

# uMobile Seminar

Jen Bourey

Denver Jasig Conference, May 22, 2011

1. The Basics
2. uPortal's Mobile Strategy
3. The Mobile Theme
4. Developing Portlet UIs
5. Native App Development
6. The Future

# The Basics

# What's Different?

- User Interface
  - Screen size
  - Touchscreen interactions
- Performance
  - Bandwidth
  - Processing power
- Content
  - Overlap with existing apps

# Touchscreen Interactions

- Tap
- Touch
- Pinch
- Swipe

# Additional Hardware

- Accelerometer
- Camera
- Microphone

# Content

AT&T 4:23 PM



Search Seattle, WA

**Right Now**  
Updated: 5/19/11 4:05 PM PDT

**Fair**

Feels Like **66°F**  
Wind **NNW at 7 mph**  
Humidity **29%**  
UV Index **3 Moderate**  
Dew Point **33°F**  
Visibility **10.0 mi**  
Sunrise **5:28 AM**  
Sunset **8:45 PM**

Share

Now Hourly 36 Hour

Weather Map Severe

AT&T 4:25 PM

Nearby Search

Filter Coffee & Tea

- 7. Energy Espresso**  
5905 15th Ave NW, Ballard  
★★★★★ 1 Review  
Coffee
- 8. Short Stop Coffee**  
336 NW 40th St, Ballard  
★★★★☆ 15 Reviews  
Coffee
- 9. 15th Ave Coffee and More**  
7001 15th Ave NW, Ballard  
★★★★☆ 2 Reviews  
Coffee
- 10. Sip & Ship**  
1752 NW Market St, Ballard  
★★★★☆ 77 Reviews  
Coffee
- 11. Cafe Besalu**  
5909 24th Ave NW, Ballard

Nearby Check-Ins Search Bookmarks

AT&T 4:26 PM

Search Post office



Search Directions

# **uPortal's Mobile Strategy**



# Primarily Web-based

- Well-defined, familiar standards
- Flexible and mature
- HTML5!

# Strategic Native Adoption

- Additional features
  - Geolocation
  - Address book / filesystem integration
  - Camera / microphone
- Potential performance improvements
  - Native controls
  - Local caching

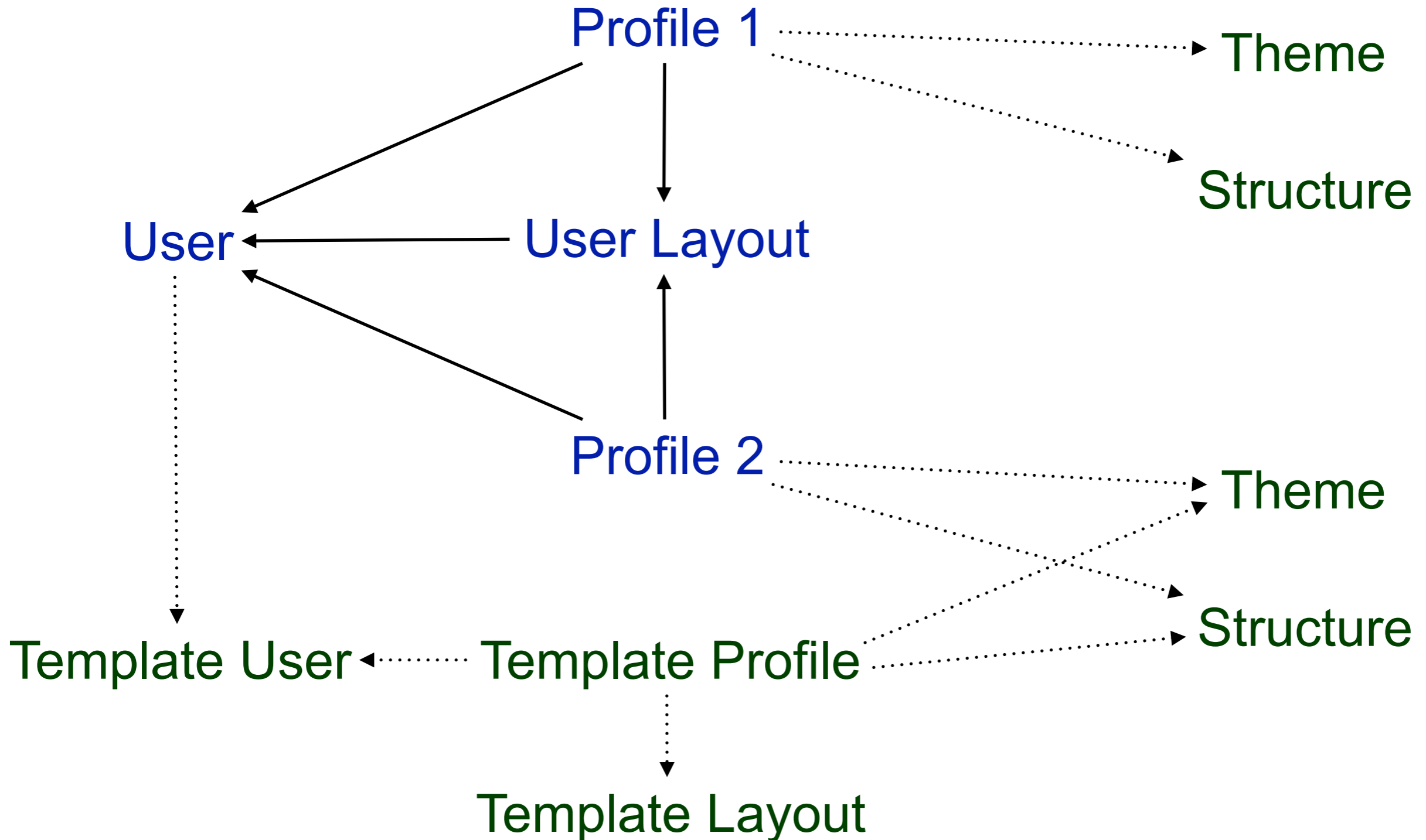
# The Mobile Theme

# User Profiles

# User Data Organization

User Data

System Data



# Sample User Profile Data

ID	User	FName	Structure	Theme	Layout
1	defaultTemplateUser	default	1	1	1
2	jbourey	default	1	1	1
3	jbourey	mobileDefault	2	2	1
4	cclark	default	1	1	1
5	cclark	mobileDefault	2	2	0

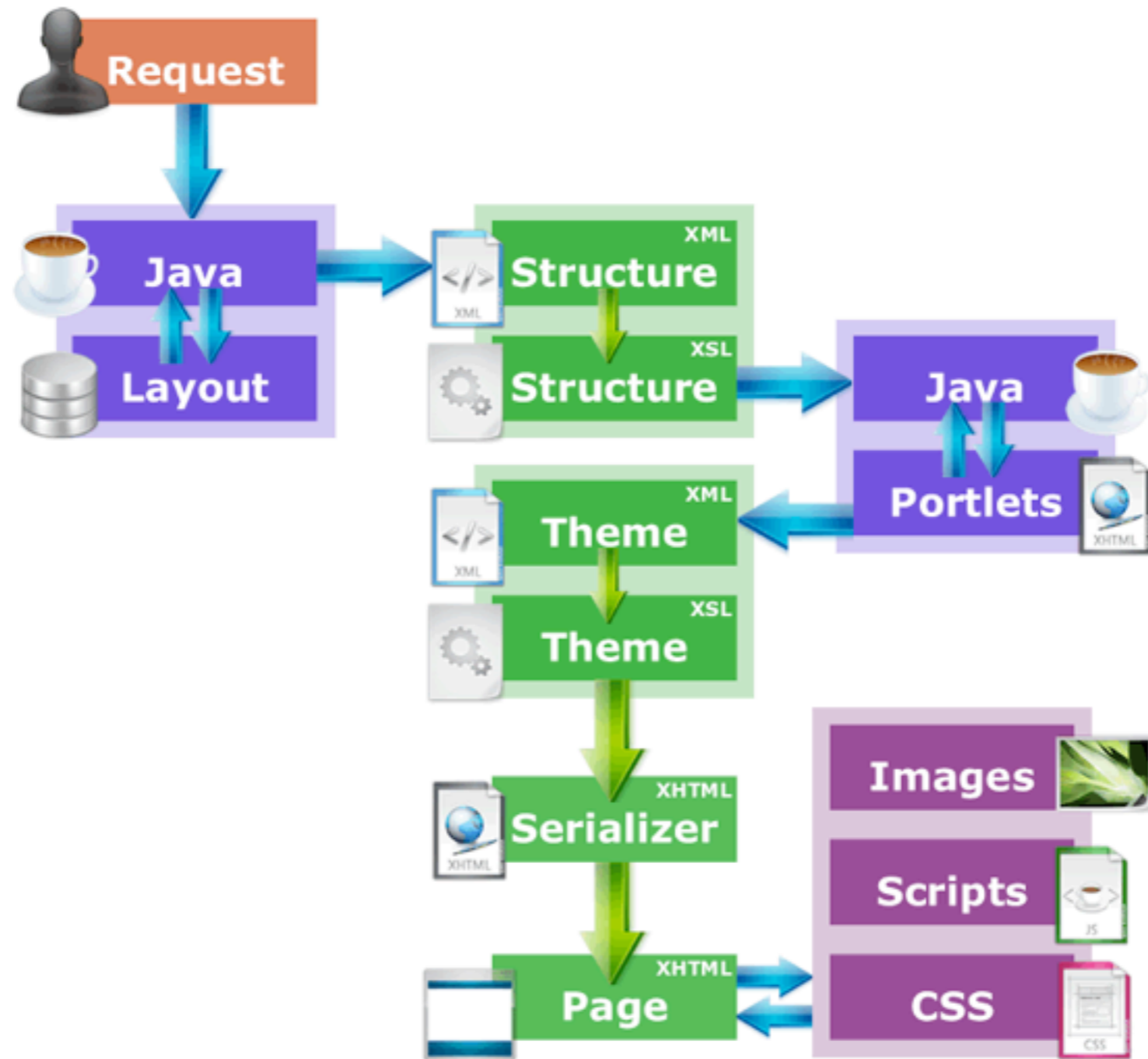
# Choosing a Profile

- Get user profile for user agent
- Get system profile for user agent
- Get user profile for browser mapping
- Get system profile for browser mapping

# Themes and Structures



# uPortal Rendering Pipeline



# User Layout Document

- XML document constructed from database
- Nested structure of folders
- Folders may contain also channels
- Nodes may contain additional attributes
  - edit/remove permissions
  - channel parameters

# Structure Transform

## User Layout

- Folder
  - Folder
    - Channel
    - Channel
  - Folder
    - Channel
- Folder
  - Folder
    - Channel
    - Channel



## After Transform

- Tab
  - Column
    - Channel
    - Channel
  - Column
    - Channel
- Tab
  - Column
    - Channel
    - Channel

# Theme Transform

- Consists of several XSLT files
- Has access to a number of parameters
- Transforms structure XML into XHTML

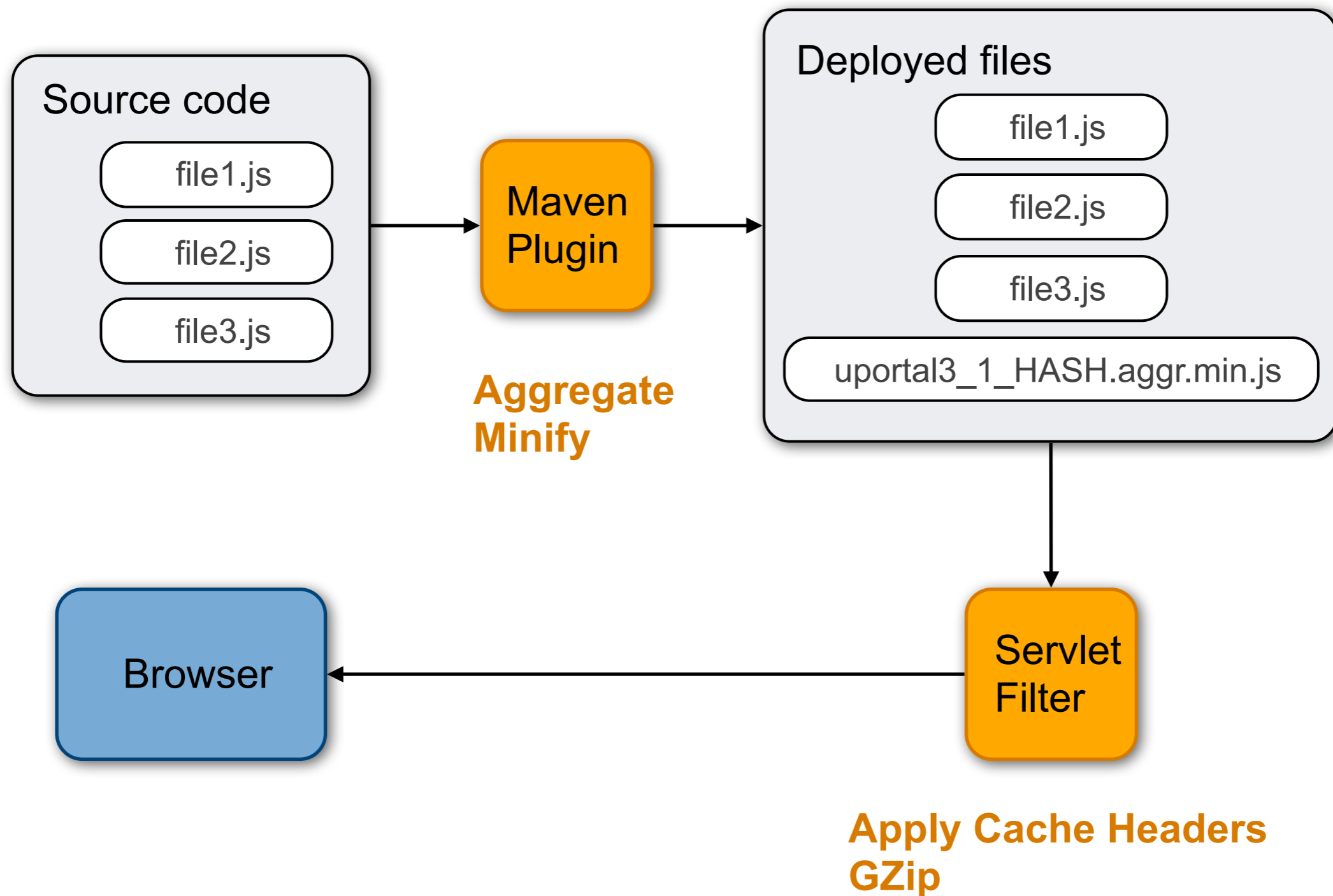
# Debugging Themes and Structures

## log4j.properties

```
## Uncomment to see the XML at various stages in the rendering pipeline  
log4j.logger.org.jasig.portal.rendering.LoggingStAXComponent=DEBUG, R  
log4j.additivity.org.jasig.portal.rendering.LoggingStAXComponent=false
```

# **uPortal Skin Aggregation**

# CSS/JS Aggregation



# Skin File Format

```
<resources xmlns="http://www.jasig.org/uportal/web/skin"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.jasig.org/uportal/web/skin
https://source.jasig.org/sandbox/resource-aggregator/tags/resource-aggregator-parent-1.0.0/resource-aggregator-api/src/main/resources/skin-configuration.xsd">

<parameter name="fss-theme">fl-theme-iphone</parameter>

<css import="true">../common/common_skin.xml</css>

<js import="true">../common/common_skin.xml</js>

</resources>
```



# Performance

Table: Mobile browser external resource cache characteristics

Browser/OS/Device	Single Component Limit	Survives Power Cycle
Android 2.2 (Nexus One)	2MB	Yes
Mobile Safari, iOS 3.1.3 (1st-gen iPhone)	4MB+	No
Mobile Safari, iOS 3.2 (iPad)	4MB+	No
Mobile Safari, iOS 4.0 (iPhone 3GS)	4MB+	No
Mobile Safari, iOS 4.0 (iPhone 4)	4MB+	No
webOS 1.4.1 (Palm Pre Plus)	~0.99MB (1,023KB)	Yes

# uPortal Mobile

# uPortal Mobile

- mobile-columns structure
  - merges all tabs/columns into a flat list
- mUniversality theme
  - renders portlets using smartphone dashboard style layout
  - displays badge numbers
  - in focus mode, hides navigation and displays a single portlet

# Mobile Structure Transform

## User Layout

- Folder
  - Folder
    - Channel
    - Channel
  - Folder
    - Channel
- Folder
  - Folder
    - Channel
    - Channel



## After Transform

- Content
  - Channel
  - Channel
  - Channel
  - Channel
  - Channel

# Disabling Content in Mobile Theme

## Global CPD Parameter



**Display Settings**

These options influence how your Portlet is displayed within the portal

Parameters	Value	User Editable
Show Portlet Window Chrome:	<input type="checkbox"/> true	
Use 'Alternate' Appearance (Theme Dependent):	<input type="checkbox"/> false	
Use Highlighted Appearance:	<input type="checkbox"/> false	
Hide in mobile theme:	<input type="checkbox"/> true	
Quicklinks index:	<input type="text"/>	

# Disabling Content in Mobile Theme

```
<!--====START: CHANNEL TEMPLATE RULE====-->  
<xsl:template match="channel">  
  <xsl:if test="not(parameter[@name='hideFromMobile']/@value = 'true')">  
    <xsl:copy-of select="."/>  
  </xsl:if>  
</xsl:template>  
<!--====END: CHANNEL TEMPLATE RULE====-->
```

# uPortal 4 Rendering Pipeline

- New componentized pipeline
  - Very flexible
  - SUPER COOL!

# Native Theme Tweaks

- `/Login?isNativeDevice=true` triggers native theme
- Outputs unfocused view as an XML-wrapped JSON fragment instead of as HTML
- In the future will create parallel rendering pipeline for this use case



# Badge Numbers

- Displays an optional number indicating the number of new items contained in a portlet
- Invitation to users to visit portlets for new information
- Implemented as a portlet response property

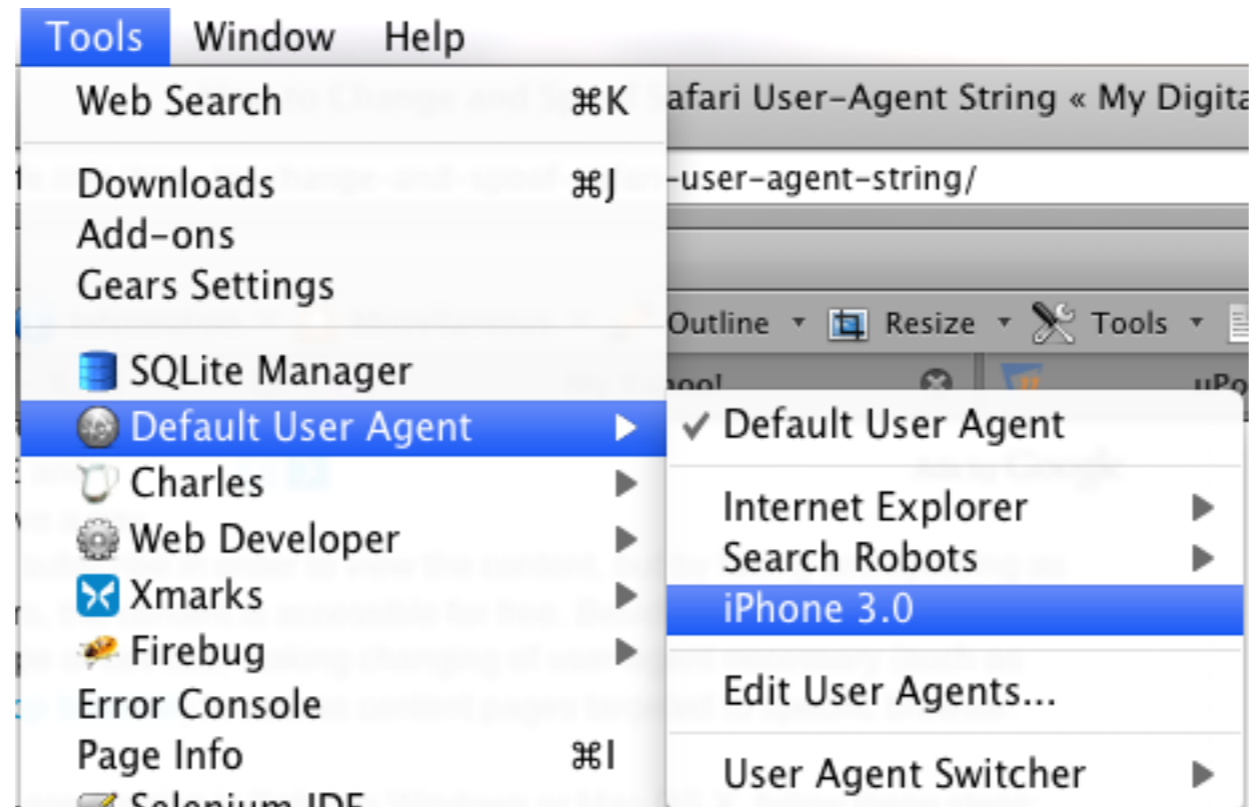
# SASS

- Framework for developing CSS
- Use variables, mixins
- Introduced in uPortal 4 for both desktop and mobile themes
  - You don't have to use it if you don't want to

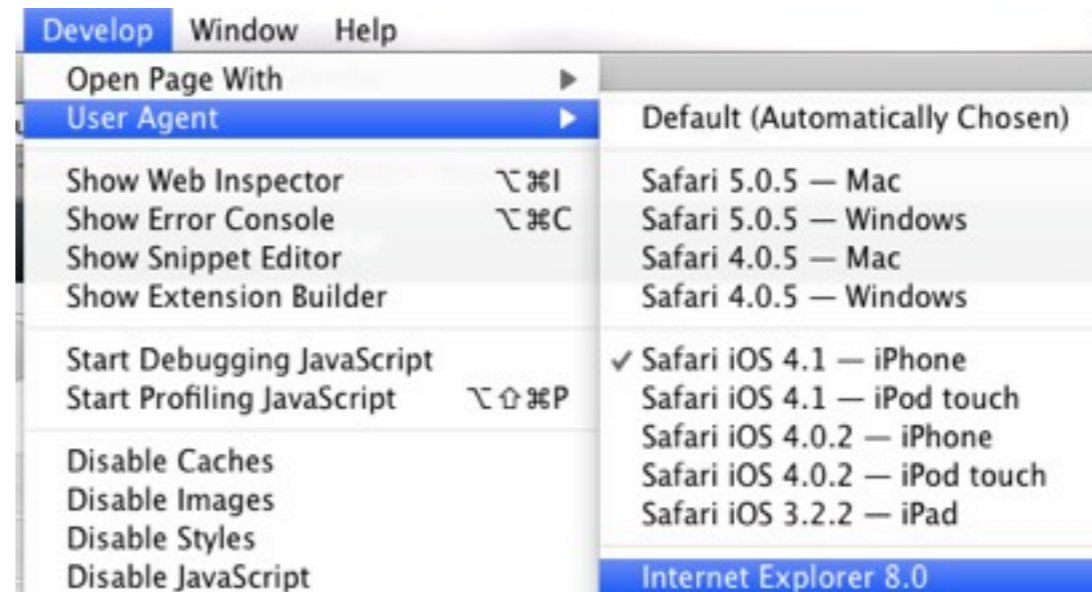
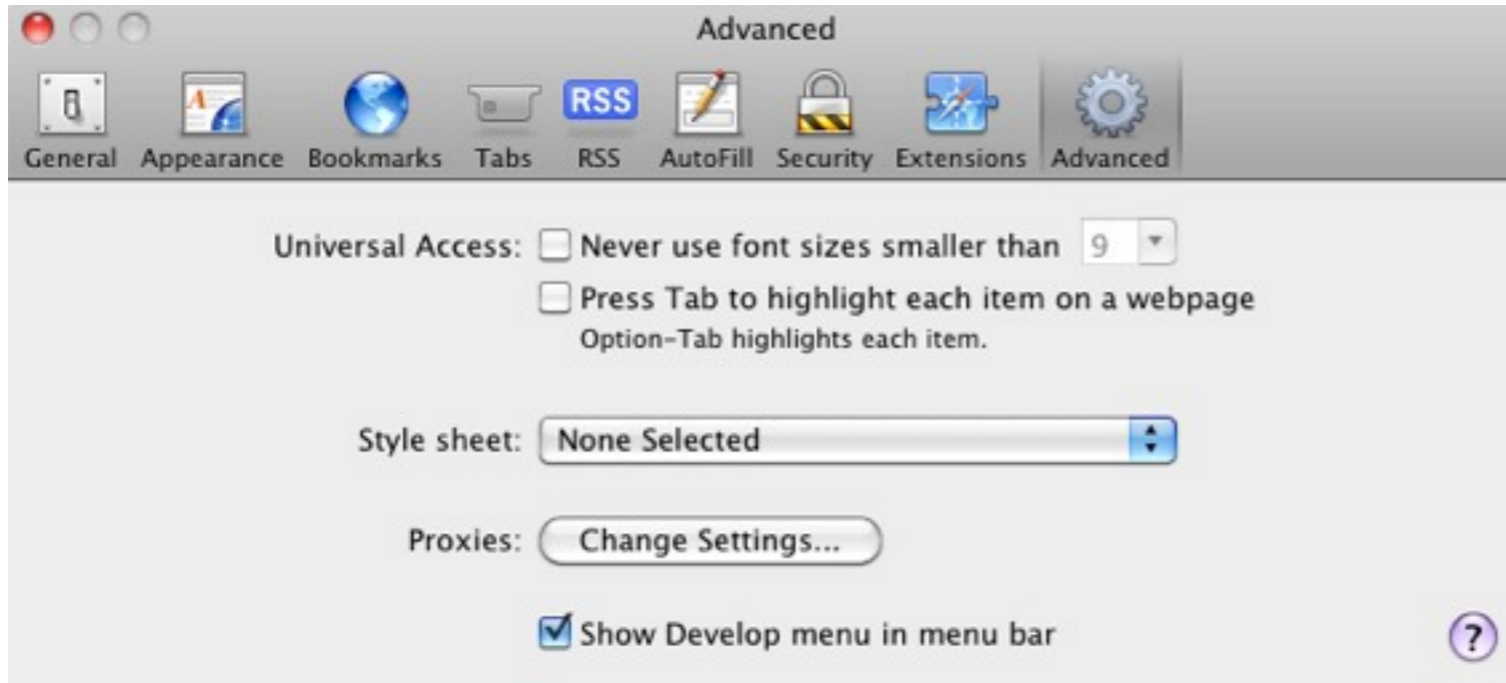
# Testing

- Real actual phone
  - Best way to test, need to do this before releasing
  - Should consider multiple versions
- Emulator
  - iPhone: OS X only
  - Android: all platforms
- Browser with spoofed user agent
  - Safari
  - Firefox with User Agent plugin

# Firefox User Agent



# Safari User Agent



# Mobile Portlet UIs

# Fluid Skinning System

- About the project
  - Pure CSS framework
  - Only supports WebKit-based browsers
- Status
  - Used for uPortal 3.2 mobile theme
  - Hasn't had a lot of recent updates
  - Fluid intends to work with jQuery Mobile in the future

# jQuery Mobile

- About the project
  - Javascript/CSS framework
  - Support for a number of modern smartphone browsers
  - Adds touchscreen event listeners
- Status
  - Alpha release, early in project lifecycle
  - Lots of active development



# jQuery Mobile Documentation

- URL
  - View in a mobile browser to use as an example
- Hint: Get markup by removing hash tag

# Not Using All Features

- jQuery Mobile Theming
- AJAX Navigation

# jQM Configuration

- jQM is loaded *\*very\** early
  - To alter default configuration, must include a script *\*before\** loading jQM

# jQM uPortal-Level Configuration

uportal-war/src/main/webapp/media/skins/muniversality/common/javascript/up-mobile-config.js

```
$(document).bind("mobileinit", function(){
  $.extend( $.mobile , {
    ajaxEnabled: false,
    ajaxFormsEnabled: false
  });
});
```

# Example jQM Portlets

- uPortal
  - Login / password reset (single view)
  - Search (multi-view)
- Jasig Portlets
  - News
  - Calendar
  - Widgets Video (YouTube)

# jQM + Fluid

- The trick: allow jQM to modify markup
- Example: YouTube portlet

# Using a Single View

- Combine desktop and mobile markup to display the same content to both clients
- Best for simple content
- Reasonable strategy for developing new portlets
  - Developing for mobile first helps keep your interface focused and simple

# Single View Markup Template

```
<!-- Portlet -->
<div class="fl-widget portlet portlet-content" role="section" data-role="content">

  <!-- Portlet Body -->
  <div class="fl-widget-content portlet-body" role="main">

    <!-- Portlet Section -->
    <div id="{n}search" class="portlet-section" role="region">

      <div class="portlet-section-body">

        <p>Hello World!</p>

      </div>

    </div>

  </div>

</div>

</div>
```



# Multiple Views

- Sometimes you need a different feature or interface on one platform
- Share backend logic as much as possible
  - Business logic
  - Controllers
- Render separate view layers as needed

# Is it Mobile?

- Parse user agent
  - User agent available as request property
  - Strategy in uPortal 3.2 and below
  - Maybe more portal-agnostic
  - Potential for mismatch between theme and portlet
- Make decisions based on reported theme
  - Theme is provided by portal as a request property
  - Requires mapping theme name -> mobile in portlet
  - Used by uPortal 4+

# Parsing the User Agent

```
private List<Pattern> mobileDeviceRegexes = null;

/**
 * Set a list of regex patterns for user agents which should be considered
 * to be mobile devices.
 *
 * @param patterns
 */
public void setMobileDeviceRegexes(List<String> patterns) {
    this.mobileDeviceRegexes = new ArrayList<Pattern>();
    for (String pattern : patterns) {
        this.mobileDeviceRegexes.add(Pattern.compile(pattern));
    }
}

@Override
public boolean isMobile(PortletRequest request) {
    String userAgent = request.getProperty("user-agent");

    // check to see if this is a mobile device
    if (this.mobileDeviceRegexes != null && userAgent != null) {
        for (Pattern regex : this.mobileDeviceRegexes) {
            if (regex.matcher(userAgent).matches()) {
                return true;
            }
        }
    }

    return false;
}
```

# Using the Theme Name

```
protected static final String THEME_NAME_PROPERTY = "themeName";
protected static final String MOBILE_THEMES_KEY = "mobileThemes";
protected static final String[] MOBILE_THEMES_DEFAULT = new String[]{ "UniversalityMobile" };

@Override
public boolean isMobile(PortletRequest request) {
    String[] mobileThemes = request.getPreferences().getValues(MOBILE_THEMES_KEY,
        MOBILE_THEMES_DEFAULT);
    String themeName = request.getProperty(THEME_NAME_PROPERTY);
    if (themeName == null) {
        return false;
    }

    for (String theme : mobileThemes) {
        if (themeName.equals(theme)) {
            return true;
        }
    }

    return false;
}
```

# jQM-only Markup Template

```
<div class="portlet">  
  
  <div data-role="header" class="titlebar portlet-titlebar">  
    <a href="javascript:;" data-role="button" data-icon="back" data-inline="true">Back</a>  
    <h2 class="title news-reader-feed-title">Title</h2>  
  </div>  
  
  <div data-role="content" class="portlet-content">  
  
    <p>Hello World!</p>  
  
  </div>  
</div>
```

# Using the Feed Class

```
<div class="portlet">
  <div data-role="content" class="portlet-content">

    <ul data-role="listview" class="feed">
      <c:forEach items="${ feed.entries }" var="entry">
        <li>
          <a href="${ entry.link }">
            
            <h3 class="title">${ entry.title }</h3>
            <p>
              ${ entry.description.value }
            </p>
          </a>
        </li>
      </c:forEach>
    </ul>

  </div>
</div>
```

# Providing Badge Numbers

- Set portlet response property from portlet
- Can implement using a handler interceptor

# Badge Numbers

```
public class MinimizedStateHandlerInterceptor extends HandlerInterceptorAdapter {

    @Autowired(required = true)
    private ICoursesDao coursesDao;

    @Override
    public boolean preHandleRender(RenderRequest request, RenderResponse response,
        Object handler) throws Exception {

        if (WindowState.MINIMIZED.equals(request.getWindowState())) {

            CourseSummaryWrapper summary = coursesDao.getSummary(request);
            int newCount = summary.getNewAnnouncementCount();
            response.setProperty("newItemCount", String.valueOf(newCount));

            return false;
        }

        return true;
    }
}
```



# Including jQM in a Portlet

```
<script type="text/javascript">
$( "[data-role=page]" ).live( 'pagebeforecreate', function( event ){
    return false;
} );
</script>
<script type="text/javascript"
    src="<rs:resourceURL value="/rs/jquery-mobile/1.0a3/jquery.mobile-1.0a3.js"/>">
</script>
```

# Optimizing JS/CSS Libraries

- Reduce browser traffic as much as possible
  - Minify
  - GZip
  - Cache
- Use Jsig filters
  - Only apply long-term cache headers to resources that will really, truly never change

# Compress Inline JS

```
<script type="text/javascript"><rs:compressJs>  
  // your code goes here!  
</rs:compressJs></script>
```

# ETags

- String marks version of a page/resource
- Browser only re-downloads if the tags don't match
- Use Spring's `ShallowETagHeaderFilter` to automatically set ETags

# Native App Development

# Titanium Appcelerator

- JavaScript platform for creating mobile apps
- Compiles OS-independent JS into Android and iPhone projects
- Open source Apache 2 licensed project

# Titanium APIs

- Views
- Network
- Map
- Contacts

# Anatomy of a Project

- Driven by a main app.js file
- Can include other javascript files



# Titanium Documentation

- <http://developer.appcelerator.com/documentation>
- Includes APIs, Getting Started guide
- For API, use PDF, not web

# **uMobile Native Architecture**

# uMobile Documentation

- <https://wiki.jasig.org/display/UPC/uMobile>
- Documentation on
  - Native Architecture
  - Installing uMobile
- Work in progress . . .

# Facade

- Simple dictionary, created in app.js
- A singleton with access to all views, models, and some controllers.
- Also provides access to other information (localization, stylesheet)
- Usually called "app" inside controllers. Gets into window controllers by creating an "app" attribute when creating the window in app.js

# Local Dictionary

- Allows translation of all native text (not text provided from web services or web views)
- In project root: localization.js
- Currently contains en\_US
- Could easily add another language, and switch the property in config.js:  

```
Titanium.App.Properties.setString('locale','en_US');
```
- Implement in any controller or view that has access to the facade, `app.localDictionary.key`. Will automatically display the correct language.

# Models/Proxies

- Are responsible for managing data for native windows (map points for search, directory entries for search).
- Found in js/models.
- Aren't aware of controllers, provide public methods to access data and search for data
- Broadcast application-level events (for now) so that controllers can respond to model events (such as search begin, search complete)

# Controllers

- Each window that is opened in `app.js` is bound to a controller, found in `js/controllers`.
- Responsible for initializing views, reacting to events, and displaying data.
- Currently, tightly coupled with views, although much view logic has been abstracted into factory-type scripts in `js/views`.

# Views / Factories

- Accept two arguments: facade instance, and optional attributes object to be passed into Titanium to render view.
- Found in js/views
- Views are usually added as references to the facade, sometimes as instances (as in the case of the global alert window which has no reason to have multiple instances.).
- Usually instantiated with new keyword, titleBar = `new app.views.GenericTitleBar(app, {backgroundColor: '#f00'});`

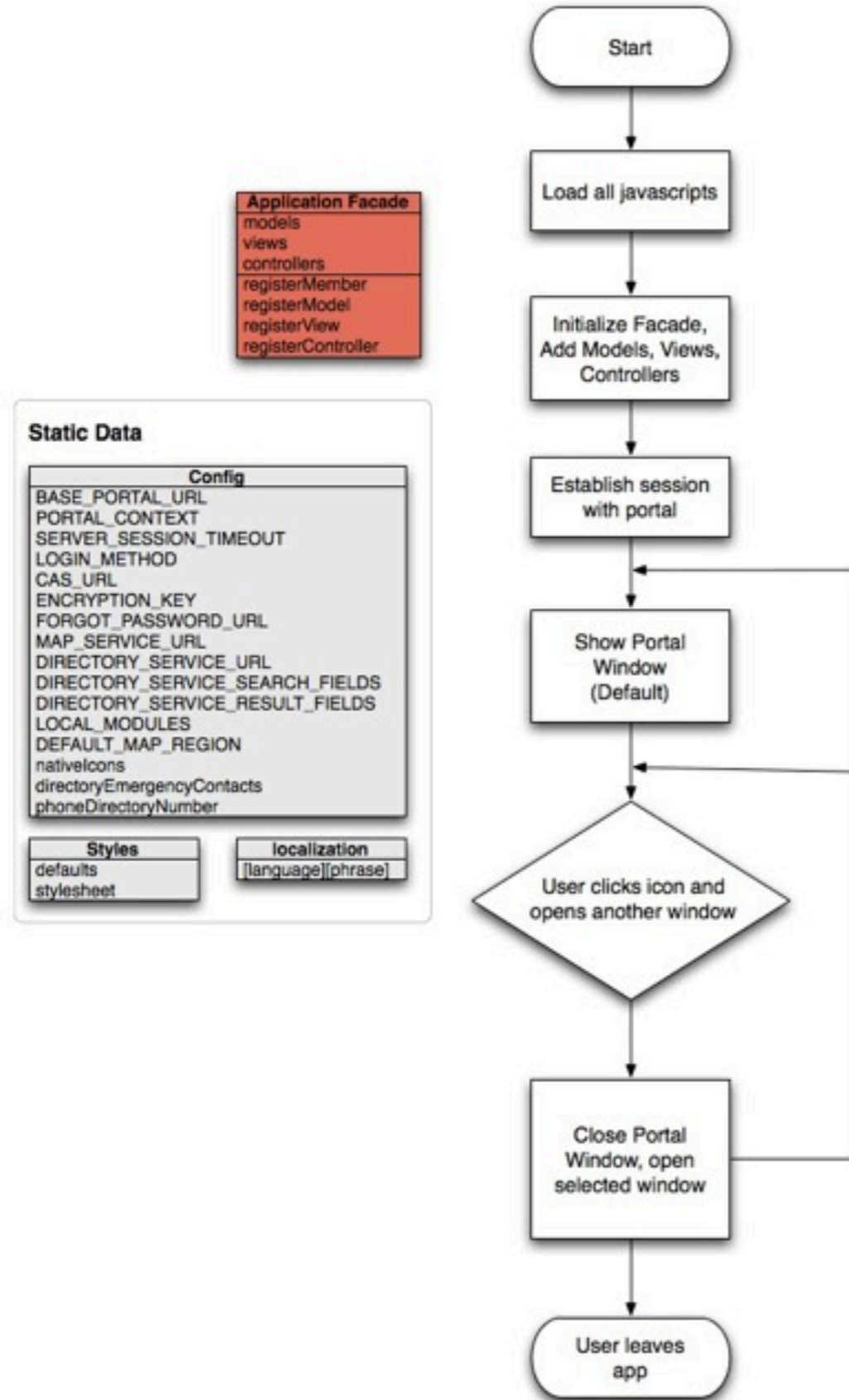


# Stylesheet

- A Javascript dictionary of styles, attached to the facade as "styles".
- Global stylesheet in resources root: style.css
- Not automatically applied to elements with selectors, each "class" must be passed in as object to view factories during creation, such as `searchButton = Titanium.UI.createButton(app.styles.searchButton);`

# Application Lifecycle and Entities

## uMobile v1.0



**Application Facade**

- models
- views
- controllers
- registerMember
- registerModel
- registerView
- registerController

**Static Data**

**Config**

- BASE\_PORTAL\_URL
- PORTAL\_CONTEXT
- SERVER\_SESSION\_TIMEOUT
- LOGIN\_METHOD
- CAS\_URL
- ENCRYPTION\_KEY
- FORGOT\_PASSWORD\_URL
- MAP\_SERVICE\_URL
- DIRECTORY\_SERVICE\_URL
- DIRECTORY\_SERVICE\_SEARCH\_FIELDS
- DIRECTORY\_SERVICE\_RESULT\_FIELDS
- LOCAL\_MODULES
- DEFAULT\_MAP\_REGION
- nativelcons
- directoryEmergencyContacts
- phoneDirectoryNumber

**Styles**

- defaults
- stylesheet

**localization**

- [language][phrase]

**Views**

**UI**

- createSearchBar
- createTitleBar
- createSecondaryNavBar
- createActivityIndicator

**Person Detail Table View (Extends Ti.UI.TableView)**

- data
- update
- appendRow

**Map Detail Top**

- update

**Shared Web View**

- getWebView
- show
- hide
- stopLoading
- setTop
- getExternalUrl
- getLocalUrl
- onWebViewLoad

**Controllers**

**Portal Window Controller**

- Key
- Open
- Close

**Directory Window Controller**

- Key
- Open
- Close

**Map Window Controller**

- Key
- Open
- Close

**Settings Window Controller**

- Key
- Open
- Close

**Portlet Window Controller**

- Key
- Open
- Close

**Directory Detail Controller (extends Ti.UI.View)**

- initialized
- construct
- update

**Map Detail View Controller (extends Ti.UI.View)**

- updateAndShow

**Models / Proxies**

**Portal Proxy**

- getShowPortletFunc
- getPortlets
- getIconURL
- getPortletsForUser

**Directory Proxy**

- getPeople
- getEmergencyContacts
- search
- clear

**Map Proxy**

- requestErrors
- init
- search
- getAnnotationByTitle
- loadMapPoints
- getMapCenter

**Login Proxy**

- sessionTimeContexts
- loginMethods
- getCredentials
- isValidNetworkSession
- isValidWebViewSession
- saveCredentials
- updateSessionTimeout
- establishNetworkSession
- getLocalLoginURL
- doLocalLogin
- getCASLoginURL
- doCASLogin

**Session Proxy**

- isActive
- createSessionTimer
- resetTimer
- stopTimer
- validateSessions

**App & Device Models**

**Resource Proxy**

- getResourcePath
- getPortletIcon

**Device Proxy**

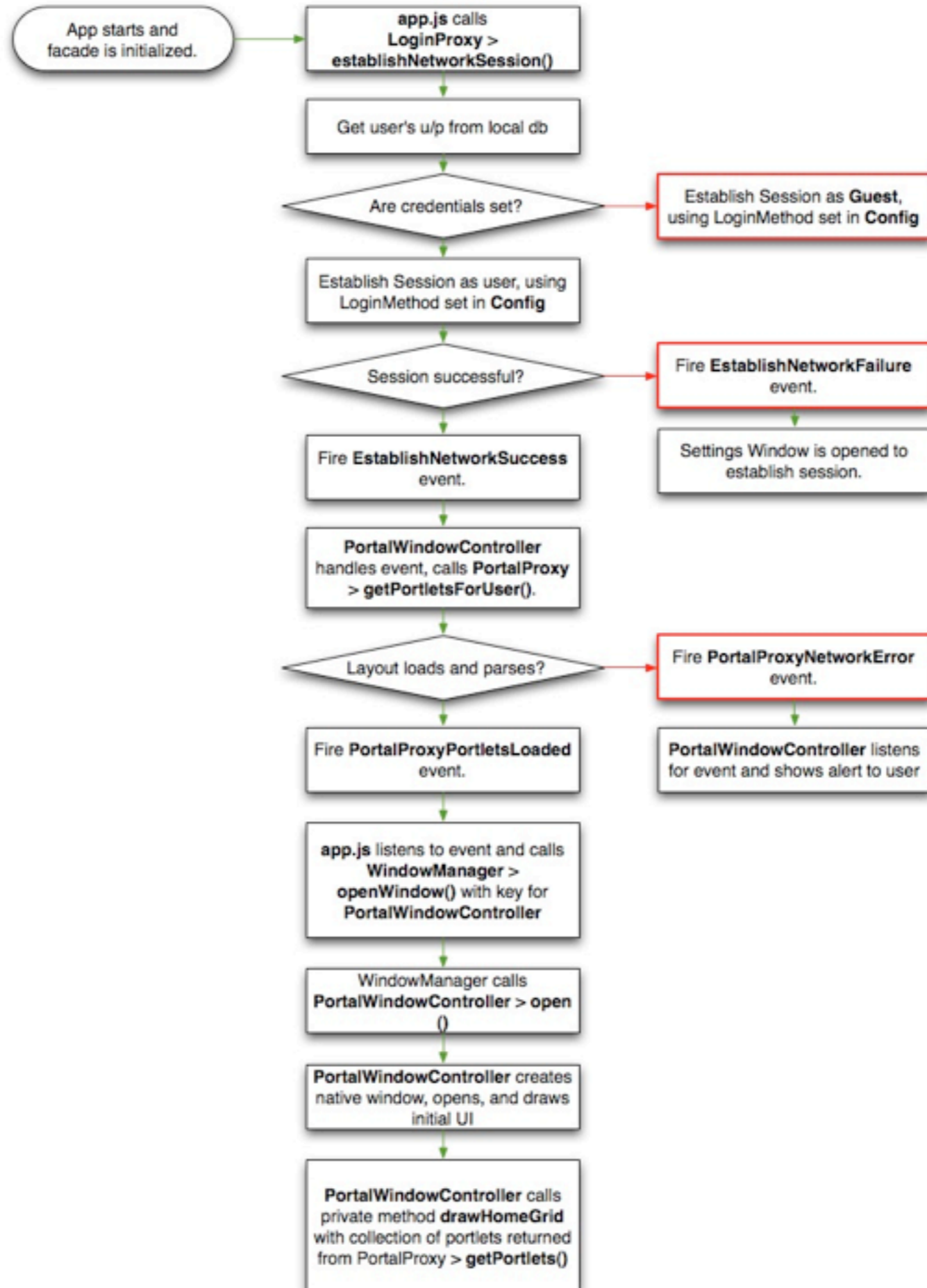
- checkNetwork

**Window Manager**

- getCurrentWindow
- getCurrentPortlet
- getPreviousWindow
- addWindow
- openWindow
- goBack

# App Session Initialization Process

uMobile v1.0



# Getting Started

# Preparing for Native Development

- Install Titanium
  - Lots of extra steps for Windows :-)
- Install an Emulator
  - Android on all platforms
  - iPhone on OS X only
- Create a new project
- Replace the /Resources directory with code from SVN

# Some Tips

- Troubleshooting: When in doubt, clean!
  - for the iPhone, delete build/iphone
  - for Android . . . ?
- Android emulator performance
  - Don't press “stop”, just relaunch
  - jlewis's virtual box instance

# Configuring uMobile

- Most settings in config.js
  - Base portal URL
  - Authentication method
  - Person directory information
  - Emergency contact numbers

# Skinning uMobile

- Controlled by style.js
- CSS-like javascript file provides style attributes for new objects



# Localizing uMobile

- Messages in localization.js
  - Simple dictionary of translations
- Configure default language in config.js
  - In the future, maybe get default locale from portal?
- Using the messages
  - `app.localDictionary.translatedTextStringKey`

# Adding External Sites

config.js

```
UPM.LOCAL_MODULES.twitter = {  
  title: 'Twitter',  
  fname: 'twitter',  
  url: 'http://mobile.twitter.com/searches?q=jasig11',  
  externalModule: true  
};
```

# Overriding Portlets

config.js

```
UPM.LOCAL_MODULES.directory = {  
  title: 'directory',  
  fname: 'directory',  
  iconUrl: 'icons/directory.png',  
  doesRequireLayout: true,  
  window: 'directory'  
};
```

# Adding Native Modules

- Follow tutorial at <https://wiki.jasig.org/display/UPC/Build+a+Native+Module+-+Hello+World>
- High-level steps
  - Create controller
  - Add facade to window manager
  - Add to config and set icon

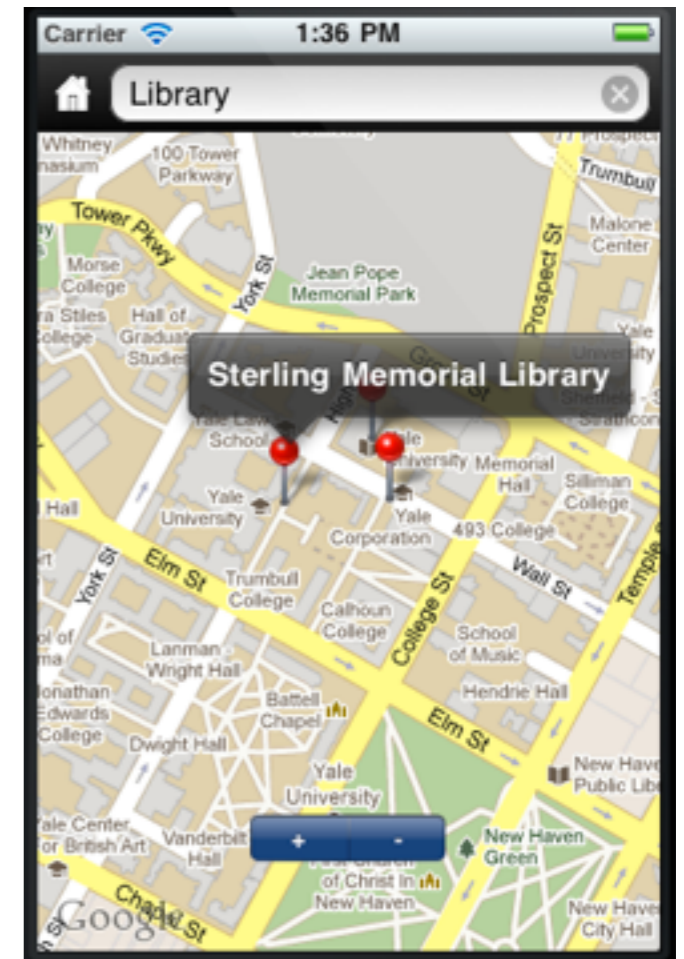
# uPortal Server

Request data from  
uPortal map service

Return location  
data as JSON

# uMobile App

Present native components  
based on portal data



# The Apple Store

- Developing
  - Need to pay \$99 for a developer account
  - Must provision devices under account
- Submitting
  - Make sure to handle network errors and other exceptions
  - Use at least a 1.0 version number
  - Time frame: ~1 week?

# The Future

# Performance

- Implement caching and gzipping for resource requests
- Investigate JS loading and parsing performance



# Layout Support

- Multiple Layouts / User
  - Requires fix to uPortal codebase
  - Development item for 4.1
- Add mobile theme layout editing