Development of Web Applications

Principles and Practice

Vincent Simonet, 2013-2014
Université Pierre et Marie Curie, Master Informatique, Spécialité STL

1 Architecture of Web Applications

Vincent Simonet, 2013-2014
Université Pierre et Marie Curie, Master Informatique, Spécialité STL

Objectives of the course

- Have an overall knowledge of the **principles** and **technologies** for the development of web applications.
- Practice by developing one complete web application.

Objectives of the course

The challenge: There is a multitude of technologies for developing web applications.

The solution: Explain principles, give an overview of the market, and focus on one example: Java Servlets.

Why Java Servlets?

- Widely used,
- Java,
- Basic mechanisms remain visible,
- Cloud hosting is possible.

Contents

1. Architecture of web applications
2. Communication
3. Server Technologies
4. Client Technologies
5. Web Development Frameworks
6. Practical Aspects
7. Project Presentations

Prerequisites

- Java programming,
- Basics in HTML and CSS,
- Basics in JavaScript.

If you're not familiar with these technologies, follow the tutorials referenced in the lecture notes.

Evaluation

- Continuous evaluation: **50%** (surprise tests!)
- Project: **50%**

Missing a test without acceptable justification = 0

References and Further Reading

Books:

- A few general books (see the list in the lecture notes),
- A multitude a technology-specific books!

The best documentation is probably on the web! (and free :) See in the lecture notes (especially Wikipedia).

Contact information

vincent.simonet@vtst.net

http://www.normalesup.org/~simonet/teaching/upmc-master

Web applications?

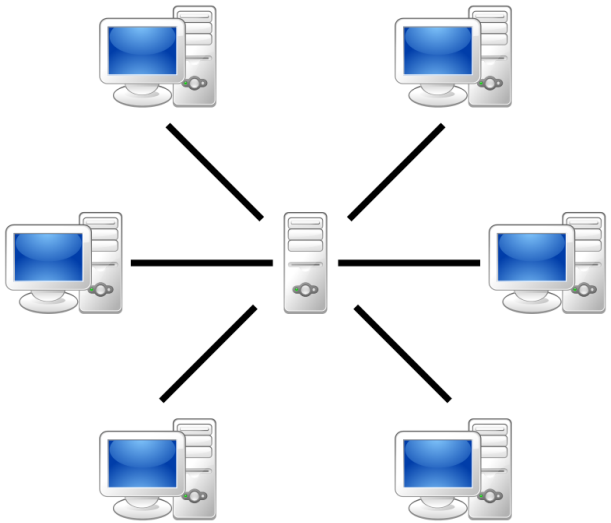
Client/server: a software definition

Servers (*a.k.a.* services or daemons) execute by waiting for requests from **client** programs to arrive, and then processing those requests.

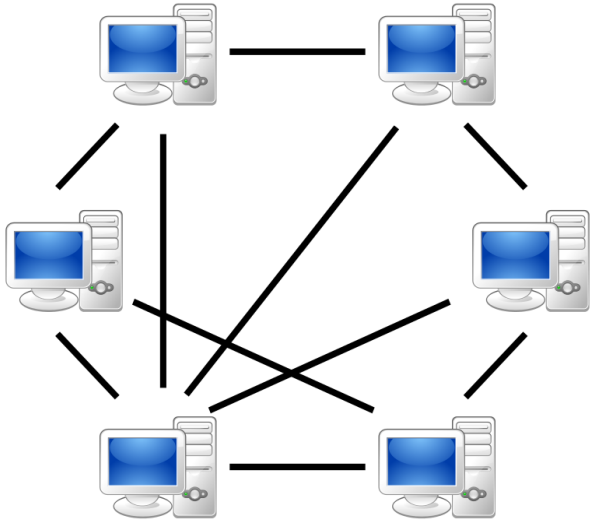
Client programs might be applications used by human beings, or they could be servers that need to make their own requests.

Client/server: a hardware definition

Client/server



Peer-to-peer

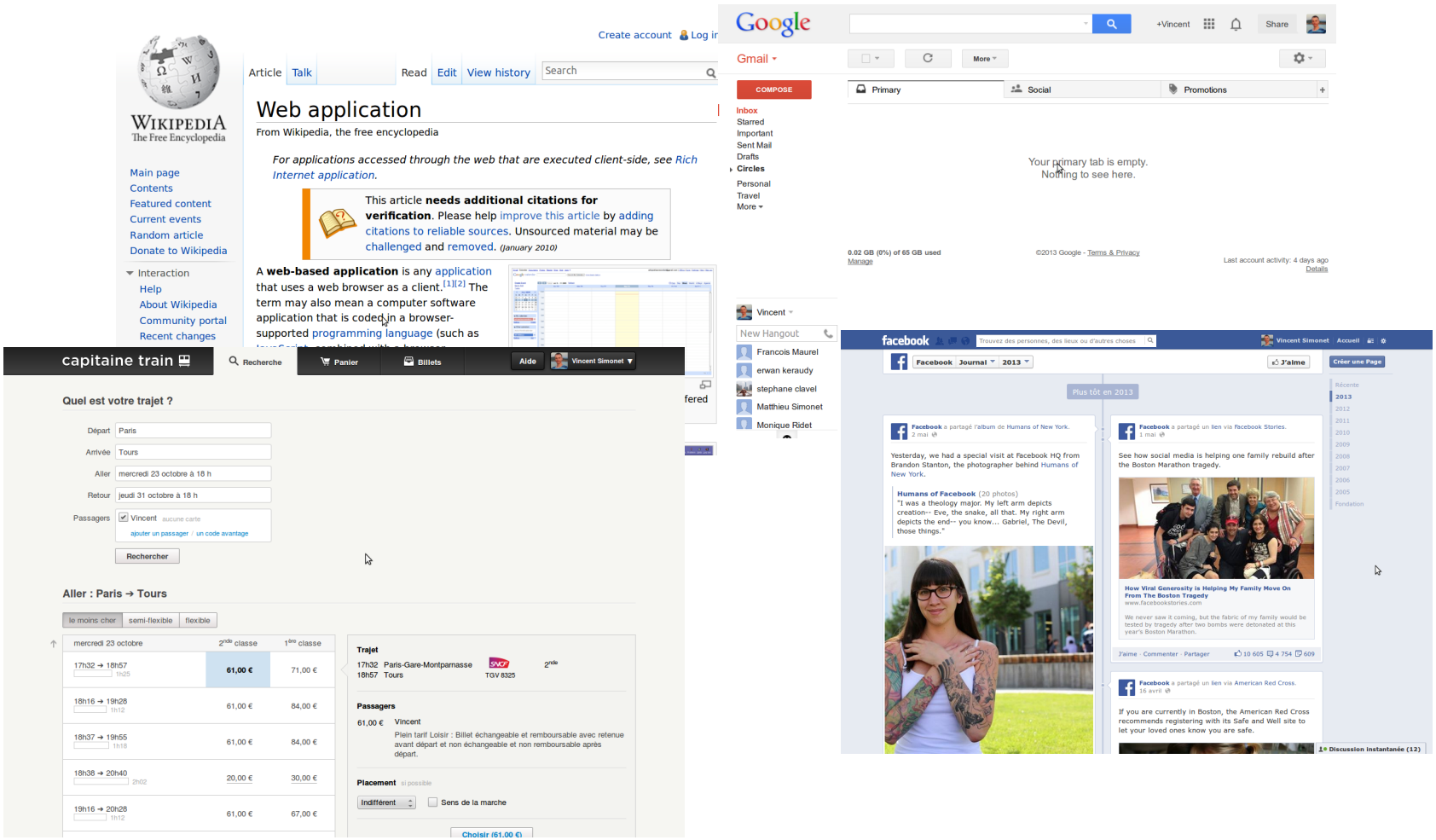


What is a web application?

It is a client/server application that uses a web browser as its client program, and performs an interactive service by connecting with servers over the Internet (or Intranet).

A web site simply delivers content from static files. A web application presents dynamically tailored content based on request parameters, tracked user behaviors, and security considerations.

Examples of web applications



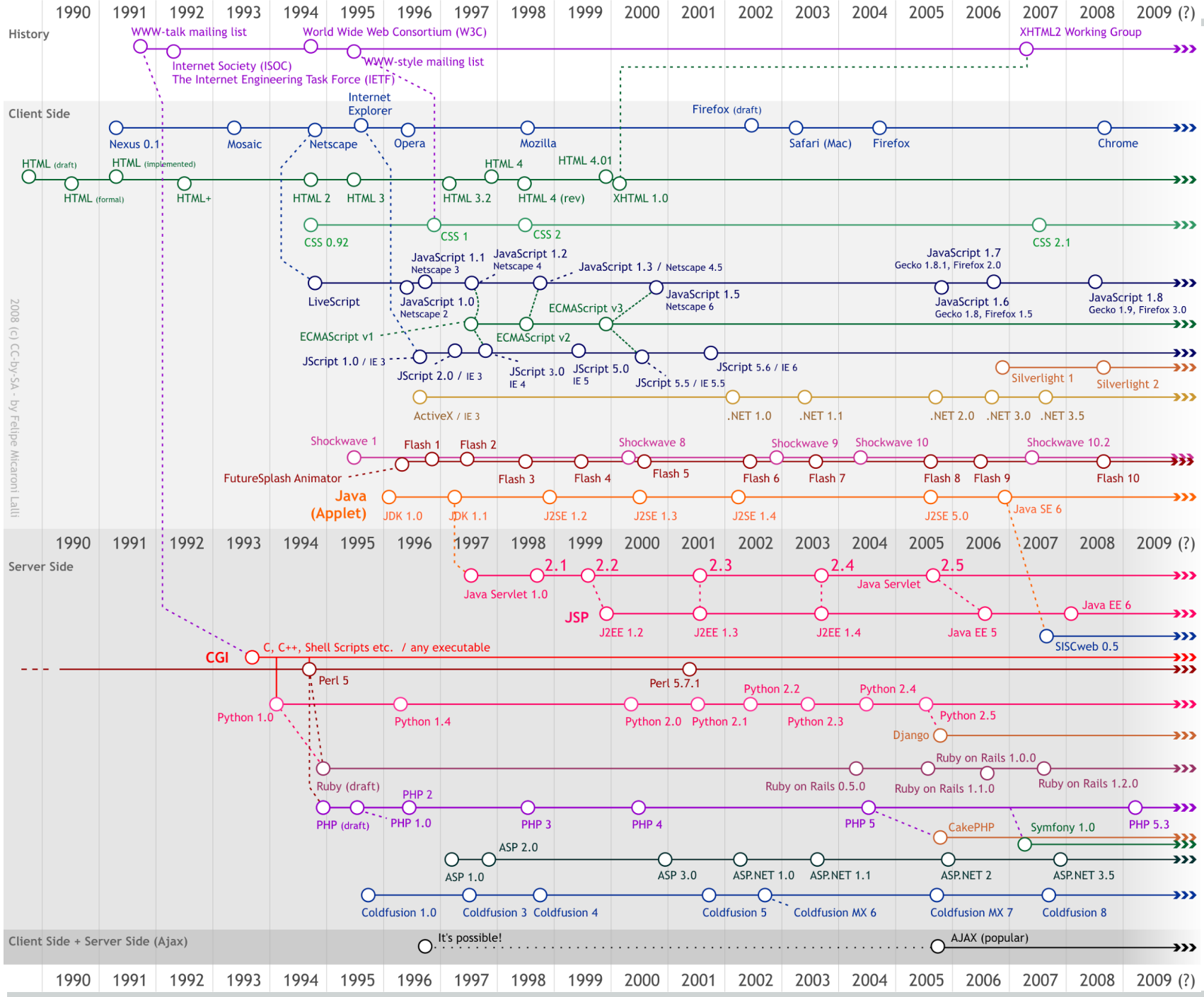
Benefits

- Easy to deploy and upgrade,
- Cross-platform compatibility,
- Limited resources on client side,
- Interoperability.

Drawbacks

- Limitations on user interface compared to natve Graphical User Interface,
- Compatibility issues with some web browser,
- Require a network connection,
- The user does not own the software.

A Brief History


1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 (?)
History
WWW-talk mailing list
World Wide Web Consortium (W3C)
XHTML2 Working Group
Internet Society (ISOC)
The Internet Engineering Task Force (IETF)
WWW-style mailing list
Client Side
Internet Explorer
Firefox (draft)
Nexus 0.1
Mosaic
Netscape
Opera
Mozilla
Safari (Mac)
Firefox
Chrome
HTML (draft)
HTML (implemented)
HTML 4
HTML 4.01
HTML (formal)
HTML+
HTML 2
HTML 3
HTML 3.2
HTML 4 (rev)
XHTML 1.0
CSS 0.92
CSS 1
CSS 2
CSS 2.1
JavaScript 1.1
Netscape 3
JavaScript 1.2
Netscape 4
JavaScript 1.3 / Netscape 4.5
JavaScript 1.7
Gecko 1.8.1, Firefox 2.0
LiveScript
JavaScript 1.0
Netscape 2
JavaScript 1.5
Netscape 6
JavaScript 1.6
Gecko 1.8, Firefox 1.5
JavaScript 1.8
Gecko 1.9, Firefox 3.0
ECMAScript v3
ECMAScript v1
ECMAScript v2
JScript 1.0 / IE 3
JScript 2.0 / IE 3
JScript 3.0
IE 4
JScript 5.0
IE 5
JScript 5.5 / IE 5.5
JScript 5.6 / IE 6
Silverlight 1
Silverlight 2
ActiveX / IE 3
.NET 1.0
.NET 1.1
.NET 2.0
.NET 3.0
.NET 3.5
Shockwave 1
Flash 1
Flash 2
Shockwave 8
Shockwave 9
Shockwave 10
Shockwave 10.2
FutureSplash Animator
Flash 3
Flash 4
Flash 5
Flash 6
Flash 7
Flash 8
Flash 9
Flash 10
Java (Applet)
JDK 1.0
JDK 1.1
J2SE 1.2
J2SE 1.3
J2SE 1.4
J2SE 5.0
Java SE 6
2008 (c) CC-by-SA - by Felipe Micaroni Lalli
1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 (?)
Server Side
2.1
2.2
2.3
2.4
2.5
Java Servlet 1.0
Java Servlet
JSP
Java EE 6
J2EE 1.2
J2EE 1.3
J2EE 1.4
Java EE 5
SISCweb 0.5
CGI
C, C++, Shell Scripts etc. / any executable
Perl 5
Perl 5.7.1
Python 2.2
Python 2.4
Python 1.0
Python 1.4
Python 2.0
Python 2.1
Python 2.3
Python 2.5
Django
Ruby on Rails 1.0.0
Ruby (draft)
Ruby on Rails 0.5.0
Ruby on Rails 1.1.0
Ruby on Rails 1.2.0
PHP 2
PHP (draft)
PHP 1.0
PHP 3
PHP 4
PHP 5
CakePHP
Symfony 1.0
PHP 5.3
ASP 2.0
ASP 1.0
ASP 3.0
ASP.NET 1.0
ASP.NET 1.1
ASP.NET 2
ASP.NET 3.5
Coldfusion 1.0
Coldfusion 3
Coldfusion 4
Coldfusion 5
Coldfusion MX 6
Coldfusion MX 7
Coldfusion 8
Client Side + Server Side (Ajax)
It's possible!
AJAX (popular)
1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 (?)

Key dates

1993: Mosaic browser, CGI

1995: PHP 1.0

1996: JavaScript 1.0

1999: Web Application, Java Servlet (server)

2005: AJAX

2008: HTML5 first public working draft

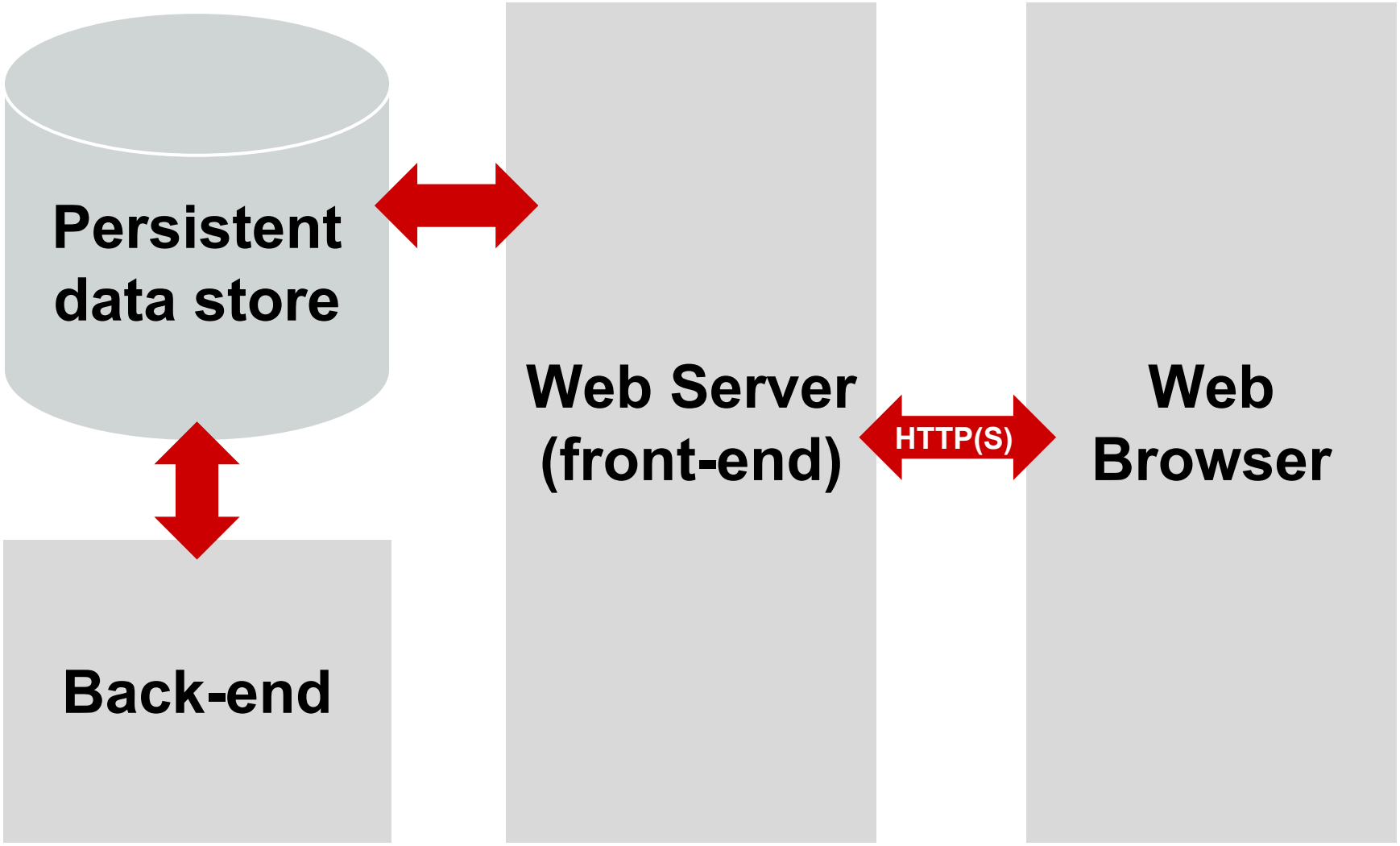
2014?: HTML5 specification

(User) client vs (remote) server

- **70s:** Light user terminals, everything is done by the server.
- **80s/90s:** Personal computers. Everything happens on client side.
- **90s/2000s:** Light client (web browser), all logic in server.
- **2000s:** Logic is back in the client ("Web 2.0").
- **2010s:** Mobile applications.

Overall Architecture

Typical architecture of a web application



Web browser

- Mainly user interface,
- Short term state (in general),
- May implement some logic, especially for fast response time (but untrusted),
- Communicate with the web server using HTTP(S),
- Executing HTML, CSS and JavaScript code.

Web server (front-end)

- Answers to HTTP(S) requests from the web clients,
- Stateless,
- Reads and writes data in a persistent data store,
- Performs most of the business logic,
- Consists in a general of a server/container (Apache, Tomcat) and a framework (PHP, Java Servlets, etc.) running business logic.

Data store

- The state of the web application,
- Historically a (My)SQL database, some more recent evolutions,
- The synchronisation point.

Back-end

- All what needs to be done in the server, but which is not triggered by a client request.

Typical architecture of a web application