
Google™
10

How Maps API v3 came to be

Tips, tricks, and lessons learned in developing a cross platform desktop and mobile API

Susannah Raub, Marc Ridey

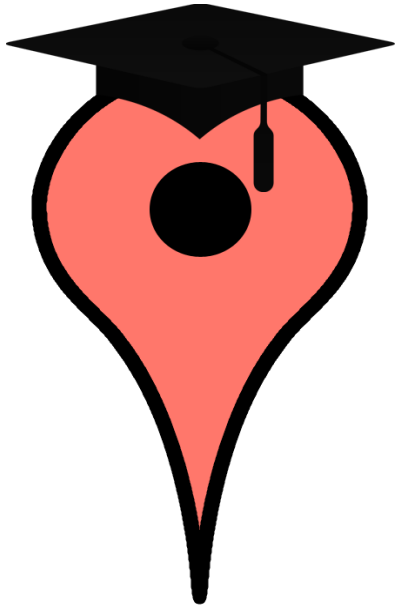
May 20, 2010

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

http://bit.ly/cnQiok

Introductions

Marc Ridey



Maps API v3

Susannah Raub

Google I/O 10

Content

- Latency
- Architecture
- Technology
- Debugging

Experience: Maps API v2

Google I/O 10

Experience: Maps API v2

- 175 kB JavaScript
- 4-6 map images

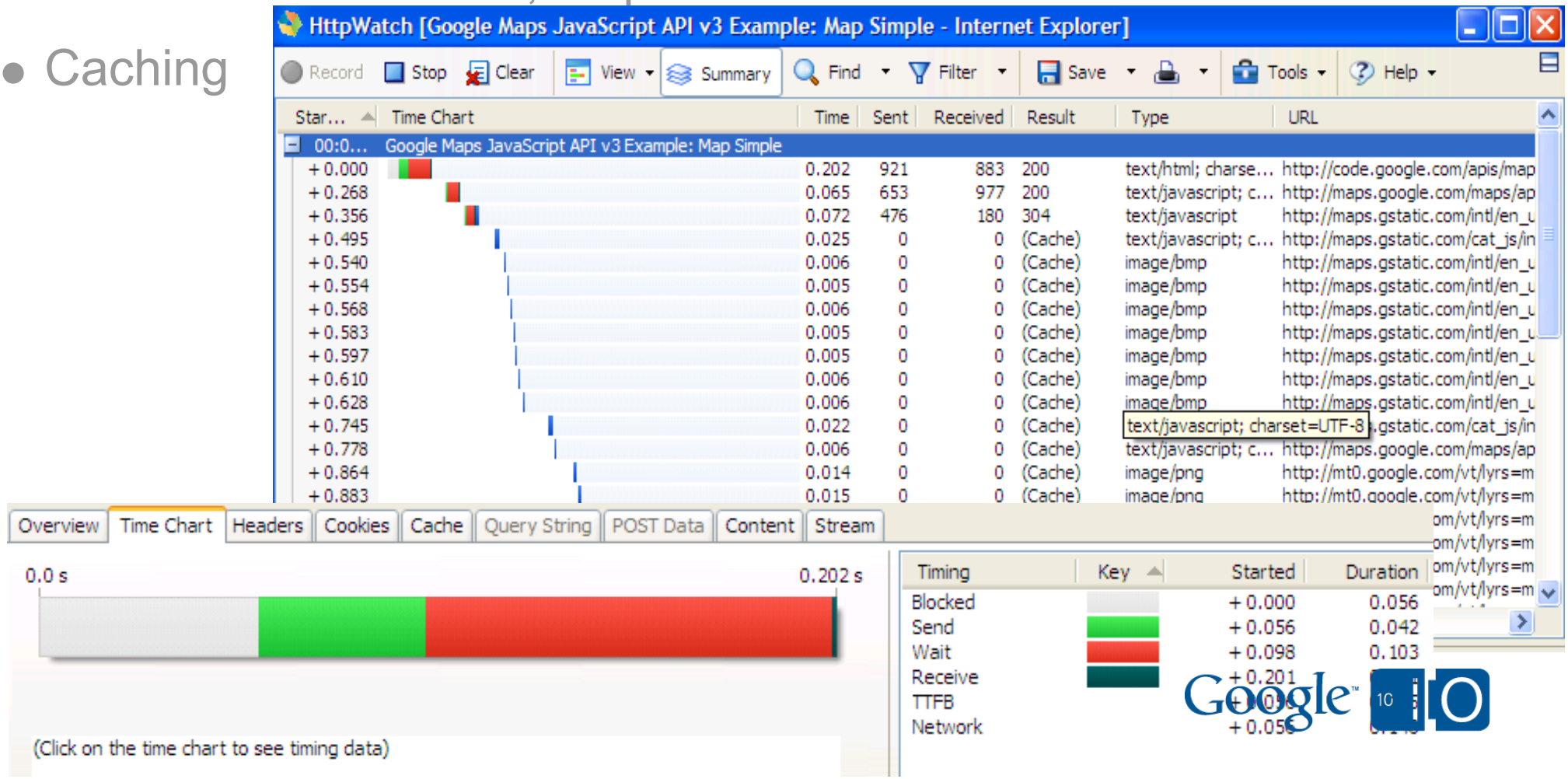
Google I/O 10

Defining Latency

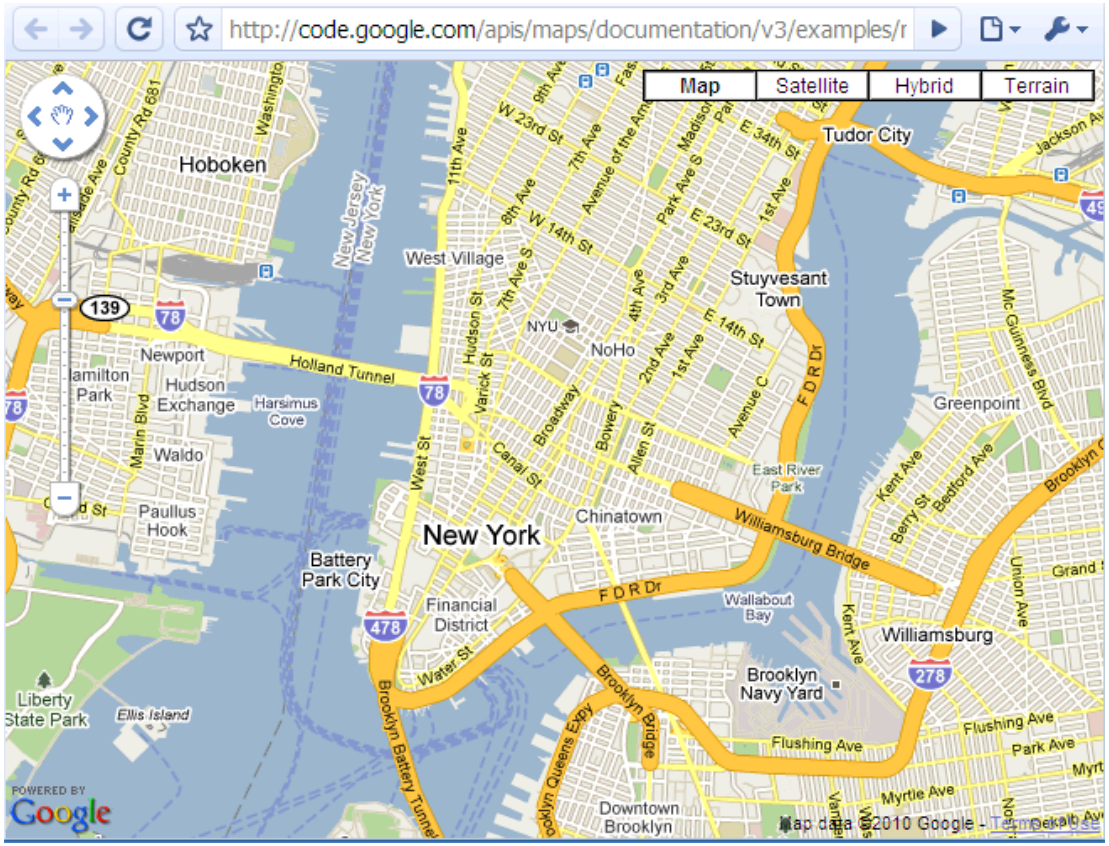
- User perceived latency
  - Page appears usable
- Page ready time
  - Page is usable
- Page load time
  - All elements are present

Measuring Latency: Desktop

- HTTP Watch
  - Internet Explorer and Firefox
- Load time breakdown, sequence
- Caching



Measuring Latency: Desktop vs Mobile



≠

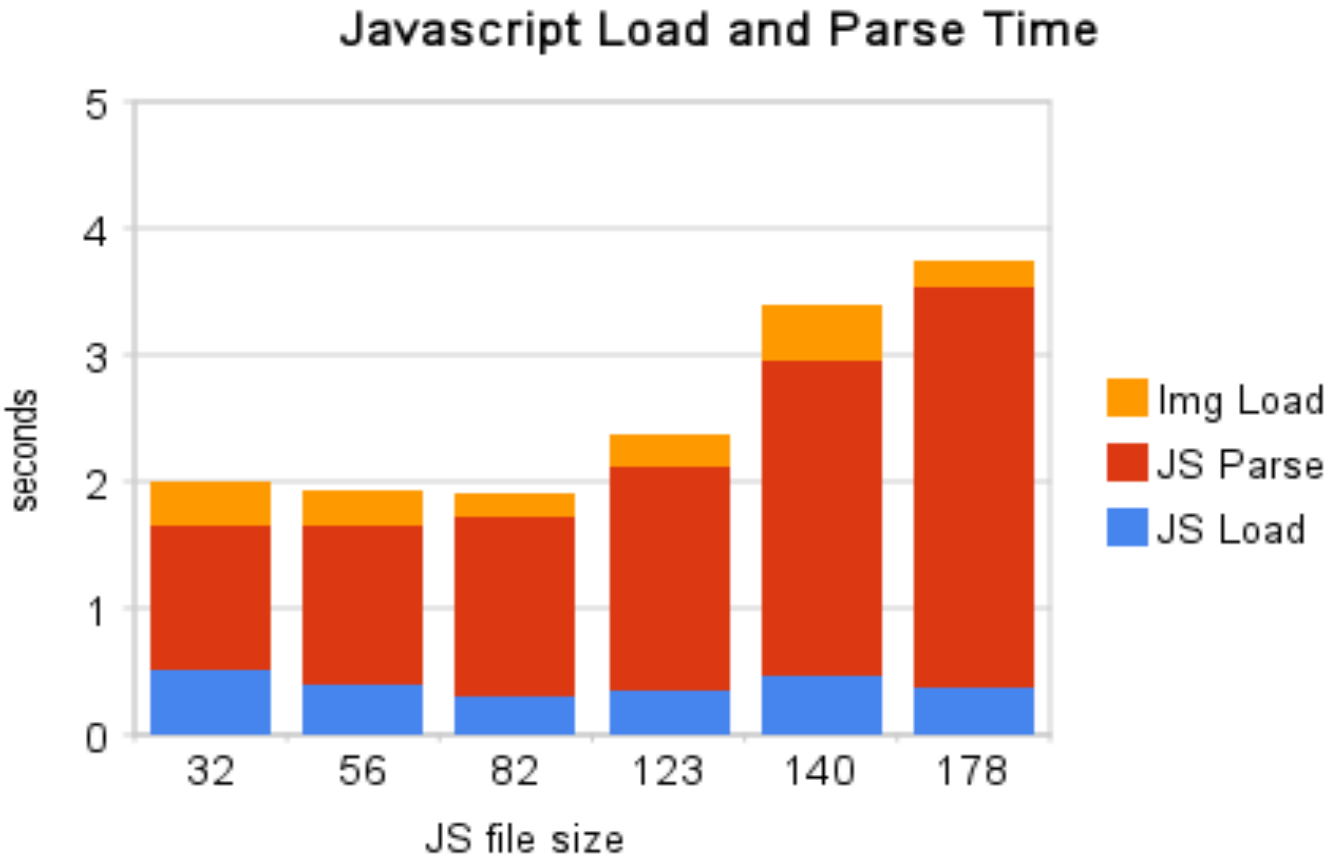
Google I/O 10

Measuring Latency: Mobile

- Using a Fiddler Proxy
- Can run over 3G

Measuring Latency: Mobile

- Takes 1s + ~5 ms/kB to load and parse JavaScript on iPhone 3 *dependent on hardware and OS version

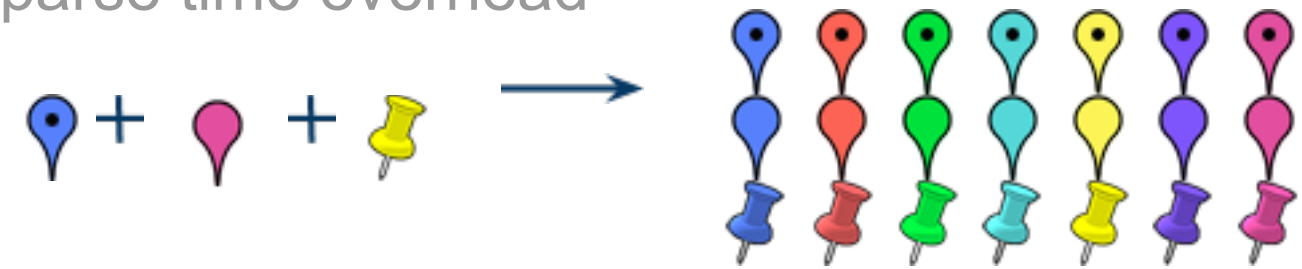


Reducing Latency: Compilers

- Obfuscates property names, reducing code size
- Optimizes for code size
- Plus, lots of helpful error checking
  - Closure - http://code.google.com/closure/compiler/
  - GWT - http://code.google.com/webtoolkit/
  - YUI, Packer, Shrinksafe, and more

Reducing Latency: Reducing downloads

- Techniques
  - Image spriting
  - Combining JS files
    - 1 HTTP request for 57kB
    - Instead of 3 requests for 15-25kB each
- Outcome
  - Fewer HTTP requests
  - Less parse time overhead



Reducing Latency: Mobile Caching

- More constrained than desktop
  - File size restrictions
  - Total size limited
- Cleared more frequently
- Improving with each hardware and OS update

Reducing Latency: HTML5 Database

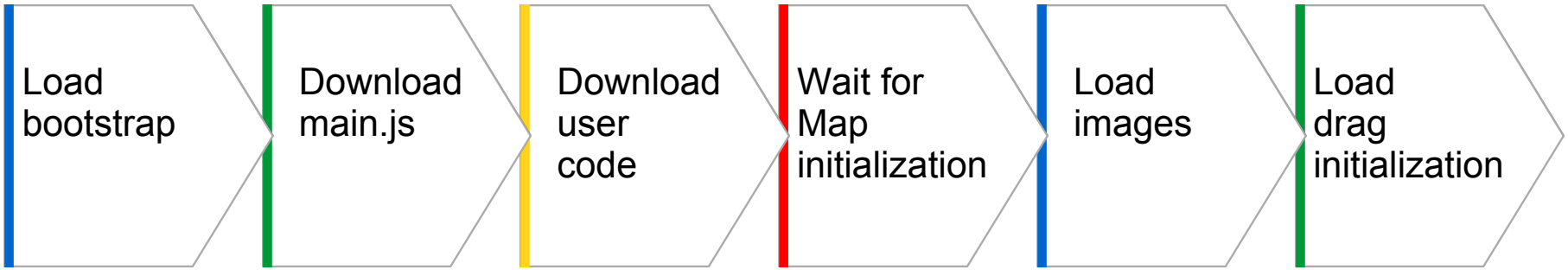
- Allows for data storage within a site
- Great for static files
- Not so great for dynamic content
- Not easy to use cross-domain

Content

- Latency
- Architecture
- Technology
- Debugging

Architecture: Shortcomings of API v2

- Large synchronous public interface
- Slow to start loading map tiles

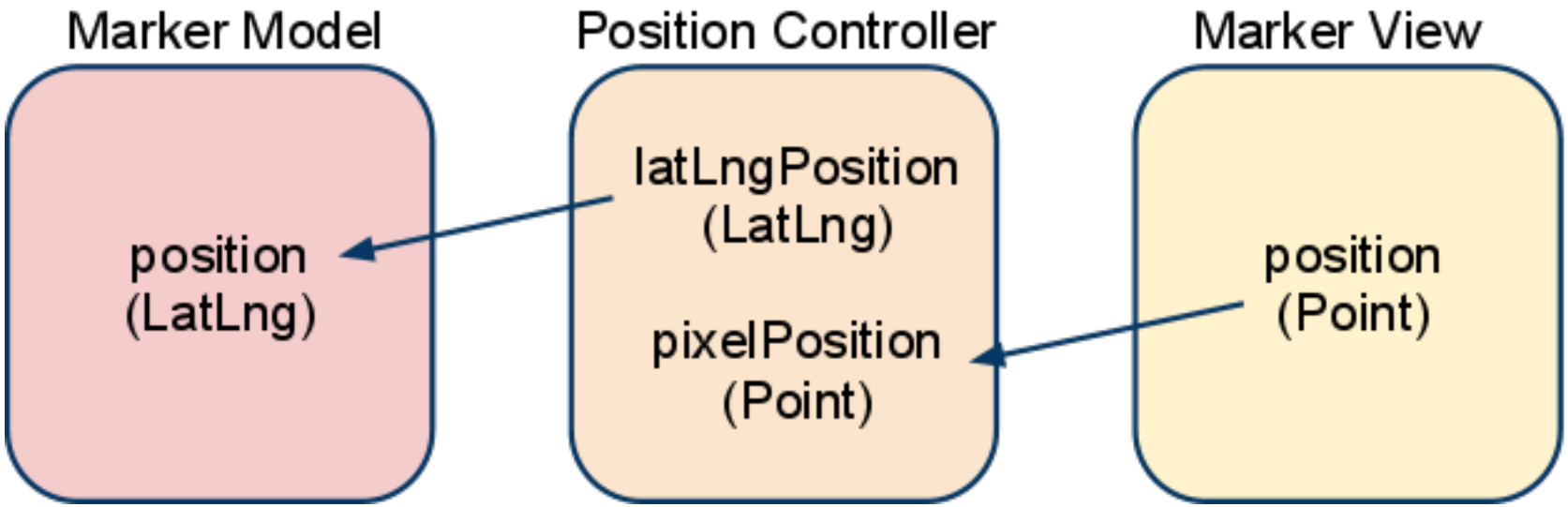


Architecture: Model-View-Controller (MVC)

- Models store state synchronously
- Views render objects asynchronously
- Controllers are go-betweens
- Initial download contains small models
- Views and controllers are loaded on-demand

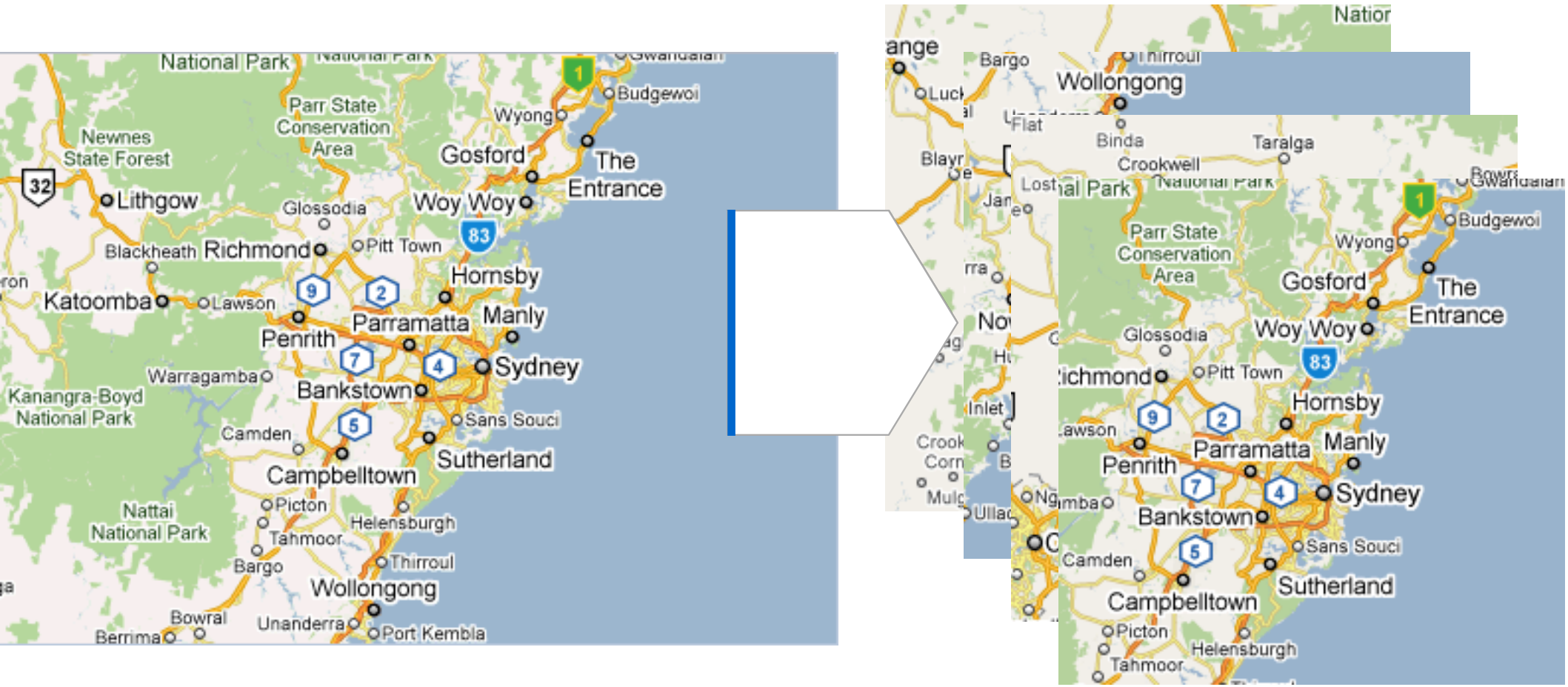
Architecture: Model-View-Controller (MVC)

- Models: Map, Marker, InfoWindow
- Views: DOM rendering
- Views are ignorant of "Map" properties (projection, LatLng, etc)



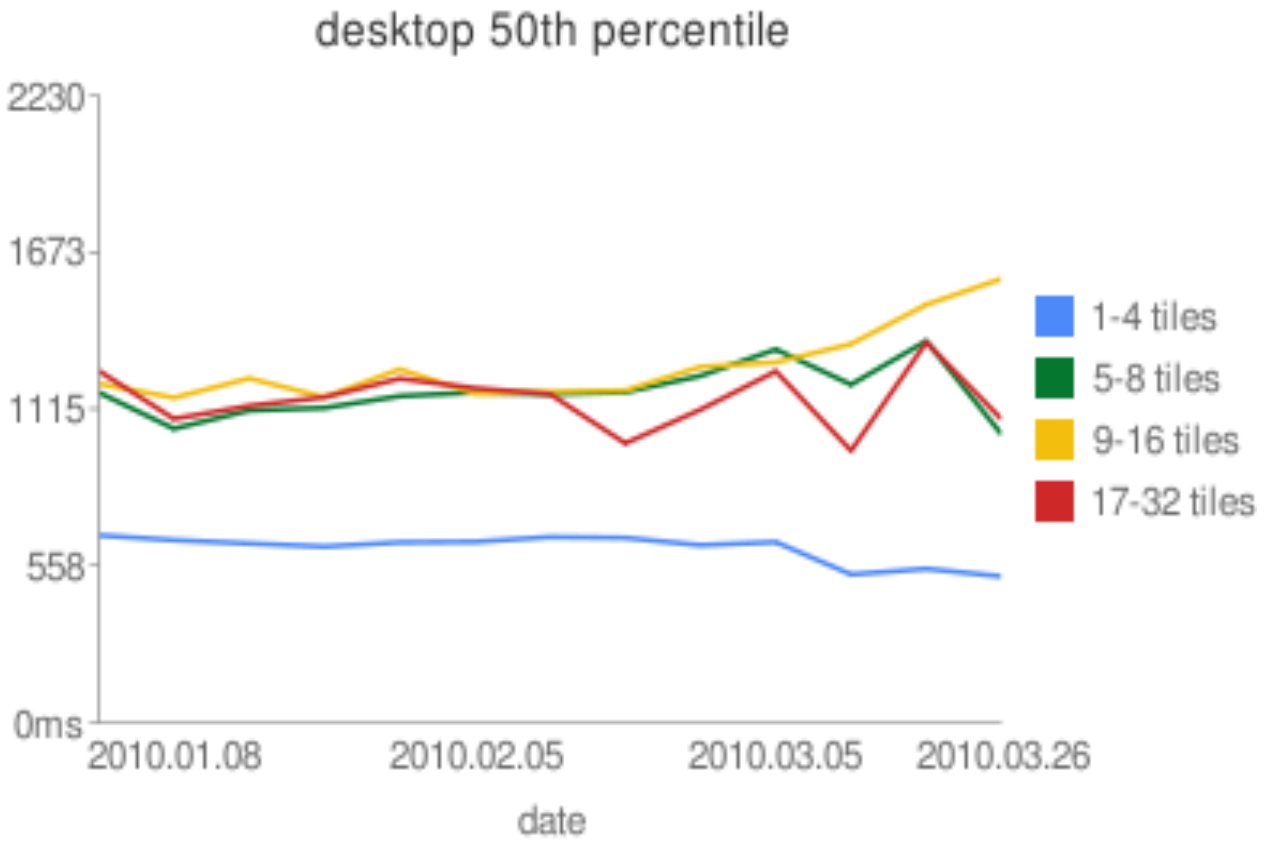
Optimizing

- Loading tiles: 4-6 HTTP requests X 25kB
- Loading one image: 1 HTTP request X 40kB



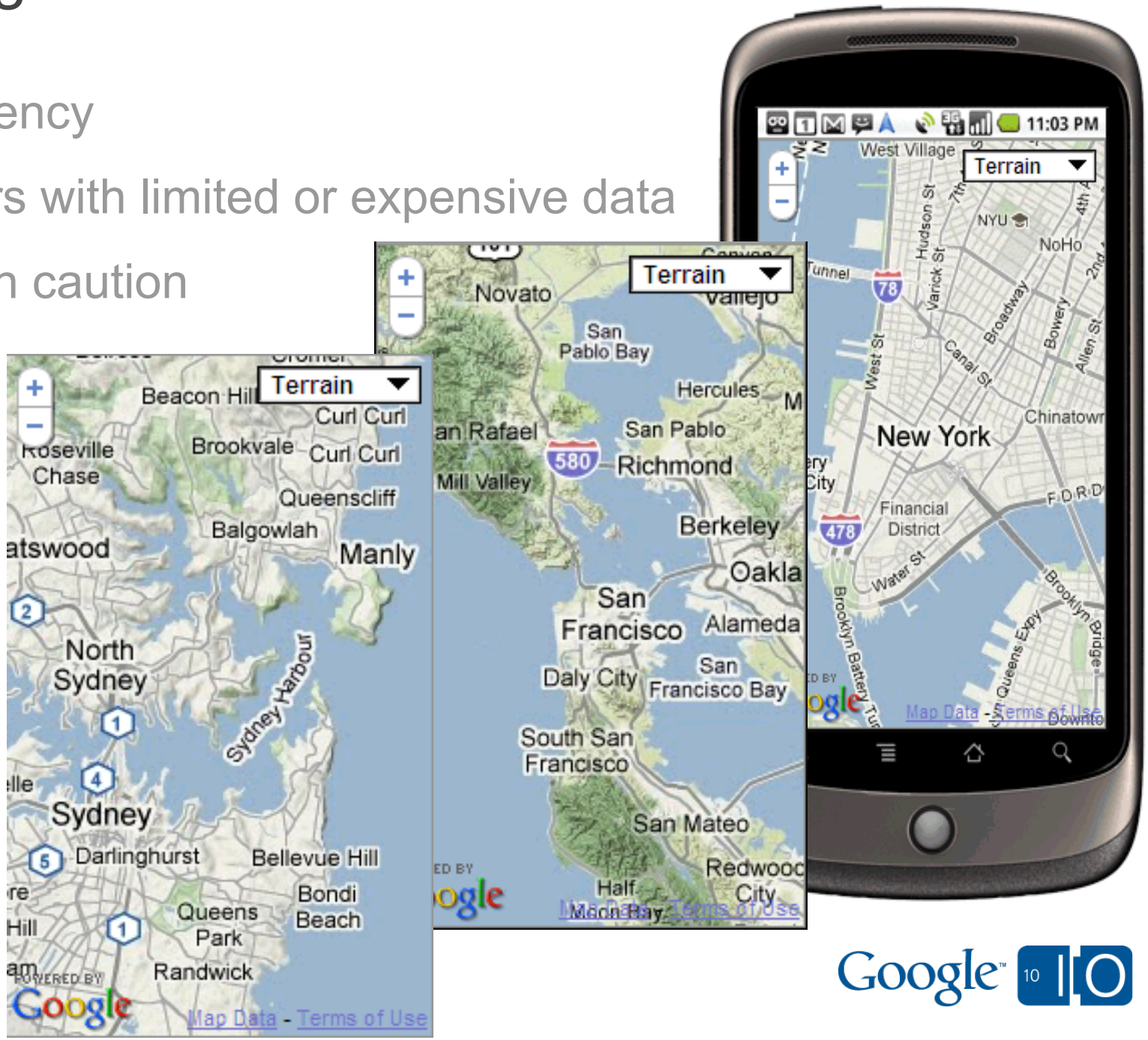
Meeting Goals

- Real world measurements
- Tracking multiple variables



Prefetching

- Good for latency
- Bad for users with limited or expensive data
- Prefetch with caution



Google I/O 10

Content

- Latency
- Architecture
- Technology
- Debugging

Technology

- Graphics
  - Scalable Vector Graphics (SVG)
  - Vector Markup Language (VML)
  - Canvas 2D
  - WebGL, Canvas 3D
- Geolocation
- Touch events

Technology - Graphics

- Mixed technologies: SVG, VML, Canvas, CSS transforms
- Mixed support: Webkit, IE, iPhone, Android
- Mixed mouse/touch events handling capabilities

Used by Maps API v3 in polys, StreetView and the compass control.

Technology - Graphics

Comparison Chart

	Pros	Cons	Coverage
SVG	...	...	...
VML	...	...	...
Canvas 2D	...	...	...
Canvas 3D	...	...	...
CSS Transforms	...	...	...

Technology - Graphics

Comparison Chart

	Pros	Cons	Coverage
SVG	Fast for polys, retained mode, DOM based	Opaque for mouse/touch events, slow for images	Not IE and Android
VML	...	...	...
Canvas 2D	...	...	...
Canvas 3D	...	...	...
CSS Transforms	...	...	...

Technology - Graphics

Comparison Chart

	Pros	Cons	Coverage
SVG	Fast for polys, retained mode, DOM based	Opaque for mouse/touch events, slow for images	Not IE and Android
VML	*same as SVG*	*same as SVG*	Only IE
Canvas 2D	...	...	...
Canvas 3D	...	...	...
CSS Transforms	...	...	...

Technology - Graphics

Comparison Chart

	Pros	Cons	Coverage
SVG	Fast for polys, retained mode, DOM based	Opaque for mouse/touch events, slow for images	Not IE and Android
VML	*same as SVG*	*same as SVG*	Only IE
Canvas 2D	Fast for images	Slow for polys/mobiles, complex events, immediate mode	Not IE
Canvas 3D	...	...	...
CSS Transforms	...	...	...

Technology - Graphics

Comparison Chart

	Pros	Cons	Coverage
SVG	Fast for polys, retained mode, DOM based	Opaque for mouse/touch events, slow for images	Not IE and Android
VML	*same as SVG*	*same as SVG*	Only IE
Canvas 2D	Fast for images	Slow for polys/mobiles, complex events, immediate mode	Not IE
Canvas 3D	Very fast, retained geometry, 3D transforms	complex events	latest nightly builds only. Not IE
CSS Transforms	...	...	...

Technology - Graphics

Comparison Chart

	Pros	Cons	Coverage
SVG	Fast for polys, retained mode, DOM based	Opaque for mouse/touch events, slow for images	Not IE and Android
VML	*same as SVG*	*same as SVG*	Only IE
Canvas 2D	Fast for images	Slow for polys/mobiles, complex events, immediate mode	Not IE
Canvas 3D	Very fast, retained geometry, 3D transforms	complex events	Not IE. latest nightly builds only
CSS Transforms	Good for animations	Mixed performance, Not useful for polys	Not IE

Technology - Graphics

Comparison Chart

	Pros	Cons	Coverage
SVG	Fast for polys, Retained mode, DOM based	Opaque for mouse/touch events, slow for images	Not IE and Android
VML	*same as SVG*	*same as SVG*	Only IE
Canvas 2D	Fast for images	Slow for polys/mobiles, complex events, immediate mode	Not IE
Canvas 3D	Very fast, retained geometry, 3D transforms	Complex events	Not IE. latest night builds only
CSS Transforms	Good for animations	Mixed performance, not useful for polys	Not IE

Technology - Canvas 2D

Render time for StreetView panorama

per frame	Chrome	Firefox	Safari	Android
Linux	25ms (v5.0)	56ms (v3.0)	--	--
MacBook Pro	280ms (v5.0)	292ms (v3.6)	67ms (v4.0.5)	--
Windows XP	33ms (v4.1)	863ms (v3.0)	105ms (v4.0.5)	--
iPad	--	--	801ms	--
iTouch v3.1.2	--	--	1904ms	--
Nexus One	--	--	--	340ms

Desktop and iPad tests done with a 600*400 pixels view, iTouch with 320*400 pixels, and Nexus One with 600*400 pixels.

Linux Chrome: 400*300: 20ms, 600*400: 25ms, 800*600: 52ms, 1200*800: 79ms.

Technology - Canvas 2D and WebGL

Render time for StreetView panorama

per frame	Chrome	Firefox	Safari	Android
Linux	25ms (v5.0) **0.9ms (WebGL)**	56ms (v3.0)	--	--
MacBook Pro	280ms (v5.0)	292ms (v3.6)	67ms (v4.0.5) **1.5ms (WebGL)**	--
Windows XP	33ms (v4.1)	863ms (v3.0) **1.1ms (WebGL)**	105ms (v4.0.5)	--
iPad	--	--	801ms	--
iTouch v3.1.2	--	--	1904ms	--
Nexus One	--	--	--	340ms

Desktop and iPad tests done with a 600*400 pixels view, iTouch with 320*400 pixels, and Nexus One with 600*400 pixels.

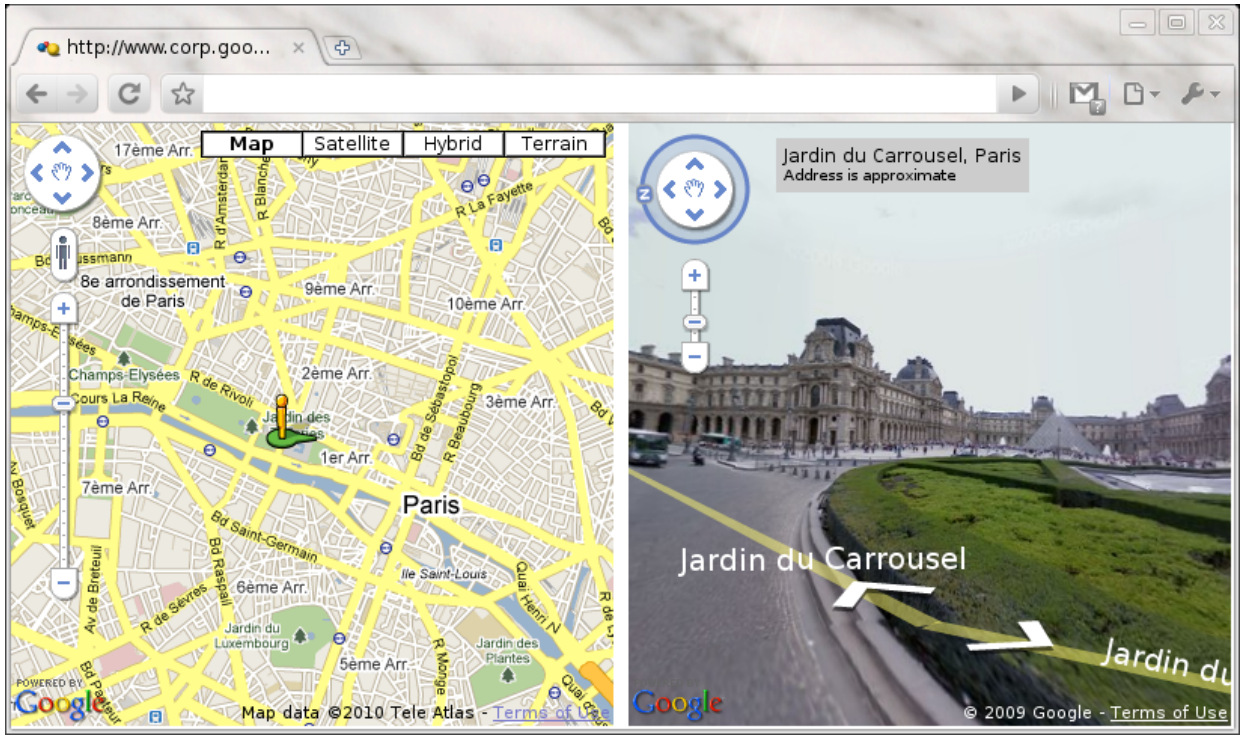
Linux Chrome: 400*300: 20ms, 600*400: 25ms, 800*600: 52ms, 1200*800: 79ms.

Technology - What to use

	Technology	Browsers
Polygons	SVG / VML	Not Android
Animations	CSS Transforms	Not IE
Image Transforms	Canvas	Not IE
3D Transforms	WebGL	Coming soon, not IE

Technology - Graphics Demo

http://code.google.com/apis/maps/documentation/javascript/examples/streetview-simple.html



Google I/O 10

Technology - Graphics beware

- Canvas 2D performance drops for large canvas or large number of Javascript calls.
- iPhone 3D Transform bug with touch events (version 3.1)
- Android Canvas bug with images until Froyo.
- WebGL only supported in nightly builds
- Bug with Embed objects and CSS transform.

Technology - Geolocation

- Uses W3C on Mobiles and Firefox 3.6+
- Falls back to Google Gears if installed
- Used in the API by the Places API
- Not exposed directly as an API

```
var geolocation;
 if (navigator && navigator.geolocation) {
   geolocation = navigator.geolocation;
 } else {
   var factory = initGears();
   if (factory) {
     geolocation = factory.create('beta.geolocation');
   }
 }
 if (geolocation) {
   try {
     geolocation.getCurrentPosition(function(position) { ... });
   } catch (err) {
     ...
  }
 } else {
   ...
 }
```

Technology - Mouse and Touch events

- iPhone, iTouch, iPad and Android so far.
- S60, Palm OS and other mobiles making progress.
- Multi-touch and gestures, Apple only.
- Mouse events and touch events.

iPhone events sequence:

- drag:
  touchstart,
  touchmove*,
  touchend.

- click:
  touchstart,
  touchend,
  mousemove,
  mousedown,
  mouseup,
  click.

- pinch:
  touchstart,
  touchmove*,
  gesturestart + touchstart,
  (touchmove + gesturechange)*,
  gestureend + touchend,
  touchmove*,
  touchend

Technology - Mouse and Touch events

	drag	pinch	click
iPhone/iTouch	✓	✓	✓
iPad	✓	✓	✓
Android 2.x	✓	–	✓
Samsung Wave	–	–	✓
Palm OS	–	–	✓
Nokia X6	*	–	✓

*: Simulated hover mouse events with mousedown/mouseup/click triggered together on finger up.

Content

- Latency
- Architecture
- Technology
- Debugging

Debugging

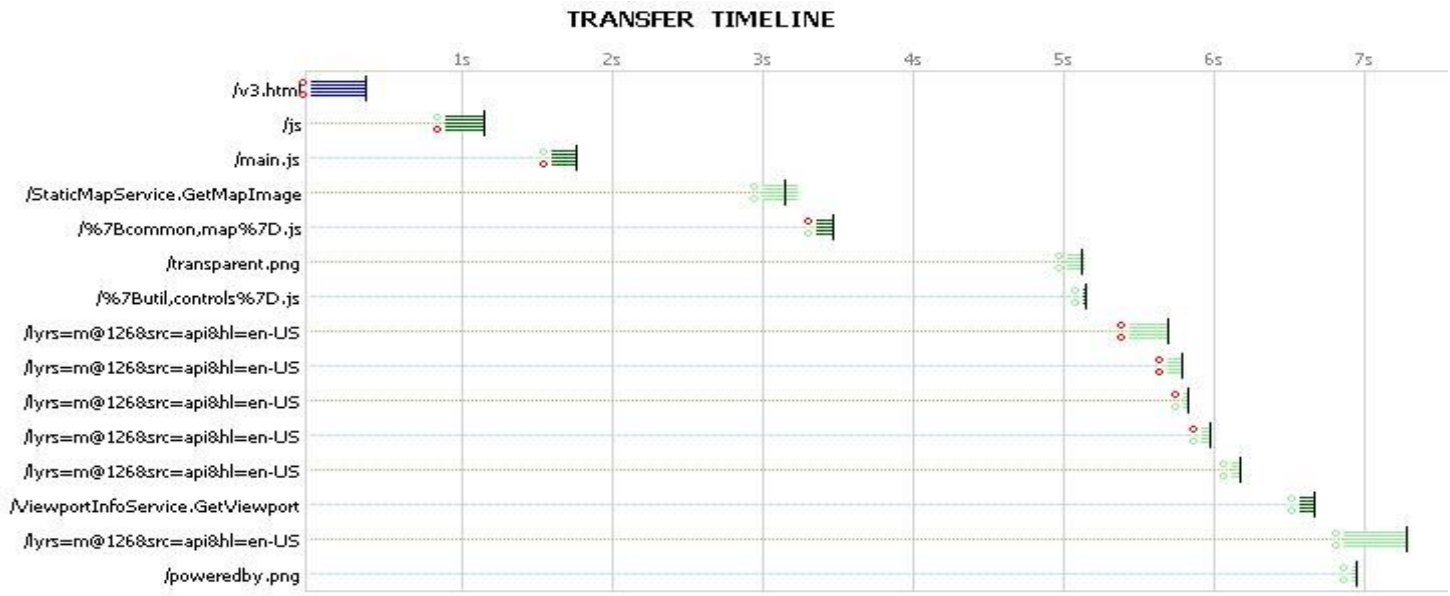
- Debugging
  - Desktops
  - iPhone
  - Android

Debugging desktops

- Firefox: Firebug
- Internet Explorer: Visual Debugger, Dev toolbar
- Chrome: Developer Console, SpeedTracer
- Safari: Web Inspector

Debugging iPhone

- Console
  - Go to Settings/Safari/Developer
  - Set Debug Console ON
  - Use window.console.log to add trace messages.
- HTTP proxy
  - Available for 3G connection only, no Wifi.
  - Use the iPhone Configuration Utility.



Debugging Android

- adb logcat
  - Download Android SDK
  - Run setup and download USB drivers
  - Set Android device to accept USB debugging
  - Use adb tool.
- HTTP proxy
  - 3G only, no Wifi
  - Setup proxy in Network APNs
  - Note: Fails on current Android release 2.1, fix coming soon.

Google Developer Qualification

Chrome Extensions

Gadgets

Search

App Engine

JS Maps API

KML

3D

Google 10 I/O

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

http://bit.ly/cnQiok


Google™
10