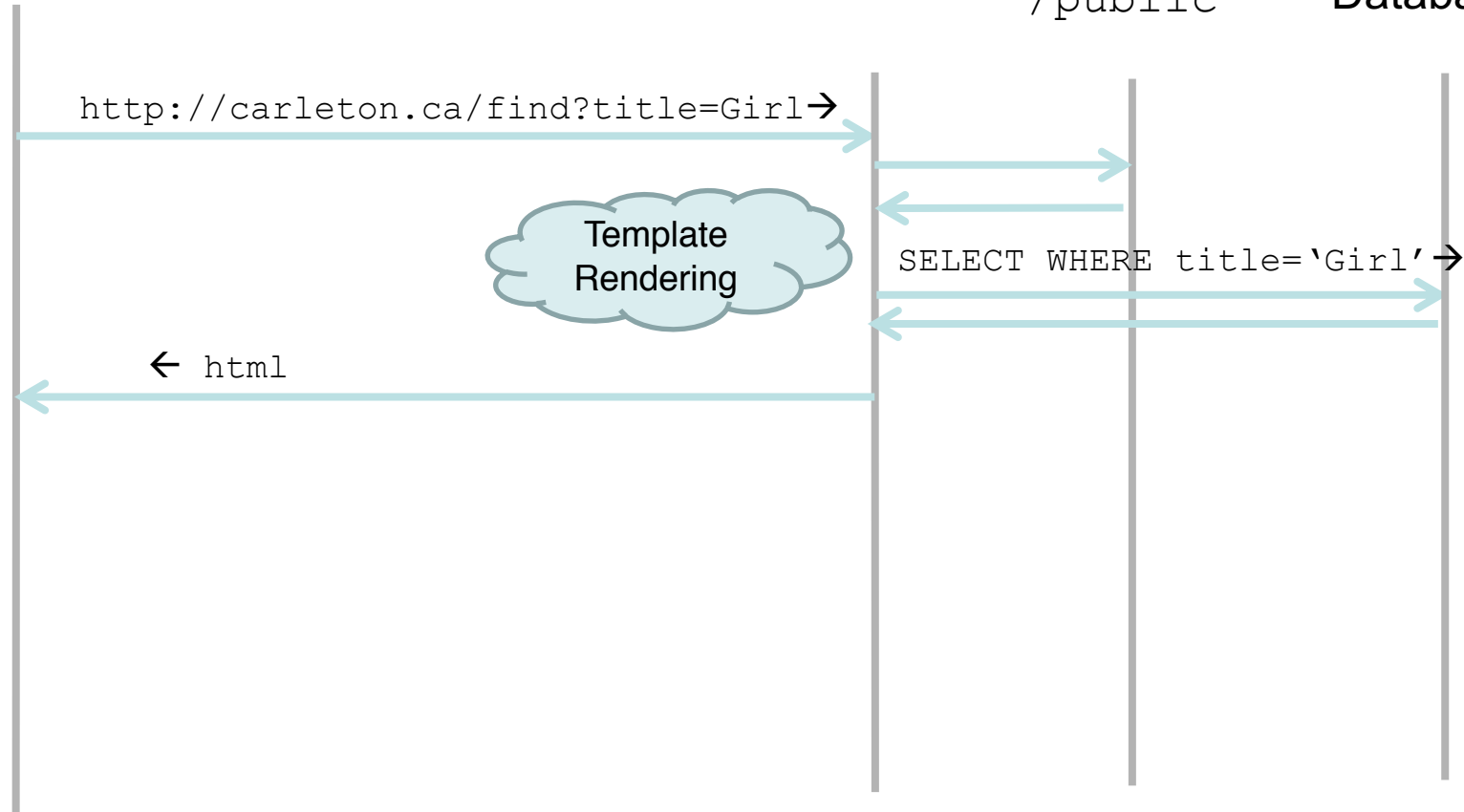
Server



/public



Database



Server synthesizes web pages from templates and database data

Use Case: Dynamic Web Page (e.g. RESTful API)

Client

126.24.17.01



Server



/public



Database

`http://carleton.ca/find?title=Girl` →

← `html, .js, .css`

mouse clicks →

`http://carleton.ca/api/find?title=Girl` →

`SELECT WHERE title='Girl'` →

← `JSON data`

execute
javascript

Client-Side
Rendering

Server provides “raw” JSON data for client to render; Server is just a data API

Use Case: Asynchronous Updates(e.g. AJAX)

Client

126.24.17.01



Browser



Browser



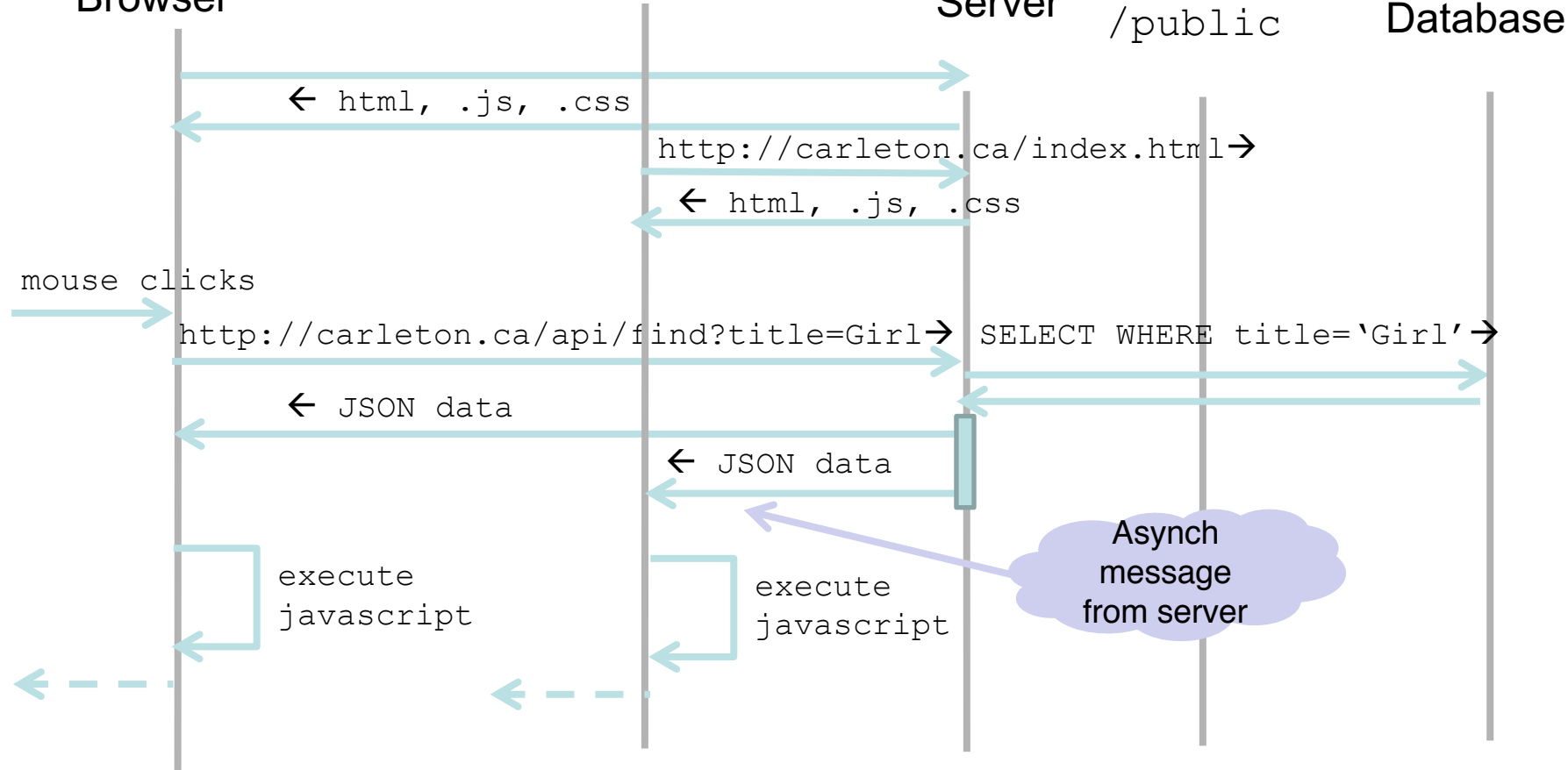
Server



/public



Database



Client can receive unsolicited messages from server, perhaps in response to what some other client is doing (e.g. real-time collaboration, games)

Course Technologies



Technologies from COMP 2406

Client Side

- ~~XHTML~~ (HTML5)
- CSS Cascaded Style Sheets
- Javascript (jQuery –javascript library??)

-
- HTTP, HTML, javascript ~~XML~~ JSON

Server Side

-
- ~~Apache~~ Node.js, NPM, Express.js
 - ~~PHP~~ Javascript
 - ~~MySQL~~ MongoDB, SQLite
-

Technologies from COMP 2406

(Add on technologies)

Client Side

- ~~XHTML~~ (HTML5)
- CSS Cascaded Style Sheets
- Javascript (jQuery –javascript library)
- Bootstrap (CSS/jQuery wrapper)

Server Side

- HTTP, transporting HTML, javascript ~~XML~~ JSON
- ~~Apache~~ Node.js, NPM, Express.js
- ~~PHP~~ Javascript
- ~~MySQL~~ MongoDB, SQLite
- Jade, Handlebars (HTML templating)