

Google™



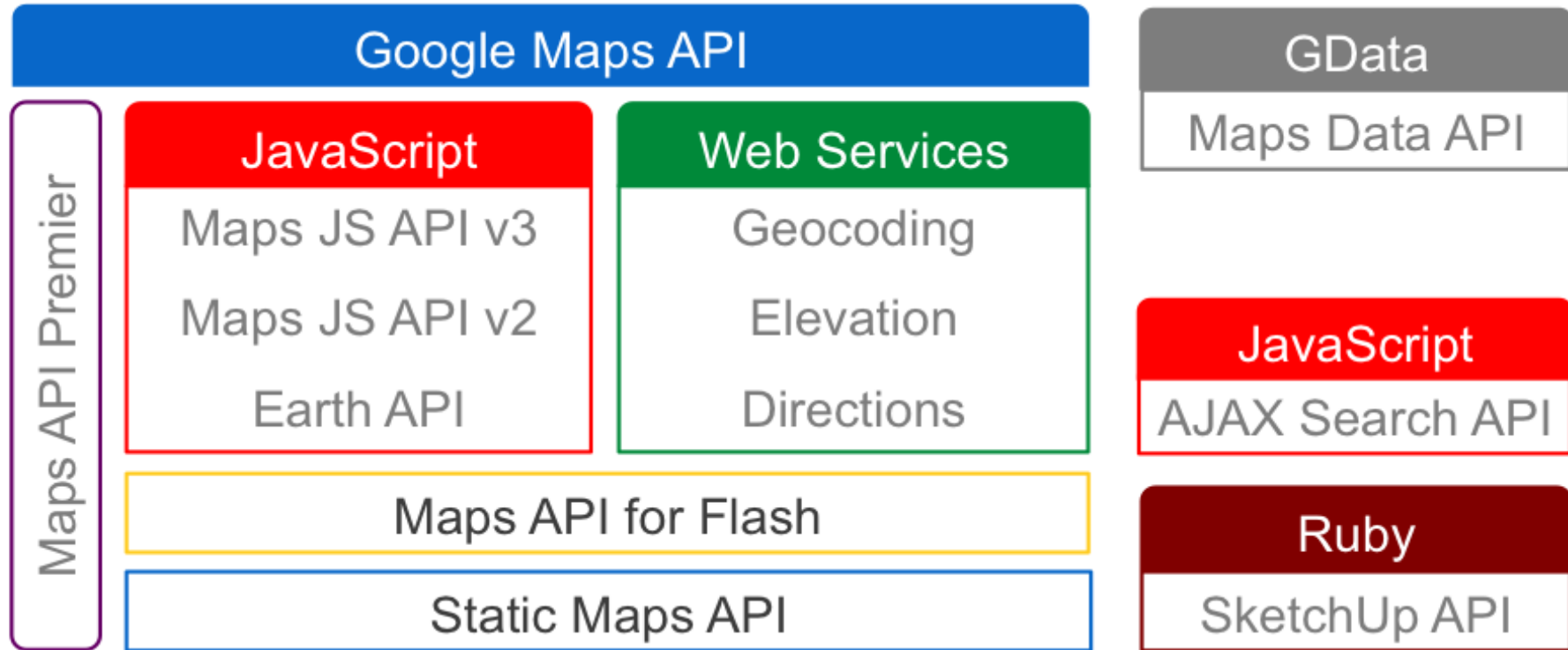
Map once, map anywhere: Developing geospatial applications for both desktop and mobile

Mano Marks and Chad Killingsworth
May 19th, 2010

Write Once, Run Anywhere

- Mobile is hot
- Desktop still hot
- Save development time
- Faster development turn around
- More featured API

Overview of Geo APIs



Agenda

- Overview of Geo APIs
- Different options for mobile
- UI Considerations
- Geolocation
- Real World App

Questions and Notes

*View live notes and ask questions about
this session on Google Wave*

<http://bit.ly/97ZfnR>

Options for Google Maps on Mobile

- Maps API V3 in Browser
- iPhone Native MapKit
- Android Native MapView
- Hybrid Native with Browser
- Static Maps API

Google Maps APIs Comparison

Feature	Google Maps API V3	Google Maps on iPhone	Google Maps on Android	Static Maps API
Add a Map	X	X	X	X
Markers	X	X	X	X
Geocoding	X		X	X
Reverse Geocoding	X	X	X	
Polygons/PolyLines*	X	4.0 SDK only		X
Custom Map Tiles	X			
Elevation	X			
KML/GeoRSS Layers	X			
Directions	X			
Traffic Overlay	X			

And, of course, Street View

Feature	Google Maps API V3	Google Maps on iPhone	Google Maps on Android	Static Maps API
Street View	New!	4.0 and later	X	

Street View just added to Maps API V3

- HTML 5 Canvas 2D
- HTML 4
- WebGL

Street View Demo in V3

Wave Questions and Discussion: <http://bit.ly/97ZfnR>

Street View code on map

```
map.setOptions({  
  streetViewControl: true  
})  
  
var panorama = map.getStreetView();  
  
panorama.setPosition(  
  new google.maps.LatLng(48.85969,  
    2.29720));
```

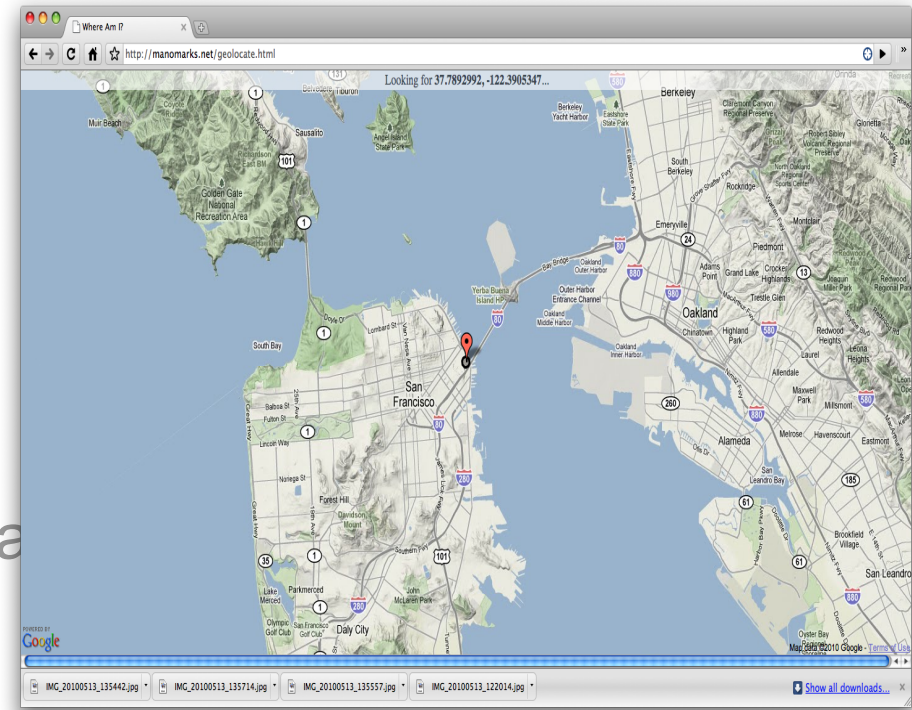
Street View code - without map

```
var div = document.getElementById('panoramadiv');
```

```
var panorama = new  
  google.maps.StreetViewPanorama(div);
```

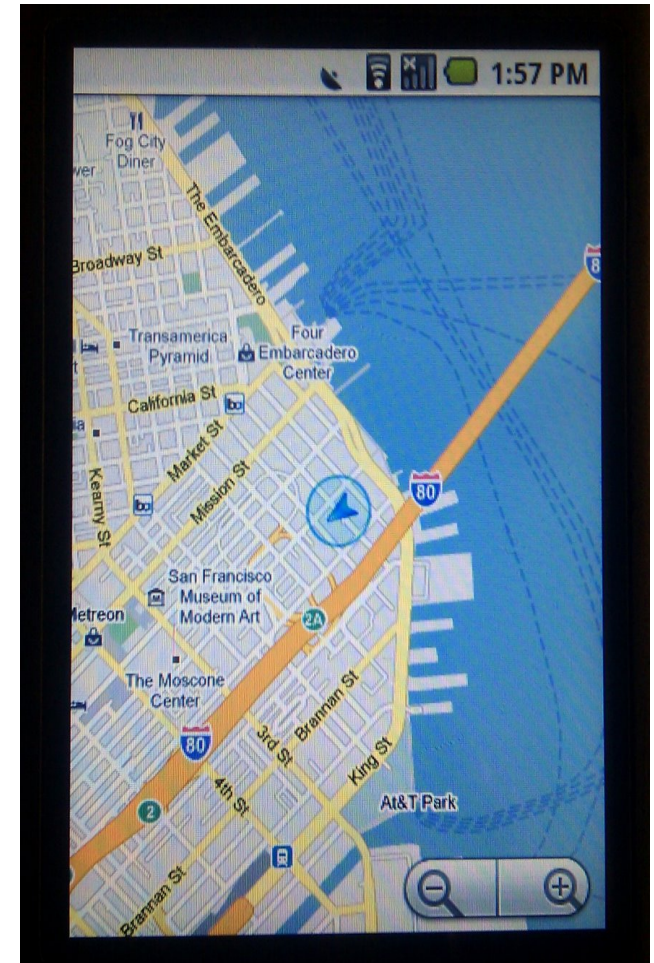
Browser based maps

- Full JavaScript browsers
- Access to some phone features
- HTML 5
- Write once
- Rapid development
- No App Store/Marketplace process
- No App Store/Marketplace discovera



Native APIs

- MapKit on iPhone
- MapView on Android
- App Store/Marketplace discoverability
- App Store/Marketplace launch process
- Harder development
- No support outside of platform



Hybrid Native Apps with Embedded Browser

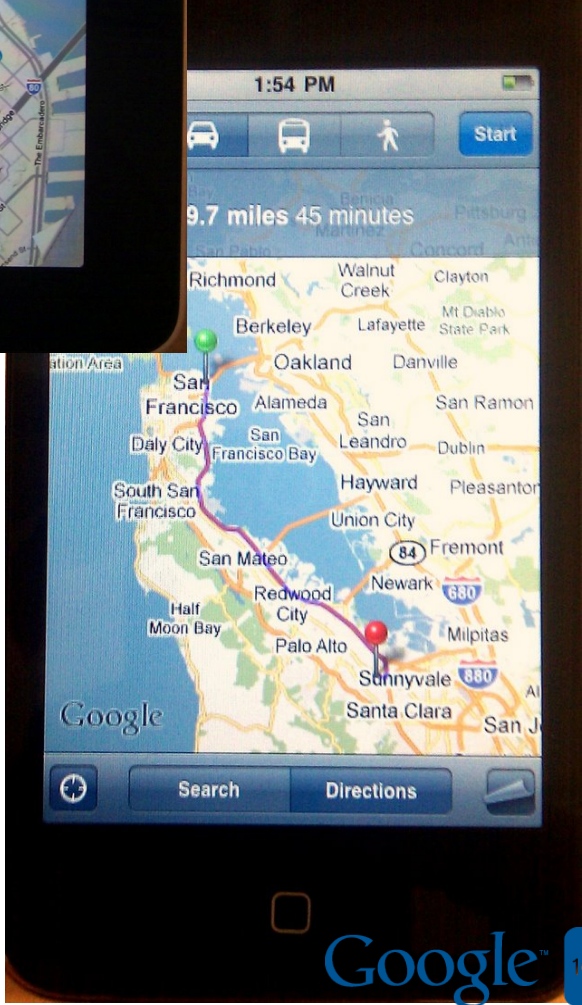
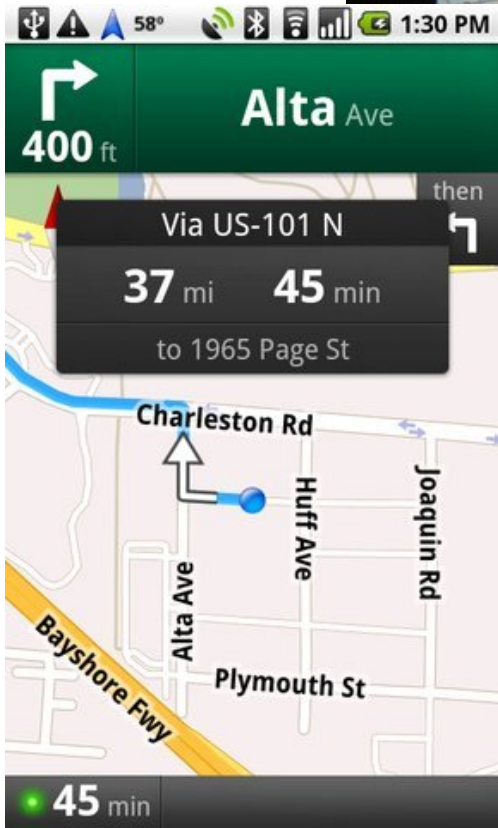
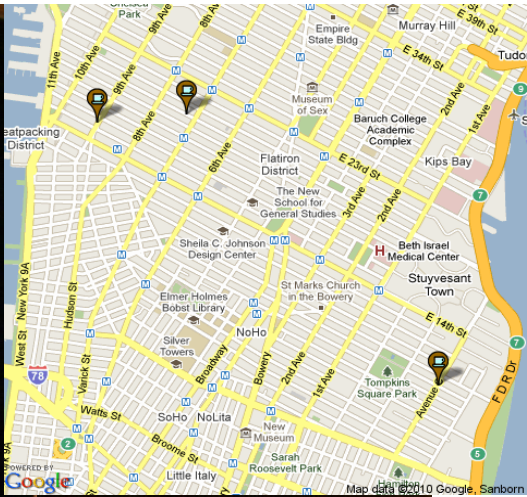
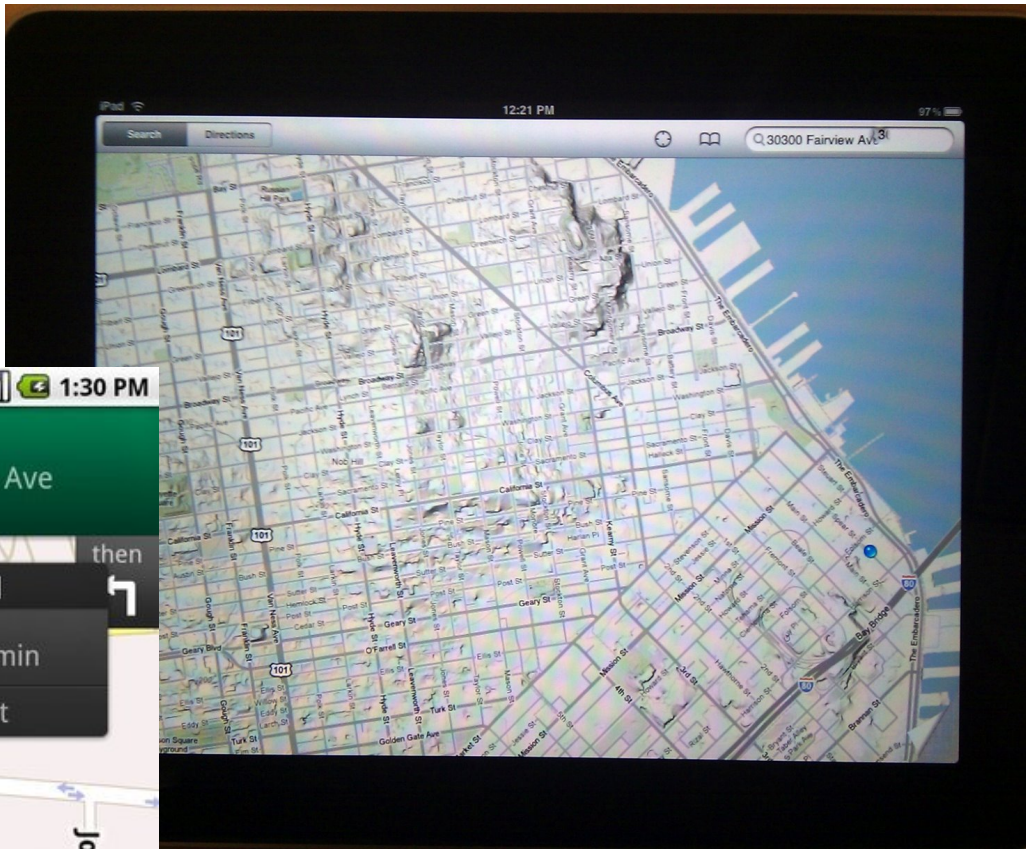
- WebView on Android
- uiWebView in iPhone
- Access to additional features of phone
- Rapid development of map
- Map is write once, but app is write per platform
- App Store/Marketplace discoverability/launch process

Static Maps API

- Any browser
- Lightweight and fast
- No features of modern APIs
- Write once, run anywhere
 - and really, anywhere

UI Considerations

- Size of screen layout
 - Make your div's flexible
 - Vary your chrome by browser or screen size
- Touch events
- Native vs. Browser look and feel



Wave Questions and Discussion: <http://bit.ly/97ZfnR>

Geolocation

- HTML 5
 - Device provides location
 - Mobile often gives GPS location
 - Desktop browser gives ip or wifi
- IP Based
 - IP lookup
 - coarse
 - Google Ajax API ClientLocation
 - or other provider

HTML 5 Geolocation Basics

```
function showMap(position) { // Show a map centered  
    at position  
} // One-shot position request navigator.geolocation.  
getCurrentPosition(showMap);
```

But, there's more!

```
navigator.geolocation.watchPosition( function)
```

```
navigator.geolocation.getCurrentPosition  
    (successCallback,  
     errorCallback,  
     {maximumAge:3600})
```

```
position.timestamp
```

```
position.coords.accuracy
```

But, Wait, what about the Desktop?

HTML 5 geolocation works in some browsers, mostly IP or Wifi based.

Or:

`google.loader.ClientLocation`

Even better:

Run anywhere by using both, failing back to `ClientLocation`

More HTML 5 Goodness

Android 2.0 for instance

- Database API support, for client-side databases using SQL
- Application cache support, for offline applications
- Geolocation API support, to provide location information about the device
- <video> tag support in fullscreen mode

Demo

Campus Map

- Wide variety of audiences using map
- Visitors and guests will visit the web site
- Frequent users prefer the convenience of an application
- Only enough resources to maintain one code base

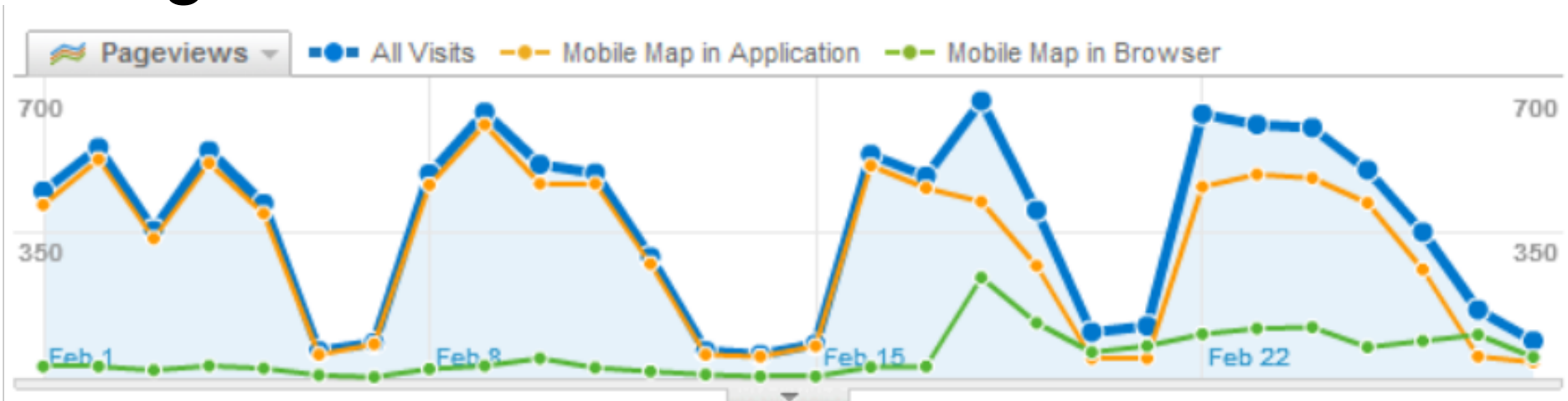
Version 3 Migration

- Shared datasets with version 2 map
- Early on - lack of features
- User interface on a small touch screen

Hybrid Application

- Performance concerns
- It felt like “cheating”

Usage Statistics



Optimizations

- Use KML Layers for complicated data
- Compress JavaScript (Closure-Compiler)
- Delay loading the Maps API so as not to block page rendering
- Use Google Page Speed

Google Developer Qualification



Chrome Extensions



Gadgets



Search



Android



JS Maps API



KML



3D

Questions and Notes

*View live notes and ask questions about
this session on Google Wave*

<http://bit.ly/97ZfnR>

Google™

