6 Google APIs for Don Isidro Parodi

Let's help Don Isidro build a better web site using 6 Google APIs

Google Code

Google Maps

Google Calendar BETA

Google Earth

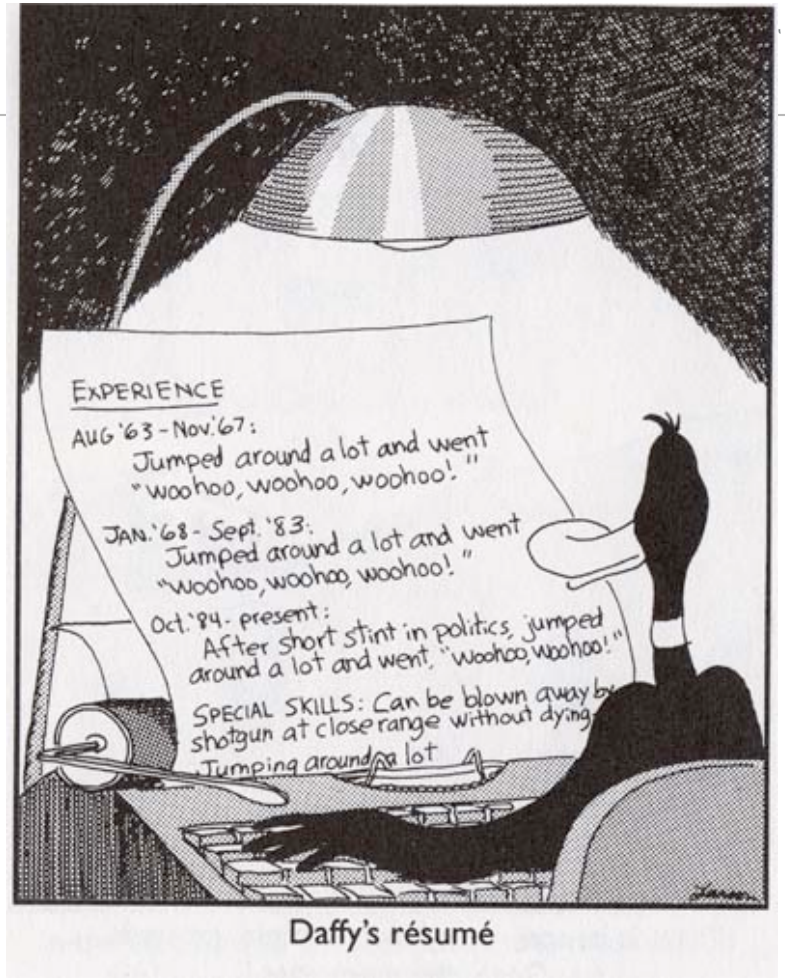
Google Checkout

Patrick Chanezon, Google, chanezon@google.com
Checkout API Evangelist

May 15th 2007

P@ in a slide

- Yo no soy Porteño:-) Paris - San Francisco
- API Evangelist - Google Checkout
- Software plumber
- Family man: married, 3 kids
- Java geek… in scripting rehab: Ruby, JavaScript, PHP, Python
- Open Source: ROME, AdWords (java, C#, ruby)
- Sun: Blogs, Portals, eCommerce
- Netscape/AOL: LDAP, Calendar, App Servers, CMS, MyNetscape (RSS)
- More on my blog http://wordpress.chanezon.com



Daffy's résumé

- Links and slides tomorrow at http://del.icio.us/chanezon/ba2007

Agenda

Google

Gracias Globant

Borges, Casares, Don Isidro and Web Services

API: Platforms and Bridges

API technologies: SOAP, REST and Ajax

What: Google Ajax Search API

Where (2D): Google Maps API (Ajax)

Where (3D): Google Earth and KML

When: Google Calendar Data API (REST)

Commerce: Google Checkout API (REST)

How: Google Web Toolkit (Ajax)

Gracias Globant

Globant

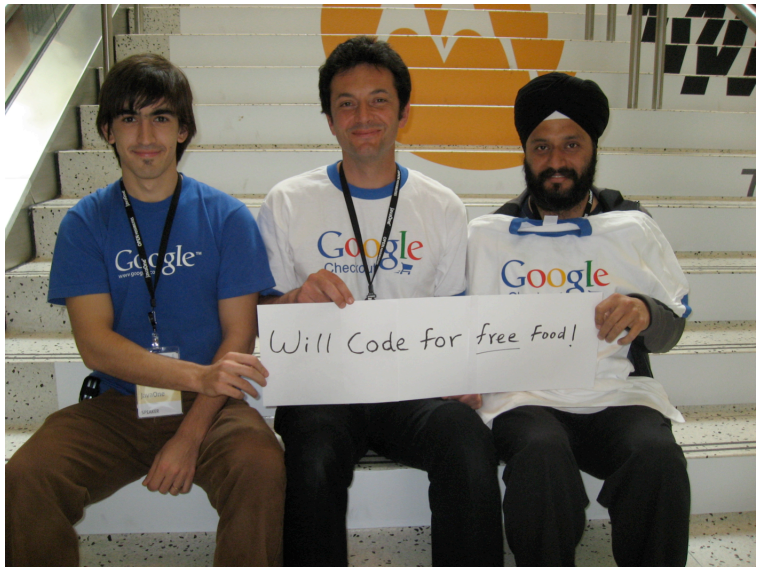
Google

First company used by Google for IT outsourcing

Work on several projects

I work with them since november 2005 on Checkout API Testing and open source libraries

Ignacio Blanco presented with us at JavaOne last week: new Mendoza open source project



Argentina for most french guys

Maradona, World Cup 1986

Dos Grandes… Jorge Luis & Adolfo Bioy

Borges is the most Googley of all writers

"the librarian deduced that the Library is "total"--Perfect, complete, and whole--and that its bookshelves contain all possible combinations of the twenty-two orthographic symbols (a number which, though uninimaginably vast, is not infinite)--that is, all that is able to be expressed, in every language."

The Library of Babel, Ficciones

25 symbols (22 plus space, comma, symbol)

410 pages (+2 for the covers) x 40 lines x 80 letters = 1 318 400

25 ^ 1 318 400 > 10^100 (1 Googol)

Google Books: only the ones that have been written:-)

Morell's invention by Casares prefigures Google Earth and 2nd Life

Don Isidro as a metaphor for Web Services

Google

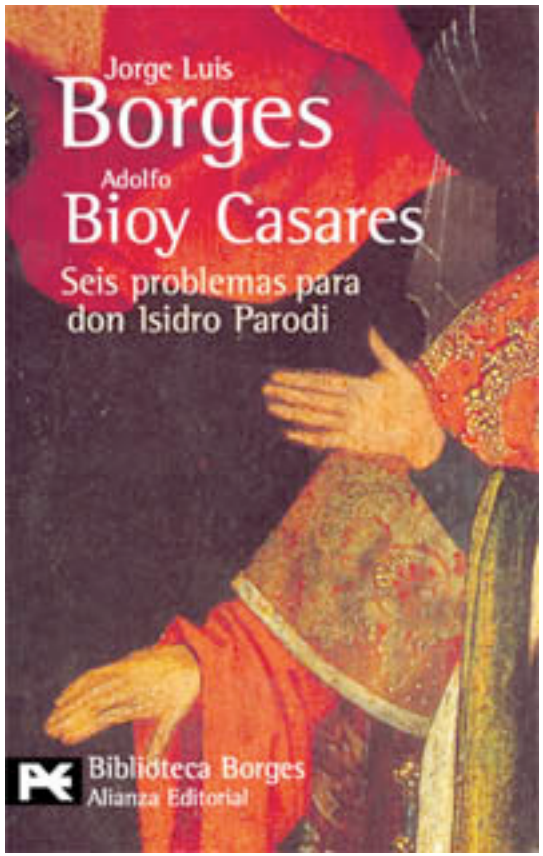
Detective, rational mind, knows a lot, infers the rest

Algorithms

People look for his advice on issues

Visit him in prison

Get answers, can get going with their lives



Google APIs

Why are we doing do it?

Our mission: “Organize the world’s information and make it universally accessible and useful”

- We can’t organize it all ourselves

32 Google APIs on http://code.google.com… and counting!

- Checkout API
- Data API
  - 8 services
- Maps API
- AdWords API
- Ajax Search API
- Ajax Feed API
- Desktop SDK
- Enterprise APIs
- Homepage API
- YouTube API
- Sitemaps
- Talk (XMPP)
- Toolbar API
- Google Web Toolkit
- …

API Metaphor: Platforms

Google

One stop shop, soup to nuts

API Metaphor: Platforms

Google

Islands of data

API Metaphor: Platforms

Google

Build tall structures on top

API Metaphor Bridges

Google

User - Search results (Ajax Search)

Ajax

User - Maps (Maps)

User - Data (Gdata)

REST

Buyers - Sellers (Checkout)

Advertiser Ads (AdWords)

SOAP

Publishers - Ads (AdSense)

Build your own bridge (Google Web Toolkit)

API Metaphor Bridges

User - Search results (Ajax Search)

User - Maps (Maps)

Ajax

User - Data (Gdata)

Buyers - Sellers (Checkout)

REST

Advertiser Ads (AdWords)

Publishers - Ads (AdSense)

SOAP

Build your own bridge (Google Web Toolkit)

API Metaphor Bridges

User - Search results (Ajax Search)

User - Maps (Maps)

Ajax

User - Data (Gdata)

Buyers - Sellers (Checkout)

REST

Advertiser Ads (AdWords)

Publishers - Ads (AdSense)

SOAP

Build your own bridge (Google Web Toolkit)

API Metaphor Bridges

Google

User - Search results (Ajax Search)

User - Maps (Maps)

Ajax

User - Data (Gdata)

Buyers - Sellers (Checkout)

REST

Advertiser Ads (AdWords)

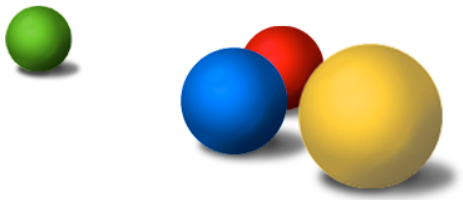
Publishers - Ads (AdSense)

SOAP

Build your own bridge (Google Web Toolkit)

The Google AJAX Search API

Mark Lucovsky
Technical Director, Engineering
Google Inc.
July 2006



Google AJAX Search API

Easy way to add Google search to your page

Parallel search over Web, Local Listings, Google Video, and Blogs

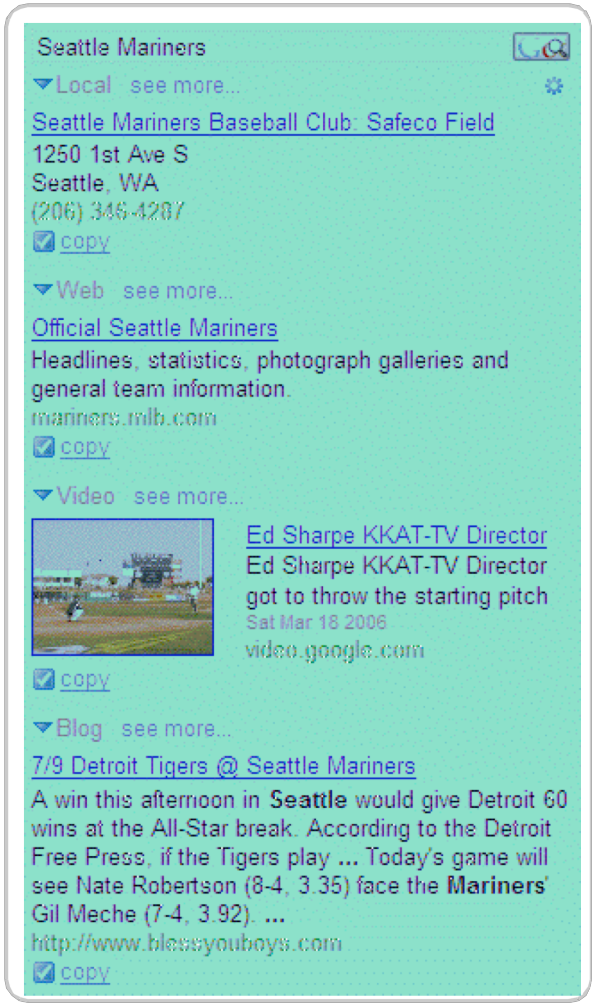
Supports “Clipping” of Search Results

Buzzword Soup Compliant

- AJAX
- JSON
- HTML Microformats…
- Free

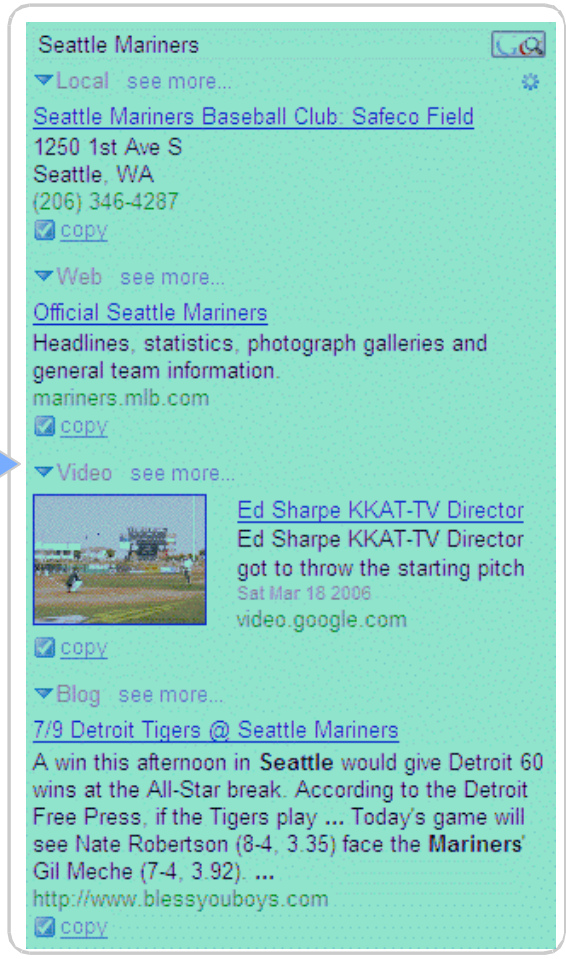
Sample to right is:

- ~9 Lines of JavaScript
- ~10 Lines of HTML

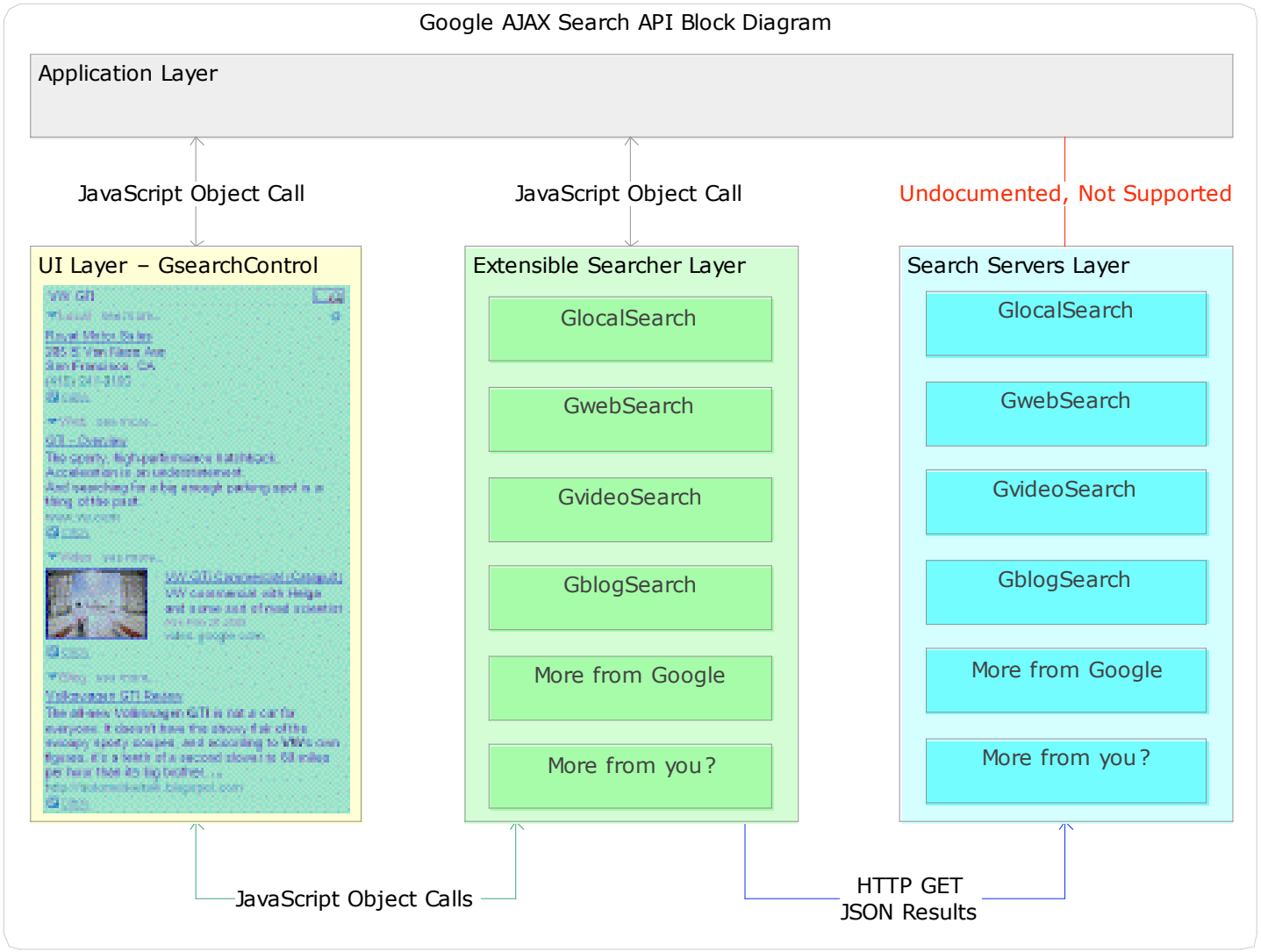


“Hello World”

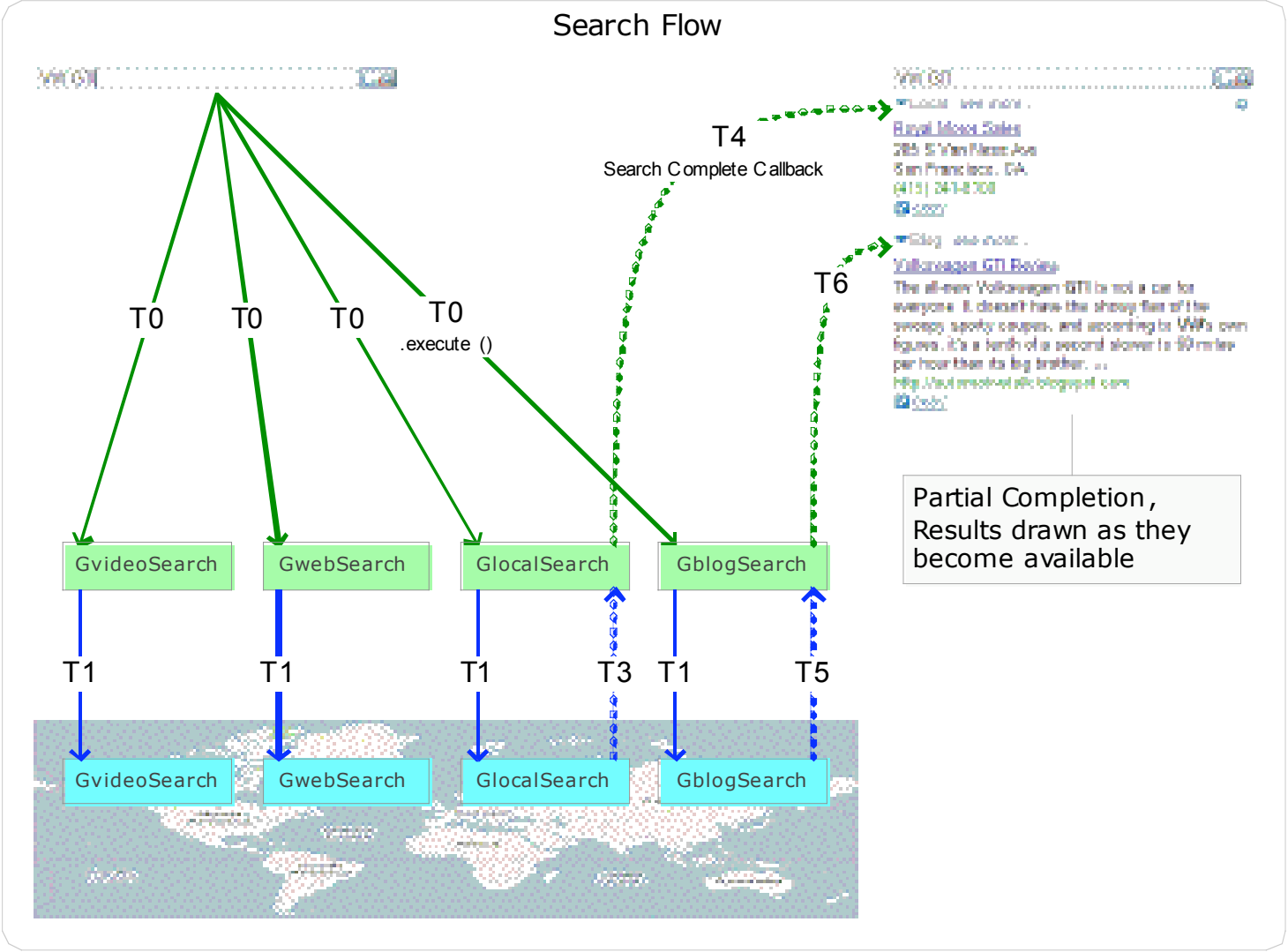
```
function OnLoad() {
  var sc = new GSearchControl();
  sc.addSearcher(new GlocalSearch());
  sc.addSearcher(new GwebSearch());
  sc.addSearcher(new GvideoSearch());
  sc.addSearcher(new GblogSearch());

  sc.draw(searchContainer);
  sc.execute(“Seattle Mariners”);
}

<body onload=“OnLoad()”>
 <div id=“searchContainer”/>
</body>
```



Architecture Block Diagram



Execution Timeline

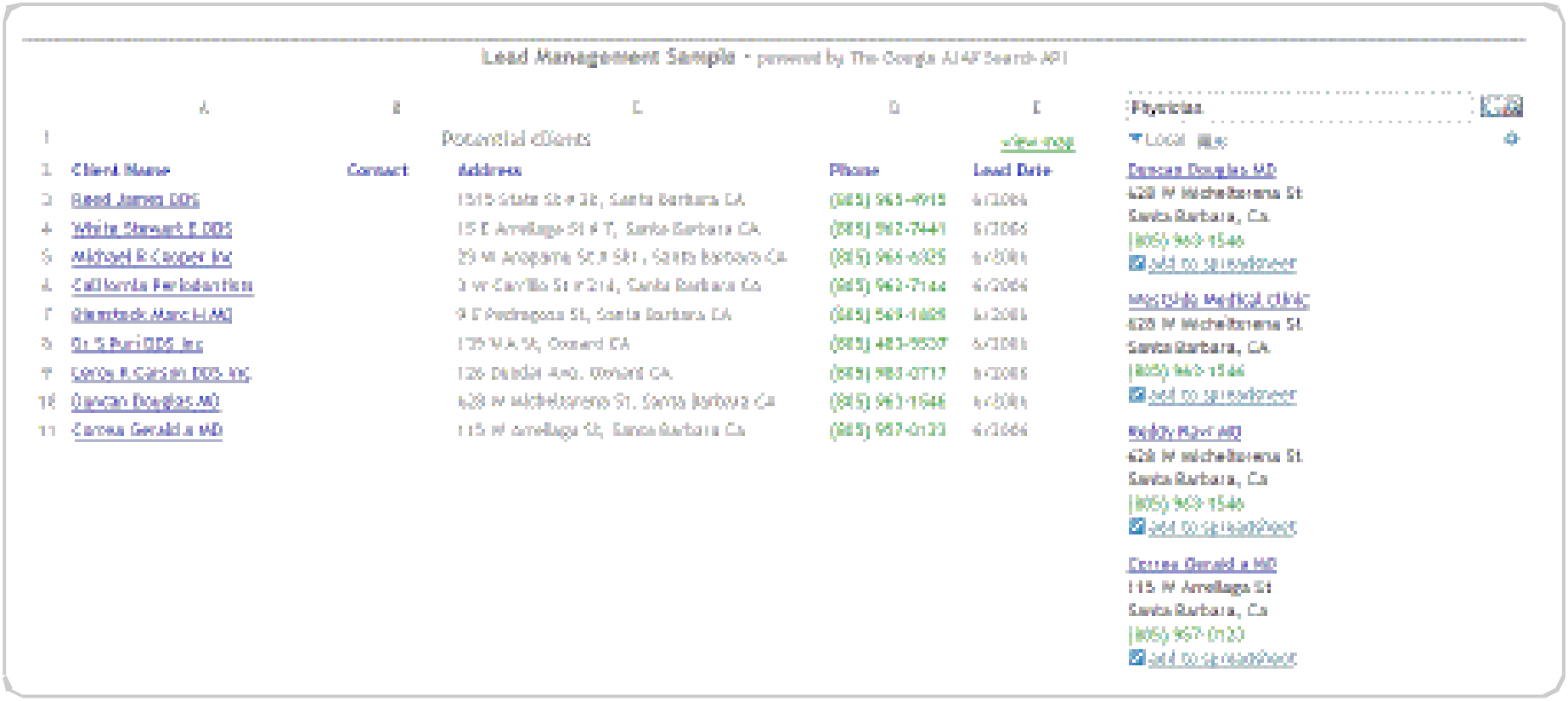


“Keep Handling” – How do I do it?

Google

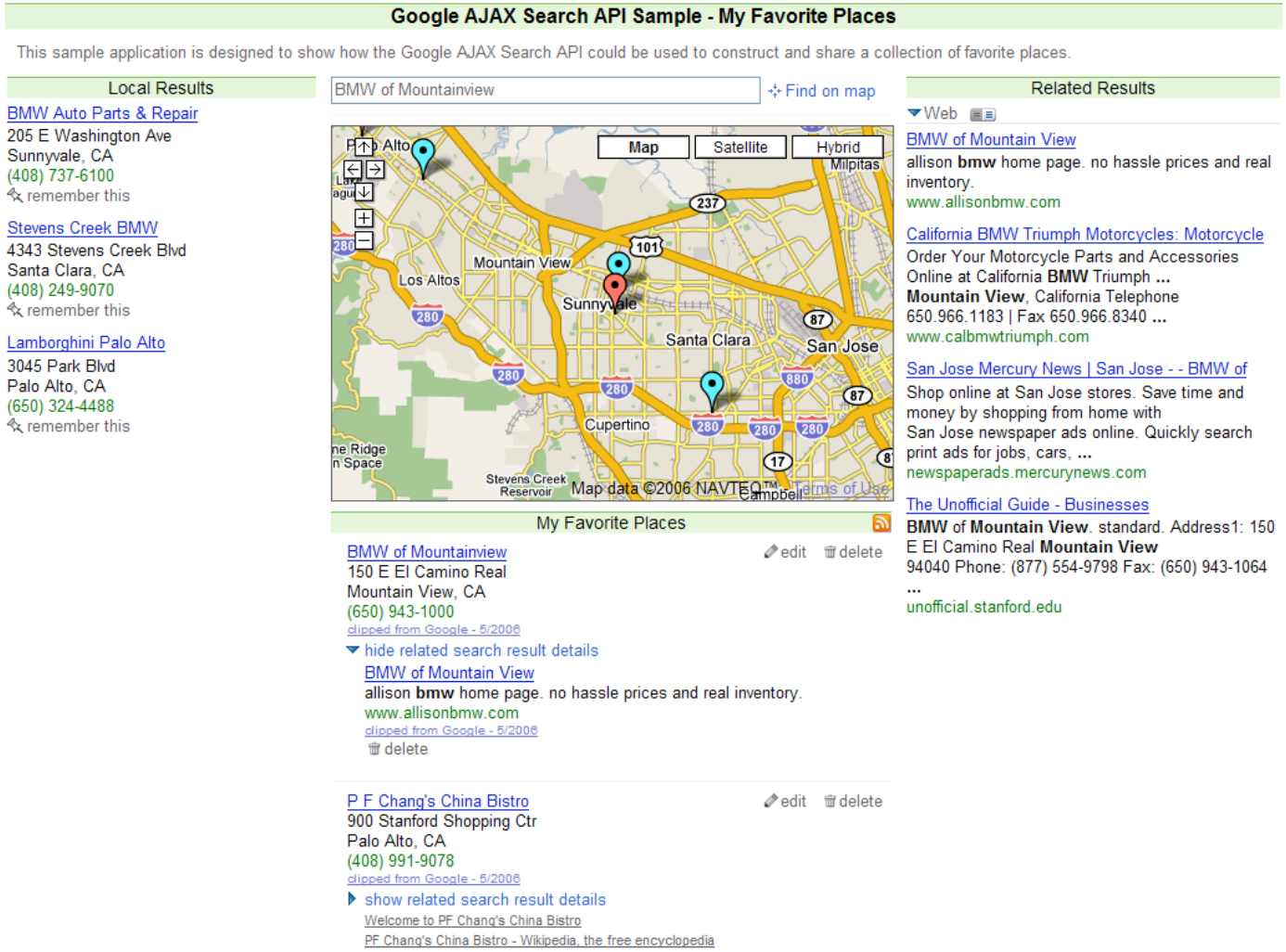
```
function OnLoad() {
  var sc = new GSearchControl();
  …
  // establish keep handler
  sc.setOnKeepCallback(this, MyKeepHandler);
  …
}

function MyKeepHandler(result){
  // clone the .html property
  var node = result.html.cloneNode(true);

  // append into my document
  savedResults = document.getElementById(“saveArea”);
  savedResults.appendChild(node);
}
```

Building a “Lead List” using Search



http://www.google.com/uds/samples/random/lead.html

Third Party App, Favorite Places List



http://www.google.com/uds/samples/places/index.html

More Information

Google Code

- http://code.google.com/

AJAX Search API Documentation and Samples

- http://code.google.com/apis/ajaxsearch/
- http://code.google.com/apis/ajaxsearch/samples.html

Search API Blog

- http://googleajaxsearchapi.blogspot.com/

Search API Developer Forum

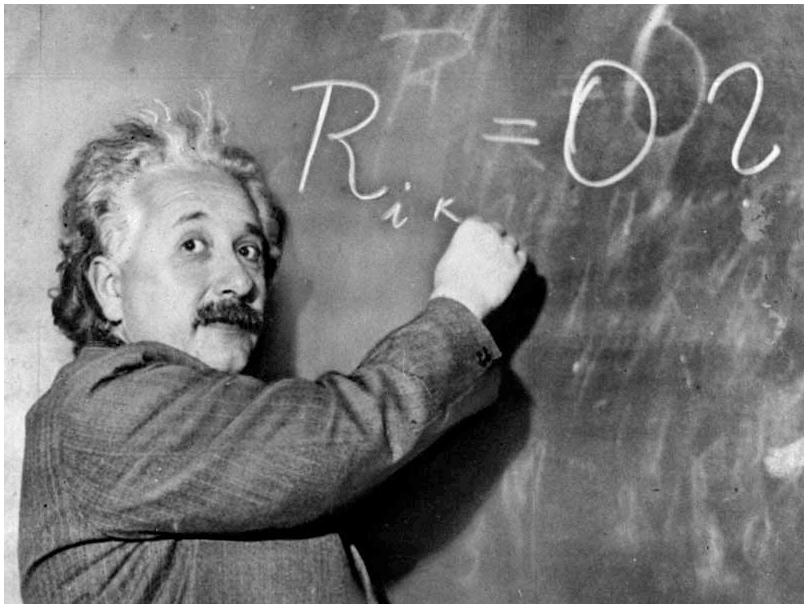
- http://groups.google.com/group/Google-AJAX-Search-API

Where and When?

Google

Space and Time

- 2 important dimensions to organize and present information



"Space and time are not conditions in which we live; they are simply modes in which we think."

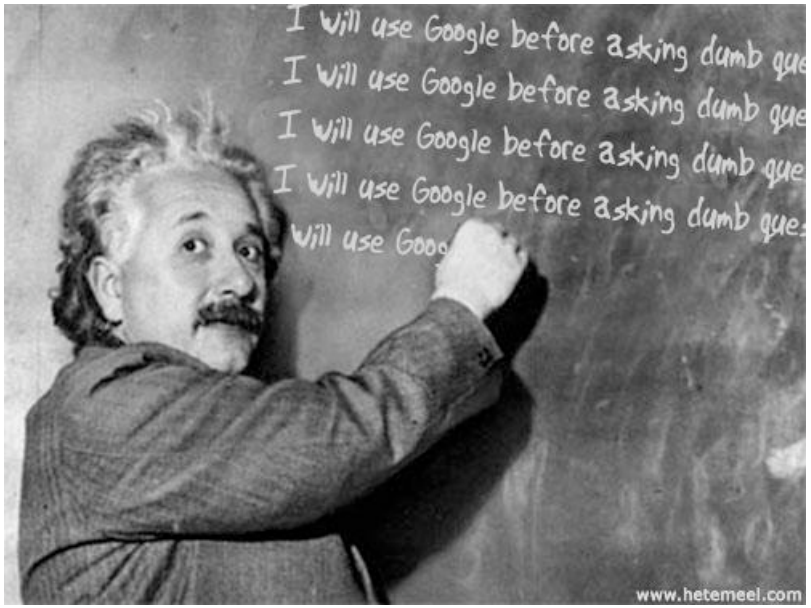
Albert Einstein

How?

Google

This is fine and dandy but how do I use that in my webapp?

- APIs = Plumbing



"If I had my life to live over again, I'd be a plumber."

Albert Einstein

Where and When Google style

Google

Google Maps API

- Triggered the Ajax revolution
- Now everybody's doing it

Google Maps API

- HTML, JavaScript, (XML)

Google Calendar

- An Ajaxy Calendar

Calendar APIs

- REST is the best: HTTP verbs for the rest of us
- Atom syntax: GET
- Atom Publishing Protocol: POST, PUT, DELETE
- Calendar is the first Google Data API

What's in a name?

Google

AJAX

- Asynchronous
- JavaScript
- XML

… but

- Only the second part (JavaScript) is really used in Google Maps
- All are completely independent
- All existed in browsers for years before Suggest, Gmail, and Maps
- XML? A data format decision that is completely unimportant relative to the other two
- But a bad name is better than no name!
- Ajax is a great Meme: thanks Jesse James Garrett!

What makes an AJAX Application?

Google

Classic web application

- User clicks on a link / submits a form
- HTTP request
- HTTP response **replaces** the document

AJAX web appplication

- User triggers an event (mouse click, keyboard click, etc)
- Event invokes scripted event handler
- Event handler *may* initiate a data transfer
- Event handler or data transfer callback **updates** the document

Consequences of AJAX-style development

Google

Sophisticated user interaction

- Display can be partially updated, modified, or animated
- Complex manipulations of page state and UI are possible

Client-side session state

- Transient session state stored on the client
- Persistent user state stored on the server
- *Enables a much more natural architecture for applications*

Technical Components

Google

DOM – Document Object Model

- API for structured text

CSS – Cascading Style Sheets

- Defines visual layout properties, etc.

JavaScript

- Flexible scripting language that is both powerful and fragile

HTTP

- Transport for background data transfer
- Typically done with IFRAME elements or XMLHttpRequest

Transfer Data Format

- JavaScript object literals (JSON) or XML

Illustration: XML vs. JSON – Text Format

Google

XML

```
<data>
<location lon="13.4156" lat="52.5206"/>
</data>
```

```
var data = {
      location: {
            lon: 13.4156,
            lat: 52.5206
      }
};
```

Illustration: XML vs. JSON – API

Google

XML (DOM)

```
var lon =
datanode.firstChild.getAttribute('lon');
```

XML (E4X)

```
var lon = data.location.@lon;
```

JSON

```
var lon = data.location.lon;
```

But E4X only in Firefox and Flash

Standards based technologies? Sort of.

Google

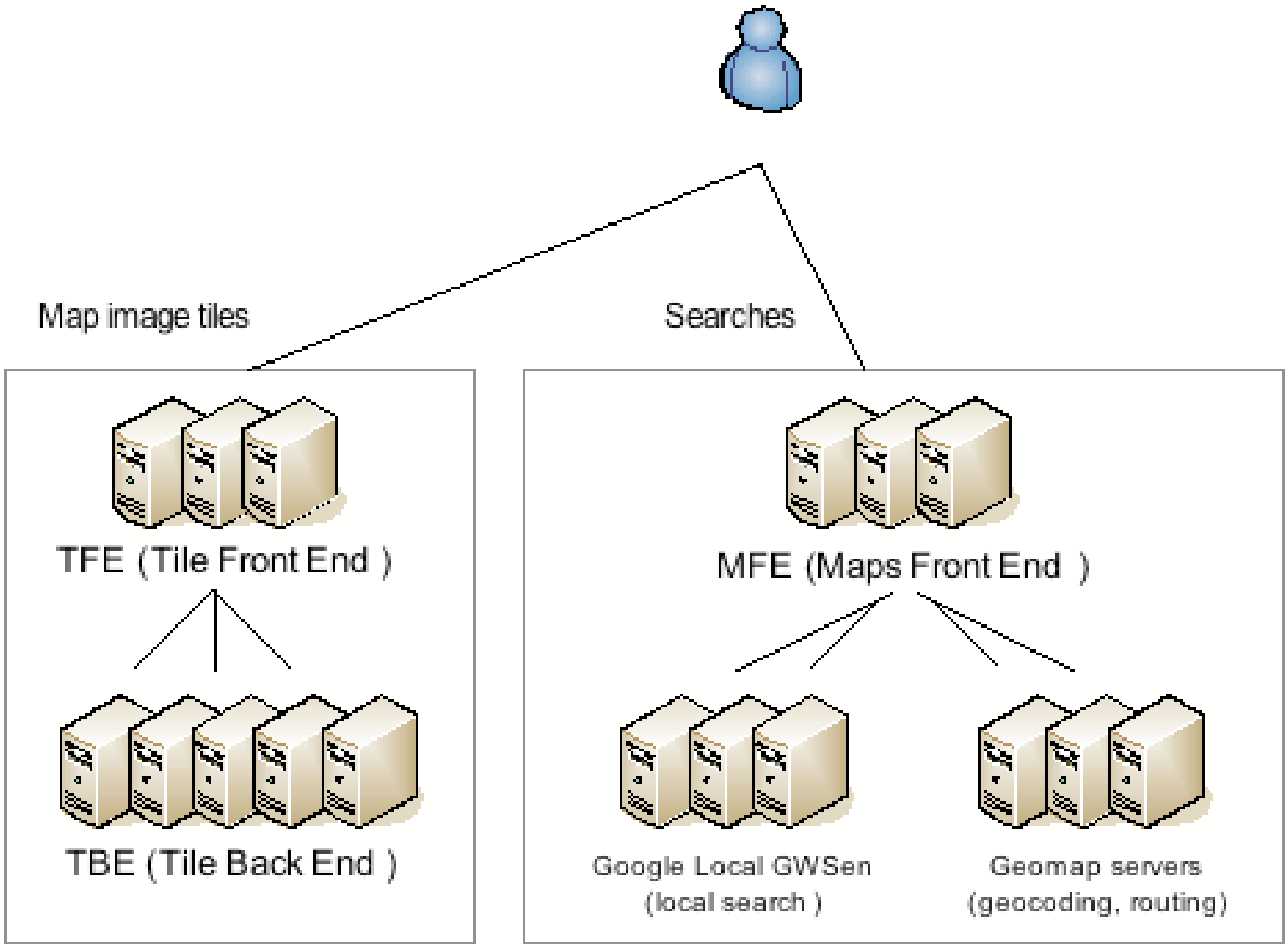
AJAX is a collection of successful technologies being used for things they were never intended for… like the web itself

- The HTML Document Object Model (DOM) was designed for textual documents, not rich graphical interfaces
- XMLHttpRequest was introduced to Internet Explorer by the Microsoft Outlook team to solve a specific problem for Outlook Web Access

Each component has several implementation differences across browsers and platforms

- AJAX Developer is typically synonymous with PhD in Browser Quirks
- See http://www.quirksmode.org/

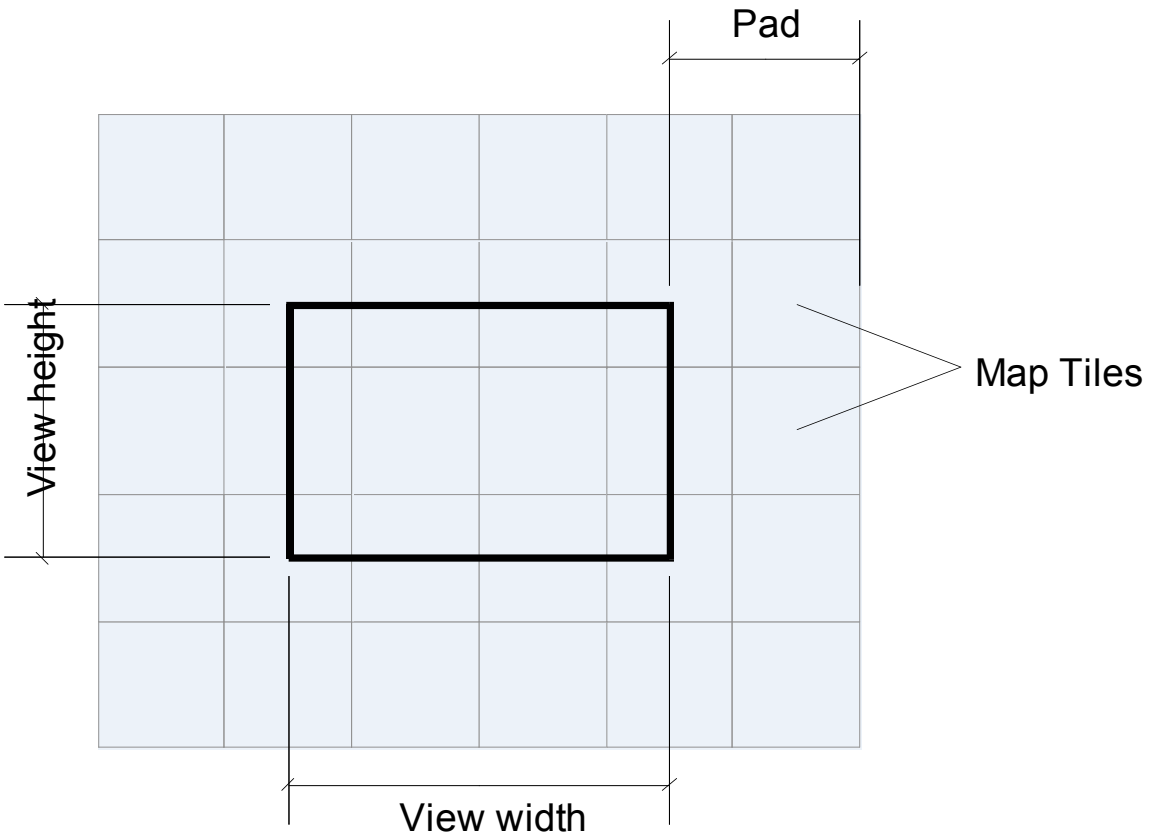
So how does Maps do work?



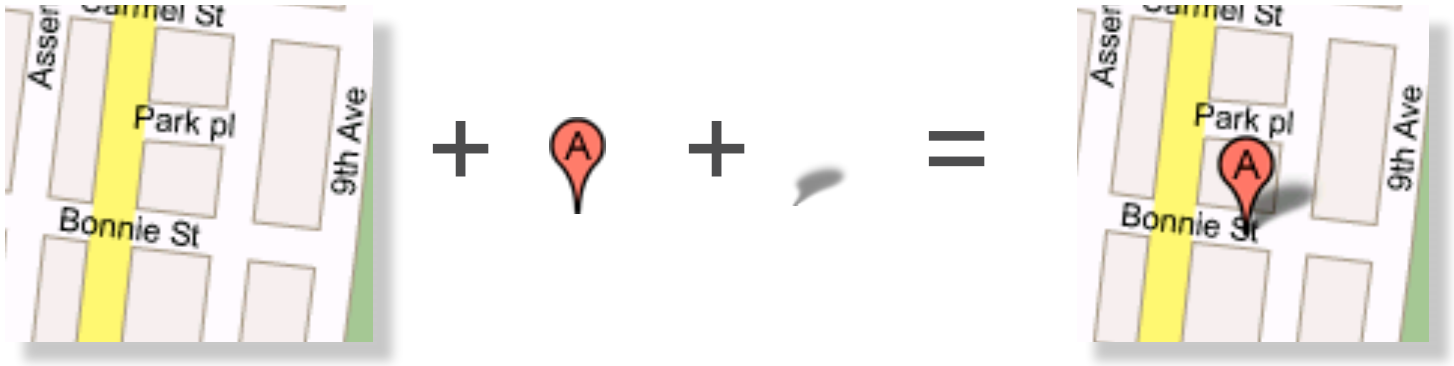
Maps JavaScript architecture

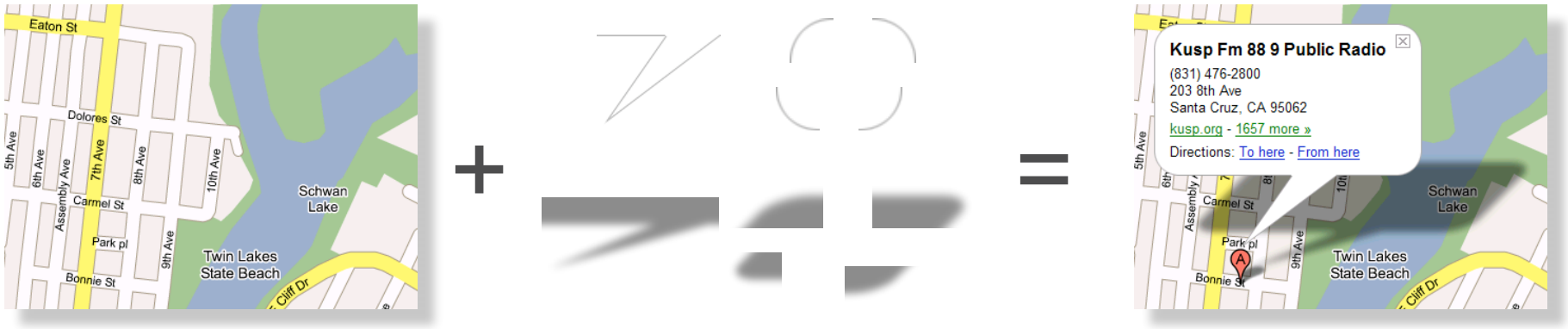
Google

CSS positioned map tiles with a cropped at the map border



Maps JavaScript architecture





Google Maps API

Google

What is it?

- Put Google Maps on your site for free, just like Frappr.com, housingmaps.com and chicagocrime.org
- Overlay custom content, write custom functionality
- Free to all web sites that are free to consumers (including commercial web sites)
- V1 in june 2005, V2 in April 2006

What you can do with the Google Maps API

Google™

Switch between maps

- "Map", "Satellite" and "Hybrid" modes supported by default

Overlays

- Markers
  - Custom icons, shapes, etc
- Polylines
  - e.g., driving directions lines
- Custom overlays

"Info window", tabbed

- You know, that cool window with the shadow underneath it

Customizability

- Custom map controls, overlays, and map types

Overview map

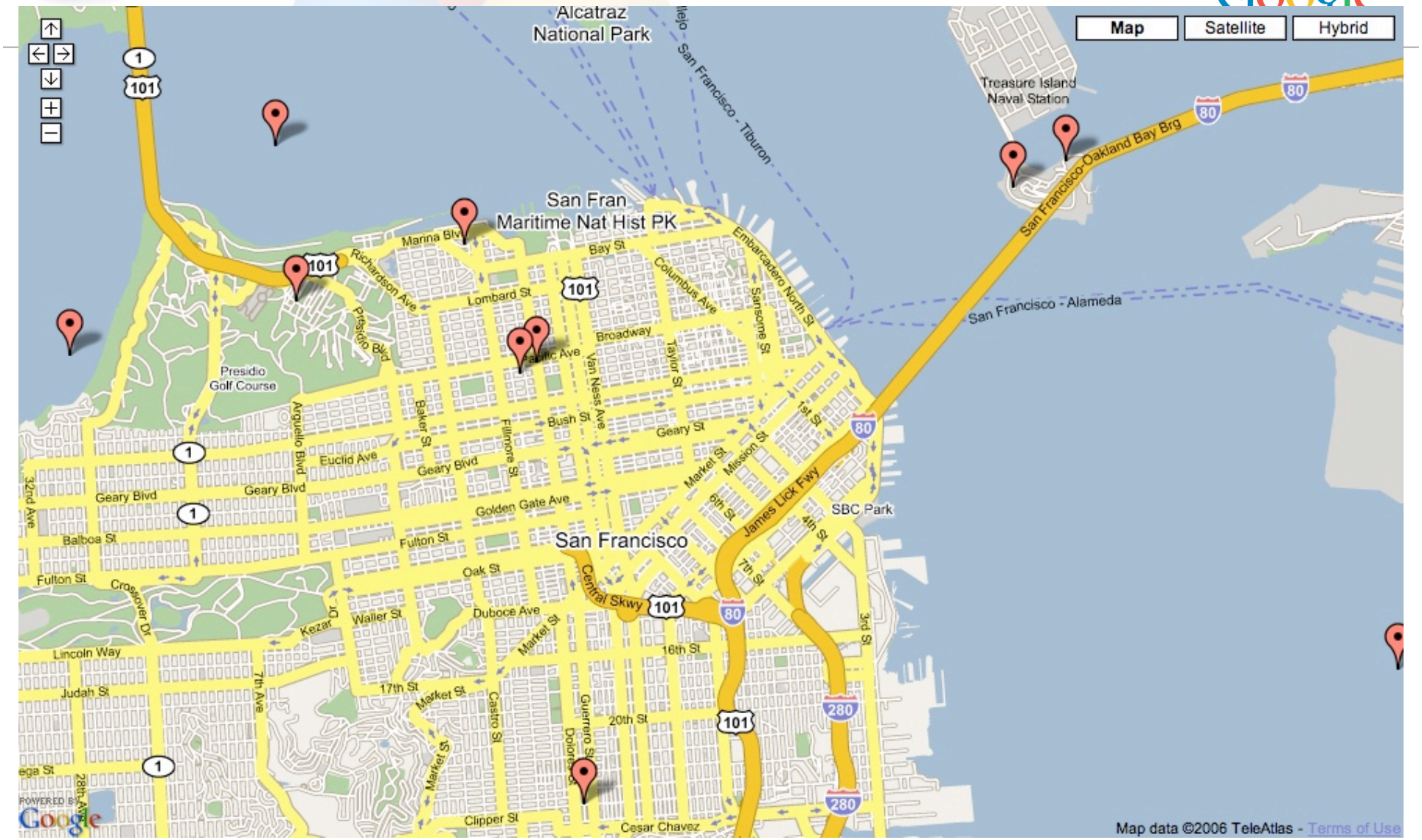
- New GOverviewMapControl control displays an attractive and collapsible map in the corner of the screen

Support for a subset of KML (Google Earth data format)

Emphasis on Simplicity

Google

```
// Center the map on Westin Hotel in San Francisco
var map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.setCenter(new GLatLng(37.787742,-122.408295), 13);
// Add 10 markers to the map at random locations
var bounds = map.getBounds();
var southWest = bounds.getSouthWest();
var northEast = bounds.getNorthEast();
var lngSpan = northEast.lng() - southWest.lng();
var latSpan = northEast.lat() - southWest.lat();
for (var i = 0; i < 10; i++) {
        var point = new GLatLng(southWest.lat() + latSpan * Math.random(),
        southWest.lng() + lngSpan * Math.random());
        map.addOverlay(new GMarker(point));
}
```


Google
Map
Satellite
Hybrid
Alcatraz
National Park
San Francisco - Tiburon
Treasure Island
Naval Station
San Francisco-Oakland Bay Brg
San Fran
Maritime Nat Hist PK
San Francisco - Alameda
Presidio
Golf Course
San Francisco
SBC Park
Map data ©2006 TeleAtlas - Terms of Use

A few cool Google Maps applications

Google

Useful/Cool

- http://del.icio.us/chanezon/ajaxian+google+maps+app+cool

Technically interesting

- http://del.icio.us/chanezon/ajaxian+google+maps+app+tech

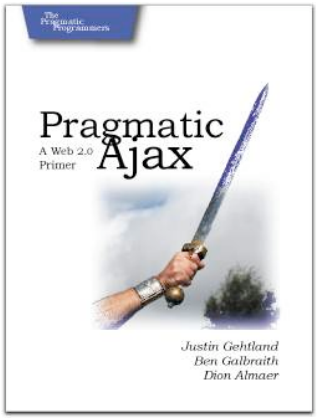
Maps API Web Resources

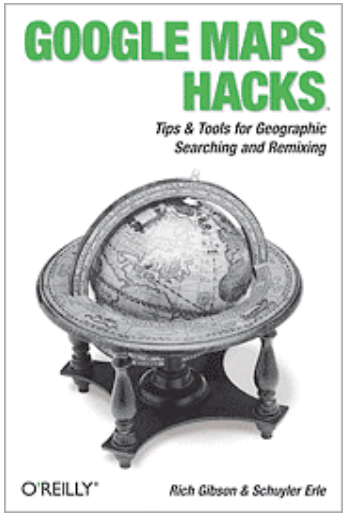
Google

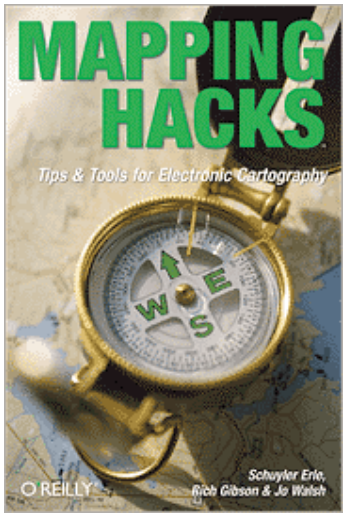
- http://code.google.com/
- http://www.google.com/apis/maps
- Google-Maps-API Developer Forum: http://groups.google.com/group/Google-Maps-API?lnk=li
- Maps API blog: http://googlemapsapi.blogspot.com/
- Mike William's tutorials: http://www.econym.demon.co.uk/googlemaps/
- http://del.icio.us/chanezon/google+maps

Maps API Books

- The Ajaxians book: Pragmatic Ajax
  - implement your own Map UI in a day… then you have to do the server side:-)
- The O'Reilly ones:
  - Google Maps Hacks
  - Mapping Hacks







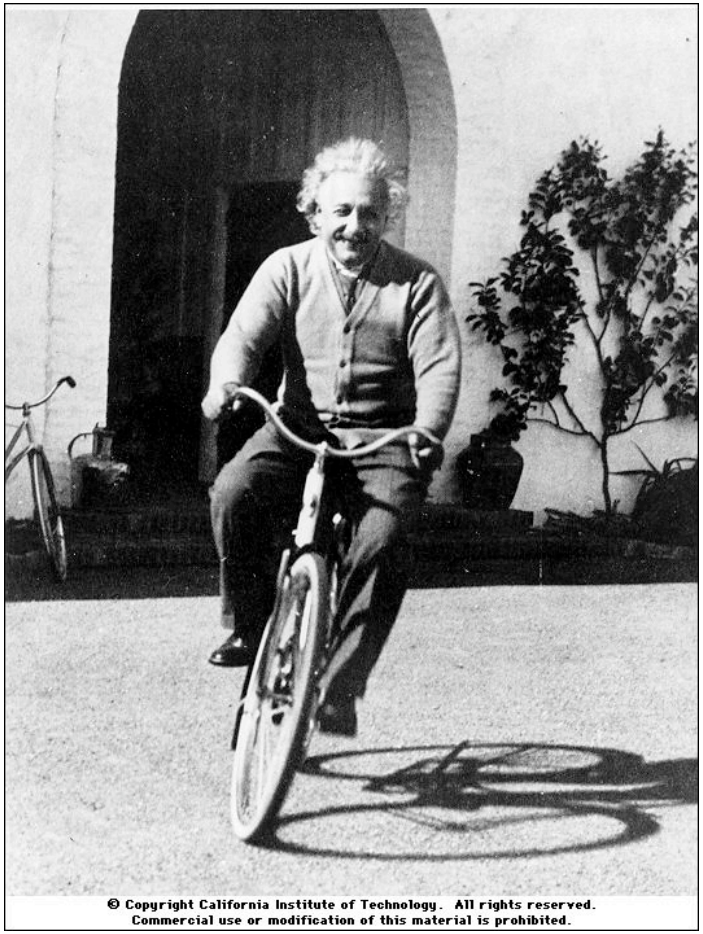
Google Earth: KML

Google

- The Google Earth team don’t do 2D slides
- Let’s do the presentation in KML:-)

Google Data API

REST API, Based on Atom and the Atom Publishing Protocol



"The release of atom power has changed everything except our way of thinking…"

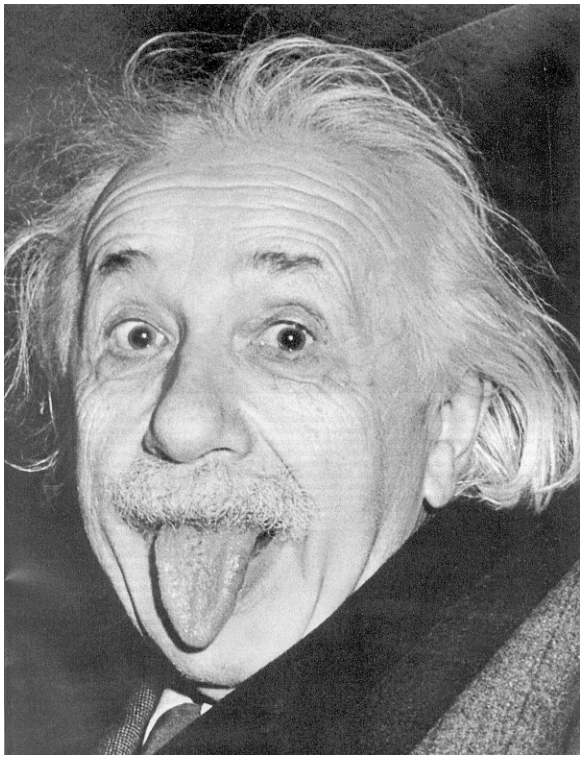
Albert Einstein

First Google Data API: Calendar

Google

GET, POST, PUT, DELETE Calendar Entries

Released in April 2006



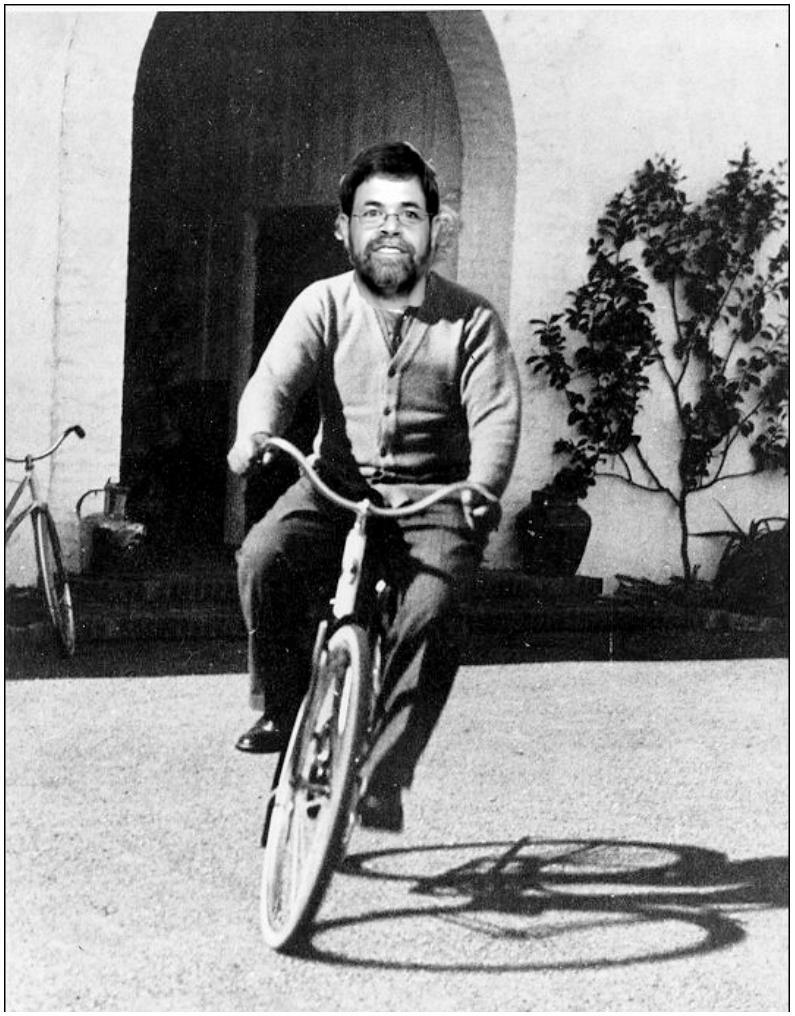
"...the solution to this problem lies in the heart of mankind. If only I had known, I should have become a watchmaker."

Albert Einstein

Google Data API

frank mantek (fmantek@google.com)

quotes...



Adam Bosworth on a bike

"'Google Data APIs Protocol' is a horrible name. You can tell that ex-Microsoft employees had a hand in this effort. :)"

Dare Obasanjo (Microsoft)

more quotes...

Google

"The benefit is that you'd have a single API that could be used to query, update, and index structured data on the web--anywhere on the web. It's a pretty powerful vision and something I didn't expect to see for a couple more years" - **Jeremy Zawodny (Yahoo)**

Data plumbing technologies

Google

ODBC

OLEDB

(P@: In my case JDBC, EJB and Hibernate… but same feeling)

Remoting technologies

XML

SOAP, WSDL

WS_* specs to your hearts content...

Google Requirements for a data API

Google

- lot of APIs available
- some are SOAP based (AdWords API)
- some are somewhat REST based
- ... all are somehow different

What do we need?

- simple data protocol
- query? Sure, i just love queries...
- updates - there is no point getting data if you can not mess with it....

simple data protocol

Google

REST style of design...

- data format should be easy to understand
- easy to consume by existing tools

And the winner is…

- the Syndication format (Atom 1.0 and RSS 2.0).

in the simple case

Google

I just enter a URI, like:

- http://www.google.com/calendar/user/public/full
- and get an Atom feed of my calendar data

the result

```
<feed>
    <id>
http://www.google.com/calendar/feeds/user/public/basic
</id>
<updated>2006-05-05T14:22:41.000Z</updated>
<title type="text">Frank Mantek</title>
<subtitle type="text">Frank Mantek</subtitle>
<link rel="http://schemas.google.com/g/2005#feed" type="application/atom+xml"
href="http://www.google.com/calendar/feeds/fmantek%40gmail.com/public/basic"/>
<link rel="self" type="application/atom+xml"
href="http://www.google.com/calendar/feeds/fmantek%40gmail.com/public/basic?max-results=25"/>
<link rel="next" type="application/atom+xml"
href="http://www.google.com/calendar/feeds/fmantek%40gmail.com/public/basic?start-index=26&max-results=25"/>
    <author>
<name>Frank Mantek</name>
<email>fmantek@gmail.com</email>
</author>
<generator version="1.0" uri="http://www.google.com/calendar">Google Calendar</generator>
<openSearch:itemsPerPage>25</openSearch:itemsPerPage>
</feed>
```

the result, again...

Google

data comes in standard Atom or RSS format

some namespaced extensions are used to identify semantic entities and data items not mappable in Atom and RSS

extensions

Google

google extensions are in their own namespace

- xmlns:gd=http://schemas.google.com/g/2005

we have **types**

and there are **kinds**

- semantic grouping of types

so, what about types?

Google

types can appear anywhere in the document

they do not have semantic meaning by themselves

e.g. a <who> element can appear in any document, but that does not make it a contact

Google kinds...

The Calendar defines

- Contacts
- Events
- Messages

example: an event

```
<entry xmlns:gd="http://schemas.google.com/g/2005">
  <category scheme="http://schemas.google.com/g/2005#kind" term="http://schemas.google.com/g/2005#event"/>
  <id>http://mycal.example.com/feeds/jo/home/full/e1a2af06df8a563edf9d32ec9fd61e03f7f3b67b</id>
  <published>2005-01-18T21:00:00Z</published>
  <updated>2006-01-01T00:00:00Z</updated>
  <title>Discuss BazMat API</title>
  <content>We will discuss integrating GData with BazMat.</content>
  <author>
    <name>Jo March</name>
    <email>jo@example.com</email>
  </author>
  <gd:when startTime='2005-01-18T21:00:00Z' endTime='2005-01-18T22:00:00Z'>
    <gd:reminder minutes='15'/>
  </gd:when>
  <gd:where valueString='Building 41, Room X'/>
  <gd:eventStatus value="http://schemas.google.com/g/2005#event.confirmed"/>
  <gd:visibility value="http://schemas.google.com/g/2005#event.public"/>
  <gd:transparency value="http://schemas.google.com/g/2005#event.transparent"/>
</entry>
```

example: an event

```
<entry xmlns:gd="http://schemas.google.com/g/2005">
  <category scheme="http://schemas.google.com/g/2005#kind" term="http://schemas.google.com/g/2005#event"/>
  <id>http://mycal.example.com/feeds/jo/home/full/e1a2af06df8a563edf9d32ec9fd61e03f7f3b67b</id>
  <published>2005-01-18T21:00:00Z</published>
  <updated>2006-01-01T00:00:00Z</updated>
  <title>Discuss BazMat API</title>
  <content>We will discuss integrating GData with BazMat.</content>
  <author>
    <name>Jo March</name>
    <email>jo@example.com</email>
  </author>
  <gd:when startTime='2005-01-18T21:00:00Z' endTime='2005-01-18T22:00:00Z'>
    <gd:reminder minutes='15'/>
  </gd:when>
  <gd:where valueString='Building 41, Room X'/>
  <gd:eventStatus value="http://schemas.google.com/g/2005#event.confirmed"/>
  <gd:visibility value="http://schemas.google.com/g/2005#event.public"/>
  <gd:transparency value="http://schemas.google.com/g/2005#event.transparent"/>
</entry>
```

Did someone say Query?

Google

REST filter model

- categories - part of the path
  http://test.com/-/xtech/talks
- full text - the **q** parameter
  http://test.com?q=GData
- author - the **author** parameter
  http://test.com?author=Frank Mantek
- **updated-min/max** parameter
  http://test.com?updated-min=2006-01-01&updated-max=2006-12-31

http://test.com/-/xtech/talks/2006?author=frank mantek&q=GData&updated-min=2006-01-01&updated-max=2006-12-31

...and direct access

Google

each entry contains a “self” URI

- <link rel=”self” href=”http://test.com/feeds/talks/idforthis”/>

using that gives you just this entity

What about updating?

Google

Updates are based on the Atom Publishing protocol draft

- http://www.ietf.org/internet-drafts/draft-ietf-atompub-protocol-11.txt
  Proposed as a standard in May
- Updates are done using optimistic concurrency

Updates...

Google

Insert a new entry in a feed

- POST to the feed's **service.post** URI

Delete an entry

- DELETE to the entries **edit** URI

Update an entry

- PUT to the entries **edit** URI

Google...

Google™

just released one property supporting the Data API

- Google calendar was well received since its April 12th launch date
- more Google properties will support this API over time

Support?

Google

Language libraries for Java & C#, including documentation

Active community at code.google.com

Support for other languages (PHP, Javascript) is in the works

more quotes...

Google

"I think GData is hot. I'd like us to use this as a building block for some of the developer platform stuff we are doing for Windows Live" - **Dare Obasanjo (Microsoft)**

I just love quotes...

Google

"The next logical questions, for me at least, are:

1. Will MySQL add native GData support to the server anytime soon?
2. Should Yahoo begin to enable a GData API on our data stores?

I hope the answer to #1 is "yes, they should" and suspect the answer to #2 is "probably--at least for some of them." - **Jeremy Zawodny (Yahoo)**

Google Data Authentication

Google

ClientLogin: Account Authentication for Installed Applications

- programmatic login
- CAPTCHA
- But username/password stored by the application: Bad

AuthSub: Account Authentication Proxy for Web Applications

- Connects the user to a Google service that requests credentials
- The service then returns a token that the web application can use
- Google (rather than the web front end) securely handles and stores the user's credentials.

Applications

Google

Outlook sync

- Daniel Tiles, RemoteCalendars

Cell Phone Sync

- Thomas Oldervoll's Gcalsync, J2ME midlet

RDFCalendar XVCD

- By Justsystems Corporation: Google, eventful, upcoming

Resources

Google

- http://code.google.com/
- http://code.google.com/apis/gdata/overview.html
- http://code.google.com/apis/gdata/calendar.html
- Google Calendar Data API Developer Forum: http://groups.google.com/group/google-calendar-help-dataapi?lnk=li
- http://del.icio.us/chanezon/gdata

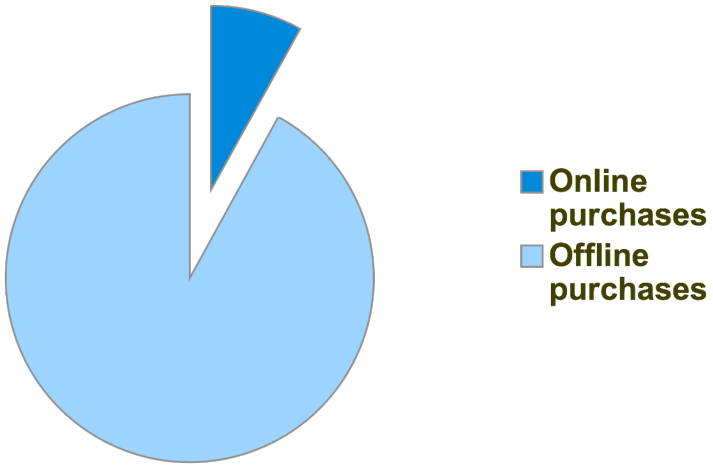
Google™ Checkout

Fast, secure checkout across the Web

Why Google Checkout

Only 8% of all retail purchases are expected to occur online in 2006*

2006 Consumer Purchases by Channel



* 2006 State of Retailing Online, Shop.org/Forrester

Problem: Discovery, Checkout

- Difficult to find trustworthy sellers
- Long, repetitive, checkout processes
- 63% of shopping carts are abandoned after beginning checkout*

* Online Customer Experience Survey, Allurent, February 2006

Opportunity: Search, Convenience

- 37% of online purchases start with search*
- 25% of search queries result in a purchase directly related to the query**
- Fast checkout can improve conversion

* 2005 Online Holiday Mood Study, Shop.org/BizRate Research
** The Role of Search in Consumer Buying, comScore/Google

DEMO

Google

Using Google Checkout

Product Benefits

Google

For Buyers…

Convenience

- Google Checkout badge helps identify merchants to buy from
- One place to track purchases

Speed

- Single log in for easy buying across the web

Confidence

- Credit card number concealment
- Fraud protection
- Email forwarding

For Merchants…

More Leads

- Google Checkout badge helps attract more customers

More Conversions

- Streamlined checkout process increases conversions

Lower Costs

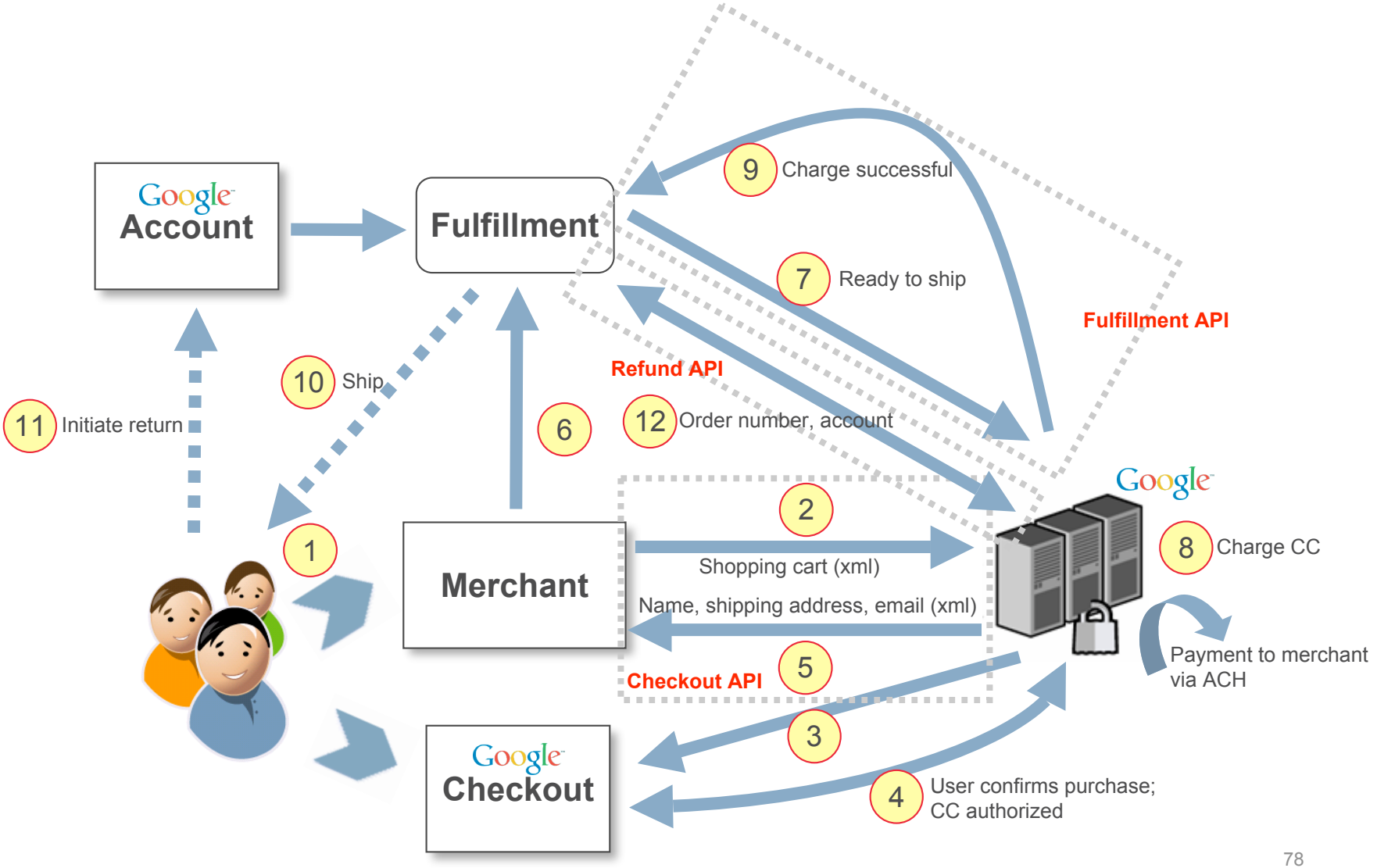
- Free transaction processing for sales up to 10X AdWords spend
- Low 2% + $0.20 per transaction fee for all other sales
- Fraud protection

AdWords and Google Checkout

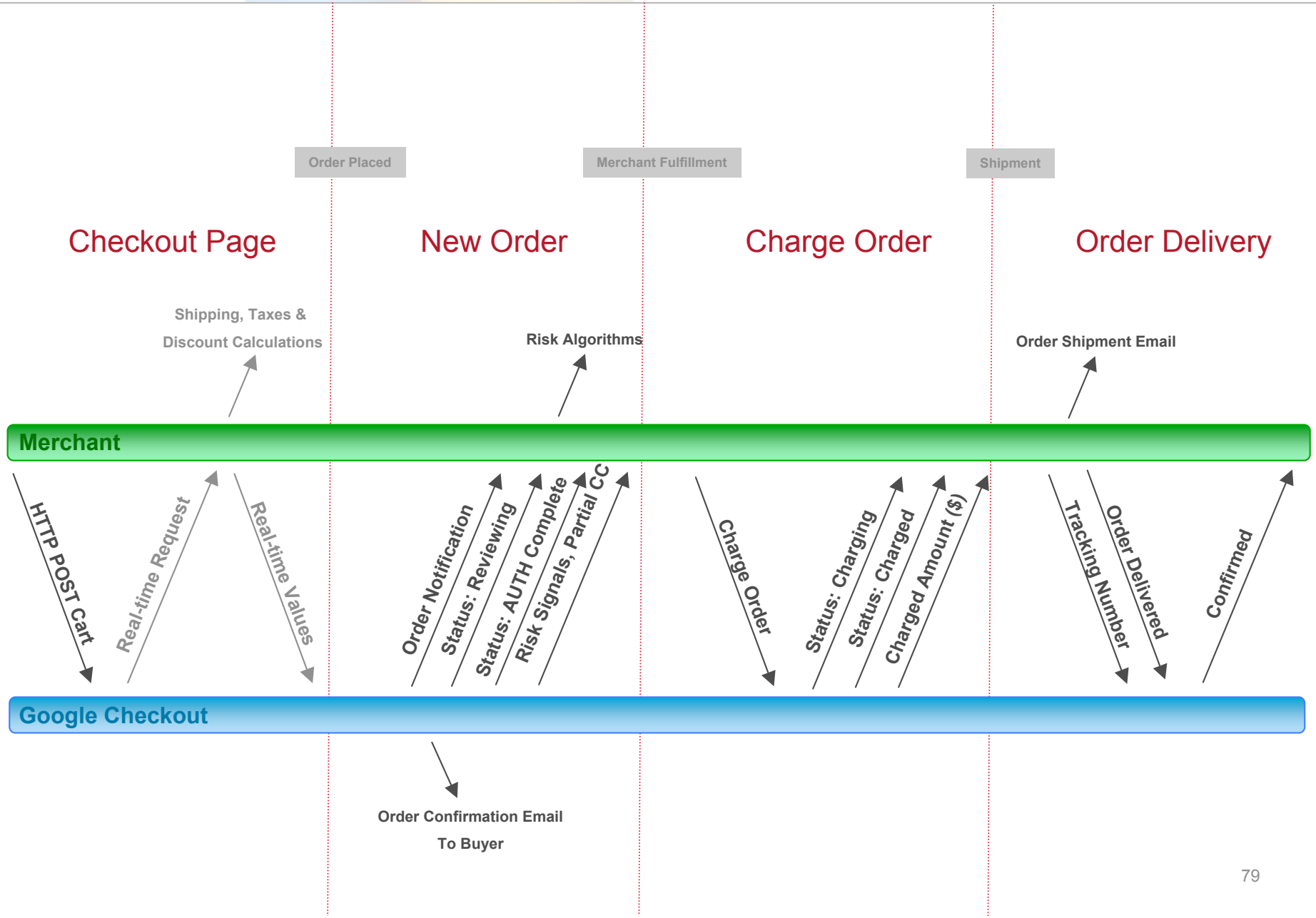
AdWords and Google Checkout work together to increase sales and lower costs.

1. Promote your business with AdWords
2. Get more traffic with the Google Checkout badge
3. Get higher sales conversions by offering a streamlined checkout process
4. Lower your costs with free transaction processing

Google Checkout Transaction Flow



Order Flows: Typical



Google Checkout Integration Options

- Buy Now buttons
- e-commerce partners
- HTML (Name/Value pair) API
- Google Checkout XML API
- Systems Integrators specialized in Checkout

http://checkout.google.com/seller/developers.html

For Developers: bonuses for bringing Google Checkout to your merchants
https://services.google.com/inquiry/checkout_ecommerce

Promotions

- Free processing for merchants in 2007
- Frequent Buyer Promotions

Google Checkout API

2 levels of integration

- Level 1: post shopping carts (taxes, shipping, coupons)
- Level 2: merchant calculations, order management, 2-ways API

REST API

- XML over HTTP
- XML Schema
- Synchronous / Asynchronous (notifications)
- HMAC-SHA1 for signature, Basic auth over SSL
- Samples and Libraries in many environments
  - PHP, Java, Windows Classic ASP (COM), .NET
  - OsCommerce, Zen Cart

Google Checkout API Level 1

Google

Static Checkout Buttons

Large – 180 x 46

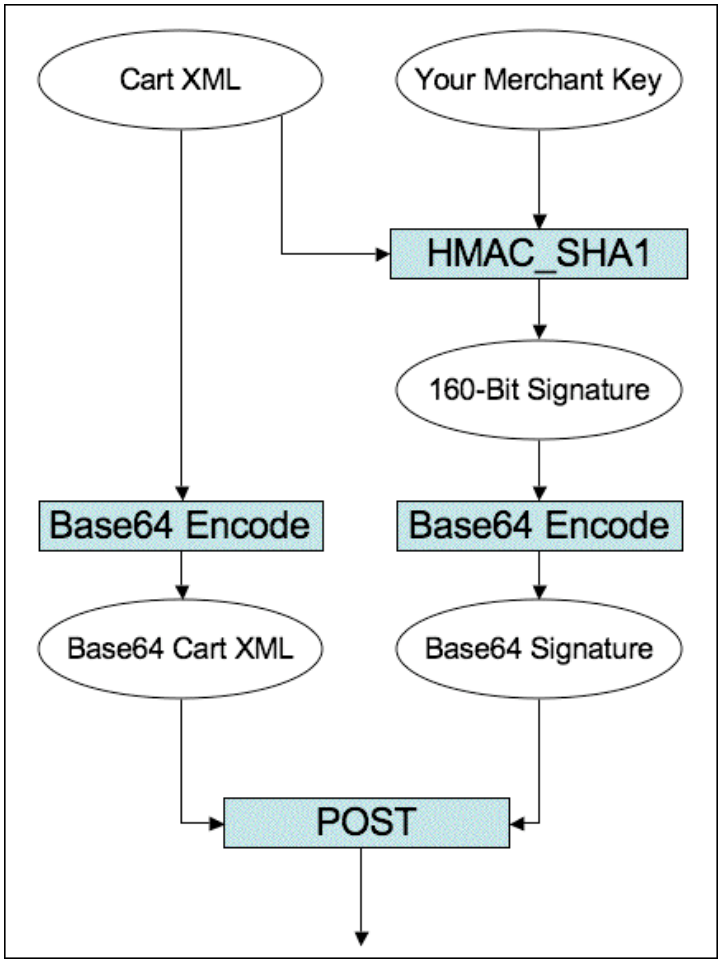
Form with hidden parameters

```
<form method="POST"
action="https://sandbox.google.com/cws/v2/Merchant/123456789
0/checkoutForm" accept-charset="utf-8">
    <input type="hidden" name="item_name_1" value="Peanut
Butter"/>
    <input type="hidden" name="item_description_1"
value="Chunky peanut butter."/>
    <input type="hidden" name="item_quantity_1" value="1"/>
    <input type="hidden" name="item_price_1" value="3.99"/>
<input type="image" …
src="http://sandbox.google.com/buttons/checkout.gif…"/>
</form>
```

Google Checkout API Level 1

Google

Post Cart XML + Signature



Level 2 Integration

Google

Level 1: Merchants use the UI to manage orders

Level 2: lets you integrate your order management system with Google

- Google calls an endpoint on the merchant side
- SSLv3 + Basic Auth

Merchant Calculations API

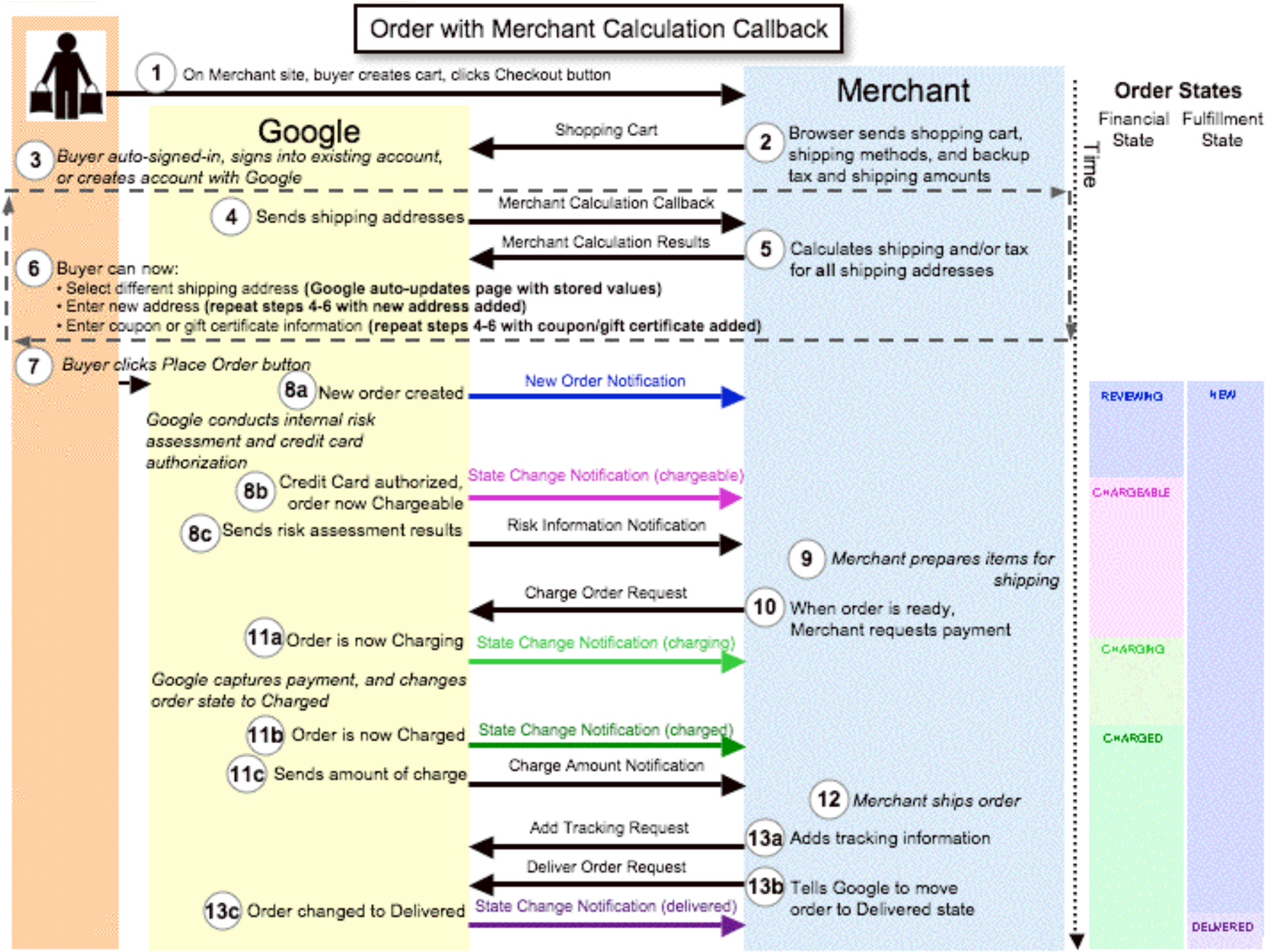
- Merchants can apply their own business logic to compute values
- Shipping, Taxes, Coupons
- 3s to answer, else Google uses default values

Notification API

- New Order, Risk Information, Order State Change, Amount
- If fails, retry policy up to 30 days

Order Processing API

- Change order state


Order with Merchant Calculation Callback
Google
Merchant
Order States
Financial State
Fulfillment State
Time
1 On Merchant site, buyer creates cart, clicks Checkout button
Shopping Cart
2 Browser sends shopping cart, shipping methods, and backup tax and shipping amounts
3 Buyer auto-signed-in, signs into existing account, or creates account with Google
Merchant Calculation Callback
4 Sends shipping addresses
Merchant Calculation Results
5 Calculates shipping and/or tax for all shipping addresses
6 Buyer can now:
• Select different shipping address (Google auto-updates page with stored values)
• Enter new address (repeat steps 4-6 with new address added)
• Enter coupon or gift certificate information (repeat steps 4-6 with coupon/gift certificate added)
7 Buyer clicks Place Order button
8a New order created
New Order Notification
Google conducts internal risk assessment and credit card authorization
8b Credit Card authorized, order now Chargeable
State Change Notification (chargeable)
8c Sends risk assessment results
Risk Information Notification
9 Merchant prepares items for shipping
Charge Order Request
10 When order is ready, Merchant requests payment
11a Order is now Charging
State Change Notification (charging)
Google captures payment, and changes order state to Charged
11b Order is now Charged
State Change Notification (charged)
11c Sends amount of charge
Charge Amount Notification
12 Merchant ships order
Add Tracking Request
13a Adds tracking information
Deliver Order Request
13b Tells Google to move order to Delivered state
13c Order changed to Delivered
State Change Notification (delivered)
REVIEWING
NEW
CHARGEABLE
CHARGING
CHARGED
DELIVERED

Google Checkout API in Java

Google

Open Source project

- http://code.google.com/p/google-checkout-java-sample-code/
  - Apache 2.0 license

Designed to work for Java 1.4 and above

- org.w3c.dom for XML
- No Generics

J2EE sample

- War sample: in memory notifications, logged to a file
- Ear sample: JMS queue for notifications

Easy to integrate in your Java application

- Let's hook up the legendary Java PetStore!

Learning from our mistakes

Google

Minimizing dependencies is good, Simplicity rules

Initially used JAXB

- Needed to maintain 2 projects
  - JWSDP 1.6 for Java 1.4 users
  - JWSDP 2.0 for Java 1.5 users
- Plus JAXB version conflicts for users

Redesign 1

- No Generics, no JAXB
- Define interfaces for business objects, fully pluggable

Redesign 2: remove excessive pluggability

- Business objects as classes
- Sample application to customize: pure servlet or MDB

Google Checkout Java Sample App

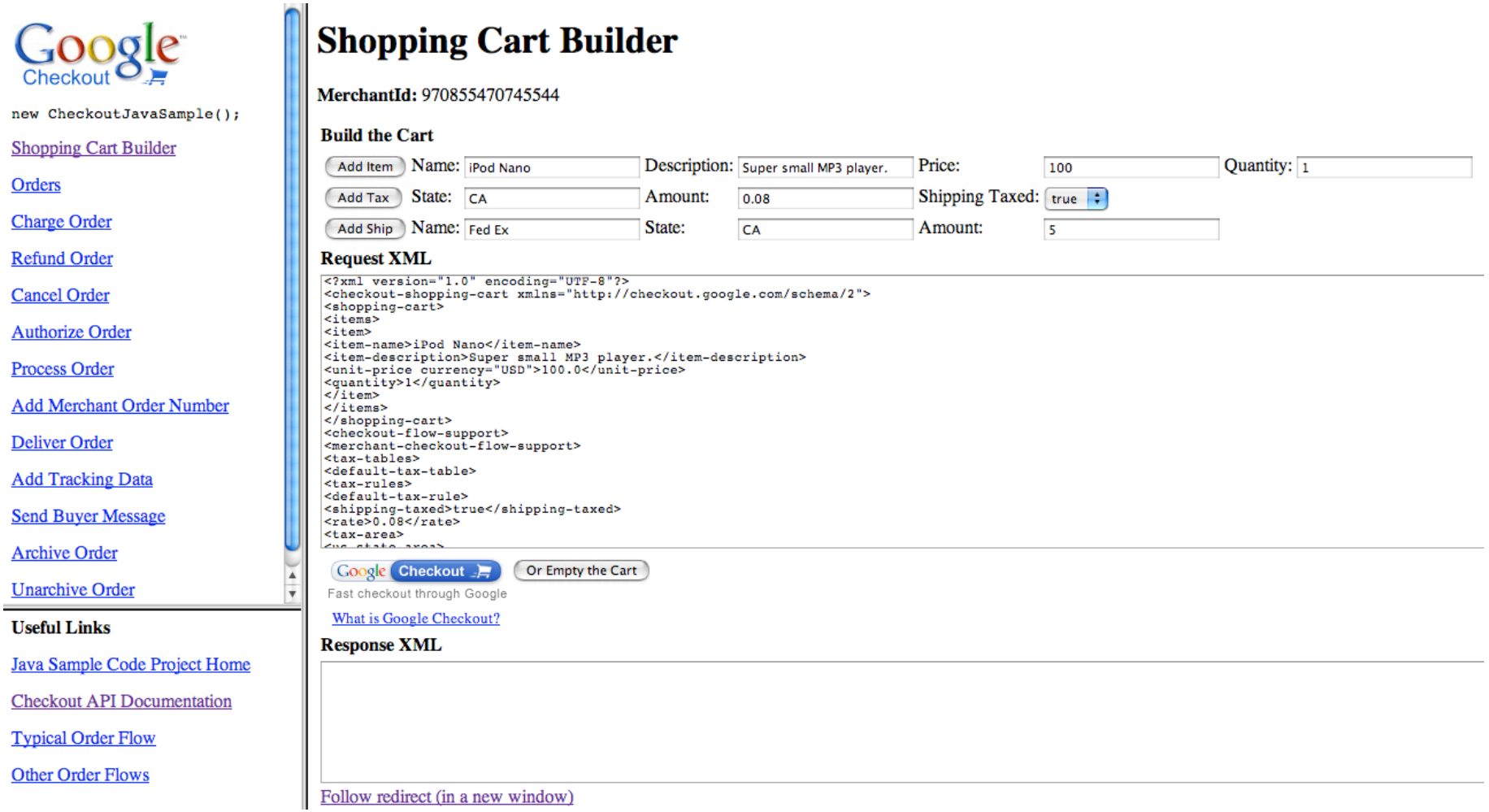
Console to experiment with the API

Console view to make calls

- UI to Build your XML messages
- Place calls
- Watch notifications

Very easy way to learn the API

Google Checkout Java Console



Checkout Java Sample Structure

Google

Use: XML Marshalling/Unmarshalling and utilities library, no need to customize


JavaCheckoutRefImpl [trunk/ref-impl/JavaCheckoutRefImpl]

Customize: Notification and Merchant Calculation processor examples


JavaCheckoutExampleCommon [trunk/examples/JavaCheckoutExampleCommon]

Web application example

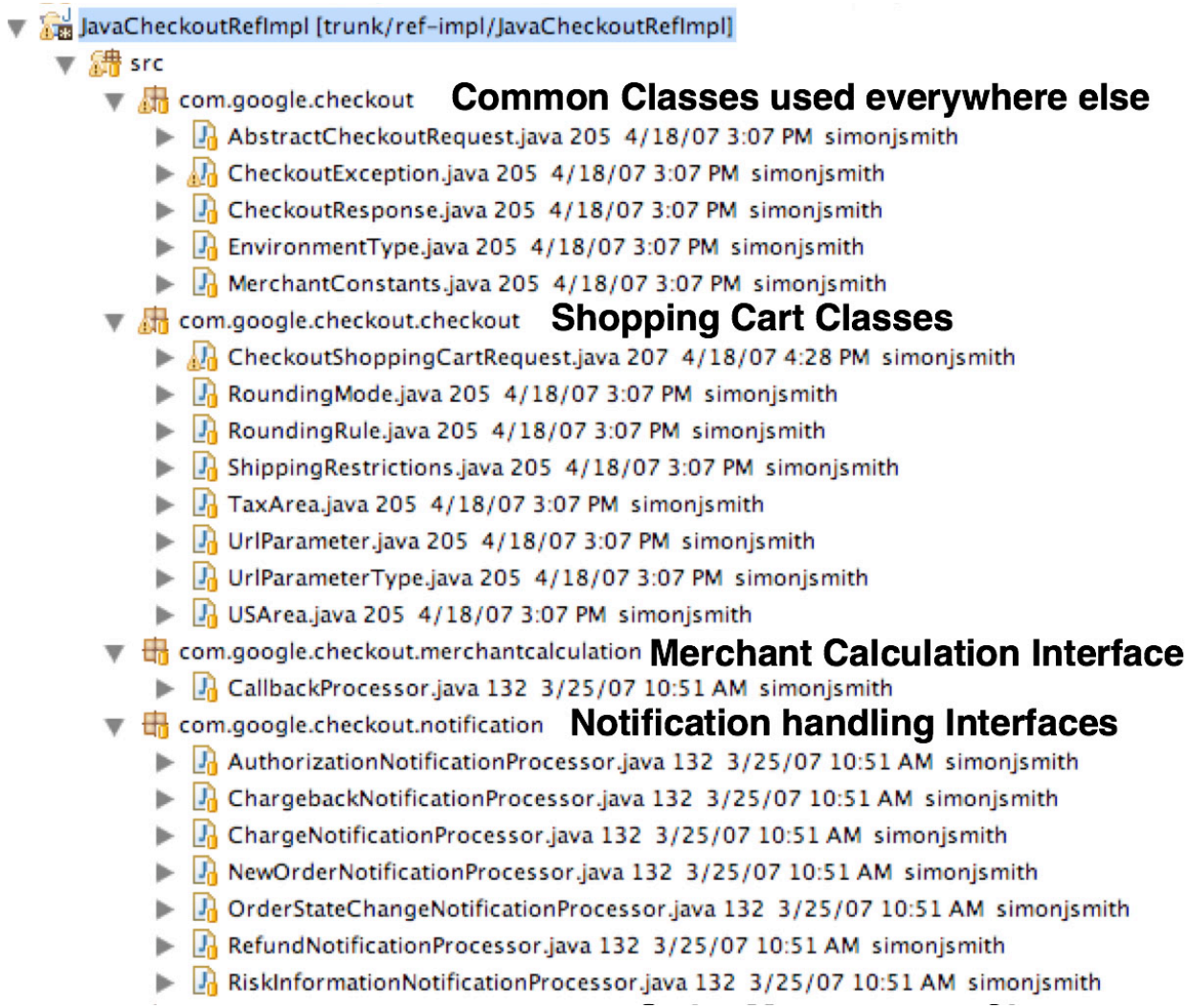

JavaCheckoutWebExampleWar [trunk/examples/JavaCheckoutWebExampleWar]

Optional: EJB and EAR examples if you want to use MDB to process notifications


JavaCheckoutJ2eeExampleEjb [trunk/examples/JavaCheckoutJ2eeExampleEjb]


JavaCheckoutJ2eeExampleEar [trunk/examples/JavaCheckoutJ2eeExampleEar]

Checkout Java Library Structure (1/2)



Checkout Java Library Structure (2/2)

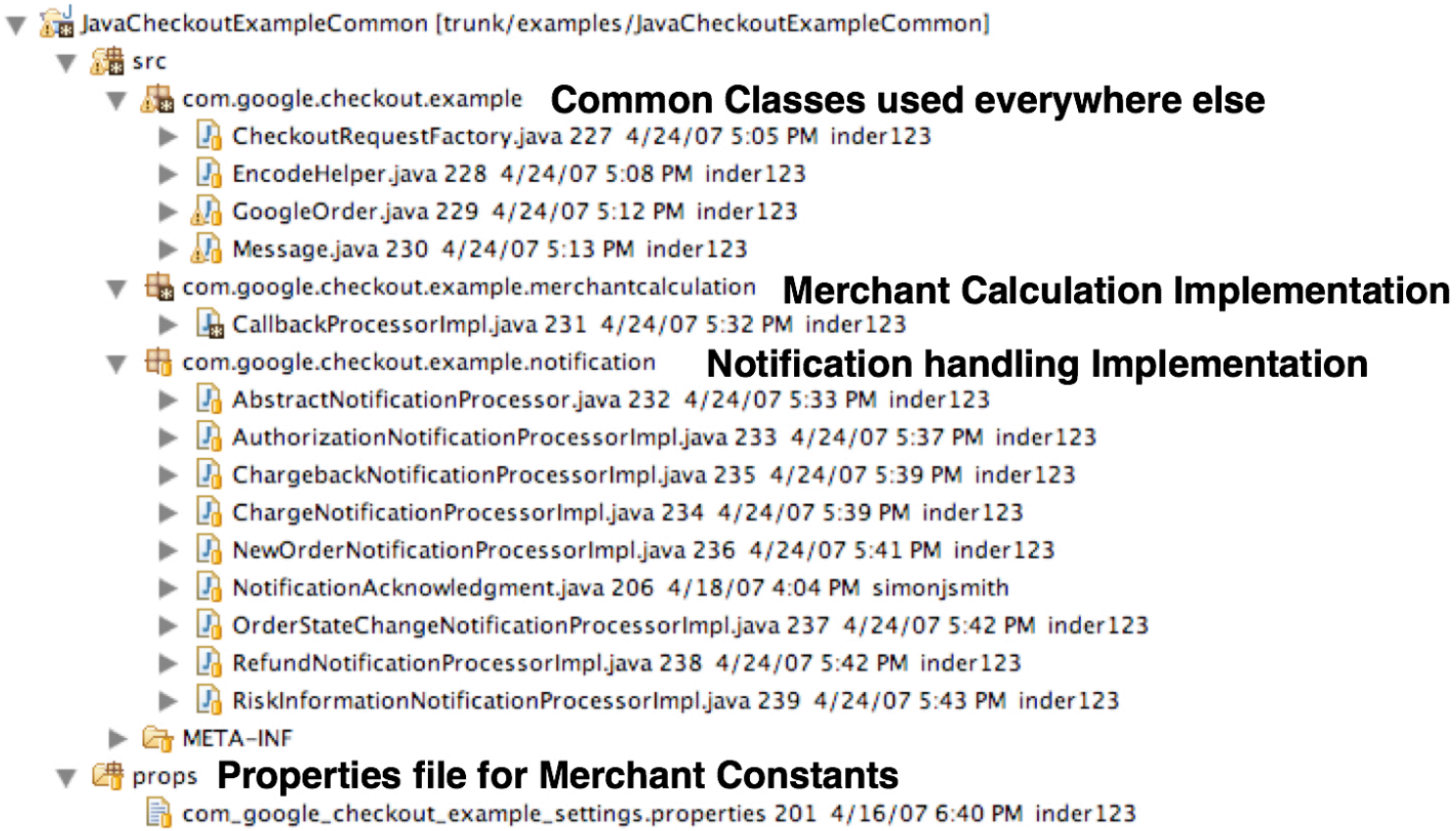
Google

Checkout Java Example Structure

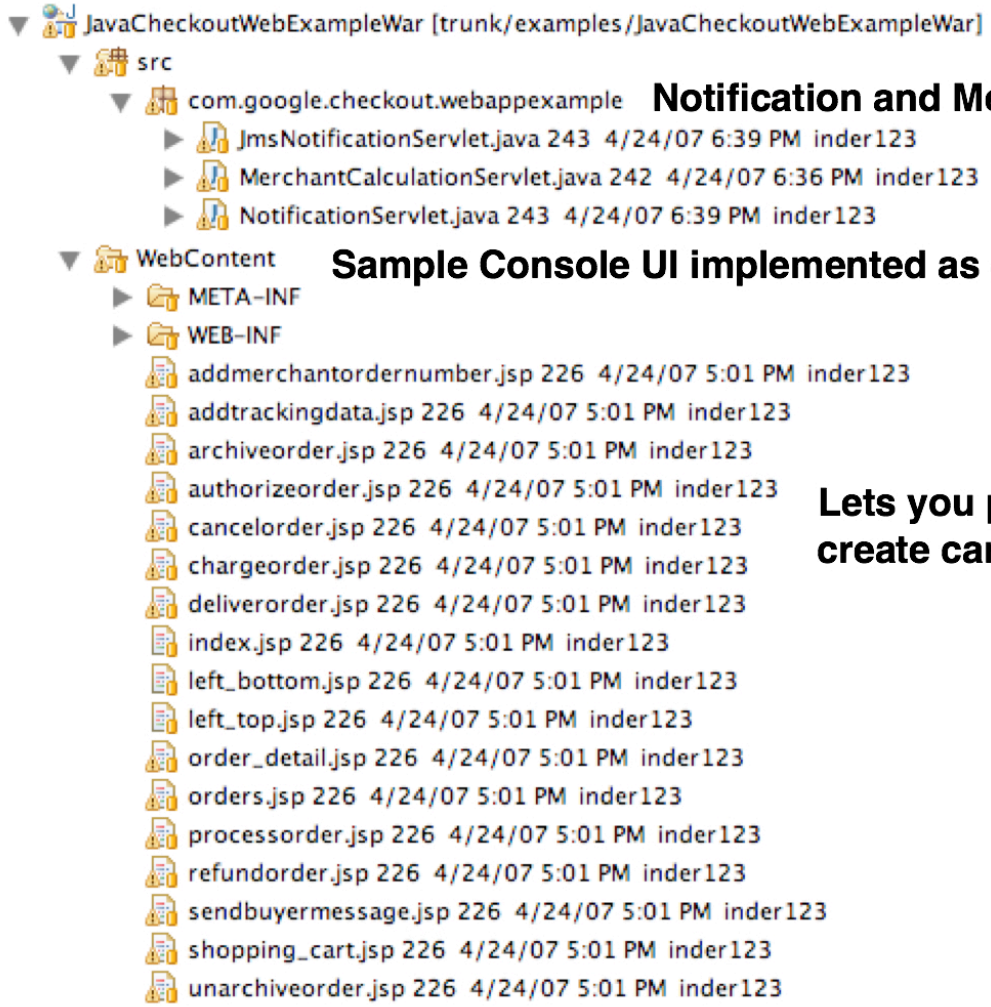
Google

Customize this project

Implement your business logic, replace the factory



Checkout Java Web Application Structure



Notification and Merchant Calculation Servlets

Dispatch processing to example project handlers

Sample Console UI implemented as JSPs

Lets you play with various aspects of the API create carts, examine XML messages

Posting a Cart in Java (1/2)

Google

```
// Set up Merchant Constants (usually from a properties file).
MerchantConstants mc = new MerchantConstants("248088419036991",
     "Q2x_yWSVUHokvFxw1MqVjp", "0", "US",
"https://sandbox.google.com/checkout/cws/v2/Merchant/checkout",
"https://sandbox.google.com/checkout/cws/v2/Merchant/merchantCheckout",
"https://sandbox.google.com/checkout/cws/v2/Merchant/request");

// Create a shopping cart
CheckoutShoppingCartRequest cart = new CheckoutShoppingCartRequest(mc);

// Add items: 2 Biscuits
cart.addItem("Biscuit", "Yummy biscuit", new Float(9.99).floatValue(),
2);
// Add shipping: a shipping method restricted to California.
ShippingRestrictions sr = new ShippingRestrictions();
sr.addAllowedStateCode("CA");
cart.addFlatRateShippingMethod("USPS Ground", new Float(3.99), sr);
```

Posting a Cart in Java (2/2)

```
// Add taxes: here 8.3% in California, shipping is taxed.
TaxArea ta = new TaxArea();
ta.addStateCode("CA");
cart.addDefaultTaxRule(0.08375, true, ta);

// Send the cart
CheckoutResponse response = cart.send();

if (response.isValidRequest()) {
    System.out.println(response.getXmlPretty());
    // Redirect the user to response.getRedirectUrl().
} else {
    System.out.println(response.getErrorMessage());
}
```

Google Checkout Button posts to that servlet

If the Cart post is successful redirect the user to Google Checkout!

Processing a Notification (1/3)

Google

```
// NotificationServlet dispatches to the right Processor.
public void doPost(…)
…
    String notification =
getNotificationBody(request.getInputStream());
    String result = dispatch(notification);
    PrintWriter out = response.getWriter();
    out.print(result);
…
private String dispatch(String notification) throws Exception {
    if (notification.indexOf("new-order-notification") > -1) {
      NewOrderNotificationProcessor processor =
CheckoutRequestFactory
          .newNewOrderNotificationProcessor();
      return processor.process(notification);
    }
…
```

Processing a Notification (2/3)

Google

```
// Implements an interface for pluggability.
public class NewOrderNotificationProcessorImpl extends
AbstractNotificationProcessor
    implements NewOrderNotificationProcessor {

  private MerchantConstants merchantConstants;

  // Dependency injection style constructor.
  public NewOrderNotificationProcessorImpl(MerchantConstants
merchantConstants) {
    this.merchantConstants = merchantConstants;
  }

  //This is where your business logic goes
  public String process(String callbackXML) throws CheckoutException {

    String ack = "";
    try {       // Parse the notification.
      Document document = Utils.newDocumentFromString(callbackXML);
```

Processing a Notification (3/3)

Google

```
    String orderNumber = Utils.getElementStringValue(document,
document.getDocumentElement(), "google-order-number");
…
    // Build an order.
    GoogleOrder order =
        GoogleOrder.findOrCreate(merchantConstants.getMerchantId(),
        orderNumber);
    order.setLastFulStatus(lastFulStatus);
…
    ack = getAckString();

    // Log the order in a file or database.
    order.addIncomingMessage(timestamp,
                        document.getDocumentElement().getNodeName(),
                        Utils.documentToStringPretty(document), ack);
} catch (Exception e) {
    throw new CheckoutException(e);
}
return ack;
```

Order Management
add tracking number in Java

Google

```
// Setup Merchant Constants (usually from a properties file).
MerchantConstants mc = new MerchantConstants(…);

// Create a request to add UPS tracking data to the order.
AddTrackingDataRequest atdr =
new AddTrackingDataRequest(mc, "3750535331300 49", "UPS", "12345");
System.out.println("XML To Send: "+atdr.getXml());

// Send the request.
CheckoutResponse cResponse = atdr.send();

String cResponseStr = cResponse.getXml();

if (cResponseStr != null) {
    System.out.println(new StringBuffer(
                        "XML Received: ").append(cResponseStr));
} else {
    System.out.println("No Response XML was sent.");
}
return;
```

DEMO

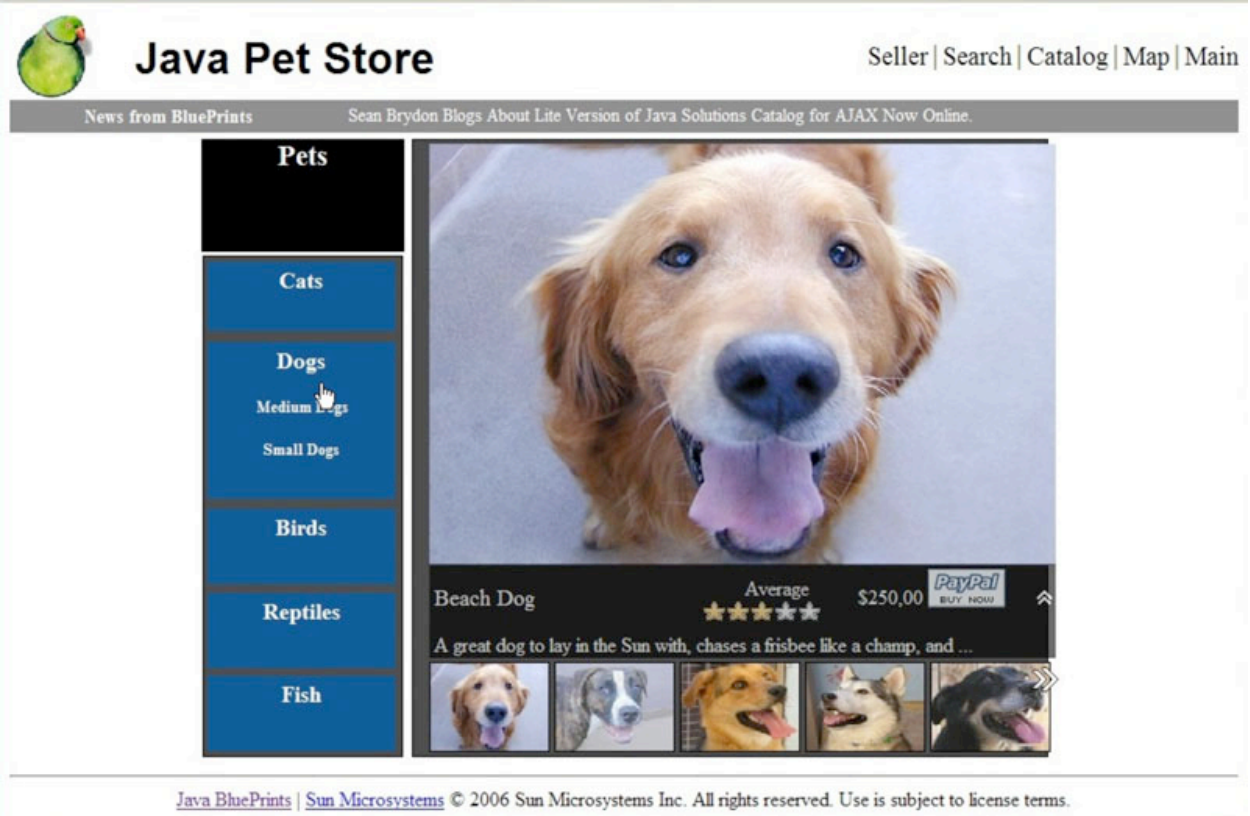
Google

Google Checkout Java Sample Console

The Java PetStore 2.0

Let's integrate it with Google Checkout!

Full of Annotations and Ajax Goodness



Mendoza: Google Checkout Test Server

Google

Integration testing for your Checkout implementation

Problem: testing synchronous and asynchronous http messages, some triggered by Ajax interactions.

Solution: What if we just add something between you and Checkout to ease the burden of manual testing integration?

Cf. Cedric Beust “Unit or Functional”
http://beust.com/weblog/archives/000444.html

Google Checkout Test Server

Integration testing for your Checkout implementation

Test Server between you and Checkout “acts as a transparent proxy”

Uses Selenium to place orders + user interaction

- as if it was a buyer using a browser
- **Test tool for web applications.**
- **Tests run directly in a browser, just as real users do (run in Internet Explorer, Mozilla and Firefox on Windows, Linux, and Macintosh).**

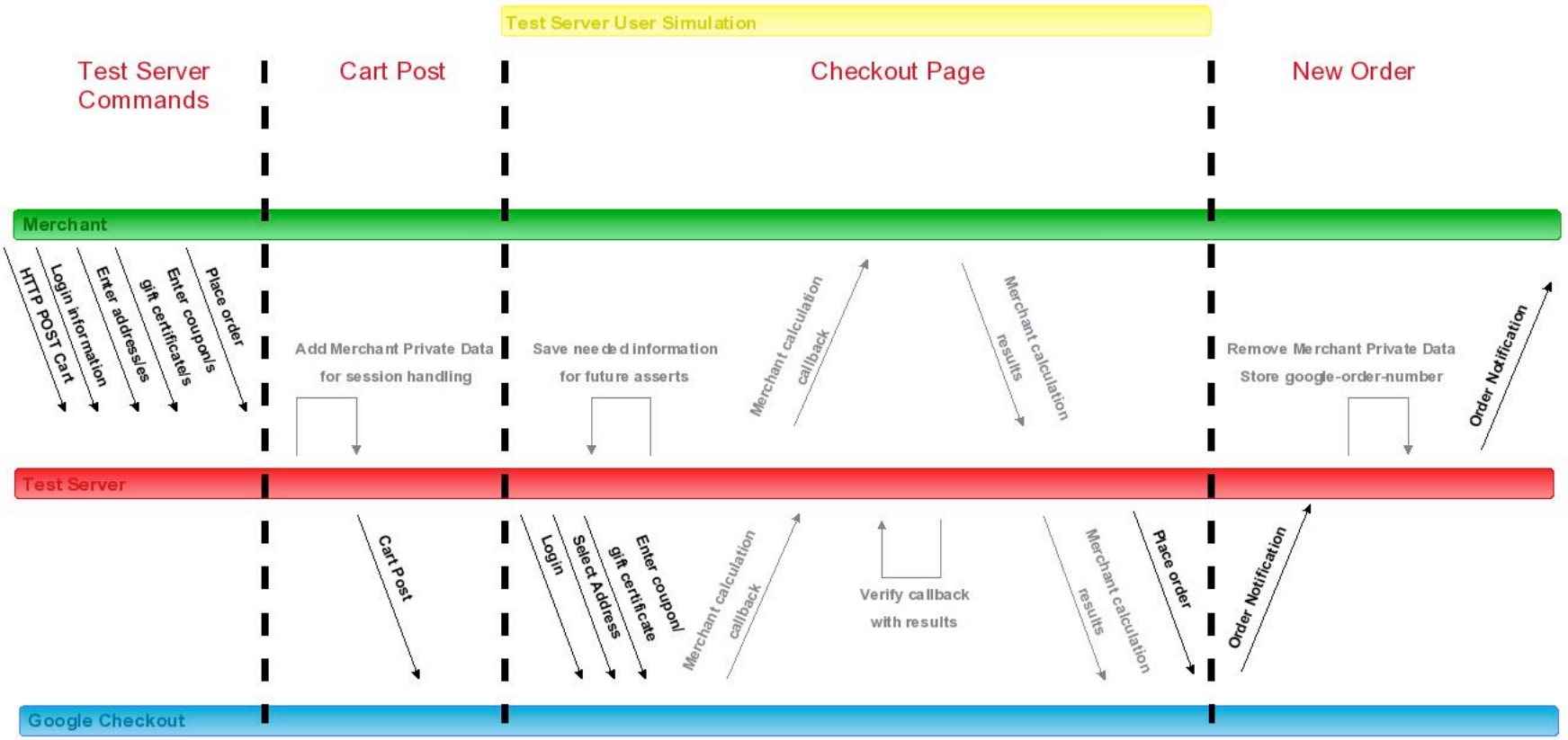
Google Checkout Test Server

Integration testing for your Checkout implementation

Test Server:

- **Simple Java stand alone application (jar).**
- **Receives http commands from the Merchant then simulates user interaction + interacts with Checkout using the API.**
- **Verifies syntax + semantics + flow of the messages.**

Google Checkout Test Server



Google Checkout Test Server

Integration testing for your Checkout implementation

Test Server future development:

- Improved web UI.
- Unit Testing Tools
- Test Suites
- Performance testing capabilities.

Malbec:

- Batch automated testing tool for the Google Checkout API integration level 2.
- Includes batch OSCommerce and ZenCart integration testing.

DEMO

Using the Google Checkout Test Server to test our Integration

Summary

Google

Google Checkout is worth integrating with

- Better click-through rate on ads sporting the shopping cart
- Free processing in 2007

Google Checkout API makes it possible

- Easy things easy: Level 1 integration
  - static Buy Now buttons, HTML buttons, signed XML
- Difficult things possible: Level 2 integration
  - Uniform HTTP + XML API
  - 2 way, synchronous and asynchronous

Sample code makes it easy

Mendoza makes testing an integration easier

Developers can make money with Checkout

- Incentive program for helping merchants integrate
- https://services.google.com/inquiry/checkout_ecommerce

For More Information

Google

Main site: http://checkout.google.com

Google Checkout Video / Demo:
https://checkout.google.com/seller/demo.html

Developers Center: https://checkout.google.com/seller/developers.html

Google Checkout API Center: http://code.google.com/apis/checkout/

Developer's Guide:
http://code.google.com/apis/checkout/developer/index.html

Java Sample
http://code.google.com/p/google-checkout-java-sample-code/

Google Checkout API Blog
http://googlecheckoutapi.blogspot.com/

Google AdWords-Checkout Business Mashup

Google

Demo

Fully Buzzword Compliant:-)

- PHP, SOAP, REST, Ajax, Business Mashup, Google

Agua Biz: shows how to integrate AdWords and Checkout in your PHP online store

Open Source

http://google-apility.sourceforge.net/agua_biz.html

SOAP vs REST

Google

	+	-
SOAP	Client toolkits: most code can be generated from WSDL Batch support: you batch inside of request	Doc/Lit support uneven
REST	url for resources Can use http libraries, no need SOAP client lib Caching Can load balance without parsing xml based on url components (like checkout API does)	Less code generation: xsd -> lang OK but still need to do http work No agreement on batch (cf Atom PaceBatch discussion)

SOAP vs REST

Google

"S for Simple": Recent blog thread summarized by Tim Bray

- http://www.tbray.org/ongoing/When/200x/2006/11/16/WS-Socratic

"Why SOAP sucks" Nelson Minar, ex-Google, designer of early Google APIs SOAP Search and AdWords

- http://www.somebits.com/weblog/tech/bad/whySoapSucks.html

"There's an amusing dialogue floating around about how simple SOAP is. As someone who bears some past responsibility for well used SOAP services (Google's APIs for search and AdWords) let me say now I'd never choose to use SOAP and WSDL again. I was wrong."

"Truly, none of this protocol fiddling matters. Just do something that works."

Don't miss Sam Ruby and Leonard Richardson's last book

"RESTful Web Services" a must read for all software developers

REST Issues

Google

Verbose: in order to update a field you must PUT the whole entry

Batch semantics controversial (Google rolled its own)

Tool support OK but less mature than SOAP

Google Web Toolkit

What, Why, and How

Joel Webber
Google, Inc.

Topics

Google

Ajax State of the Art?

The Google Web Toolkit

High-Performance Ajax

Integration with Existing Web Technologies

Parting Thoughts

The Slippery Slope of Ajax

Google

Product management and engineering decide to add script…

I begin experimenting with JavaScript

- Cool! The boss loves it – and it's fun!
- Maybe I'll get a raise!

The salespeople love it…

We're an Ajax shop?

Oh yeah, we can't just support Internet Explorer

Wait…this is hard

I hate browsers with all my heart

I quit – find another sucker to maintain this spaghetti

State of the Ajax “Art”

Google

The world of Ajax is crazy and nigh unmanageable

You need regexs to list all the technologies on one page

- HTTPS?, [DX]?HTML (3.2|4.0), CSS[1-3]
- DOM Level[0-3]
- (Java|ECMA|J|VB)Script
- (X|VR?|Math)ML
- SVG, Canvas, Flash
- JSONP?, SOAP, XML-RPC

Poor interoperation and consistency across browsers

JavaScript is too flexible for big projects and team projects

Yet browsers are a great way to distribute apps

Hello? Software Engineering?

Google

Hey, what happened to all that software engineering stuff we figured out in the last few decades?

Static type checking?

Design patterns?

Unit testing?

Code reuse?

IDEs?!?!

Debugging?!?!

How can we restore some sanity?

Requirements Laundry List

Google

Ajax features with web usability

Use the Java language, developers and technologies

Debugging, JUnit, findbugs, profiling, coverage, javadoc

Eliminate browser-specific coding with very low overhead

Simple reuse via jars

Rich remote procedure call (RPC) semantics

Minimum size and maximum speed, especially at startup

Great scalability

Basically: the impossible…

Unless you compile Java into JavaScript :-)

Topics

Google

Ajax State of the Art?

The Google Web Toolkit

High-Performance Ajax

Integration with Existing Web Technologies

Parting Thoughts

What is Google Web Toolkit (GWT)?

Google

What is GWT?

- A set of tools for building AJAX apps in the Java language
- Open Source (Apache 2.0 license)

What makes GWT interesting?

- Write, run, test, and debug everything in Java, both client-side UI code and server-side business logic

Isn't that called an applet?

- No JVM required
- GWT converts your working Java source into pure JavaScript

GWT is a compiler?

- GWT's Java-to-JavaScript compiler is a big part of it, but there's really a lot more to the story than that…

Code Sample – Hello, AJAX

Google

Without further ado…

```
public class Hello implements EntryPoint {

  public void onModuleLoad() {
    Button b = new Button("Click me", new ClickListener() {
      public void onClick(Widget sender) {
        Window.alert("Hello, AJAX");
      }
    });

    RootPanel.get().add(b);
  }

}
```

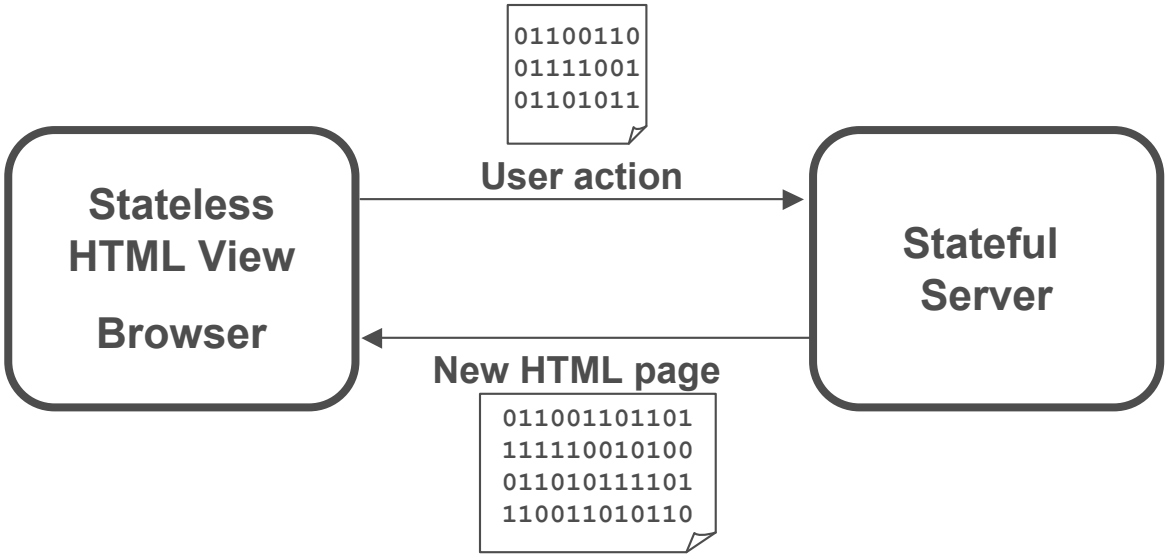
Demo

Hello, AJAX

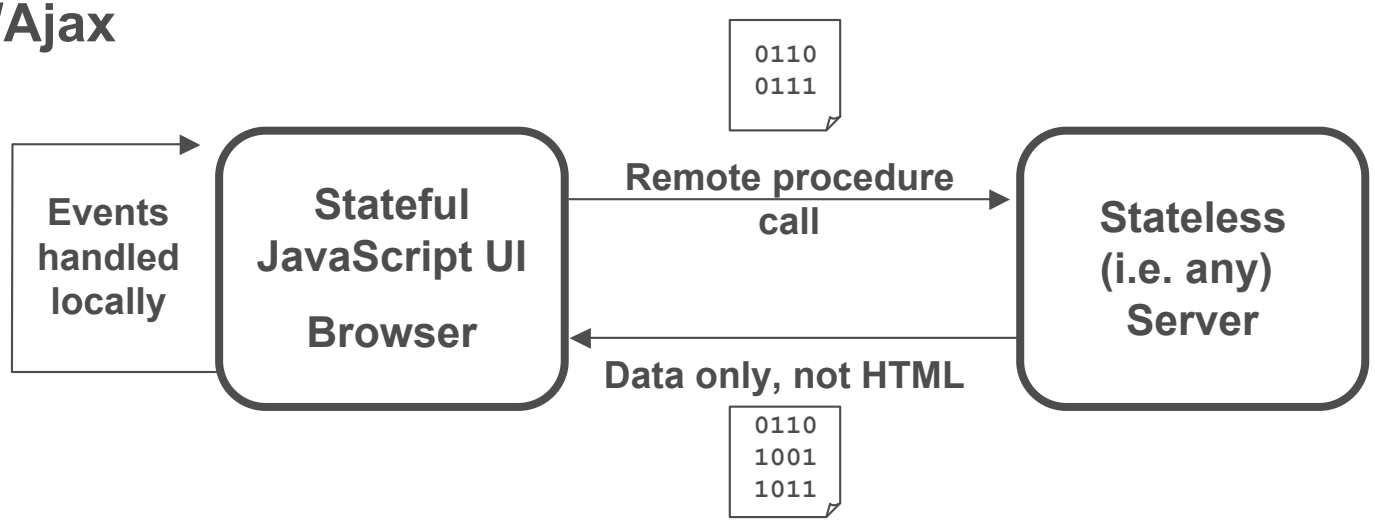
Traditional HTML vs. GWT/Ajax

Google

Traditional HTML



GWT/Ajax



Wow, That's So Much Easier

Google

Redefining the problem has been fruitful

Session state? All client…not a server issue

UI event handling? All client…no round trips

Deployment? Use any web server…just copy compiled JS

Leverage for the biggest Ajax headaches

- Mantra: Solve the problem once & wrap it in a class
- History? Create a History class
- Cross-browser? Create a DOM class
- RPC? Create an all-Java RPC mechanism

Widgets and Layout

Google

Build cross-browser widgets in straight Java

Constraint-based layout with panels

Create new widgets from existing ones

```
public class InboxWidget extends Composite {
  private EmailList list = new EmailList();
  private EmailPreview pvw = new EmailPreview();
  // combine them together in a simple panel to
  // create a new, reusable composite widget
}
```

Styled with CSS!

Remote Procedure Calls

Google

Many solutions out there (JSON, XML-RPC, …)

A pure Java RPC interface sure is nice!

```
interface SpellService extends RemoteService {
  String[] suggest(String word)
}
```

Client and server can speak the same language

Inner classes make it easy to deal with asynchronous RPCs

Remote Procedure Calls

Google

```
SpellServiceAsync spell = GWT.create(SpellService.class);

spell.suggest("compnent", new AsyncCallback() {

  void onSuccess(Object result) {
    String[] alts = (String[])result;
    if (alts.length > 0)
      showSuggestionsInGui(alts);
  }

  void onFailure(Throwable e) {
    reportProblemInGui(e);
  }

});
```

Demo

Google Checkout Buttons Generator

Topics

Google

Ajax State of the Art?

The Google Web Toolkit

High-Performance Ajax

Integration with Existing Web Technologies

GWT Internationalization

Parting Thoughts

Optimize Like Crazy

Google

Tough decision not to support reflection and class loading

And I'm so glad! Three words: Whole program optimization

For example, type tightening to eliminate polymorphism

- `Shape s = new Circle(2);  // radius of 2`
- `double a = s.getArea();`

- becomes
  - `Circle s = new Circle(2); // radius of 2`
  - `double a = (s.radius * s.radius * Math.PI);`

Inlining lets us discard the entire `getArea()` method!

Imagine those sorts of optimizations across your entire app

In JavaScript, reducing size and increasing speed are complementary goals, which makes optimizations *really* fun

Startup Time

Google

Absolutely crucial

- Should be measured in milliseconds
- If startup time isn't acceptable, nothing else matters

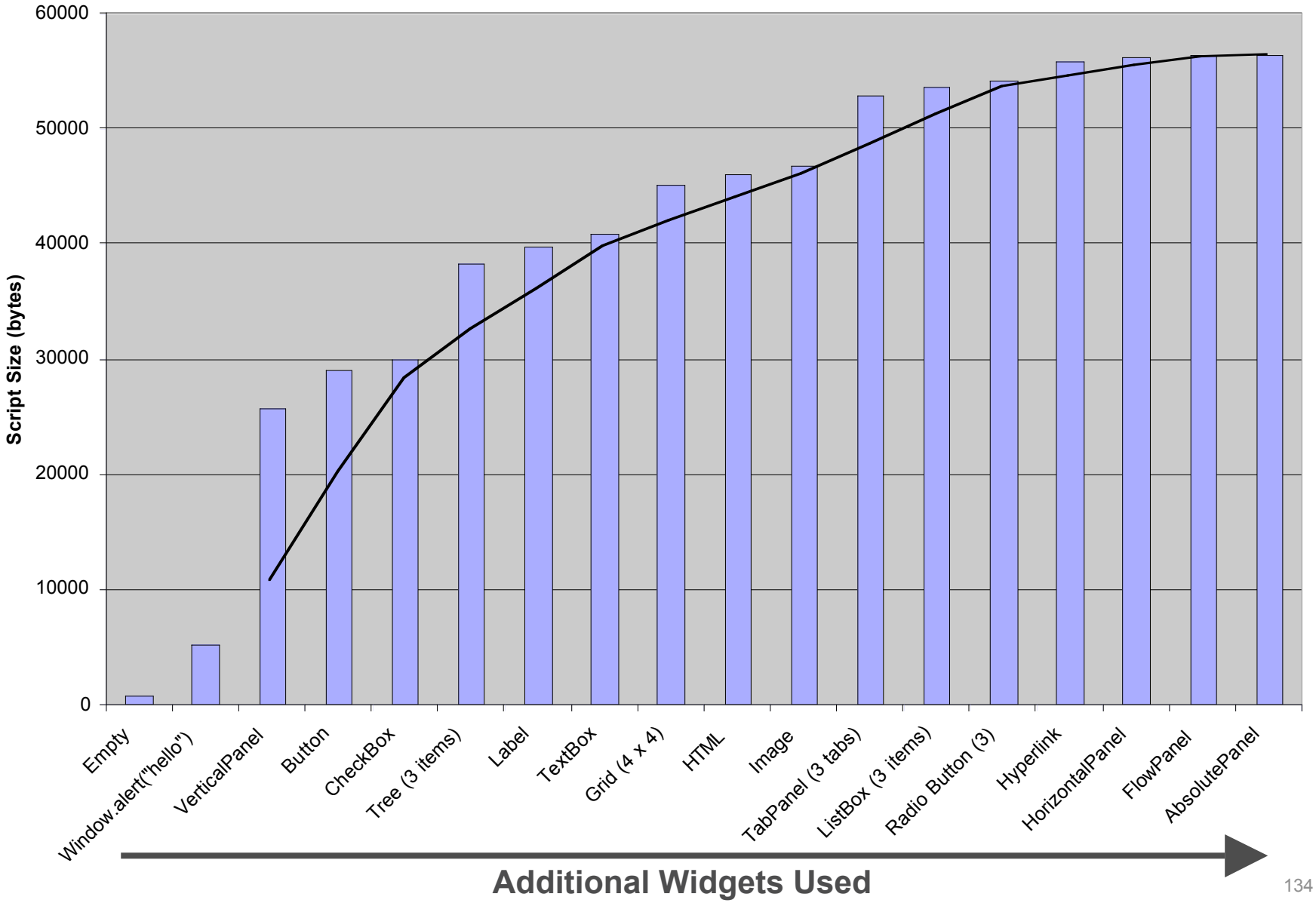
Very hard to do well

- Loading code with synchronous XHR is out of the question
- <script> tags serialize HTTP requests
- GZip your script ahead of time? Good idea, but…
- Some versions of IE6 fail on gzipped .js files
- Script versioning vs. cacheability

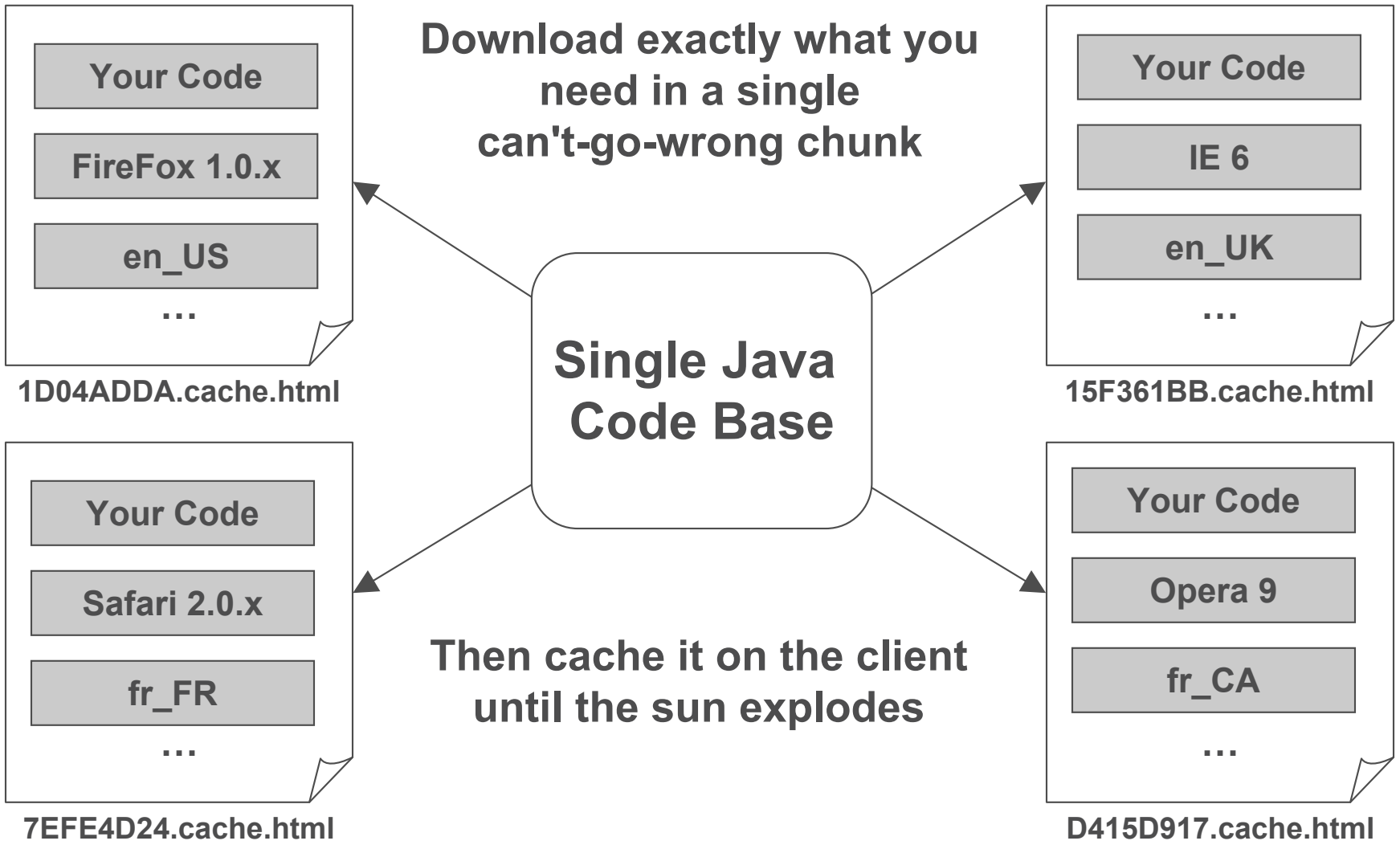
GWT gives you leverage

- Compiled output includes only what a particular user needs
- Output is JS wrapped in HTML, which is safely gzip'able
- Loads code in an <iframe> in parallel with the page
- Scripts are named uniquely and are perfectly cacheable

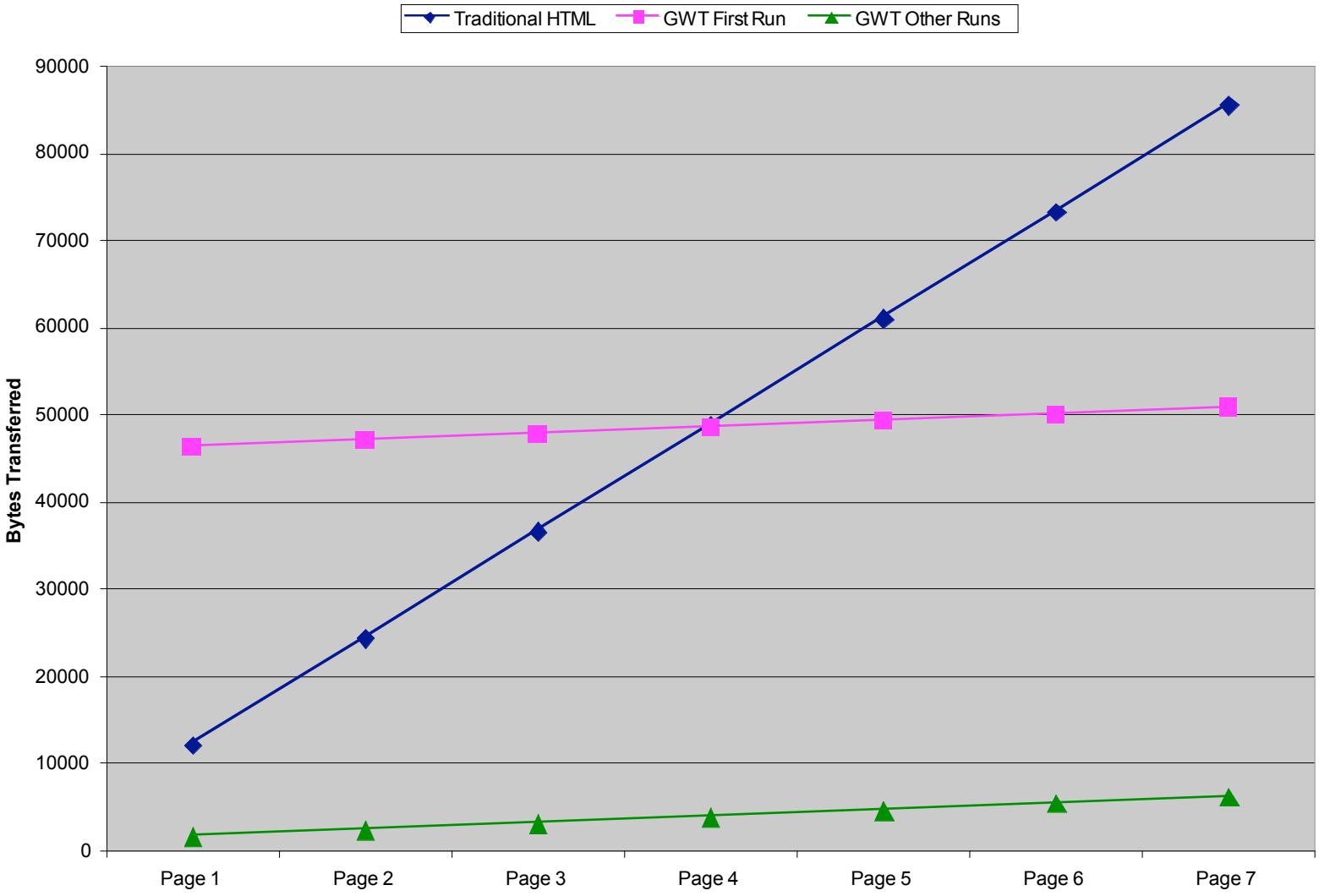
Compilation: Only Pay for What You Use



Compilation: Optimized Per Client



Less Bandwidth: Lower Costs and Faster Apps



Topics

Google

Ajax State of the Art?

The Google Web Toolkit

High-Performance Ajax

Integration with Existing Web Technologies

Parting Thoughts

Why Do Users Love Web Apps?

Google

No installation

- Every application is just a URL away
- No such thing as "DLL Hell" – it just works
- Capable of super-fast startup time

Simple usability

- Pages have a simple, friendly look and feel
- Not much to learn: back, forward, buttons, links, URLs

History and Bookmarks

Google

History is the first thing to go in most AJAX apps

Excruciating hidden <iframe> and/or timer tricks

Different solutions work best in each browser

Solve it once and reuse

- `History.addHistoryListener(myController);`

History support leads to bookmark support

- http://google.com/gulp.html#beta_carroty

Not All-or-Nothing

Google

GWT does not force you to start over!
Attach code to existing pages with a <meta> tag

```
<html>
…<meta name="gwt:module" content="…"/>
…<h1>Welcome to GWTravel Services</h1>
…<div id="reservationWizard">
…</html>
```

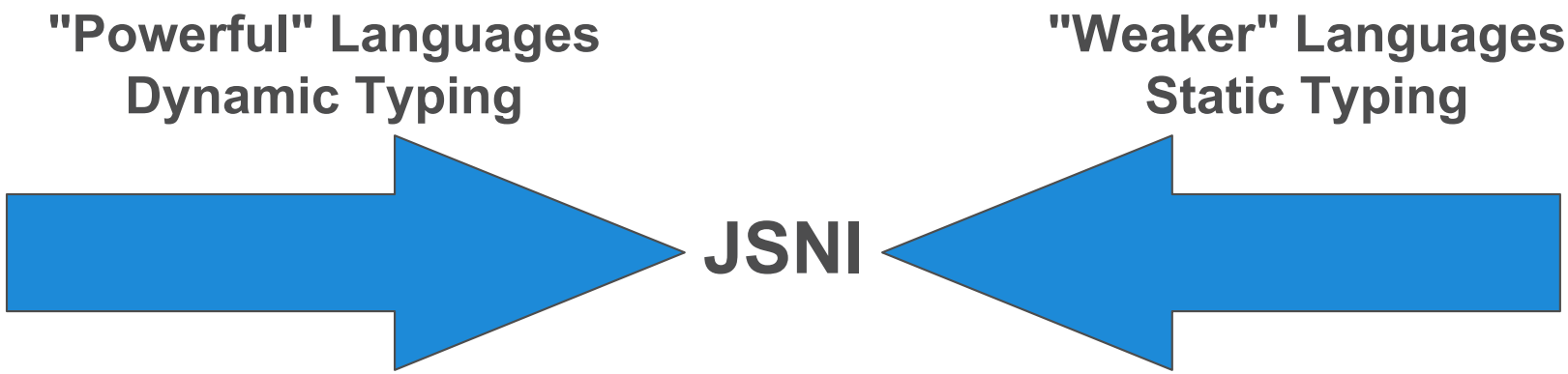
Your Java source is loosely-coupled

```
Panel p = RootPanel.get("reservationWizard");
Wizard wiz = new ReservationWizard();
p.add(wiz);
```

Works with any HTML-generating server approach

JavaScript Native Interface (JSNI)

Google



Include JavaScript directly in your Java source!

- And you can still debug
- Write no-compromise JS and make it reusable
- Expose existing JavaScript libraries into Java projects

Topics

Google

Ajax State of the Art?

The Google Web Toolkit

High-Performance Ajax

Integration with Existing Web Technologies

Parting Thoughts

Not Enough Time to Demo Everything

Google

I18N

- Properties file, Java Interface to use in your code
- `FeelingLuckyMessages.properties, public interface FeelingLuckyMessages extends Messages {`
- GWT Designer JUnit integration
- Works in hosted mode (i.e. debuggable Java)
- Work in web mode (i.e. tests the compiler output)

Automatic, dynamic dependency inclusion

- Slurp in external CSS
- Slurp in external JS

Wire-format libraries

- XML and JSON for integration with existing servers

Growing GWT Community

Google

Community and Support

- Over 6500 members on the developer forum and growing
- Books and articles
- Meta-sites (e.g. gwtPowered.org)

Libraries and Applications

- GWT Widgets on SourceForge
  43 projects on Google Code Project Hosting
- Diverse products built completely with GWT
- JetBrains' JET markup framework for GWT

Tools, Tools, Tools

- IntelliJ IDEA support for GWT built into Version 6.0
- Instantiations GWT Designer
- WindowBuilderPro GUI designer for GWT
- VistaFei for GWT
- Googlipse, an open source Eclipse plug-in for GWT

Summary

Google

Ajax is a lot of work…
Make sure to build a code base you're glad to own

Lots of rewards, lots of risks

Leverage is needed to use Ajax well with minimum risk

PhD in browser quirks is no longer an Ajax prereq

Turn Ajax hacking into software engineering

We will share our best work and ideas with you, and we hope you will return the favor

Much more to come… see you online!

Conclusion

Google

Many bridges, using different technologies

- No new SOAP API planned
- REST for data and message APIs: low and high REST
- Ajax to surface customizable controls and enable mashups
- Google Web Toolkit to build your own Ajax applications easily

Exposing many Google services… and more to come: give it a try!

- What: Ajax Search, Ajax Feed
- Where: Maps, KML
- When: Calendar
- Commerce: Checkout
- How: Google Web Toolkit

Credits

This presentation is a mashup of presentations by many Googlers and Google friends

Mark Lucovsky
Technical Director, Engineering, Ajax Search

Bret Taylor
Product Manager, Maps, GWT

Steffen Meschkat
Software Engineer, Maps

Frank Mantek
Software Engineer, GData

Thomas Steiner, tsteiner@google.com
Intern AdWords API

Ignacio Blanco, ignacio.blanco@globant.com
Globant, Checkout API Testing

Jacob Moon, jacobm@google.com
Checkout API support

Joel Webber
Software Engineer, Google Web Toolkit

Questions