

# USO COMBINATO DI PHP, JAVASCRIPT E WEB SERVICES

*Adriano Cerocchi*

*cerocchi@dis.uniroma1.it - www.dis.uniroma1.it/~cerocchi*

## PIATTAFORME SOFTWARE DISTRIBUITE



**SAPIENZA**  
UNIVERSITÀ DI ROMA



# INTRODUCTION

- javascript overview
- simple javascript example
- ajax
- ajax example
- web services using php and ajax
- GeoNames example
- PHP, ajax and web-services all together in a real example

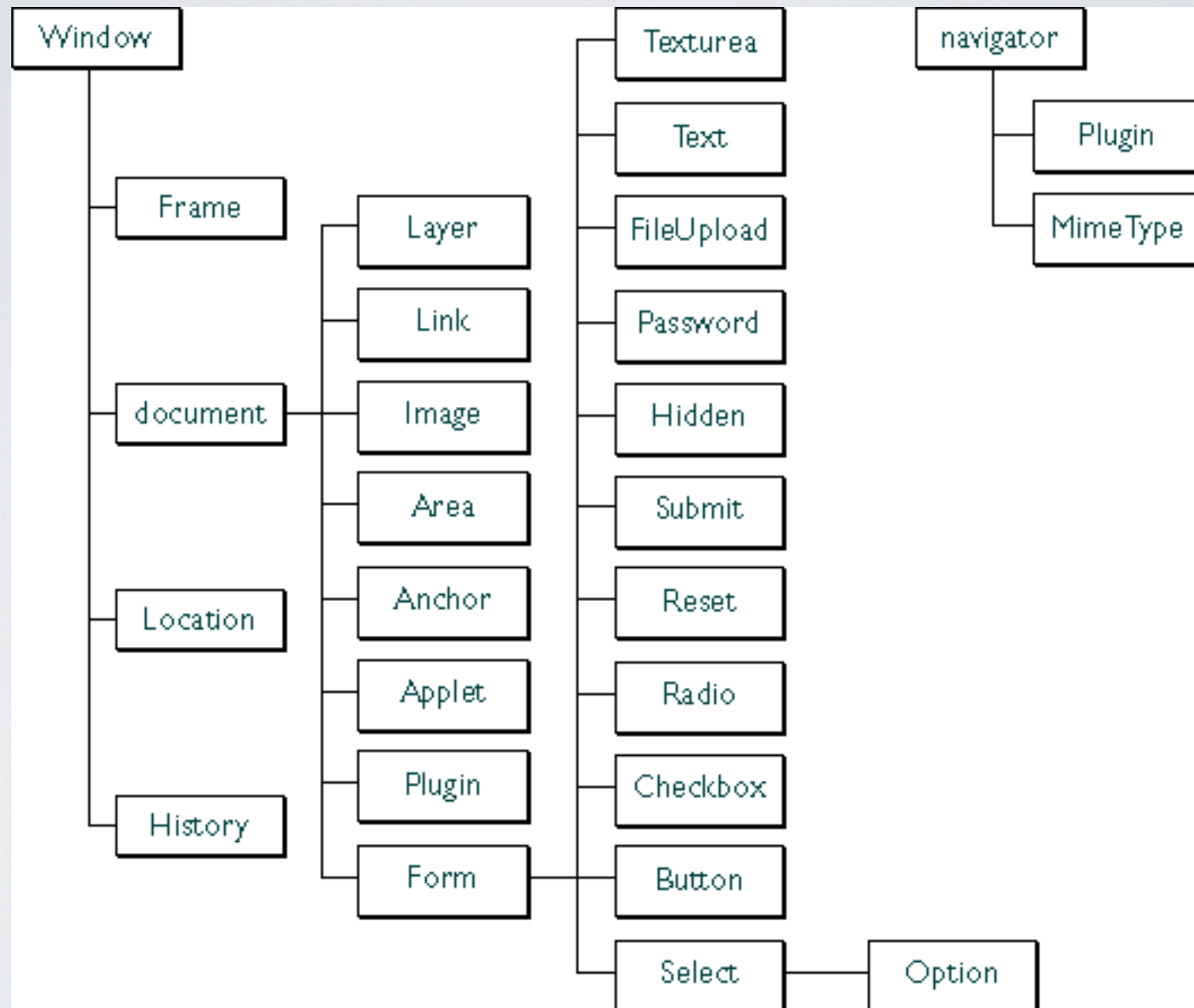
# JAVASCRIPT OVERVIEW

- it is a scripting language (it is interpreted)
- it was developed by Netscape Communication (it was named LiveScript)
- first standard in 1997, last one in 1999
- it is client-side
- it is designed to add interactivity to HTML pages
- usually embedded directly into HTML code

# JAVASCRIPT OVERVIEW

- it is object oriented:
  - build-in objects
  - **the page is an object**
  - custom object
- it is event-driven: procedures are triggered when an event occurs (ex: page loaded, click, key press, ...)

# JAVASCRIPT OVERVIEW



# JAVASCRIPT OVERVIEW

Attribute	The event occurs when...
<a href="#">onabort</a>	Loading of an image is interrupted
<a href="#">onblur</a>	An element loses focus
<a href="#">onchange</a>	The user changes the content of a field
<a href="#">onclick</a>	Mouse clicks an object
<a href="#">ondblclick</a>	Mouse double-clicks an object
<a href="#">onerror</a>	An error occurs when loading a document or an image
<a href="#">onfocus</a>	An element gets focus
<a href="#">onkeydown</a>	A keyboard key is pressed
<a href="#">onkeypress</a>	A keyboard key is pressed or held down
<a href="#">onkeyup</a>	A keyboard key is released

<a href="#">onload</a>	A page or an image is finished loading
<a href="#">onmousedown</a>	A mouse button is pressed
<a href="#">onmousemove</a>	The mouse is moved
<a href="#">onmouseout</a>	The mouse is moved off an element
<a href="#">onmouseover</a>	The mouse is moved over an element
<a href="#">onmouseup</a>	A mouse button is released
<a href="#">onreset</a>	The reset button is clicked
<a href="#">onresize</a>	A window or frame is resized
<a href="#">onselect</a>	Text is selected
<a href="#">onsubmit</a>	The submit button is clicked
<a href="#">onunload</a>	The user exits the page

# JAVASCRIPT OVERVIEW

- javascript could be the fortune or the disaster of a web page:
  - it can enhance the user experience
  - ...but commonly it is not optimized, the page can be surfed very slow!
  - reference example: facebook
  - be careful to the browsers JS standard compliance!

# JAVASCRIPT EXAMPLE

- example:

- [www.adrianoocerocchi.it/psd/esercitazione/provaJS.php](http://www.adrianoocerocchi.it/psd/esercitazione/provaJS.php)

AJAX

# AJAX

- AJAX: Asynchronous Javascript And Xml
  - *but...ajax requests can avoid to use XML and they can also be performed in a synchronous way...*
- using javascript it is possible to make HTTP request
- HTTP reply is an event, then it could be associated to a javascript procedure
- it is performed by the interface **XMLHttpRequest**
- it becomes famous thanks to **Google Suggest**

# AJAX - PROS & CONS

- it enhances the user experience (HCI) (++)
- it avoid the full-refresh of a page for a simple update (+)
- it force the server to take in account more requests (-)
- the complexity hidden behind each request could be not negligible, then AJAX can limit the scalability of the system (--)
- for security reasons, requests can be only sent to the same domain from which the page is loaded (+?-)

# AJAX - XMLHttpRequest INTERFACE

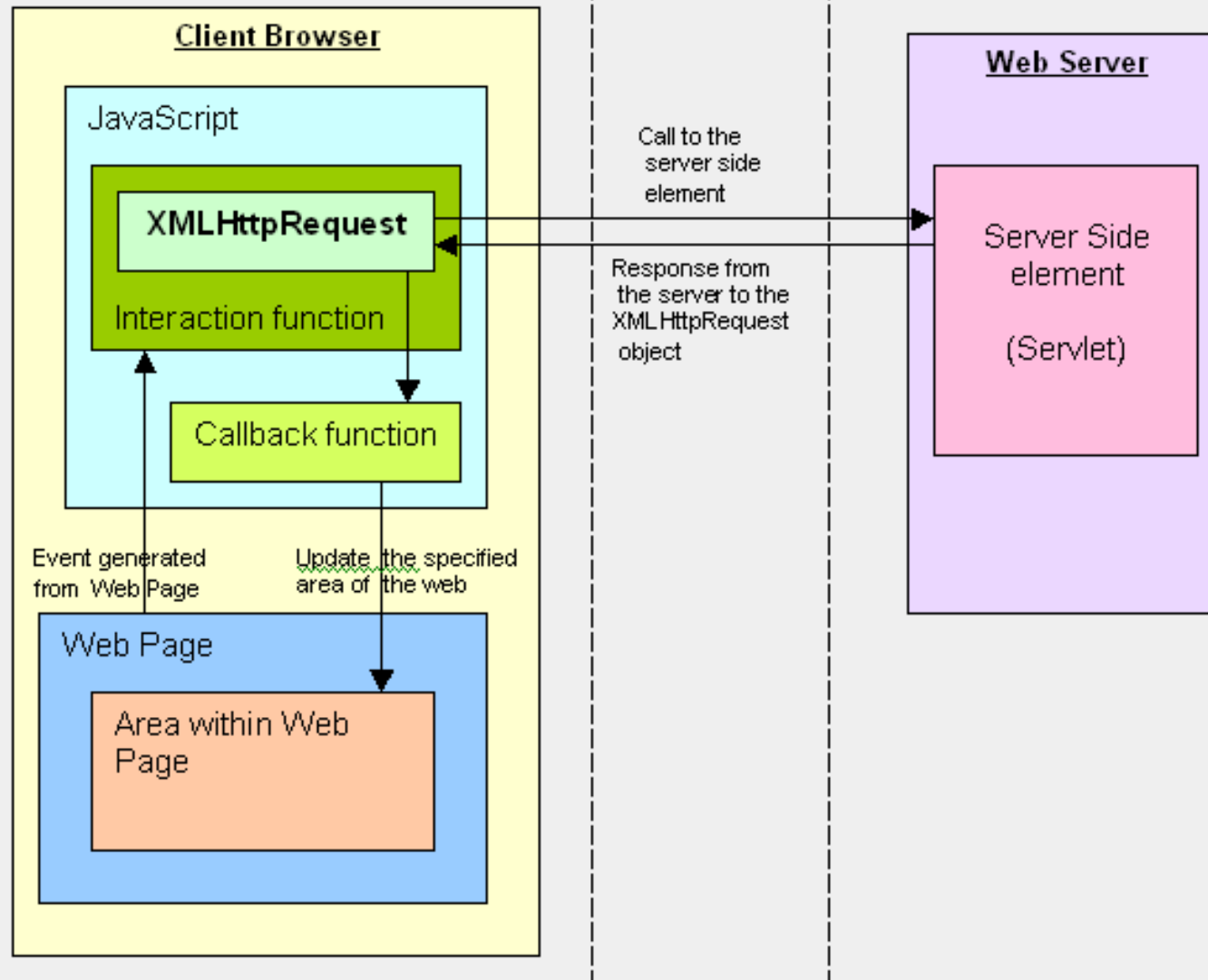
```
interface XMLHttpRequest {
    // event handler
    attribute EventListener onreadystatechange;

    // state
    const unsigned short UNSENT = 0;
    const unsigned short OPENED = 1;
    const unsigned short HEADERS RECEIVED = 2;
    const unsigned short LOADING = 3;
    const unsigned short DONE = 4;
    readonly attribute unsigned short readyState;

    // request
    void open(in DOMString method, in DOMString url);
    void open(in DOMString method, in DOMString url, in boolean async);
    void open(in DOMString method, in DOMString url, in boolean async, in DOMString user);
    void open(in DOMString method, in DOMString url, in boolean async, in DOMString user, in DOMString password);
    void setRequestHeader(in DOMString header, in DOMString value);
    void send();
    void send(in DOMString data);
    void send(in Document data);
    void abort();

    // response
    DOMString getAllResponseHeaders();
    DOMString getResponseHeader(in DOMString header);
    readonly attribute DOMString responseText;
    readonly attribute Document responseXML;
    readonly attribute unsigned short status;
    readonly attribute DOMString statusText;
};
```

# AJAX - ARCHITECTURE



AJAX interaction with XMLHttpRequest object

# AJAX EXAMPLE

- example:

- [www.adrianoocerocchi.it/psd/esercitazione/provaJSajax.php](http://www.adrianoocerocchi.it/psd/esercitazione/provaJSajax.php)

# WEB SERVICES AND PHP

# WEB SERVICES AND PHP

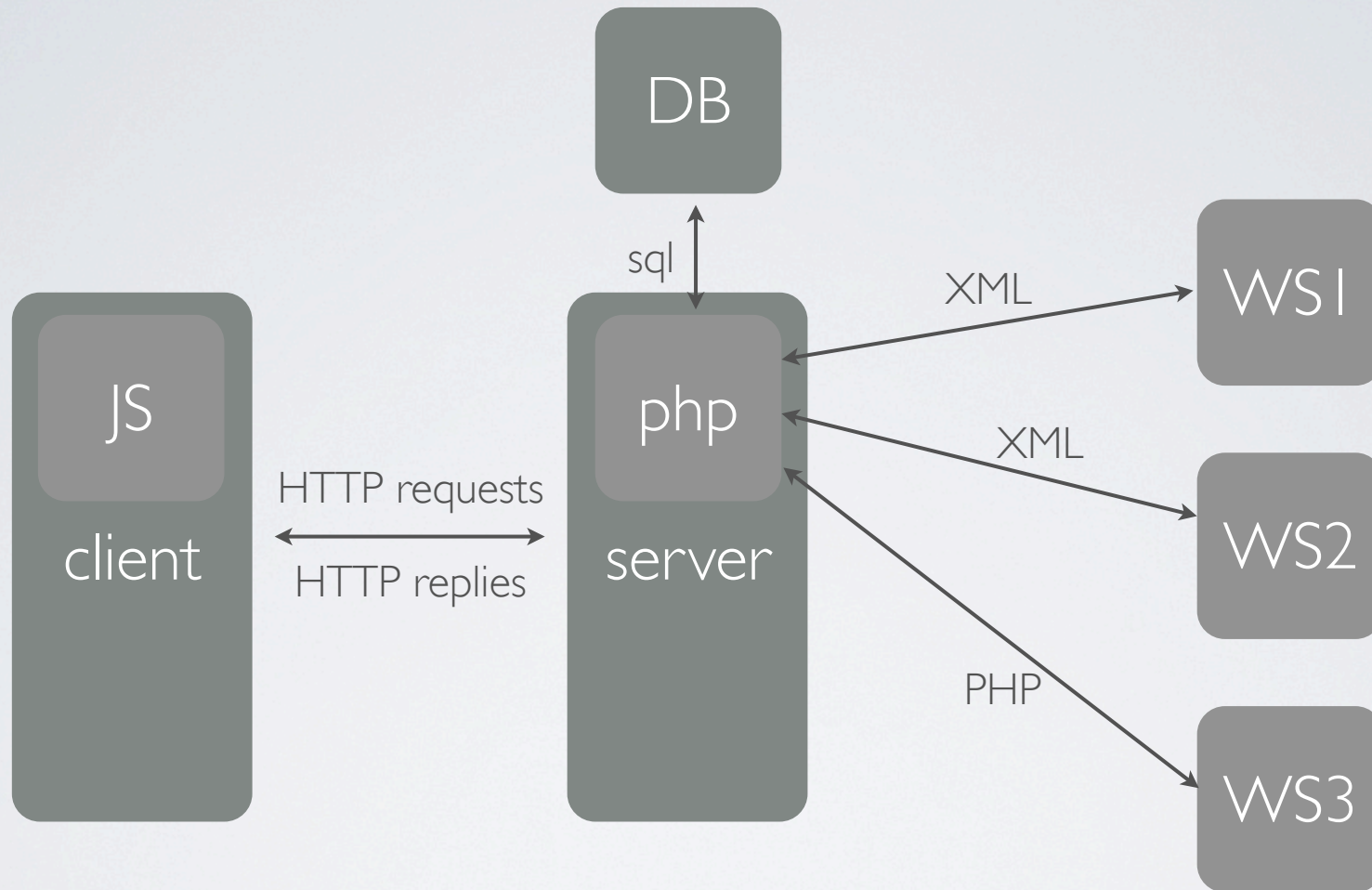
- using PHP it is possible to use external web services
- the interaction paradigm is synchronous
- request must be sent using XML
- libraries are commonly available on the website of each service providers
- some famous examples: Google APIs, **Yahoo!APIs**, GEONames, ...

# GEONAMES EXAMPLE

- example:

- [www.adrianocerocchi.it/psd/prova\\_geonames.php](http://www.adrianocerocchi.it/psd/prova_geonames.php)

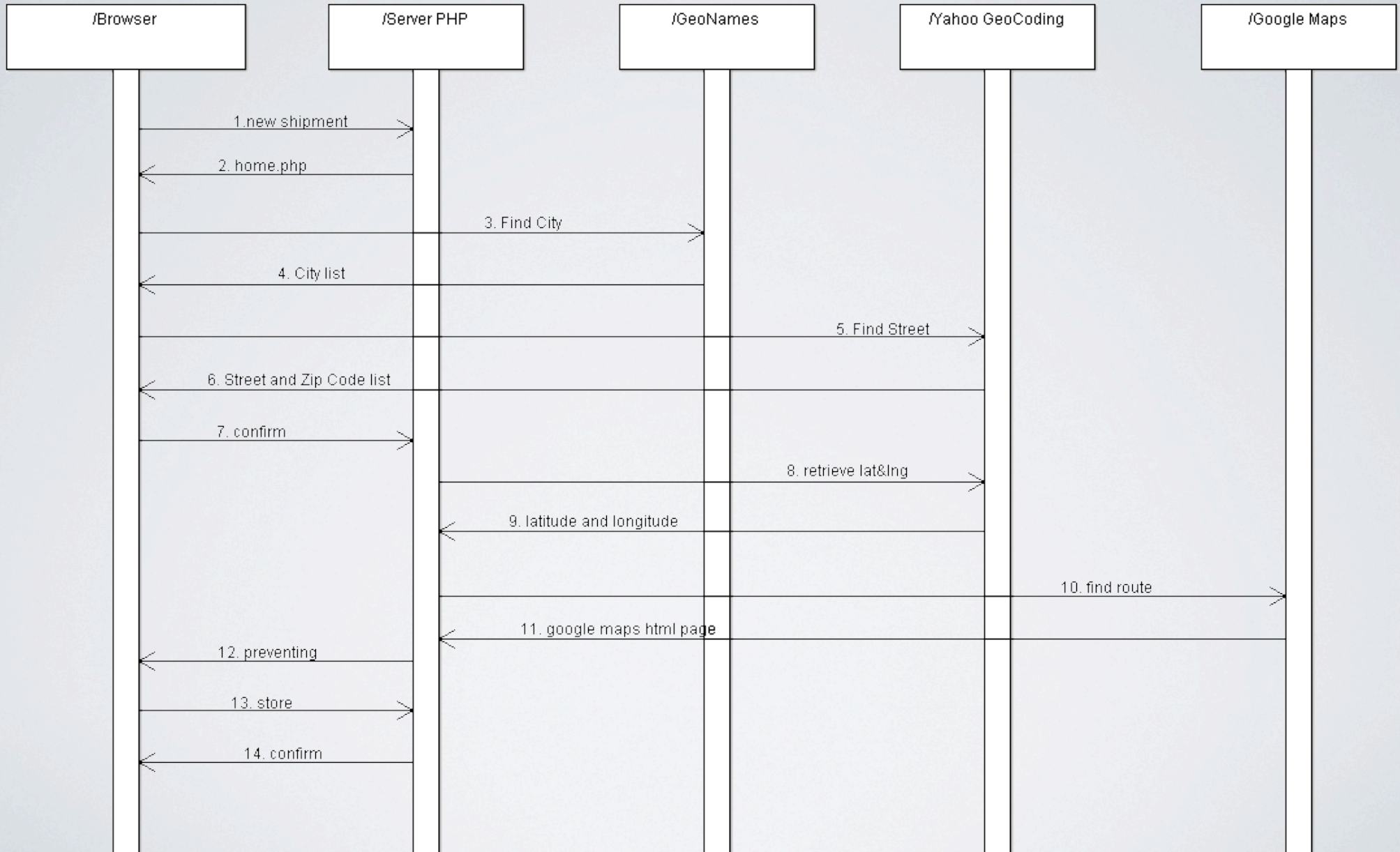
# WEB SERVICES AND PHP - GENERAL ARCHITECTURE



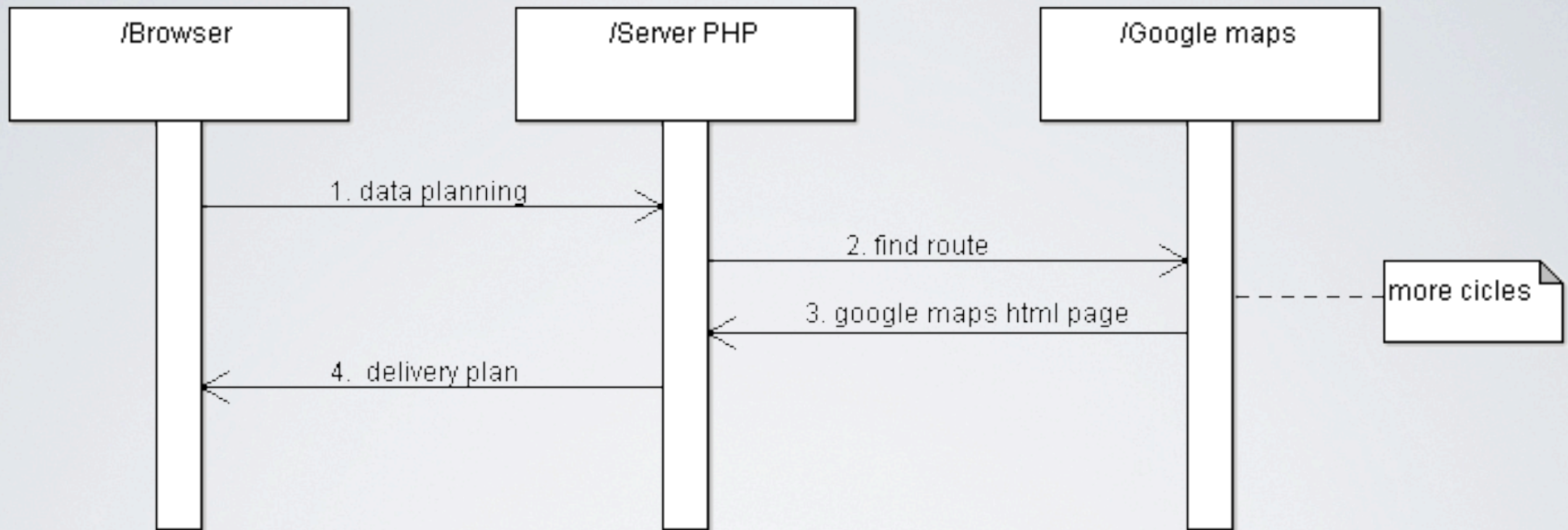
# A “ALL-INCLUSIVE” EXAMPLE

[www.adrianocerocchi.it/psd/home.php](http://www.adrianocerocchi.it/psd/home.php)

# SMART DELIVERY PLANNING



# SMART DELIVERY PLANNING



# CONCLUSION AND PROPOSALS

- javascript and ajax have a great potential (see G-DOCS) enhanced by PHP
- their diffusion is world wide
- ...
- web 3.0? we are waiting for...
- web-services sometimes unavailable
- proposal for PSD exam: ajax live chat, auto-tunable application

# REFERENCES

- all the material of the lecture is available on [www.adrianocerocchi.it/didactics.php](http://www.adrianocerocchi.it/didactics.php)
- web...