Comet

Moving towards a real-time Web

Simon Willison
Highland Fling
3rd April 2008

Header image: http://svn.xantus.org/shortbus/trunk/cometd-perl/html/

Comet

~~Moving towards a real-time Web~~

Real developers fight browsers with their bare hands!

Simon Willison

Highland Fling

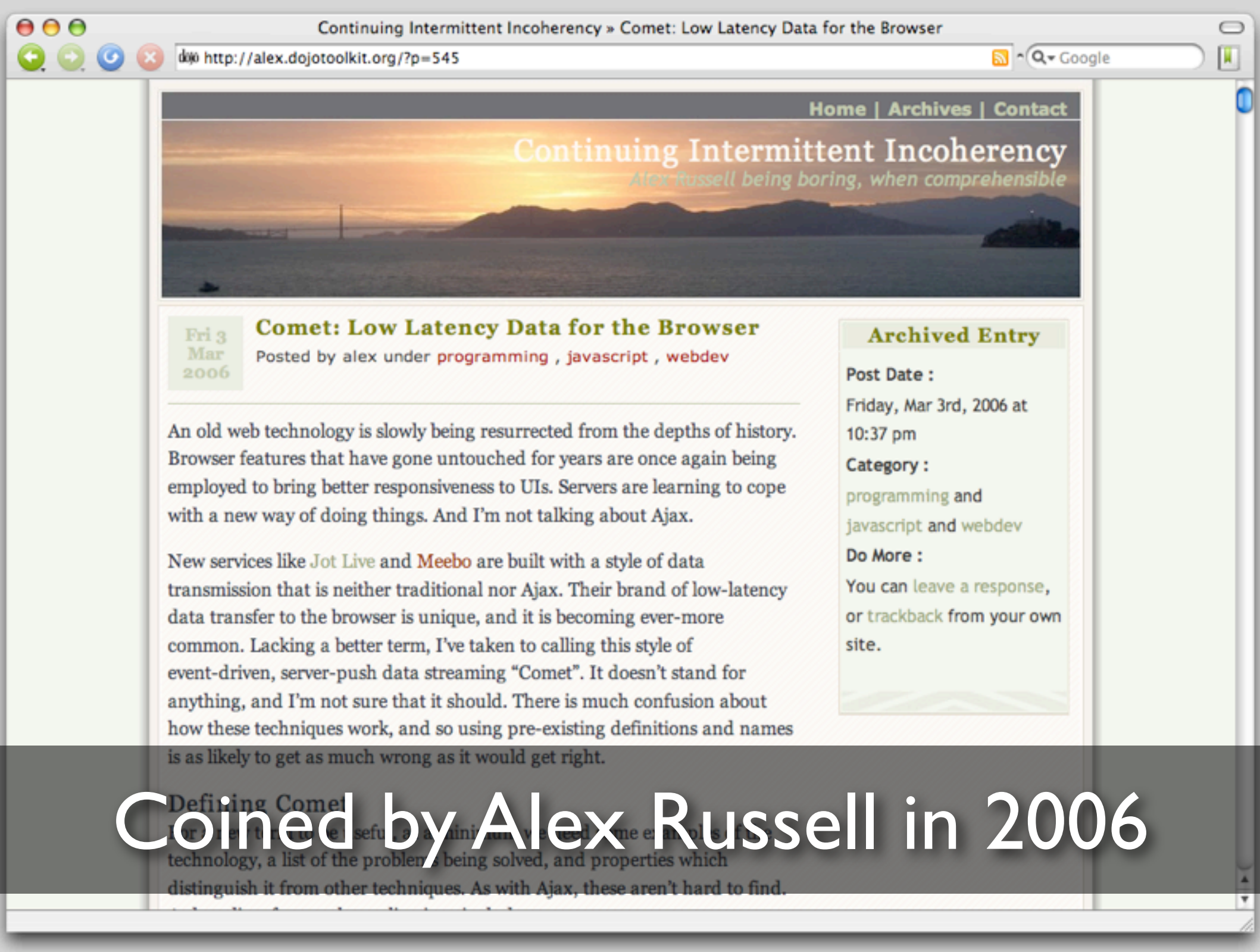
3rd April 2008

Header image: http://svn.xantus.org/shortbus/trunk/cometd-perl/html/

Comet: any technique that allows a web server to push events "live" to connected browsers

A new name for a very old technique

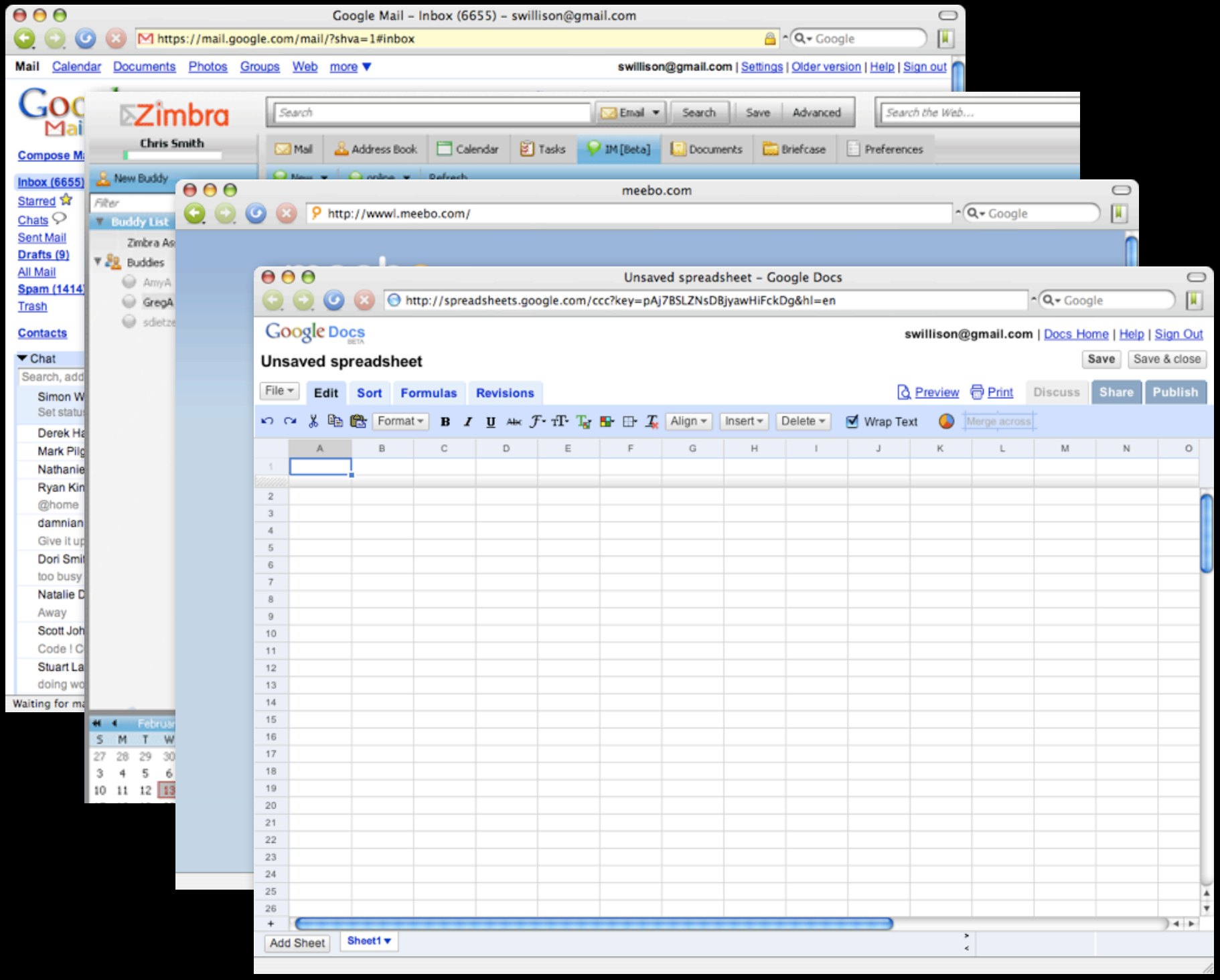
- Also known as...
  - HTTP push
  - Server push
  - Reverse Ajax
  - Pushlets

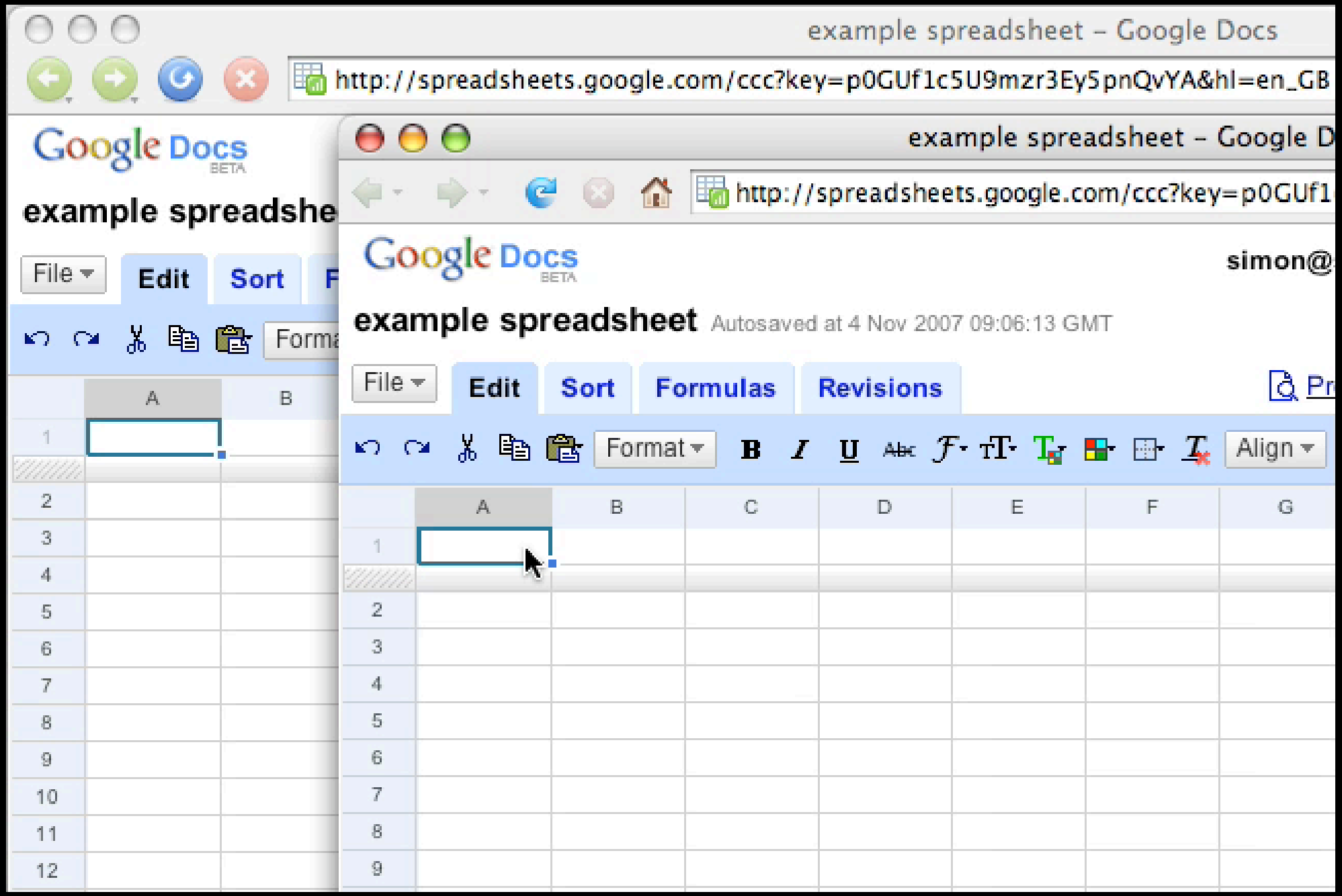


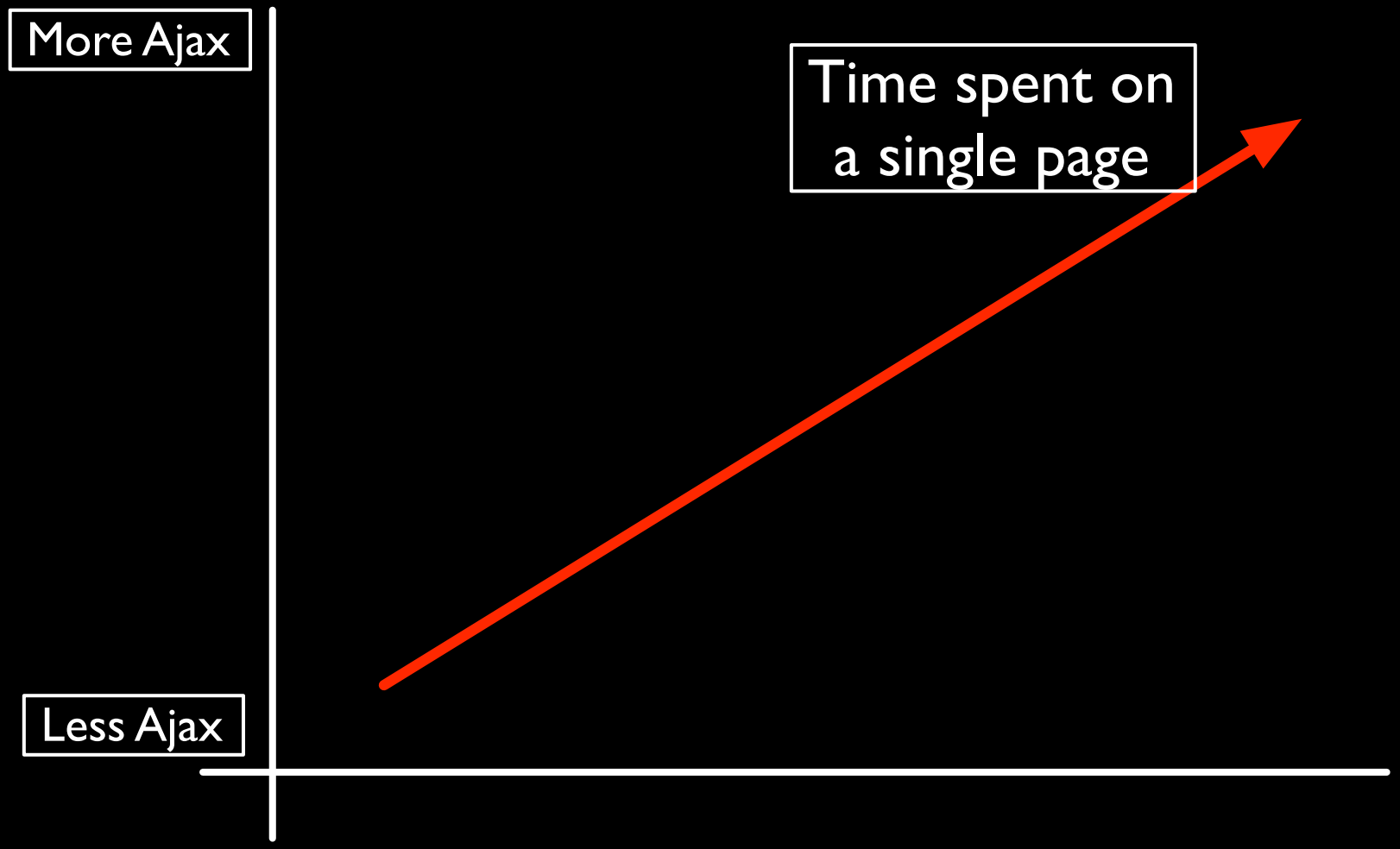
Coined by Alex Russell in 2006

Deliberately named after a kitchen cleaner

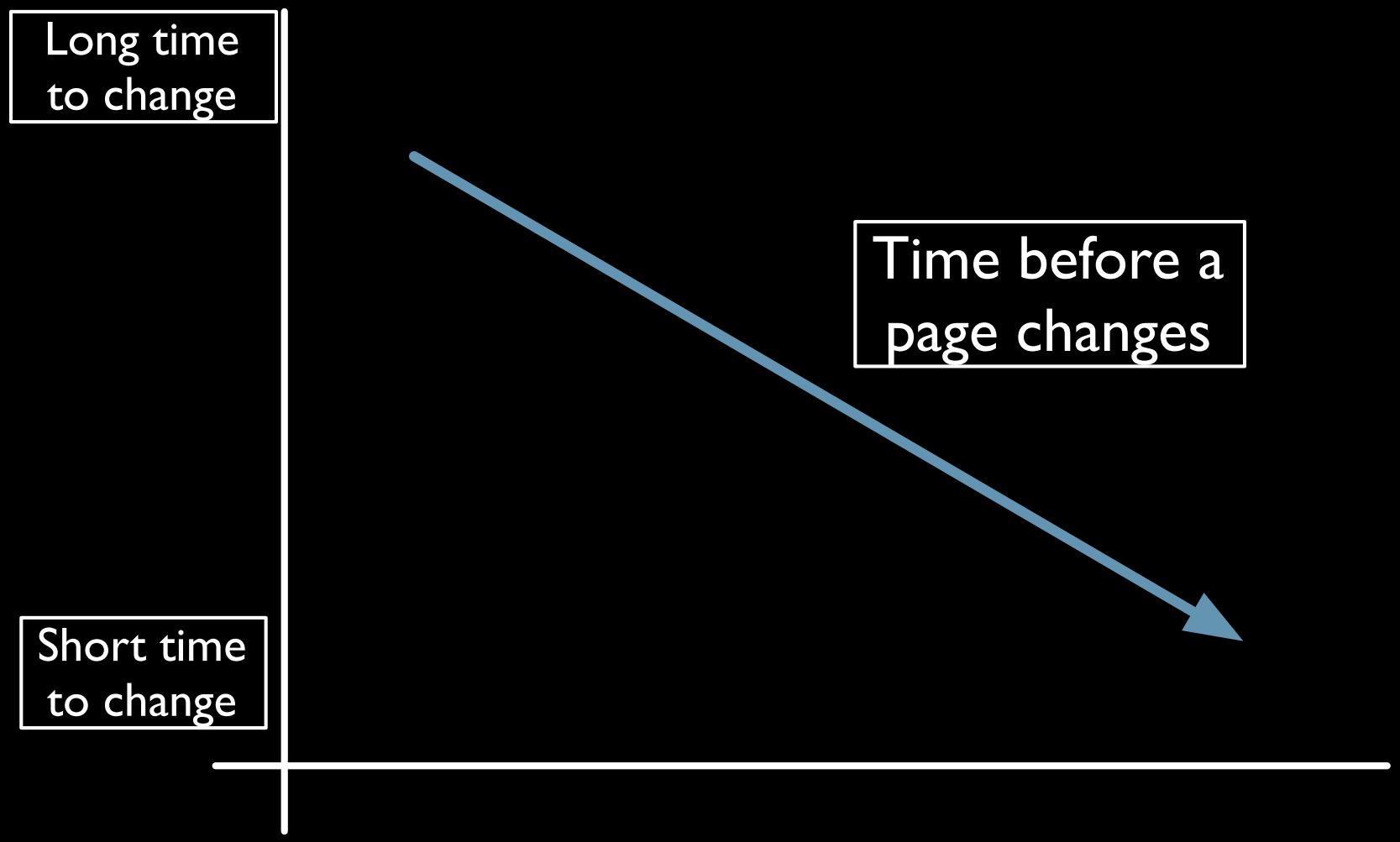
What can you do with it?


Google Mail - Inbox (6655) - swillison@gmail.com
https://mail.google.com/mail/?shva=1#inbox
Mail Calendar Documents Photos Groups Web more
swillison@gmail.com | Settings | Older version | Help | Sign out
Compose Mail
Inbox (6655)
Starred
Chats
Sent Mail
Drafts (9)
All Mail
Spam (1414)
Trash
Contacts
Chat
Zimbra
Chris Smith
Search
Email
Search
Save
Advanced
Search the Web...
Mail
Address Book
Calendar
Tasks
IM [Beta]
Documents
Briefcase
Preferences
New Buddy
Buddy List
Buddies
meebo.com
http://www1.meebo.com/
Unsaved spreadsheet - Google Docs
http://spreadsheets.google.com/ccc?key=pAj7BSLZNsDBjyawHiFckDg&hl=en
Google Docs
swillison@gmail.com | Docs Home | Help | Sign Out
Unsaved spreadsheet
Save
Save & close
File
Edit
Sort
Formulas
Revisions
Preview
Print
Discuss
Share
Publish
Format
Align
Insert
Delete
Wrap Text
Merge across
Add Sheet
Sheet1

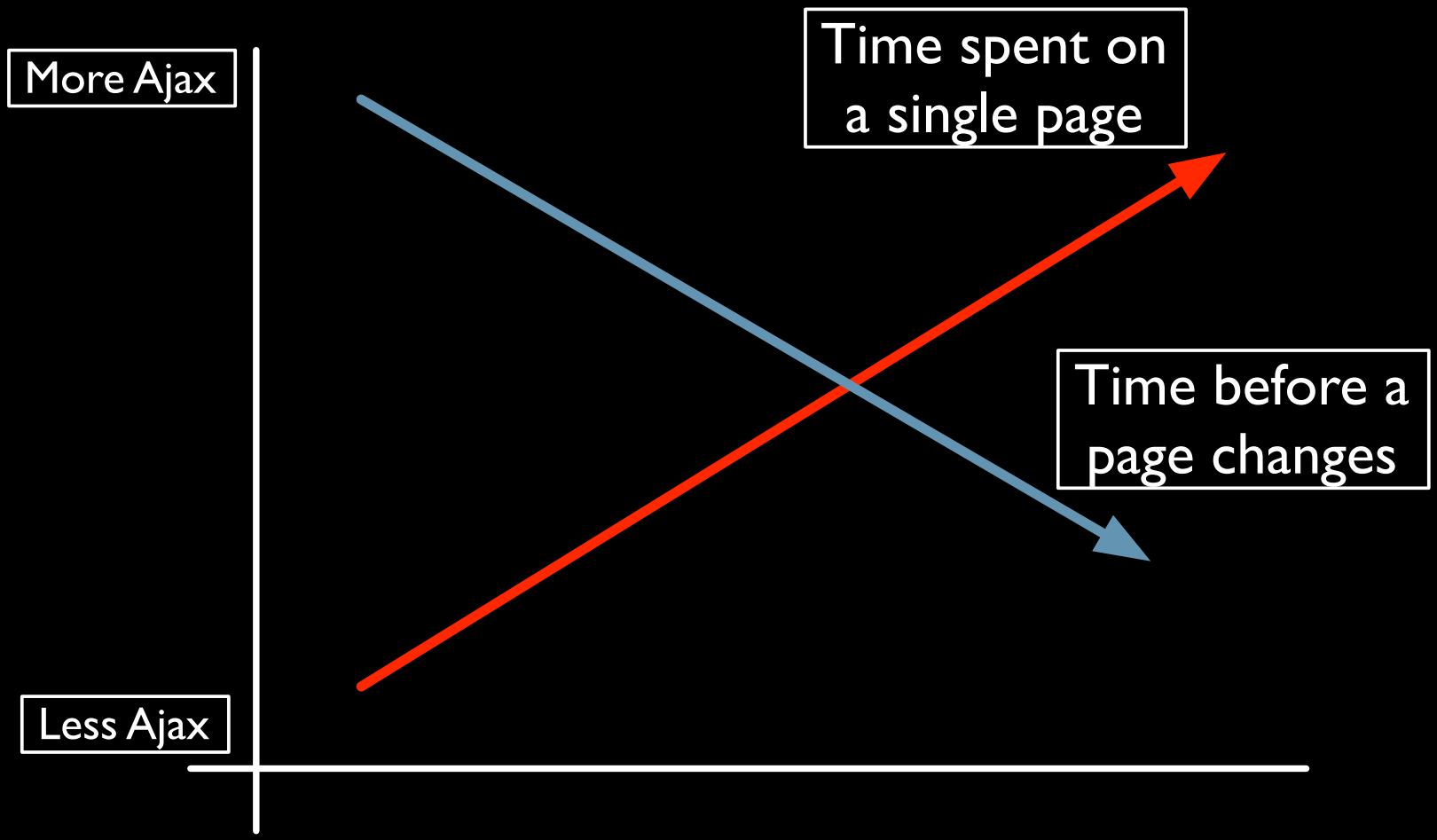

example spreadsheet - Google Docs
http://spreadsheets.google.com/ccc?key=p0GUf1c5U9mzr3Ey5pnQvYA&hl=en_GB
Google Docs BETA
example spreadshe
File
Edit
Sort
Forma
example spreadsheet - Google D
http://spreadsheets.google.com/ccc?key=p0GUf1
Google Docs BETA
simon@
example spreadsheet
Autosaved at 4 Nov 2007 09:06:13 GMT
File
Edit
Sort
Formulas
Revisions
Format
Align
A
B
C
D
E
F
G
1
2
3
4
5
6
7
8
9
10
11
12



(Slides courtesy of Joe Walker)

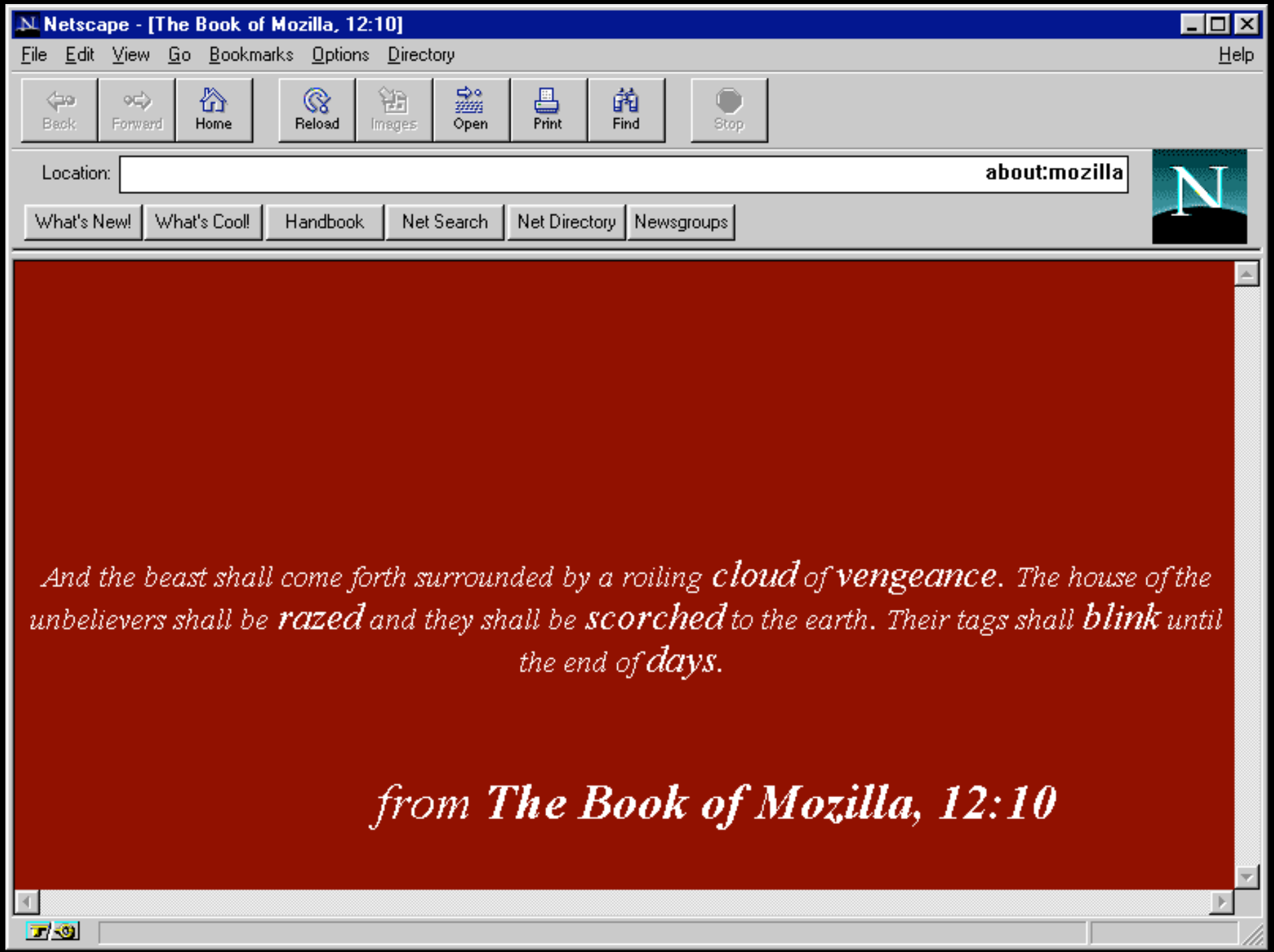


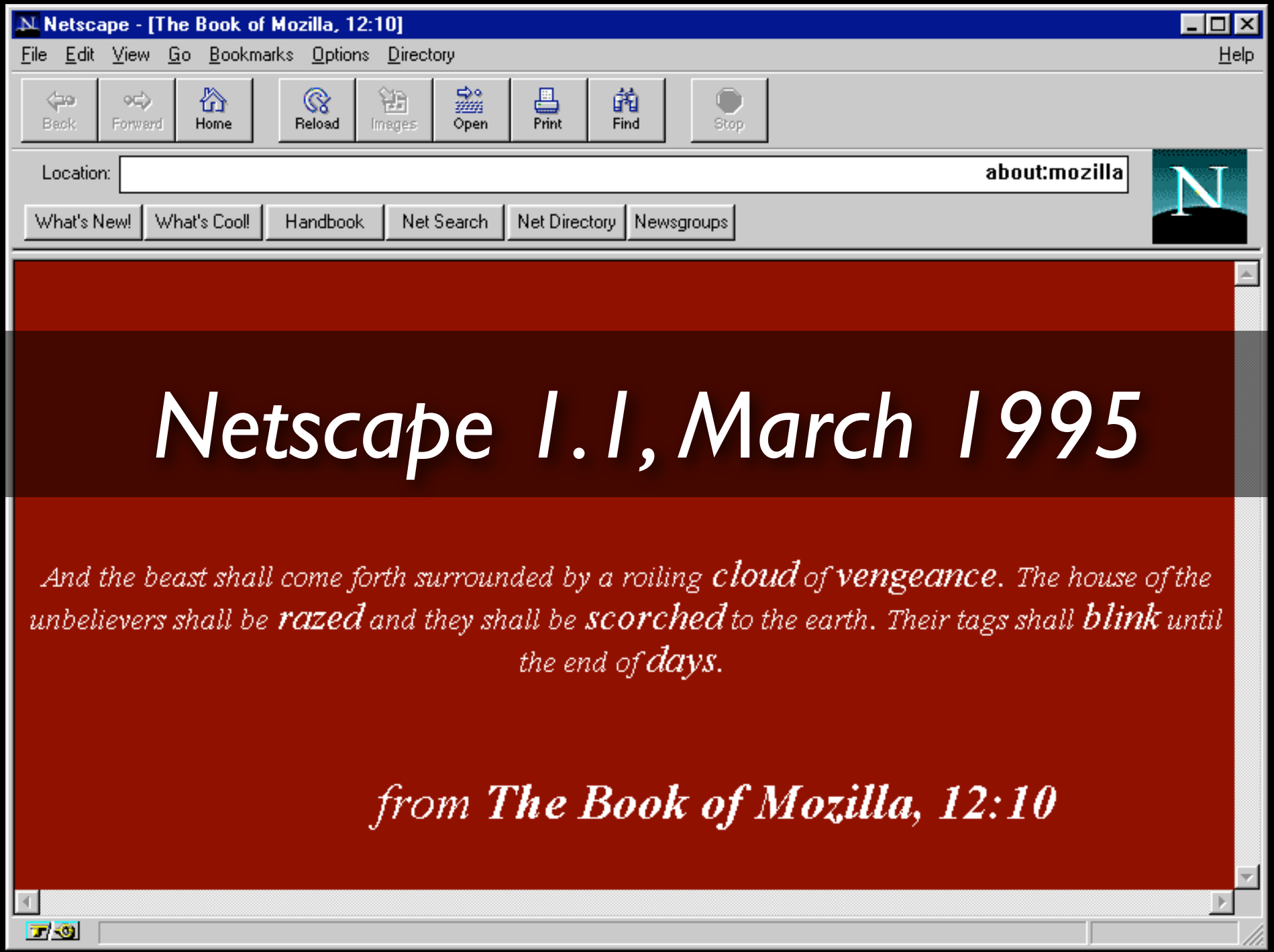
(Slides courtesy of Joe Walker)



(Slides courtesy of Joe Walker)

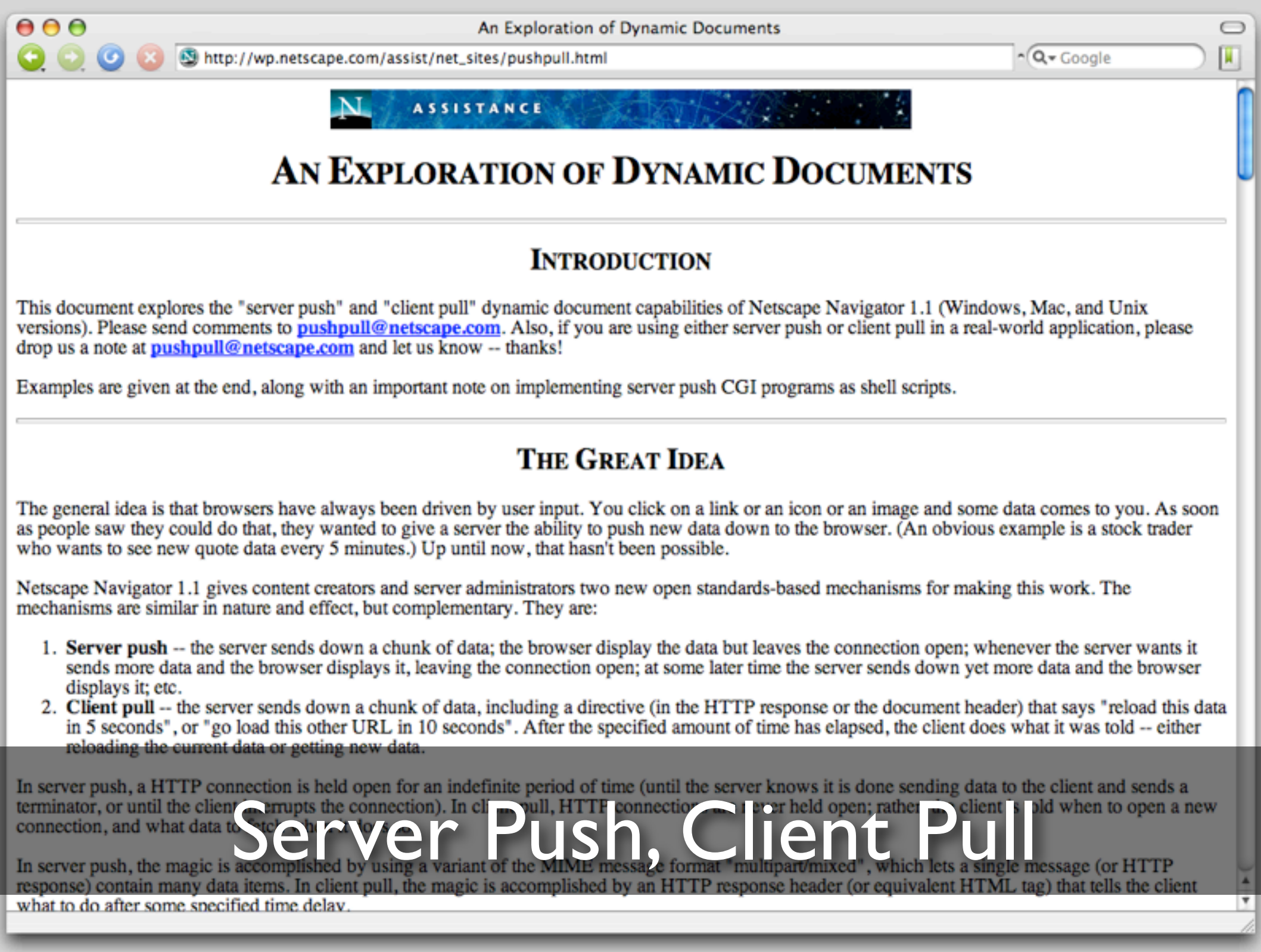
Early Comet


Netscape - [The Book of Mozilla, 12:10]
File Edit View Go Bookmarks Options Directory Help
Back Forward Home Reload Images Open Print Find Stop
Location: about:mozilla
What's New! What's Cool! Handbook Net Search Net Directory Newsgroups
And the beast shall come forth surrounded by a roiling cloud of vengeance. The house of the unbelievers shall be razed and they shall be scorched to the earth. Their tags shall blink until the end of days.
from The Book of Mozilla, 12:10


Netscape - [The Book of Mozilla, 12:10]
File Edit View Go Bookmarks Options Directory Help
Back Forward Home Reload Images Open Print Find Stop
Location: about:mozilla
What's New! What's Cool! Handbook Net Search Net Directory Newsgroups
Netscape 1.1, March 1995
And the beast shall come forth surrounded by a roiling cloud of vengeance. The house of the unbelievers shall be razed and they shall be scorched to the earth. Their tags shall blink until the end of days.
from The Book of Mozilla, 12:10

Netscape 1.1 new features

- Tables
- Background images
- “Dynamic Documents”
  - Client Pull
  - **Server Push**
- Predates JavaScript, Internet Explorer and Flash


An Exploration of Dynamic Documents
http://wp.netscape.com/assist/net_sites/pushpull.html
Google
ASSISTANCE
AN EXPLORATION OF DYNAMIC DOCUMENTS
INTRODUCTION
This document explores the "server push" and "client pull" dynamic document capabilities of Netscape Navigator 1.1 (Windows, Mac, and Unix versions). Please send comments to pushpull@netscape.com. Also, if you are using either server push or client pull in a real-world application, please drop us a note at pushpull@netscape.com and let us know -- thanks!
Examples are given at the end, along with an important note on implementing server push CGI programs as shell scripts.
THE GREAT IDEA
The general idea is that browsers have always been driven by user input. You click on a link or an icon or an image and some data comes to you. As soon as people saw they could do that, they wanted to give a server the ability to push new data down to the browser. (An obvious example is a stock trader who wants to see new quote data every 5 minutes.) Up until now, that hasn't been possible.
Netscape Navigator 1.1 gives content creators and server administrators two new open standards-based mechanisms for making this work. The mechanisms are similar in nature and effect, but complementary. They are:
1. Server push -- the server sends down a chunk of data; the browser display the data but leaves the connection open; whenever the server wants it sends more data and the browser displays it, leaving the connection open; at some later time the server sends down yet more data and the browser displays it; etc.
2. Client pull -- the server sends down a chunk of data, including a directive (in the HTTP response or the document header) that says "reload this data in 5 seconds", or "go load this other URL in 10 seconds". After the specified amount of time has elapsed, the client does what it was told -- either reloading the current data or getting new data.
In server push, a HTTP connection is held open for an indefinite period of time (until the server knows it is done sending data to the client and sends a terminator, or until the client interrupts the connection). In client pull, HTTP connections are never held open; rather, the client is told when to open a new connection, and what data to fetch when it does.
In server push, the magic is accomplished by using a variant of the MIME message format "multipart/mixed", which lets a single message (or HTTP response) contain many data items. In client pull, the magic is accomplished by an HTTP response header (or equivalent HTML tag) that tells the client what to do after some specified time delay.
Server Push, Client Pull

“Client Pull”

<META HTTP-EQUIV="Refresh" CONTENT=1>

“Server Push”

Content-type: **multipart/x-mixed-replace**;boundary=XXooXX

--XXooXX
Content-type: text/plain

Data for the first object.

--XXooXX
Content-type: text/plain

Data for the second and last object.

--XXooXX--

Adobe Resort

http://www.adoberesort.com/webcam.htm

Google

the ADOBE RESORT

• RESERVATIONS • CONTACT US • WEBCAM

▪ HOME ▪ PACKAGES ▪ DINING ▪ ROOMS & SUITES ▪ VIEW ROOMS ▪ ATTRACTIONS

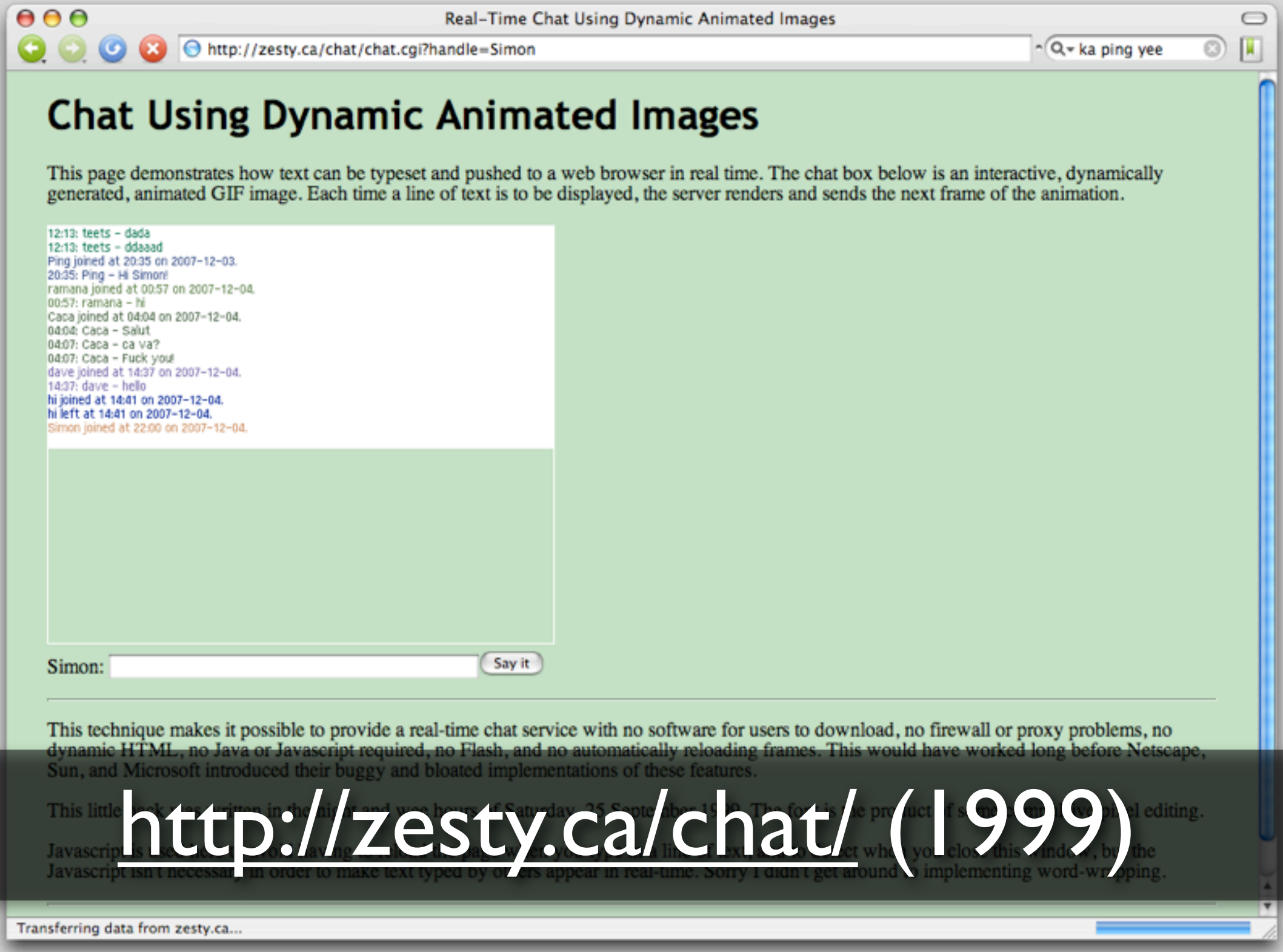
Web Cam

Adobe Resort
PO Box 219,
1555 Hwy 101,
Yachats, OR 97498.
Reservations:
1-800-522-3623
Or
(541) 547-3141
Fax: (541) 547-4234
or email us.

DIRECTIONS

Enter Address Here

TELL A FRIEND RESET


Real-Time Chat Using Dynamic Animated Images
http://zesty.ca/chat/chat.cgi?handle=Simon
ka ping yee
Chat Using Dynamic Animated Images
This page demonstrates how text can be typeset and pushed to a web browser in real time. The chat box below is an interactive, dynamically generated, animated GIF image. Each time a line of text is to be displayed, the server renders and sends the next frame of the animation.
12:13: teets - dada
12:13: teets - ddaaad
Ping joined at 20:35 on 2007-12-03.
20:35: Ping - Hi Simon!
ramana joined at 00:57 on 2007-12-04.
00:57: ramana - hi
Caca joined at 04:04 on 2007-12-04.
04:04: Caca - Salut
04:07: Caca - ca va?
04:07: Caca - Fuck you!
dave joined at 14:37 on 2007-12-04.
14:37: dave - hello
hi joined at 14:41 on 2007-12-04.
hi left at 14:41 on 2007-12-04.
Simon joined at 22:00 on 2007-12-04.
Simon:
Say it
This technique makes it possible to provide a real-time chat service with no software for users to download, no firewall or proxy problems, no dynamic HTML, no Java or Javascript required, no Flash, and no automatically reloading frames. This would have worked long before Netscape, Sun, and Microsoft introduced their buggy and bloated implementations of these features.
http://zesty.ca/chat/ (1999)
Transferring data from zesty.ca...

Modern Comet

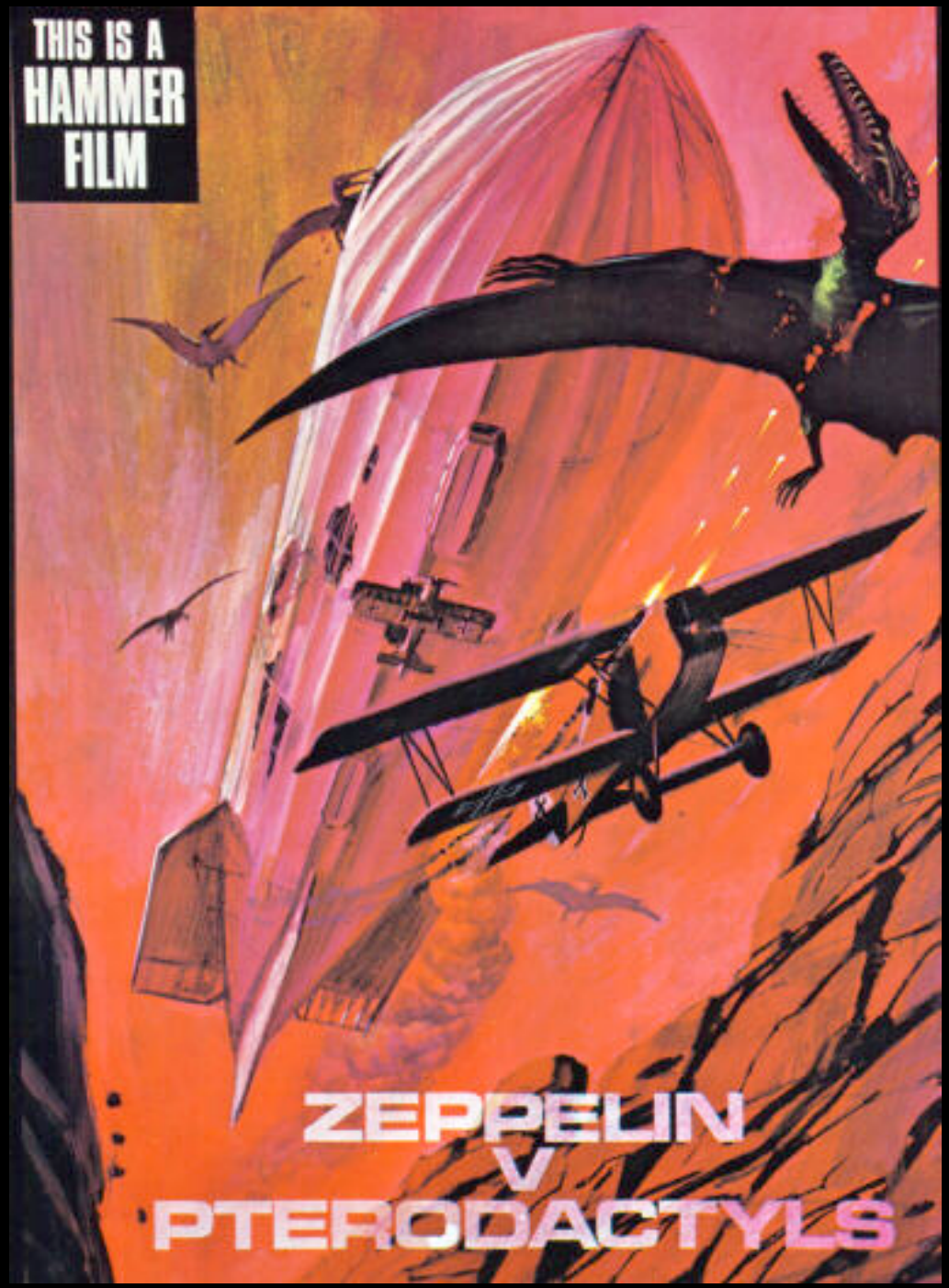
```
var xhr = new XMLHttpRequest();
xhr.open('GET', '/comet', true);

xhr.onreadystatechange = function() {
    if (readyState == 3) {
        // Process xhr.responseText
    }
};
```

If only it were
that simple...

An Epic Saga

Of geek v.s. browser



http://airminded.org/2007/05/25/the-movie-that-time-forgot/

Problem #1

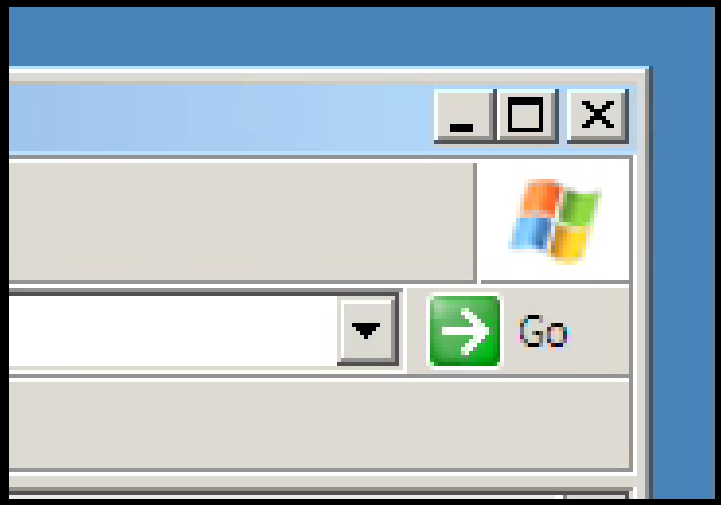
xhr.responseText isn't available until
the page has finished loading

A hidden iframe

(The forever frame technique)

- Add a hidden iframe pointing at a streaming page
- Send down chunks of <script> blocks, which the browser will execute as part of progressive rendering
- Add 1KB of junk at the start to trigger progressive rendering in IE

Problem #2



The throbber!

Problem #3

The click!

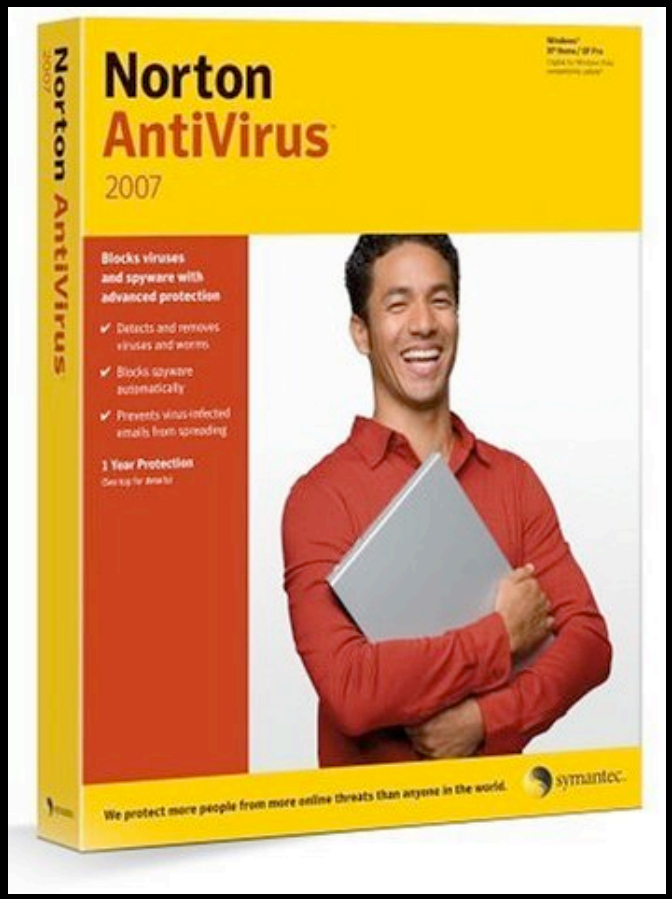
Solved for GChat

```
var htmlfile = new ActiveXObject('htmlfile');
```

- An undocumented ActiveX extension
  - Create an "htmlfile" instance
  - Create an iframe inside it
  - Set up some cross-frame scripting hooks
  - Run comet using the iframe
- No clicks! No throbber!

Confuse IE with enough nested iframes and it forgets to click

Problem #4



Proxies!

Long polling

- Make a request (iframe, XHR or even a <script> tag)
- Server pretends to be processing it really slowly...
- An event occurs; the server finally returns the response
- The browser starts a new request straight away

Problem #5

The two connection limit

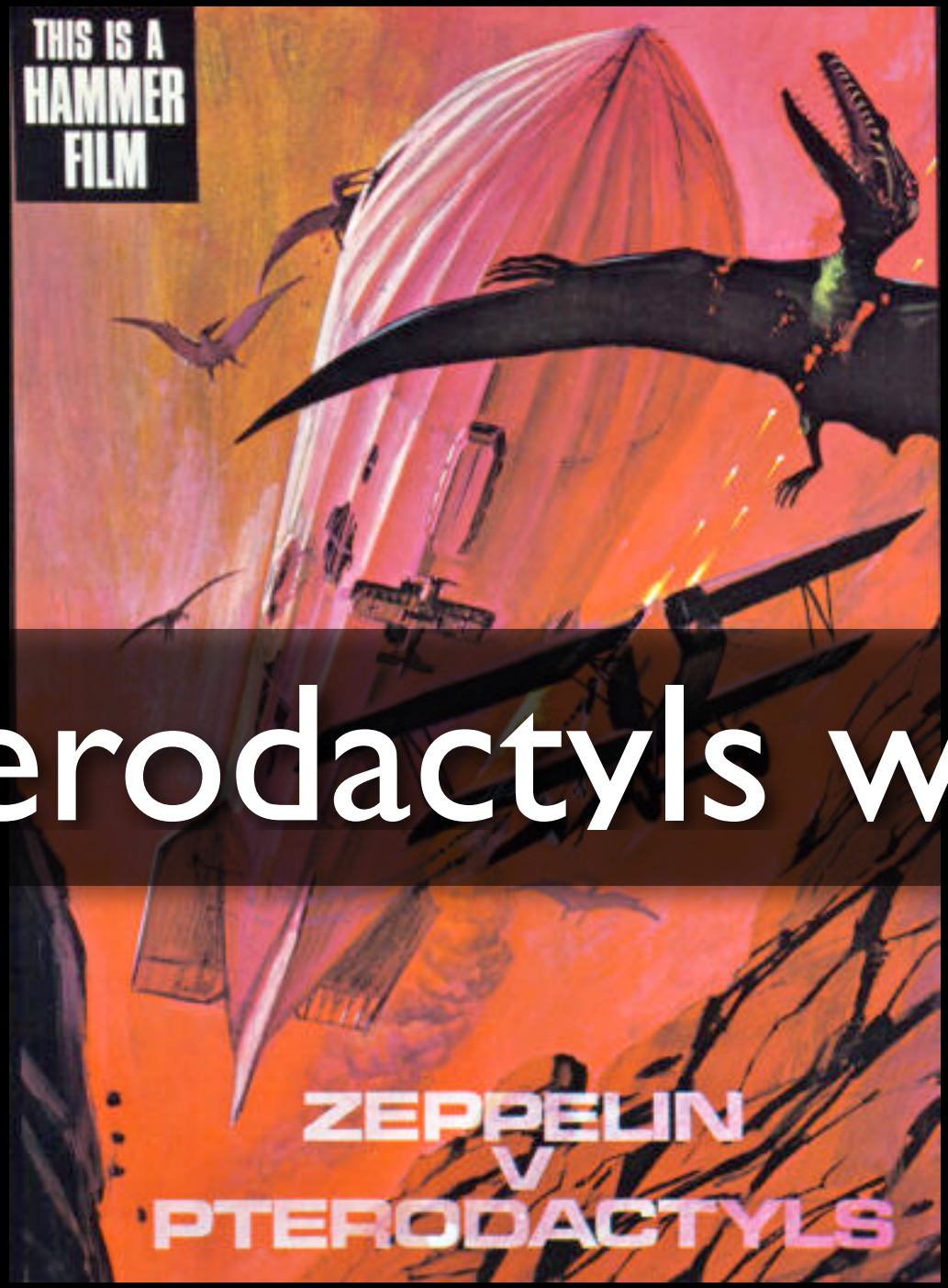
- mysite.com - regular site
- comet.mysite.com - comet server
- Cross-domain communication is easy - just add another iframe and set document.domain to enable cross-frame scripting

- mysite.com - regular site
- comet.mysite.com - comet server
- Cross-domain communication is easy - just add another iframe and set document.domain to enable cross-frame scripting
- (Then solve the resulting browser history problems with a few more iframes)

And if all else fails...

Fall back to polling

SoundAndFury http://www.flickr.com/photos/sound_and_fury/2166186492/



Pterodactyls win!

Key techniques

- Streaming (ideal)
  - XMLHttpRequest
  - Forever frame (for IE)
- Long polling (works through proxies)
- Polling (if all else fails)

Future Comet

HTML 5 event-source

http://www.whatwg.org/specs/web-apps/current-work/#event-source

```
<event-source src="/comet">
```

```
Content-Type: application/x-dom-event-stream

Event: server-time
data: [time on the server]
```

Support added in Opera 9.5


Google Gears
http://gears.google.com/
Google
Google Gears BETA
Improving Your Web Browser
Google Gears is an open source project that enables more powerful web applications, by adding new features to your web browser:
Let web applications interact naturally with your desktop
Store data locally in a fully-searchable database
Run JavaScript in the background to improve performance
Install Google Gears
Mac OS X
System requirements
Mac OS X 10.2+
Firefox 1.5+
Google Gears is available for Windows, Windows Mobile, Mac, and Linux
Developer site
Frequently asked questions
More help
Google Gears 0.2
©2008 Google - Gears Home - Privacy Policy
Done

“Wow, client-side
Comet sucks”...
that’s not the half of it

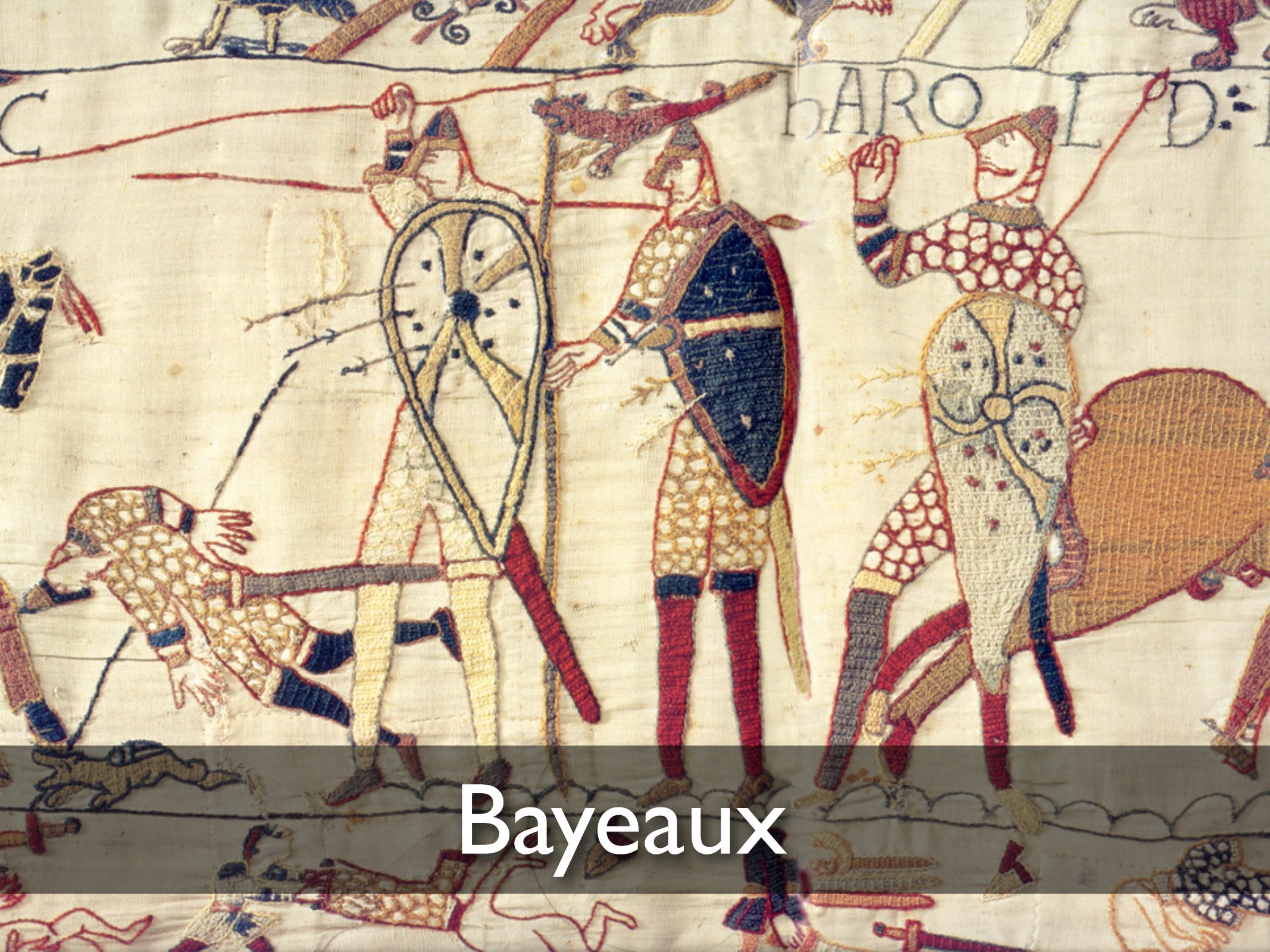
Scaling the server

http://flickr.com/photos/adrianblack/254968866/ craig1black

- Comet requires potentially tens of thousands of simultaneous HTTP connections
- Apache and other mainstream servers are designed to handle a single request as quickly as possible
  - Requests are generally served by a worker process or thread
- This absolutely will not scale to Comet

Event-based IO

- Alternative architecture designed to handle thousands of simultaneous connections
  - Twisted (Python)
  - Jetty (Java)
  - Perlbal
  - Orbited
  - Meteor ...



Bayeaux

Bayeaux

- Bayeaux is a *protocol* for Comet
- Any Bayeaux client can talk to any Bayeaux server
- The protocol is based on clients publishing and subscribing to channels
- Data is encoded using JSON
- Clients can create a permanent connection using a handshake, or send simple one-off messages

Dumb Comet servers

- The principle advantage of Bayeaux is that your Comet server can be **stupid -** it doesn't need any application logic, it just needs to route messages around
  - It's a black box which you simply drop in to your architecture
- This means your application logic can stay on your favourite platform

Cometd

- Cometd (Comet Daemon) is an umbrella project for a number of Bayeaux implementations
  - cometd-python (in Twisted)
  - cometd-perl (also a Perlbal plugin)
  - cometd-java (on Jetty)
  - dojox.cometd (JavaScript client)
- http://cometd.com/

So despite all of that...
Comet applications
are *easy* to build

How to build a Comet application (in 5 minutes)

Set up Jetty

- Download and unzip Jetty-6.1 from www.mortbay.org
- Install Maven from http://maven.apache.org/
- cd jetty-6.1.6/contrib/cometd/demo/
- mvn jetty:run
- Browse to http://localhost:8080/
- Mess around with the JS in src/main/webapp/examples/

dojox.cometd

```
dojo.addOnLoad(function() {
    dojox.cometd.init("http://127.0.0.1:8080/cometd");
    dojox.cometd.subscribe("/mychannel", function(comet) {
        alert(comet.data); // a JSON object
    });









});
```

dojox.cometd

```
dojo.addOnLoad(function() {
    dojox.cometd.init("http://127.0.0.1:8080/cometd");
    dojox.cometd.subscribe("/mychannel", function(comet) {
        alert(comet.data); // a JSON object
    });

    dojo.byId('send').onclick = function() {
        dojox.cometd.publish("/mychannel", {
            'msg': "A message sent by Comet"
        });
        return false;
    };
});
```

A slideshow in 5 lines of code

```
<script type="text/javascript">
dojo.require("dojox.cometd");
jQuery(function($) {
    dojox.cometd.init("http://127.0.0.1:8080/cometd");
    dojox.cometd.subscribe("/slideshow/change", function(comet) {
        $('#currentSlide').attr('src', comet.data.src);
    });
});
</script>
```

Conclusions

- Comet requires hacks, but they're reasonably well understood
- Bayeaux and Cometd abstract away the nastiness and make Comet accessible to sane developers
- If we survived CSS, Comet should be a cinch

More crazy hacks...

- http://meteorserver.org/browser-techniques/
- http://cometdaily.com/2007/10/25/http-streaming-and-internet-explorer/
- http://cometdaily.com/2007/11/18/ie-activexhtmlfile-transport-part-ii/
- http://orbited.org/svn/orbit/trunk/daemon/orbited/static/orbited.js
- **http://simonwillison.net/tags/comet/**

Thank you