Introduction

Joseph Mertz, Ph.D
Teaching Professor
H. John III Heinz College
Dietrich College – IS Program
Applications today

Mobile

Desktop

Tablet

Social

Interactive

Real Time

Big data

Share functionality (e.g. maps)

Interact with the physical world (e.g. body, house)
The evolution of applications

• Applications have evolved from single-computer to distributed
• E.g. History of what has become Customer Relationship Management (CRM) systems.
Contact manager application

Stand alone applications on a single computer.

- Single programming language
- Single computing platform
- Ad-hoc information storage
- No networking protocols
Shared database

Shared databases available via a LAN to all salespeople

• Single programming language
• Single computing platform
• Single DBMS
• LAN based networking
Web-based system

• CRM uses common web protocols for users to access customer information via a browser
• Multiple programming languages
  – Middleware (software on the server)
    • Java and JEE, C# and .NET, PHP, Ruby, or others
    • HTML, Javascript, CSS
• Multiple computing platforms
  – Server
  – User web browser
    • Potentially multiple web browsers with varying capabilities
• Server-side DBMS
• Web-based networking protocols: http
Customer portal into your business

- Multiple languages
  - Middleware (JEE, .NET, etc)
  - Languages of legacy systems
  - HTML, CSS, Javascript
  - Frameworks: AJAX, JQuery, Struts, Springs, Hibernate, JPA, Flash, Silverlight

- Multiple computing platforms
  - Server, legacy systems, multiple web browsers
  - Cloud-based platforms (IaaS, PaaS, SaaS)

- Multiple DBMS, some legacy

- Networking
  - HTTP, web services protocols (e.g. SOAP, REST), enterprise-internal protocols to tie to legacy systems.
Mobile.my.company.com

• Customer can access all their account information for interacting with the business via a mobile phone browser
• Multiple languages
  – Middleware
  – Languages of legacy systems
  – HTML, CSS, Javascript
  – Frameworks: AJAX, JQuery, Struts, Springs, Hibernate, JPA, Flash, Silverlight
• Multiple computing platforms
  – Server, legacy systems, multiple web browsers
  – Cloud-based platforms (IaaS, PaaS, SaaS)
  – Mobile phone platforms / browsers
• Multiple DBMS, some legacy
• Networking
  – HTTP, web services protocols (e.g. SOAP, REST), enterprise-internal protocols to tie to legacy systems,
  – XML or JSON to pass objects to/from mobile.
CRM Mobile App

- Customer can access all their account information for interacting with the business via a mobile phone
- E.g. Checking Amazon account shipments on my iPhone
- Multiple languages
  - Middleware
  - Languages of legacy systems
  - HTML, CSS, Javascript
  - Objective-C for iPhone, Java for Android and Blackberry, C# for Windows Phone 7.
  - Frameworks: AJAX, JQuery, Struts, Springs, Hibernate, JPA, Flash, Silverlight
- Multiple computing platforms
  - Server, legacy systems, multiple web browsers
  - Cloud-based platforms (IaaS, PaaS, SaaS)
  - iPhone, Android, etc mobile phone platforms.
- Multiple DBMS, some legacy
- Networking
  - HTTP, web services protocols (e.g. SOAP, REST), enterprise-internal protocols to tie to legacy systems,
  - XML or JSON to pass objects to/from mobile.
Mashup genesis

• Early CRMs were:
  – Often built by one company
    • Custom built or a commercial system
  – Only included data from within that company

• The web enabled developers to embed information and functionality from other sources in their applications
  – was called “mashup”

• First mashup: HousingMaps.com
A bit of history...

• HousingMaps.com was the first mashup.
  – By Paul Rademacher

• At the time, Craigslist real estate listings were only lists; you could not browse on a map.

• Rademacher overlaid rental and real estate information on a map he fetched from Google.

• The Google Maps team was thrilled to see how he and others were hacking (in the good sense) their map system and describing it on blogs.
At the time, Craigslist had no API?

• They still don’t have much
  – They do have a bulk-posting API

• HousingMaps was built without the aid of a published API.

• What is an API?
  – Application Programming Interface
    • A set of software specifications that enable a programmer to interact with a software system
So did HousingMaps get its data?

- **Non-optimized option:**
  - User does query for housing
  - Housingmaps does query to craigslist
  - Housingmaps processes all responses
  - Housingmaps returns page to user.

- **Apparentky optimized option:**
  - Craigslist provides RSS feeds (i.e. XML formatted response)
  - Housingmaps subscribes to Craigslist RSS feed
  - Periodically checks the RSS feed
  - When updates, retrieves and screen-scrapes and stores the information
  - This limits the number of queries HousingMaps needs to do to Craigslist.
    - (Requests are limited by Craigslist).
Status

- Craigslist now has mapping capability
  - Uses OpenStreetMap
    - [http://www.openstreetmap.org](http://www.openstreetmap.org)
- HousingMaps no longer works
  - Rademacher writes it is “taking a break”
Growth of APIs

Source: http://www.programmableweb.com/api-research
# Devpost hackathon API survey

<table>
<thead>
<tr>
<th>Communications APIs</th>
<th>Social APIs</th>
<th>Payment APIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Twilio</td>
<td>1 Facebook</td>
<td>1 Venmo</td>
</tr>
<tr>
<td>2 Yo</td>
<td>2 Twitter</td>
<td>2 Paypal</td>
</tr>
<tr>
<td>3 SendGrid</td>
<td>3 Reddit</td>
<td>3 Stripe</td>
</tr>
<tr>
<td>4 Moxtra</td>
<td>4 Instagram</td>
<td>4 Braintree</td>
</tr>
<tr>
<td>5 Mailjet</td>
<td>5 Google+</td>
<td>5 Blockchain</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Geo APIs</th>
<th>Music APIs</th>
<th>App frameworks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Google Maps</td>
<td>1 Spotify</td>
<td>1 Node.js</td>
</tr>
<tr>
<td>2 Yelp</td>
<td>2 Echo Nest</td>
<td>2 Flask</td>
</tr>
<tr>
<td>3 Google Places</td>
<td>3 SoundCloud</td>
<td>3 AngularJS</td>
</tr>
<tr>
<td>4 Esri ArcGIS</td>
<td>4 Google Play</td>
<td>4 Ruby on Rails</td>
</tr>
<tr>
<td>5 Mapbox</td>
<td>5 Rdio</td>
<td>5 Django</td>
</tr>
</tbody>
</table>

# ProgrammableWeb has a directory

## API Directory

### Featured APIs
- Facebook
- Yahoo Weather
- Google Maps
- AccuWeather
- Weather Channel
- Wikipedia
- OpenStreetMap
- Instagram

### Popular Categories
- Weather
- Mapping
- Mobile
- Transportation
- Social

[All API Categories (15,418)]

### Filter
- By Category
- By Protocol
- Most Recent
Mobile

• What does CraigsList look like on a phone?
• What does the cmu.edu/hub look like?
Mobile Content Adaptation

• Customizing presentation to fit the capabilities of any platform.
• Design goal: Design to adapt content for your full customer base.
Real-Time Collaboration Example

• Browse to:
  – http://tinyurl.com/m2c-draw

• Uses
  – HTML5 Canvas tag
  – jQuery
  – Node.js
  – Socket.io
    • which uses WebSockets (if available)

• Only
  – 36 lines of HTML
  – 120 lines JavaScript client-side
  – 36 lines JavaScript server-side
Web as a Platform

• The web has become a "platform"
• Where once we programmed only individual computers, now our applications
  – span many types of devices
  – connected by the internet
  – using open standards for easy interoperability
Evolving the CRM into the Cloud

- Shrink your internal development team
- Lower your data center operating costs
- Increase your functionality

By

- Moving to a cloud-based, software-as-a-service CRM solution (e.g. Salesforce)
What should be the course goals?

• Advances are coming rapidly:
  – new frameworks
  – new tools
  – new products

• Anything you learn about a specific language, platform, protocol, or framework will likely change in the next 5 years
  – and perhaps before you graduate!

• The goal can’t be only
  – JavaScript,
  – or Web Sockets,
  – or accessing Amazon S3 storage using REST.

• Consequently what can you learn from a class like this?
The goals must be

• To practice using specific technologies to craft solutions
• To reflect on those technologies
  – I.e. categorize, compare, contrast classes of technologies
• To reflect on those solutions
  – I.e. base solutions on
    • best-practice architectures,
    • software engineering practices,
    • and solution patterns
  – and extract new patterns from viable solutions
Learning Model Metaphor

• I'll give you an example of making dough, and we will walk through the process.
• Then I'll give you an example of using an oven and we will do an exercise in how to use it.
• I'll assume you can open a can.

Then

• I'll give you a big assignment to make and bake something that involves dough.
  – I might suggest as an idea making a creative pizza.
  – You might opt for making cinnamon buns instead.
  – In any case, I won't walk you through making pizza or cinnamon buns. The learning happens in figuring it out.
A prior student aptly named it...

• Guided Self-Learning
TA Demos
Therefore we will cover

• Developing interactive web applications
  – Improving client-side programming skills
• Developing server-side middleware
• Using public 3\textsuperscript{rd} party APIs
• Adapting to mobile devices
• While we work to understand deeper:
  – networking
  – storing persistent data
    • noSQL
  – MVC
  – security
Technologies we will use

• JavaScript
  – jQuery
  – jQuery Mobile
• Node.js
  – Express and other frameworks within Node.js
• MongoDB
• Web Sockets
  – Socket.io
• Multiple cloud-based services
Background

• My background...

• Your background - before Thursday's class:
  – Fill out the brief form at http://tinyurl.com/m2c-prior
Review course information

• Web site
  – http://www.andrew.cmu.edu/course/67-328/
  – Contact info
  – Syllabus
  – Schedule
  – Resources

• Blackboard
  – Grade book
  – Submitting assignments

• Piazza
  – Forum for discussion of classes, labs, and assignments
Where possible, *flipped* Instruction

- Use video, blogs, tutorials, and readings to get the bulk of *content*.
  - Allows you to read/watch at your own pace
- Use class time for
  - discussion, answering questions
  - reviewing demos / examples
  - working on exemplar small projects
  - getting and sharing lots of help
• Install Google Chrome on your laptop if you don't already have it
  – Read about the Chrome developer tools (links are on the course schedule)
    • Set Up Chrome DevTools
    • Navigating the Console
    • Diagnose and Log to Console
    • Inspect and Tweak Your Pages: the Basics
    • Keyboard & UI Shortcuts Reference
    • How to Add or Remove Breakpoints
    • How to Step Through the Code
• Watch the Douglas Crockford video (1:49:55 long)
  – Alternative: scan his slides to refresh your knowledge, watch when you have questions
  – Alternative: scan the JavaScript: Definitive Ch 1-9
• Come ready to write short JavaScript programs in the Chrome developer console.
• Fill out the brief form at http://tinyurl.com/m2c-prior