The Web, revisited

WEB 2.0

marco.ronchetti@unitn.it

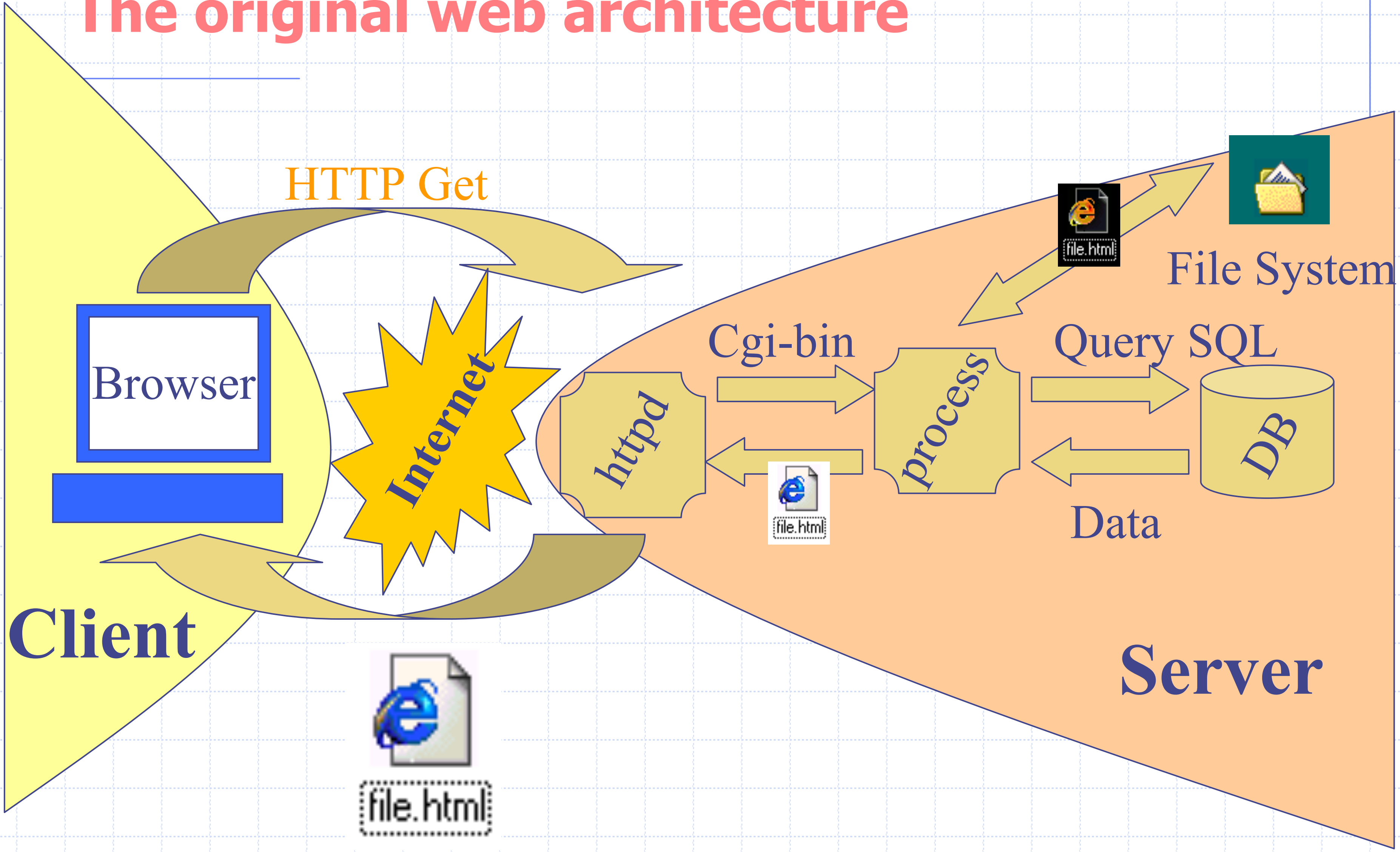
Credits: Some of the slides are based on material adapted from www.telerik.com/documents/Telerik_and_AJAX.pdf

The old web: 1994

HTML pages (hyperlinks)
+ static graphics
  + cgi (active engines)
    + some separated dynamic graphics (Applets)

HTTP is a stateless protocol: cookies

The original web architecture



Evolution of the web: 1

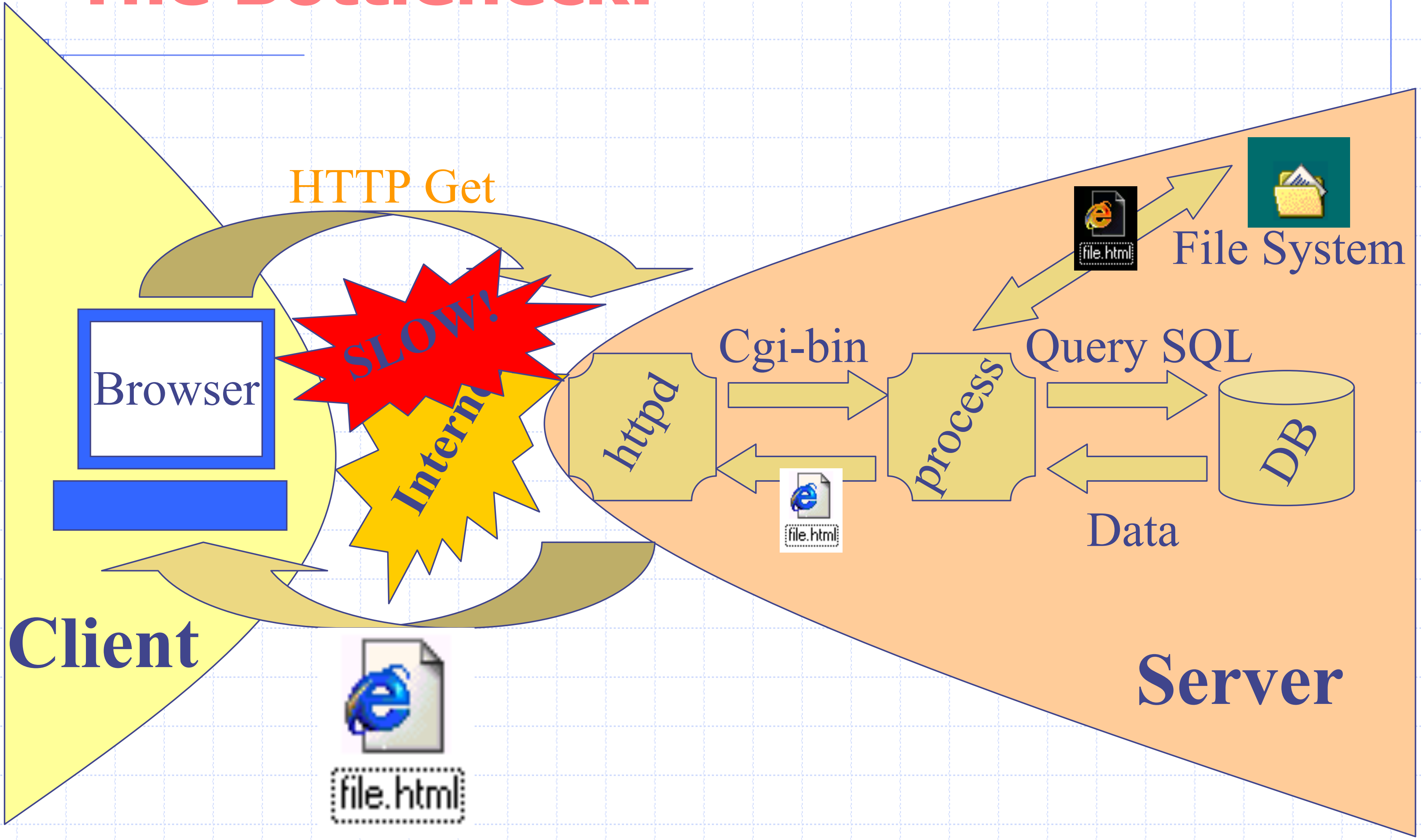
Better dynamic engines

Servlets, ASP, JSP (+ Php, Perl, Python...)

Better Server-side organization

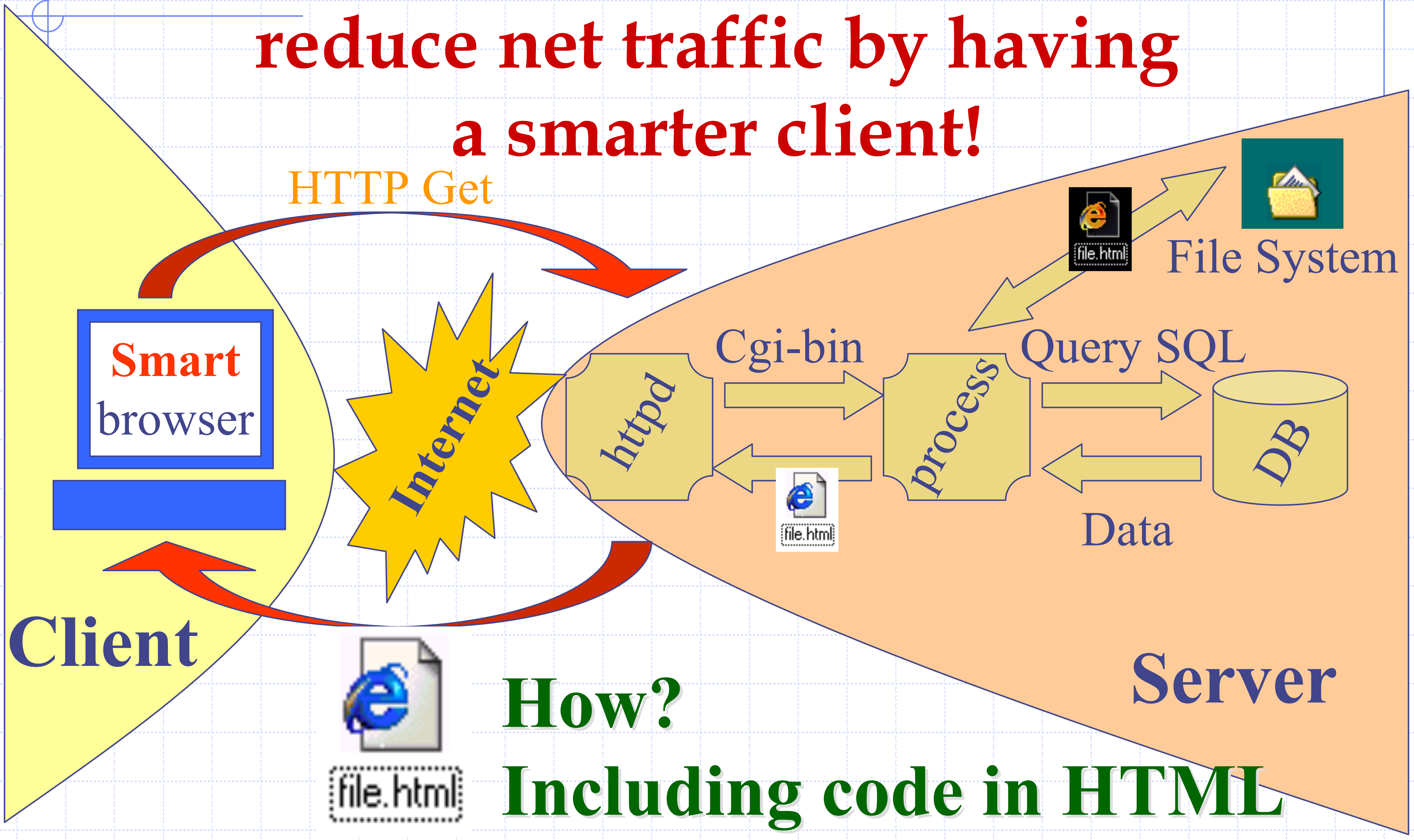
EJB, frameworks (Struts, Hybernate, Spring)

The Bottleneck!



The solution:

reduce net traffic by having a smarter client!



How?

Including code in HTML

Evolution of the web: 2

Better control of the browser

Javascript + DOM

Applet-Javascript interaction

Better separation of content and presentation

- CSS (DHTML=HTML4+Javascript+DOM+CSS)
- XML+XSLT, Cocoon (XHTML)

Evolution of the web: 3

Better construction of interfaces (widgets)

- .Net
- Java Server Faces

Are we there?

BUT ALL THIS

IS NOT YET

ENOUGH!

Key disadvantages of web apps

Poor Interactivity

users must wait for full page reloads after each interaction with the server.

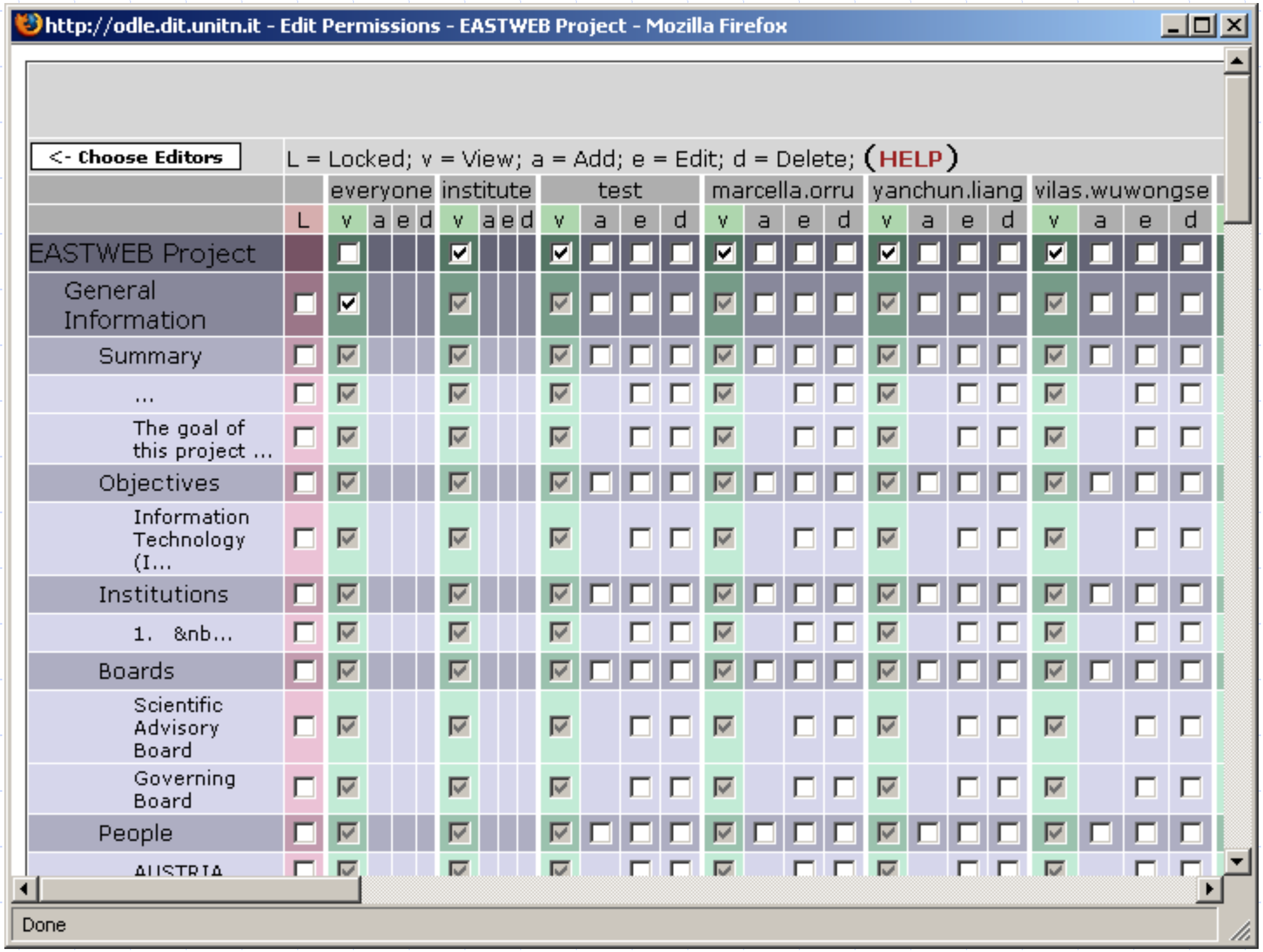
Unresponsiveness

classic web applications transfer the complete form data to the server, which in turn renders and sends back the full HTML markup of the page to the browser. Lots of bandwidth is consumed and the performance is significantly hindered. Even worse, the user will often find the page has scrolled to a different position, causing disorientation.

Simplistic Interfaces

the requirement for full page postback whenever the user interface has to be changed imposes hefty limitations on the degree of sophistication of web user interfaces. Rich and smooth interfaces with on-demand update could only be implemented using host technologies (Flash – Applets).

The form nightmare…


http://odle.dit.unitn.it - Edit Permissions - EASTWEB Project - Mozilla Firefox

<- Choose Editors

L = Locked; v = View; a = Add; e = Edit; d = Delete; (HELP)

		everyone				institute				test				marcella.orru				yanchun.liang				vilas.wuwongse			
	L	v	a	e	d	v	a	e	d	v	a	e	d	v	a	e	d	v	a	e	d	v	a	e	d
EASTWEB Project		☐				✓				✓	☐	☐	☐	✓	☐	☐	☐	✓	☐	☐	☐	✓	☐	☐	☐
General Information	☐	✓				☑				☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐
Summary	☐	☑				☑				☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐
…	☐	☑				☑				☑		☐	☐	☑		☐	☐	☑		☐	☐	☑		☐	☐
The goal of this project …	☐	☑				☑				☑		☐	☐	☑		☐	☐	☑		☐	☐	☑		☐	☐
Objectives	☐	☑				☑				☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐
Information Technology (I…	☐	☑				☑				☑		☐	☐	☑		☐	☐	☑		☐	☐	☑		☐	☐
Institutions	☐	☑				☑				☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐
1. &nb…	☐	☑				☑				☑		☐	☐	☑		☐	☐	☑		☐	☐	☑		☐	☐
Boards	☐	☑				☑				☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐
Scientific Advisory Board	☐	☑				☑				☑		☐	☐	☑		☐	☐	☑		☐	☐	☑		☐	☐
Governing Board	☐	☑				☑				☑		☐	☐	☑		☐	☐	☑		☐	☐	☑		☐	☐
People	☐	☑				☑				☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐	☑	☐	☐	☐
AUSTRIA	☐	☑				☑				☑		☐	☐	☑		☐	☐	☑		☐	☐	☑		☐	☐

Done

Evolution of the web: 4

Embedding specialized, non web objects (plug ins)

- Applets
- Active-X
- Quicktime
- RealPlayer
- Flash

The rise of the new web

"The Web as we know it is changing probably more than it has since the first graphic showed up… The idea of the webpage itself is nearing its useful end. With the way Ajax allows you to build nearly stateless applications that happen to be web accessible, everything changes."

Jesse James Garrett, February 15, 2005

http://www.adaptivepath.com/publications/essays/archives/000385.php

The new web - Examples

http://demo.script.aculo.us/

http://demos.openrico.org/

Ajax !

new development technique

- will blur the line between web-based and desktop applications.
- rich, highly responsive and interactive interfaces
- Acronym stands for “Asynchronous JavaScript + XML”.

How does Ajax work?

The core idea behind AJAX is to make the communication with the server asynchronous, so that data is transferred and processed in the background.

As a result the user can continue working on the other parts of the page without interruption.

In an AJAX-enabled application only the relevant page elements are updated, only when this is necessary.

The hart of Ajax

not actually a brand new technology!

- First used after Microsoft implemented Microsoft *XMLHTTP* COM object that was part of The Microsoft® XML Parser (IE 5.1)
- Similarly supported by a Mozilla Javascript object *XMLHttpRequest (Mozilla 1.0, Firefox, Safari 1.2 etc.)*
- Massively used by Google

Other labels for the same technology were Load on Demand, Asynchronous Requests, Callbacks, Out-of-band Calls, etc.

Ajax code

```
if (window.XMLHttpRequest) { // Mozilla, Safari,
  ...
      http_request = new XMLHttpRequest();
  } else if (window.ActiveXObject) { // IE
      http_request = new
  ActiveXObject("Microsoft.XMLHTTP");
  }
```

A more complete example? See e.g.
http://www.onlamp.com/pub/a/onlamp/2005/05/19/xmlhttprequest.html

Ajax is more than that

* dynamic presentation based on XHTML + CSS;

* dynamic display and interaction using Document Object Model;

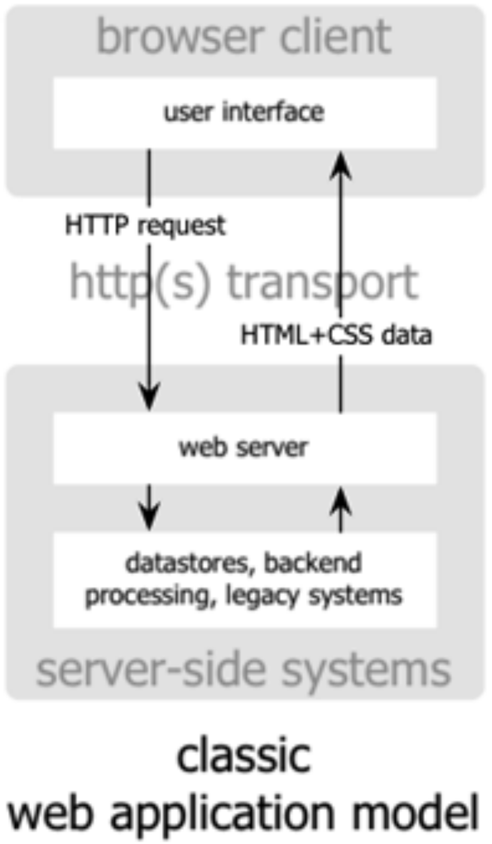
* data exchange and manipulation using XML e XSLT;

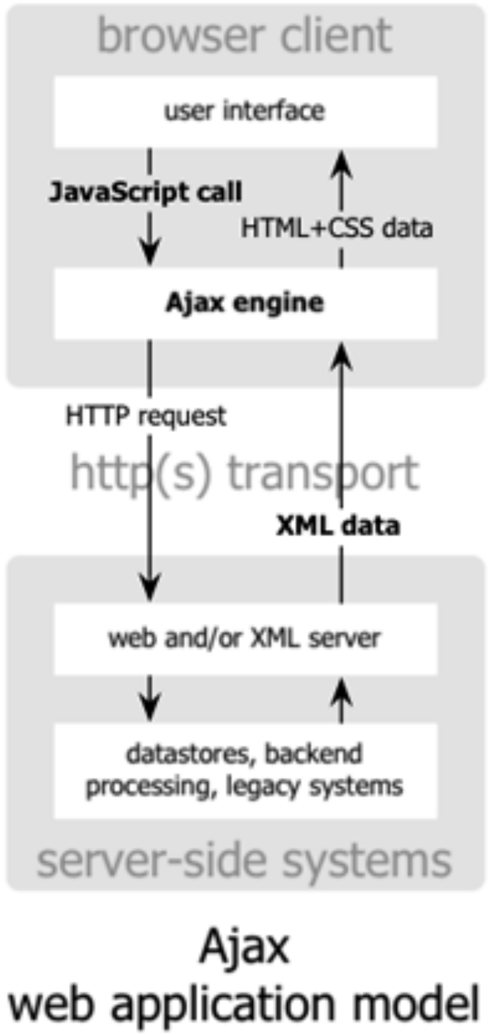
* asynchrounous data fetching using XMLHttpRequest;

* JavaScript as glue.

The paradigms

1.0



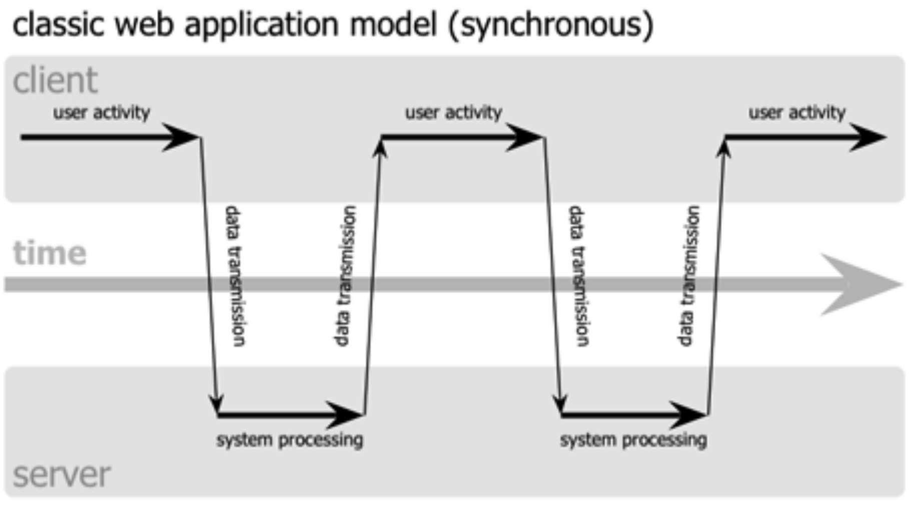


2.0

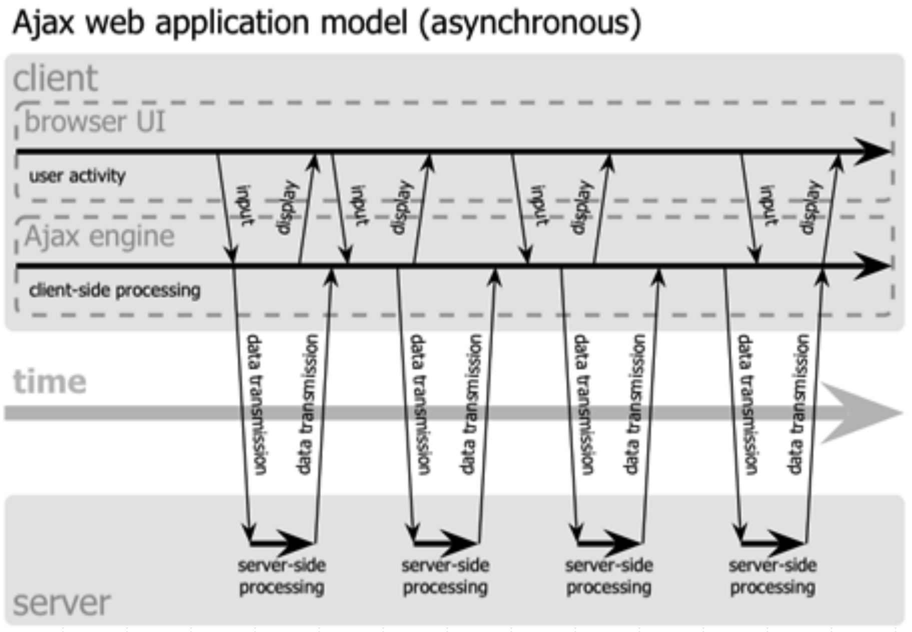
Pictures after Jesse James Garrett

The models

1.0



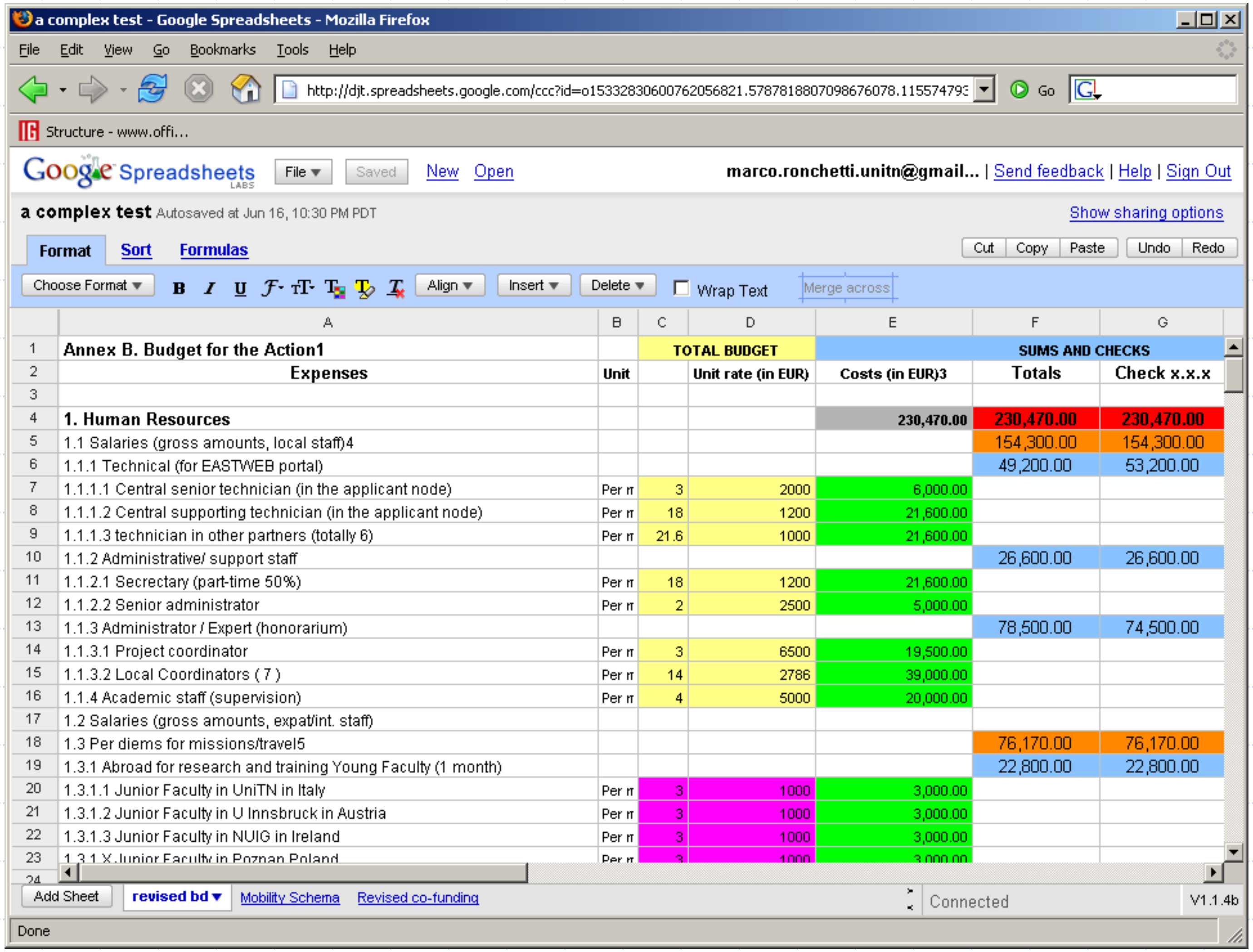
2.0



Pictures after

Jesse James Garrett

The (impressive!) result - RIA



a complex test Autosaved at Jun 16, 10:30 PM PDT

	A	B	C	D	E	F	G
1	**Annex B. Budget for the Action1**		**TOTAL BUDGET**		**SUMS AND CHECKS**		
2	**Expenses**	**Unit**		**Unit rate (in EUR)**	**Costs (in EUR)3**	**Totals**	**Check x.x.x**
3							
4	**1. Human Resources**				**230,470.00**	230,470.00	230,470.00
5	1.1 Salaries (gross amounts, local staff)4					154,300.00	154,300.00
6	1.1.1 Technical (for EASTWEB portal)					49,200.00	53,200.00
7	1.1.1.1 Central senior technician (in the applicant node)	Per m	3	2000	6,000.00		
8	1.1.1.2 Central supporting technician (in the applicant node)	Per m	18	1200	21,600.00		
9	1.1.1.3 technician in other partners (totally 6)	Per m	21.6	1000	21,600.00		
10	1.1.2 Administrative/ support staff					26,600.00	26,600.00
11	1.1.2.1 Secrectary (part-time 50%)	Per m	18	1200	21,600.00		
12	1.1.2.2 Senior administrator	Per m	2	2500	5,000.00		
13	1.1.3 Administrator / Expert (honorarium)					78,500.00	74,500.00
14	1.1.3.1 Project coordinator	Per m	3	6500	19,500.00		
15	1.1.3.2 Local Coordinators (7)	Per m	14	2786	39,000.00		
16	1.1.4 Academic staff (supervision)	Per m	4	5000	20,000.00		
17	1.2 Salaries (gross amounts, expat/int. staff)						
18	1.3 Per diems for missions/travel5					76,170.00	76,170.00
19	1.3.1 Abroad for research and training Young Faculty (1 month)					22,800.00	22,800.00
20	1.3.1.1 Junior Faculty in UniTN in Italy	Per m	3	1000	3,000.00		
21	1.3.1.2 Junior Faculty in U Innsbruck in Austria	Per m	3	1000	3,000.00		
22	1.3.1.3 Junior Faculty in NUIG in Ireland	Per m	3	1000	3,000.00		
23	1.3.1.X Junior Faculty in Poznan Poland	Per m	3	1000	3,000.00		
24							

Ajax - advantages

- Rich applications in browsers
- No issues with installation
- Built on existing infrastructure (TCP/IP, SSL, HTTP, XML…)

Ajax - advantages

- Better Performance and Efficiency
  small amount of data transferred from the server. Beneficial for data-intensive applications as well as for low-bandwidth networks.
- More Responsive Interfaces
  the improved performance give the feeling that updates are happening instantly. AJAX web applications appear to behave much like their desktop counterparts.
- Reduced or Eliminated "Waiting" Time
  only the relevant page elements are updates, with the rest of the page remaining unchanged. This decreases the idle waiting time.
- Increased Usability
- Users Can Work with the Rest of the Page while data is being transferred in the background.

Applicability Scenarios

- Highly interactive applications
  Google Spreadsheet

- Data visualization – visualizing large datasets
  Google Maps

- Data input & validation
  it's possible to validate the data the user enters, while they are entering it. They can then receive feedback (using the server's intelligence) without the page being posted back.

- Active widgets

And make sure that you…

- Preserve the Normal Page Lifecycle – as much as possible!
- Reflect Control State on the Server – in real-life scenarios there is no use of simply rendering controls on the page.
- Support Cross-Browser usage – there are different implementation of the XmlHttpRequest object. You should make sure that all AJAX components you choose operate properly on various browsers and platforms.
- Ensure proper Operation when Cookies are Disabled – support cookieless sessions.

And make sure that you…

- Give visual feedback - When a user clicks on something in the AJAX user interface, they need immediate visual feedback
- Keep the Back button – make sure that the Back button in your application functions on every page of the site.
- Use links for navigation – avoid the temptation to use links as an interface on your AJAX application to change the state of your application. Users have been trained over ten years to expect a link to “take” them somewhere, so give them what they expect.
- Limit the scope of visual changes – when an AJAX call results in one or more changes to what the user sees, keep the changes local to the place where the user would expect them to be.
- Use human-readable links – people like to pass the addresses of useful web pages to each other. Make sure your application supports URLs that people can share easily, so not too long or complex.

Adapted from: www.telerik.com/documents/Telerik_and_AJAX.pdf

And make sure that you…

- Don’t bloat the code – make sure that your application uses as little client-side scripting as possible. This reduces download time for the page and also reduces the processor requirements on the client browser, so results in a faster browser experience.
- Follow UI conventions – AJAX is a world of possibilities, but when it comes to user interface the best is invariably the familiar. If you’re creating a user interface for a specific feature, the place to start is by replicating an existing successful interface and looking at what your clients expect. Also remember that although it may be cool to implement drag-and-drop, few people may realize the interface relies on it.
- Don’t scroll – users like to feel in control, so if they have moved the scrollbar to a specific place, don’t move the page somewhere else.
- Reduce page loads – do as much as you can to reduce the number of page loads the user has to do to achieve their goal.

Adapted from: www.telerik.com/documents/Telerik_and_AJAX.pdf

Warning: Ajax has drawbacks!

- Accessibility

  the AJAX development technique fundamentally violates the requirements for accessibility.

  Since the page content is being updated dynamically, the changes may not be detected by accessibility tools like screen readers.

  Some accessibility standards prohibit the use of JavaScript altogether…

- New UI Interactivity Requires Learning

  the UI richness of AJAX-enabled application presents users with new and unexpected functionality.

  this may require some learning!

But Ajax is hard!

- Extensive use of Javascript
  - requires substantial JavaScript skills
  - lack of good debugging tools for client-side script
  - it is like debugging multithreaded Javascript!
- Breaks normal page lifecycle

  AJAX requires a different way of thinking about a web-site, since the concept of a "Page" is no longer valid. In fact, AJAX applications may be considered as closer to the desktop-applications development approach.

  The fact that a Page no longer holds constant data leads to two important consequences – the Back button and bookmarking will no longer work as expected.
- Every browser has its flavour!

How to solve the problem?

There are many proposed libraries/frameworks

Survey of AJAX/JavaScript Libraries

http://chandlerproject.org/bin/view/Projects/AjaxLibraries

WARNING: Third Party Controls with Complex JavaScript (e.g. powerful datagrids, treeviews, WYSIWYG editors, etc.) may be damaged by universal AJAX wrappers/containers!

EXT-JS

http://www.sencha.com/products/js/

Ext JS is a cross-browser JavaScript library for building rich internet applications. Build rich, sustainable web applications faster than ever. It includes:

* High performance, customizable UI widgets
* Well designed and extensible Component model
* An intuitive, easy to use API

Google’s Web Toolkit

http://code.google.com/intl/it-IT/webtoolkit/

List of the widgets:

http://code.google.com/intl/it-IT/docreader/#p=google-web-toolkit-doc-1-5&s=google-web-toolkit-doc-1-5&t=DevGuideWidgetGallery

Demo of the widgets:

http://gwt.google.com/samples/Showcase/Showcase.html

Google's Web Toolkit

The idea: write java and transform it into Javascript

Debug in "hosted mode"

If your GWT application compiles and runs in hosted mode as you expect

And GWT compiles your application into JavaScript output without complaint,

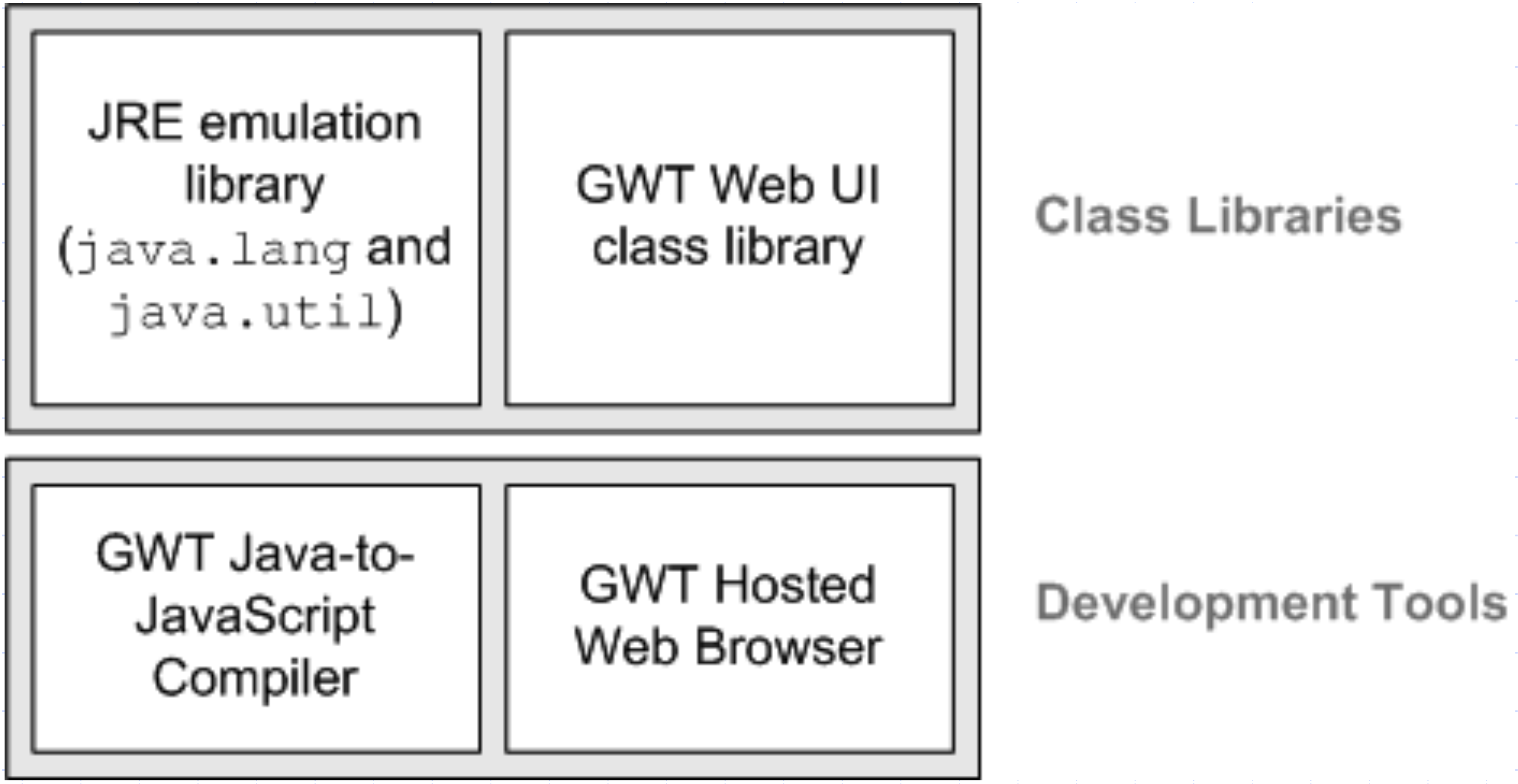
Then your application will work the same way in a web browser as it did in hosted mode.

GWT – cross browser

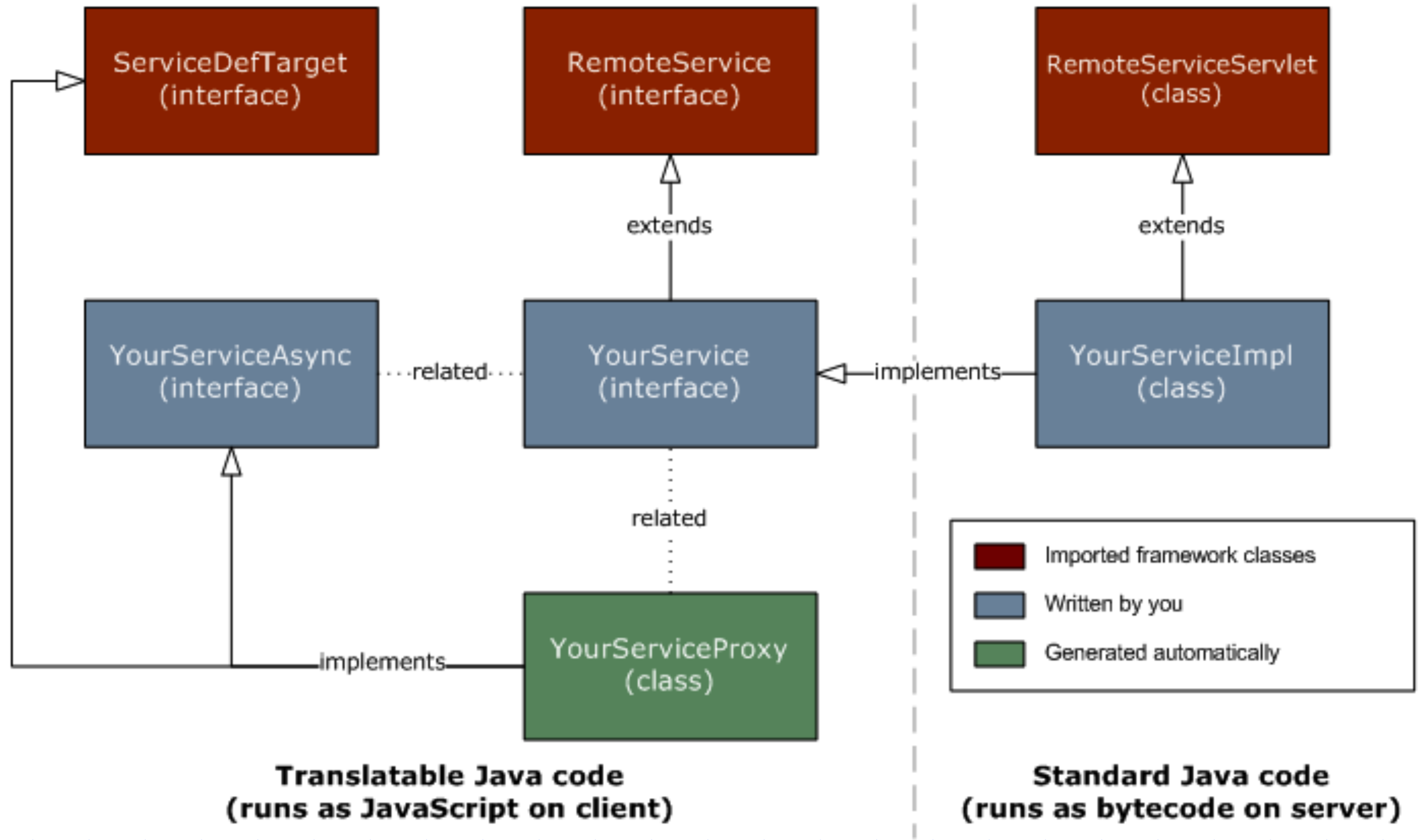
GWT shields you from worrying too much about cross-browser incompatibilities.

If you stick to built-in widgets and composites, your applications will work similarly on the most recent versions of Internet Explorer, Firefox, and Safari. (Opera, too, most of the time.)

GWT – tools


JRE emulation library (java.lang and java.util)
GWT Web UI class library
Class Libraries
GWT Java-to-JavaScript Compiler
GWT Hosted Web Browser
Development Tools

GWT service architecture


ServiceDefTarget (interface)
RemoteService (interface)
RemoteServiceServlet (class)
extends
extends
YourServiceAsync (interface)
related
YourService (interface)
implements
YourServiceImpl (class)
related
implements
YourServiceProxy (class)
Imported framework classes
Written by you
Generated automatically
Translatable Java code (runs as JavaScript on client)
Standard Java code (runs as bytecode on server)

References

About Ajax

Papers and architectural description

the first paper in the list is a must!

http://www-128.ibm.com/developerworks/java/library/wa-ajaxintro1.html

http://www-128.ibm.com/developerworks/java/library/j-ajax2/index.html

http://www-128.ibm.com/developerworks/java/library/j-ajax3/index.html

http://www-128.ibm.com/developerworks/java/library/j-ajax1/index.html

About GWT

http://code.google.com/intl/it-IT/webtoolkit/overview.html

More References

Plenty of tutorials on Ajax

http://www.maxkiesler.com/index.php/weblog/comments/60_more_helpful_ajax_tutorials/

New development (may 07)

Java FX Script

Does JavaFX Spell The End Of AJAX?

…and all the Web 2.0…

See e.g.

http://www.seomoz.org/web2.0

… and human computing…

See e.g.

http://video.google.co.uk/videoplay?docid=-8246463980976635143