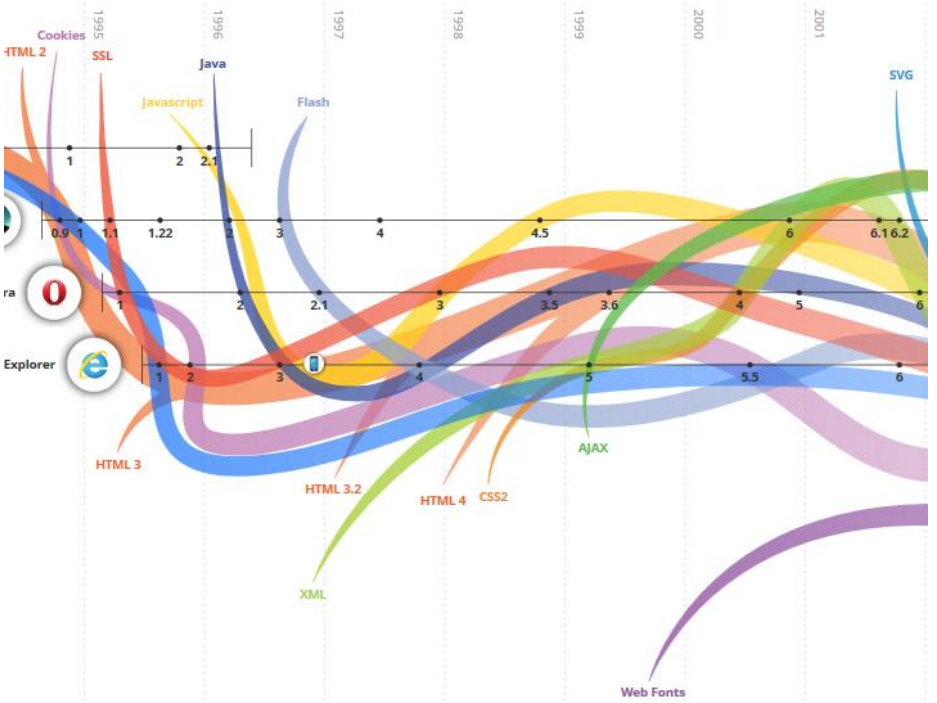


Web Architecture and Technologies

Ambient intelligence

Fulvio Corno

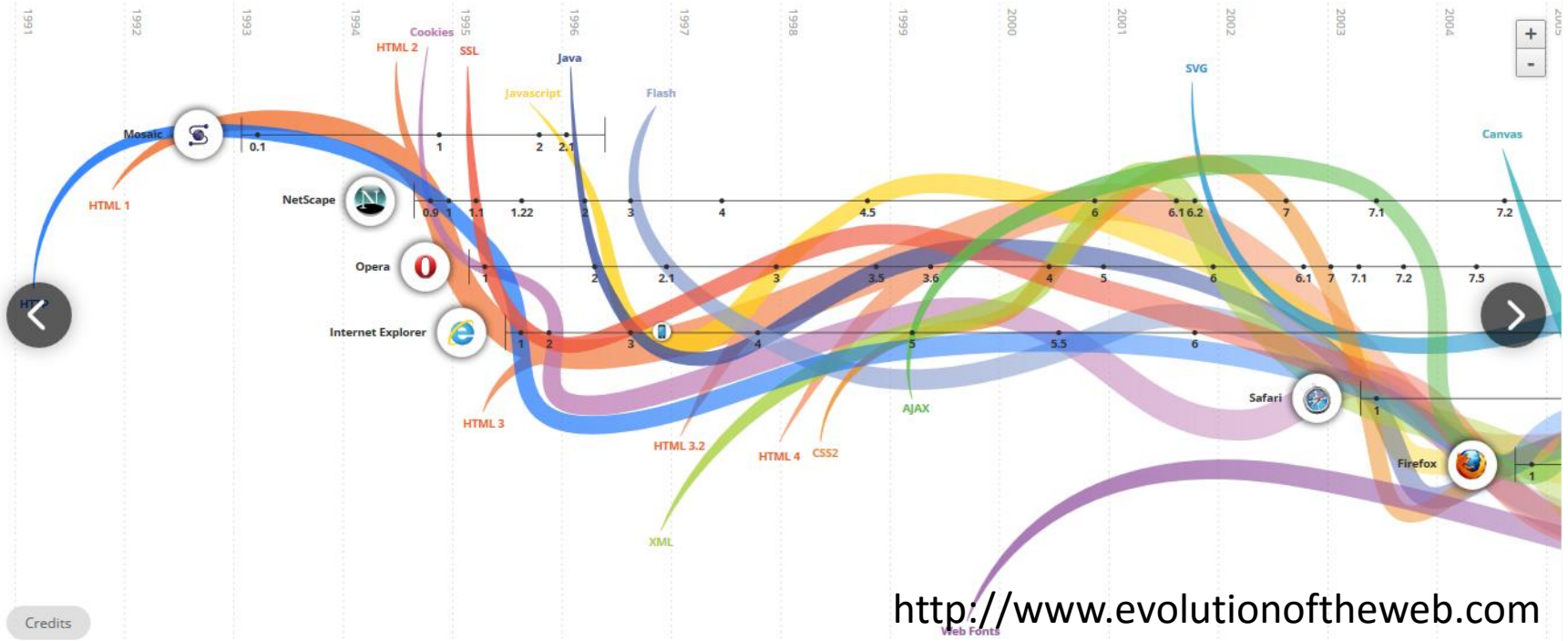
Politecnico di Torino, 2015/2016

POLITECNICO DI TORINO

e-Lite

Goal

- Understanding Web technologies
  - Adopted for User Interfaces
  - Adopted for Distributed Application Integration

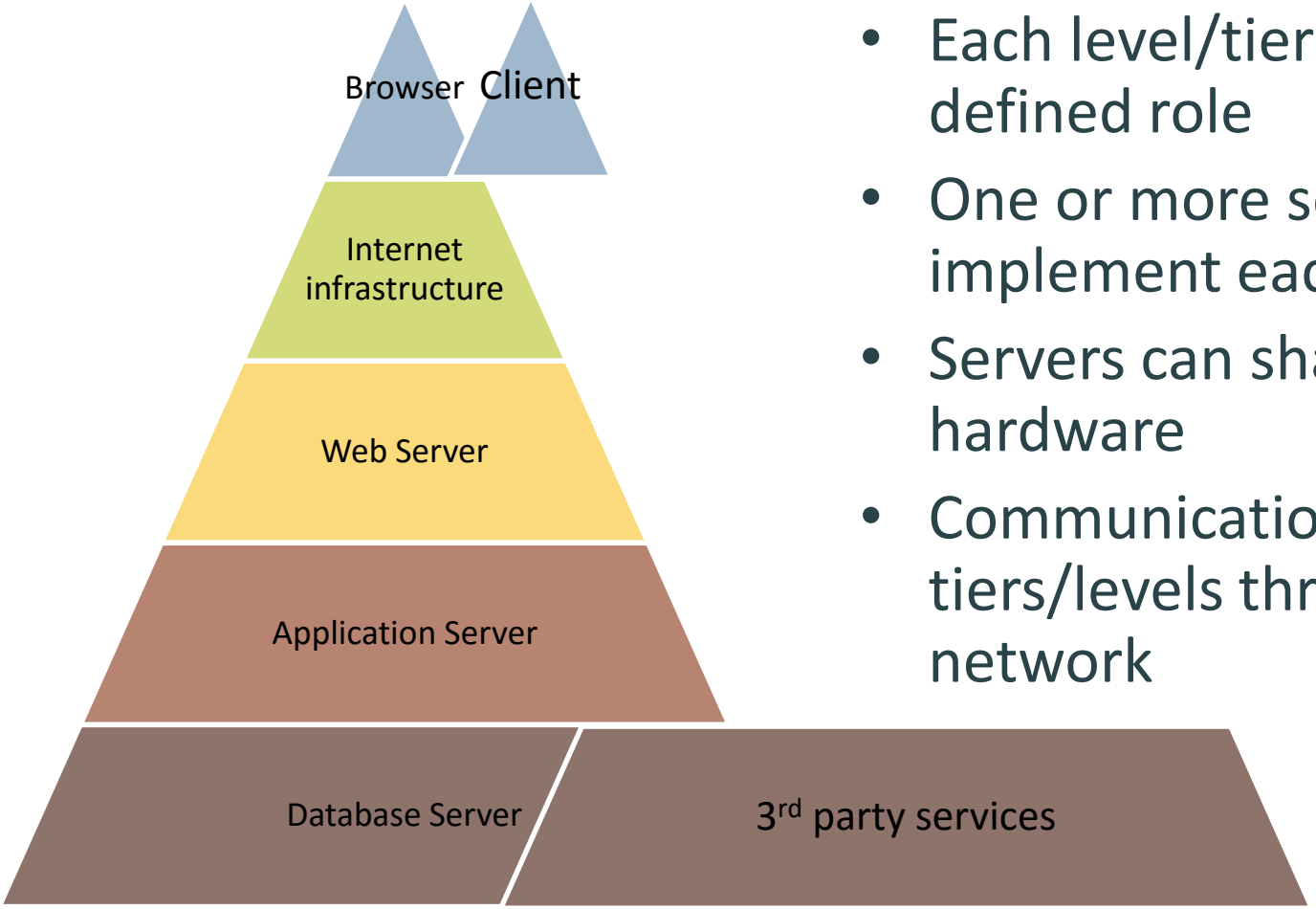


Web Architecture and Technologies

WEB ARCHITECTURE OVERVIEW

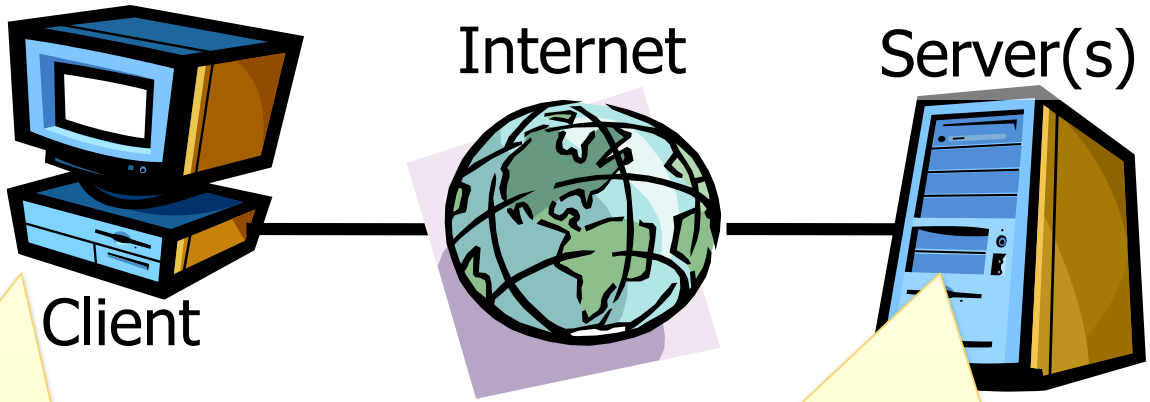
HISTORICAL EVOLUTION

N-tier (N-level) architecture



- Each level/tier has a well defined role
- One or more servers implement each tier/layer
- Servers can share hardware
- Communication between tiers/levels through the network

General Architecture



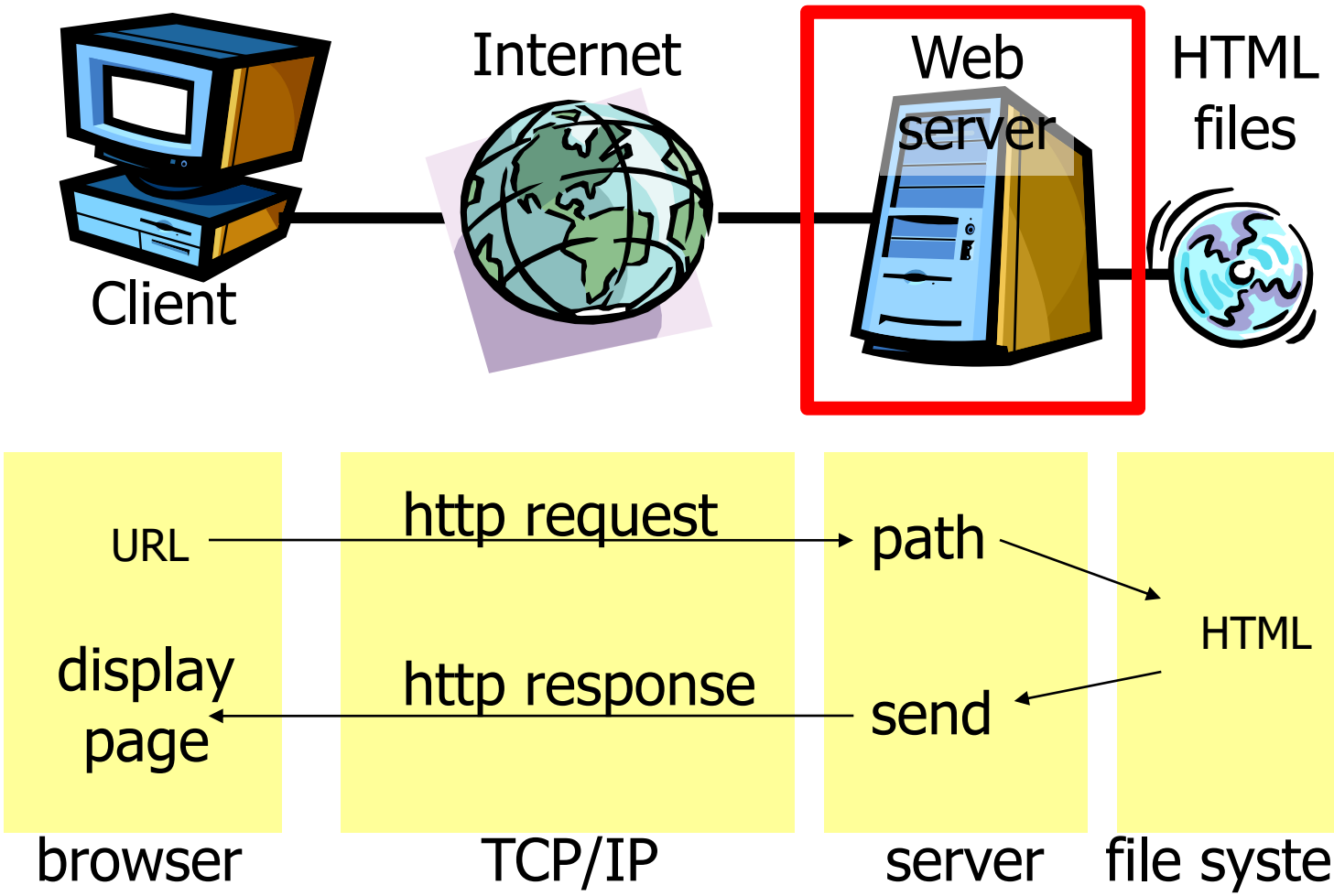
- Historically, a web browser
- But also:
  - Mobile application
  - Desktop application
  - Other server application

- Logical server:
  A process that runs on a host that relays information to a client upon the client sending it a request.
- Physical server :
  A host computer on a network that holds information (eg, Web sites) and responds to requests for information

Web server (logical)

- Manages the HTTP protocol (handles requests and provides responses)
  - Receives client requests
  - Reads static pages/contents from the filesystem
  - Activates the application server for dynamic pages/content generation (server-side)
  - Provides an file (HTML, or other) back to the client
- One HTTP connection for each request
- Multi-process, Multi-threaded or Process pool

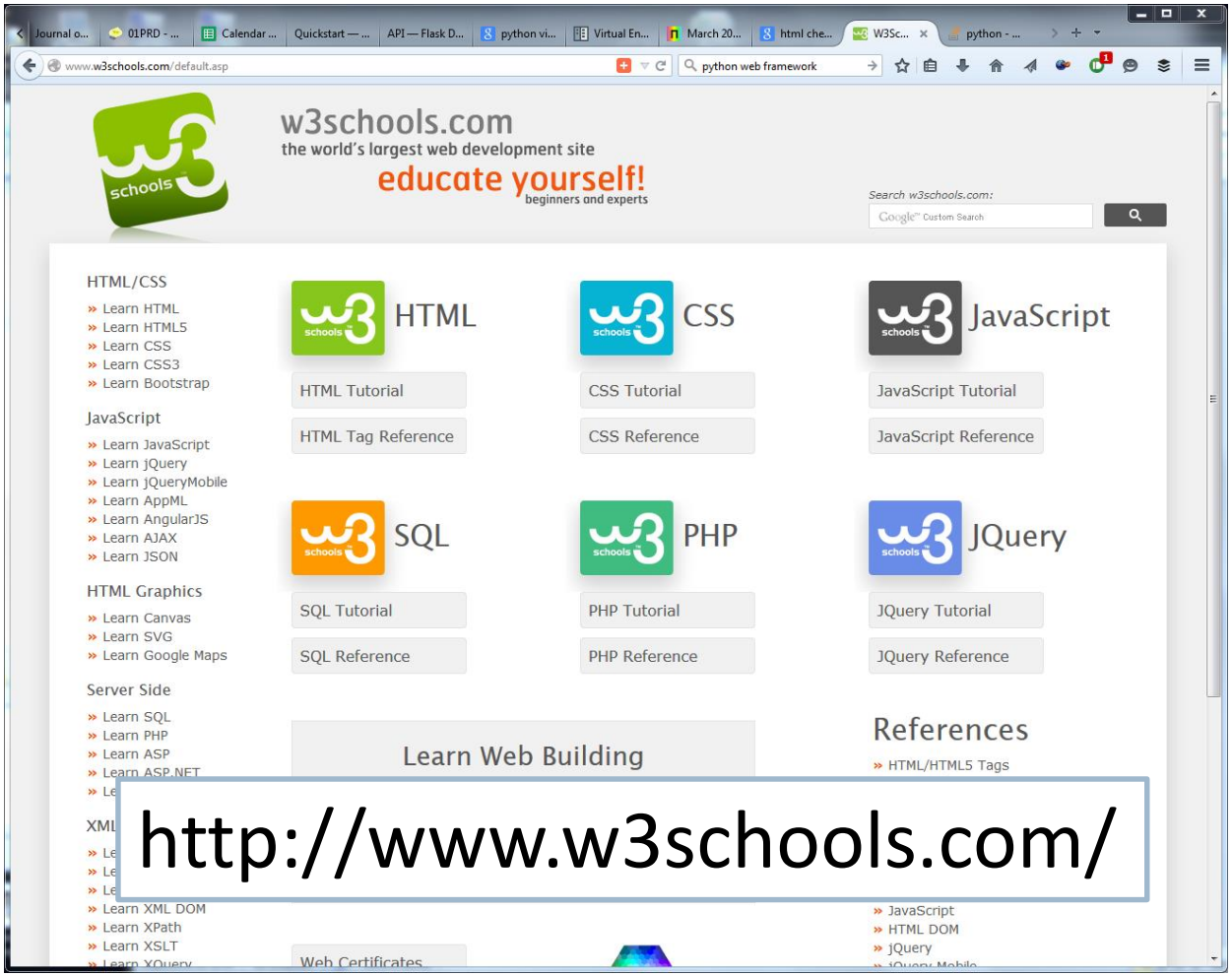
Example


Internet
Web server
HTML files
Client
URL
http request
path
HTML
display page
http response
send
browser
TCP/IP
server
file system

Adopted standards

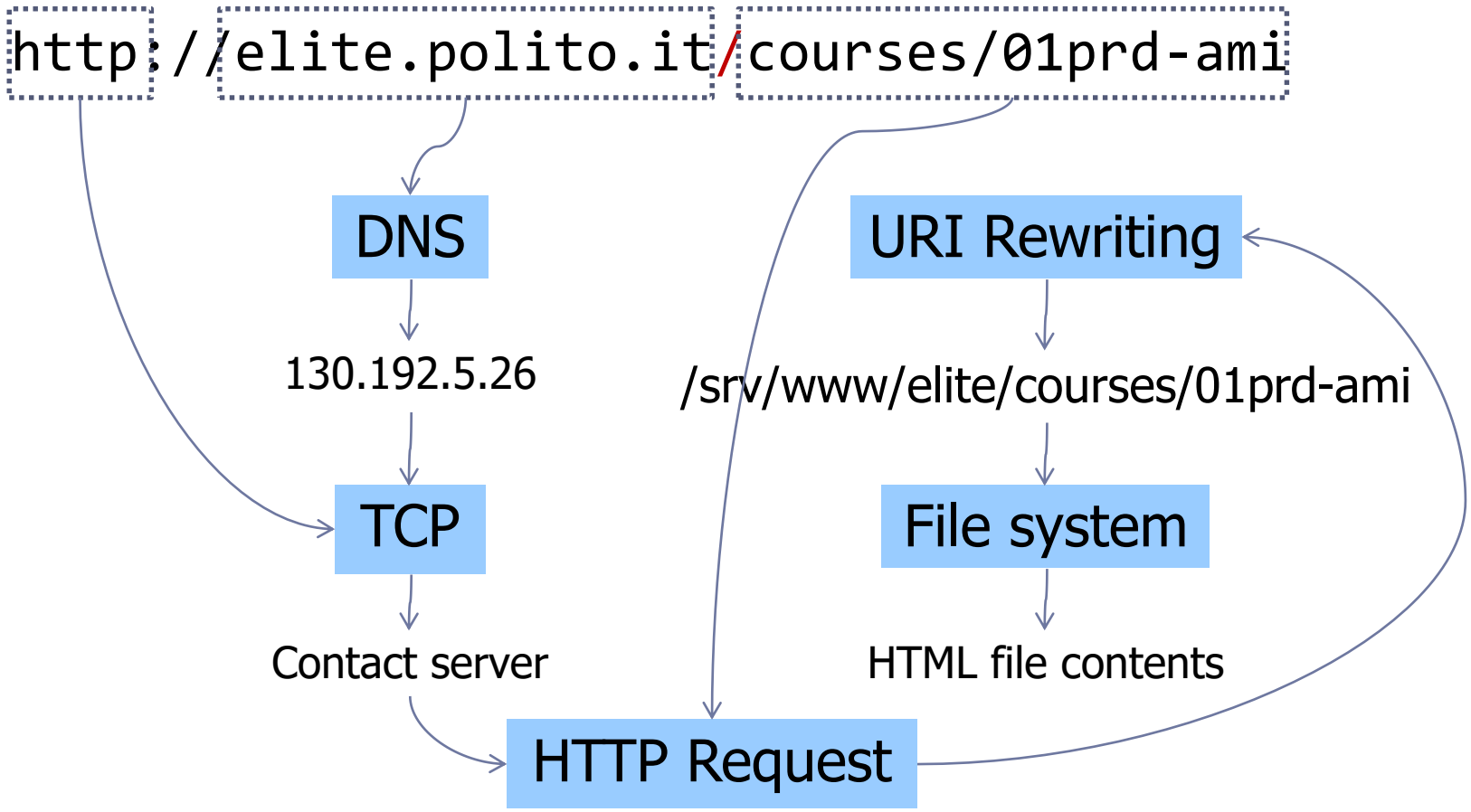
- URL (uniform resource locator) for finding web pages
- HTML (hyper text markup language) for writing web pages
- GIF (graphics interchange format), PNG (portable network graphics), JPEG, … for images
- HTTP (hyper text transfer protocol) for client-server interaction
- TCP/IP (transmission control protocol over internet protocol) for data transfer

HTML in 5 minutes



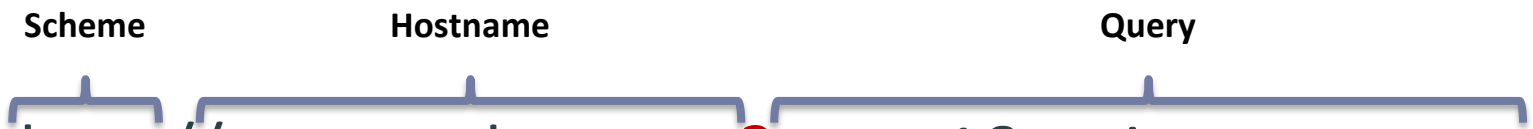
URL

RFC 2396
http://www.w3.org/Addressing/



URI Basics

- http://www.sadev.co.za/users/1/contact



- http://www.sadev.co.za?user=1&action=contact

- http://rob:pass@bbd.co.za:8044

- https://bbd.co.za/index.html#about

HTTP protocol

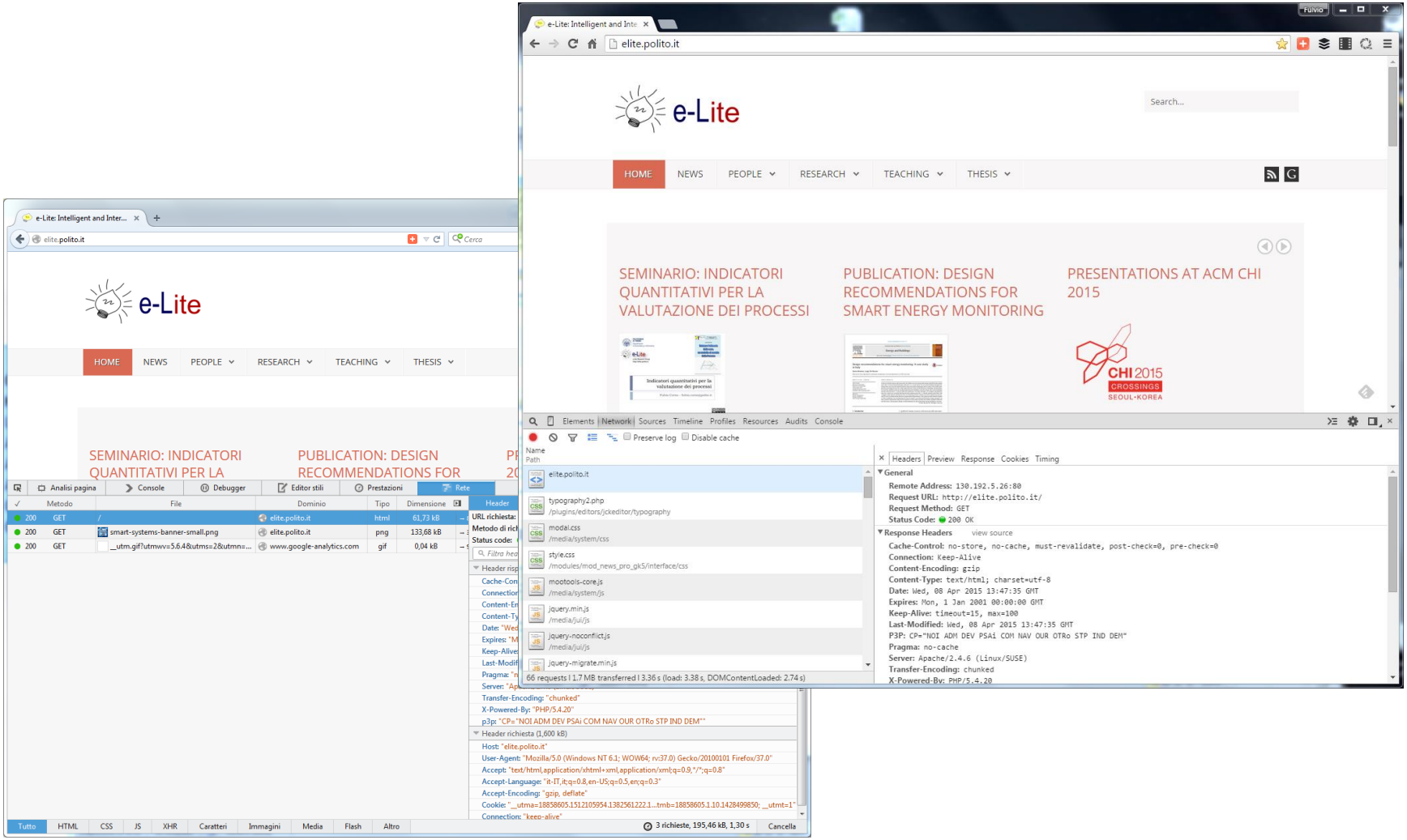
RFC 2616, RFC 2617
http://www.w3.org/Protocols

```
GET / HTTP/1.1
Host: elite.polito.it
User-Agent: Mozilla/5.0
Accept: text/html,appl
Accept-Language: it-I
Accept-Encoding: gzi
Cookie: __utma=1885
Connection: keep-a
```

```
HTTP/1.0 200 OK
Cache-Control: no-store, no-cache, must-revalidate,
Connection: Keep-Alive
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
Date: Wed, 08 Apr 2015 13:36:24 GMT
Expires: Mon, 1 Jan 2001 00:00:00 GMT
Keep-Alive: timeout=15, max=100
Last-Modified: Wed, 08 Apr 2015 13:36:24 GMT
Pragma: no-cache
Server: Apache/2.4.6 (Linux/SUSE)
Transfer-Encoding: chunked
X-Powered-By: PHP/5.4.20
p3p: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM«

<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-GB">
<head>
. . . . . .
```

Browser developer tools



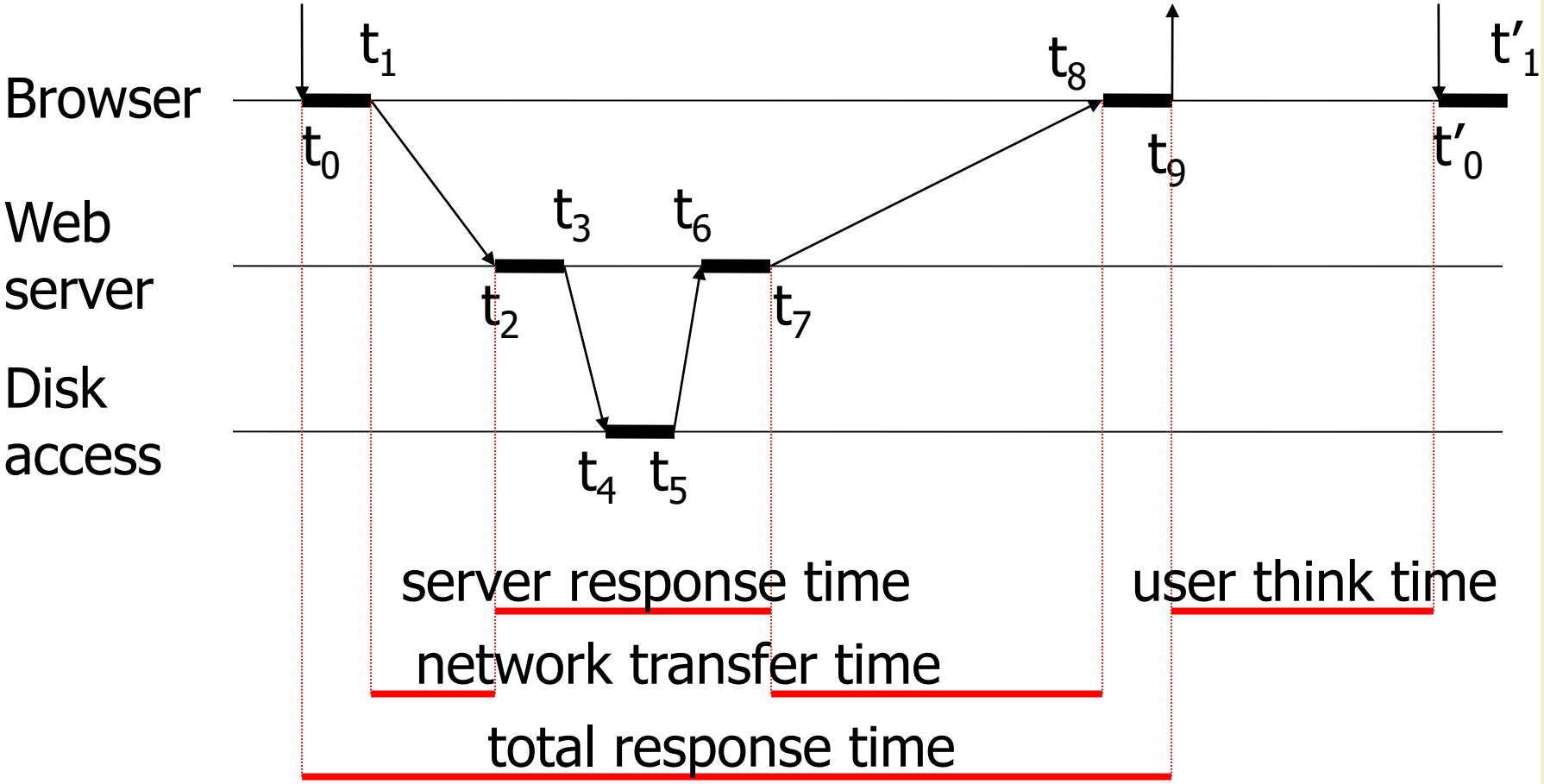
Performance measures

- **Latency**: time required for providing a 0 byte http page. Includes the server activation time, the request decoding time, the file access time, the transmission time and the time for closing the connection.
  - Unit of measure: http/s or s/http
- **Throughput**: maximum speed at which infinite-sized pages can be sent.
  - Unit of measure: Bytes (Mbytes)/s
- **#Requests / s**

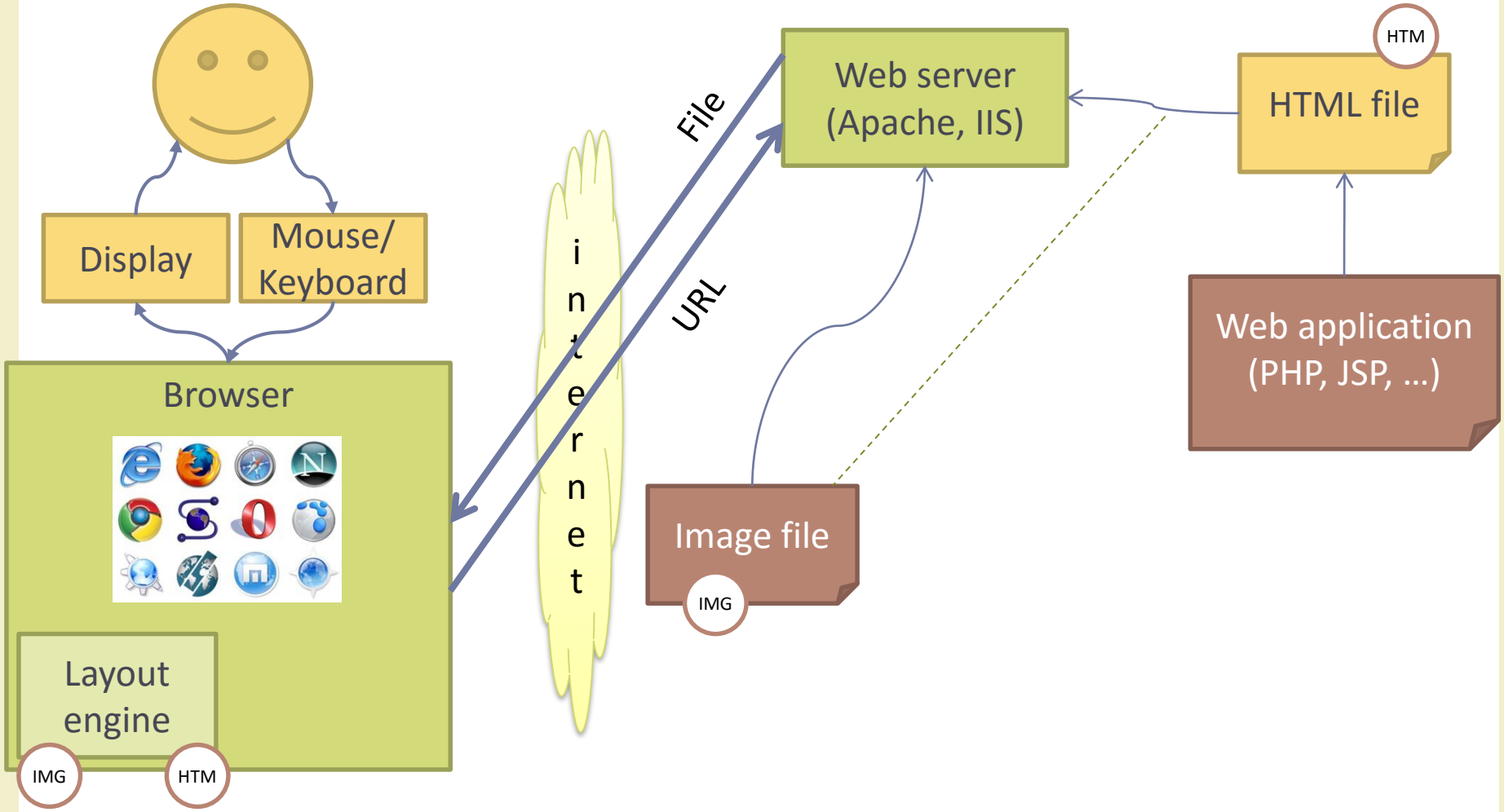
Delay time

- T = Latency + ResponseBytes / Throughput
- This equation is valid if:
  - The other architecture elements (I/O subsystem, network, ...) are not overloaded
  - The web server has not yet reached its maximum workload
- Example:
  - Latency: 0,1s
  - ResponseBytes : 100kBytes
  - Throughput: 800kBytes/s
  - T= 0,1s+ 100KBytes / 800KBytes/s =0,225s

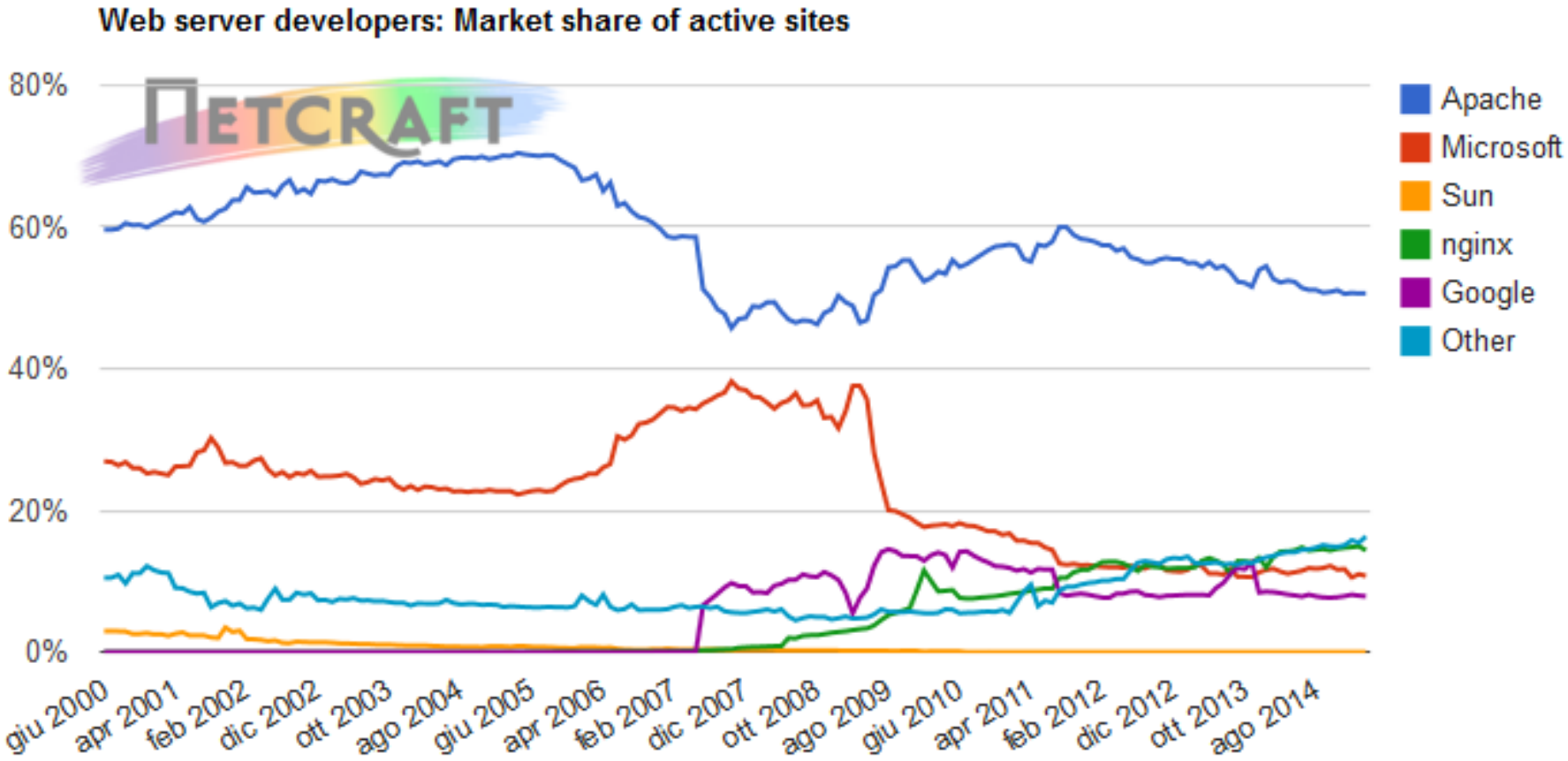
Static web transaction



General Architecture



The most adopted web servers



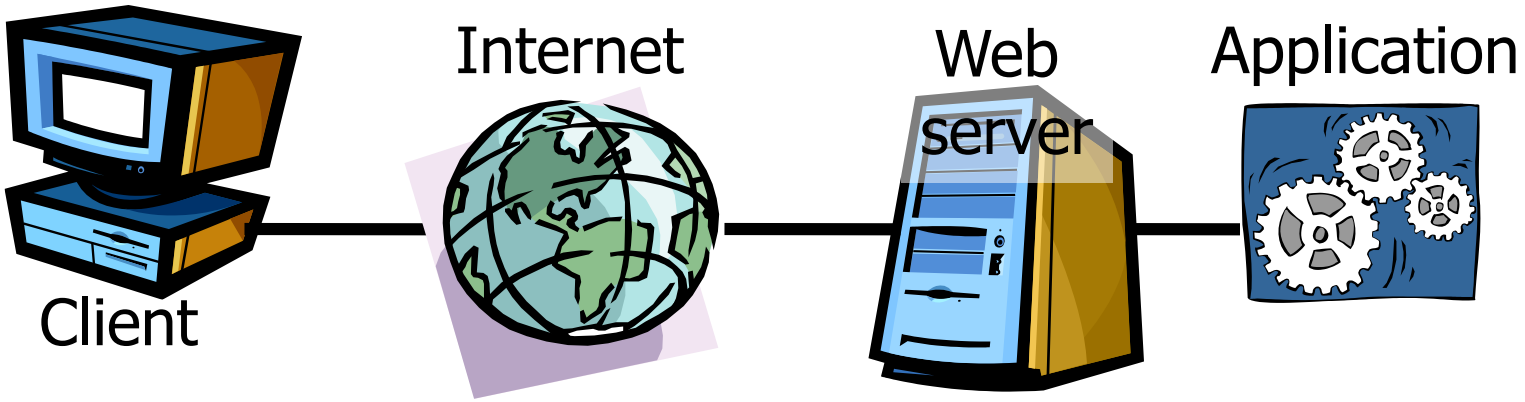
Source: http://news.netcraft.com/

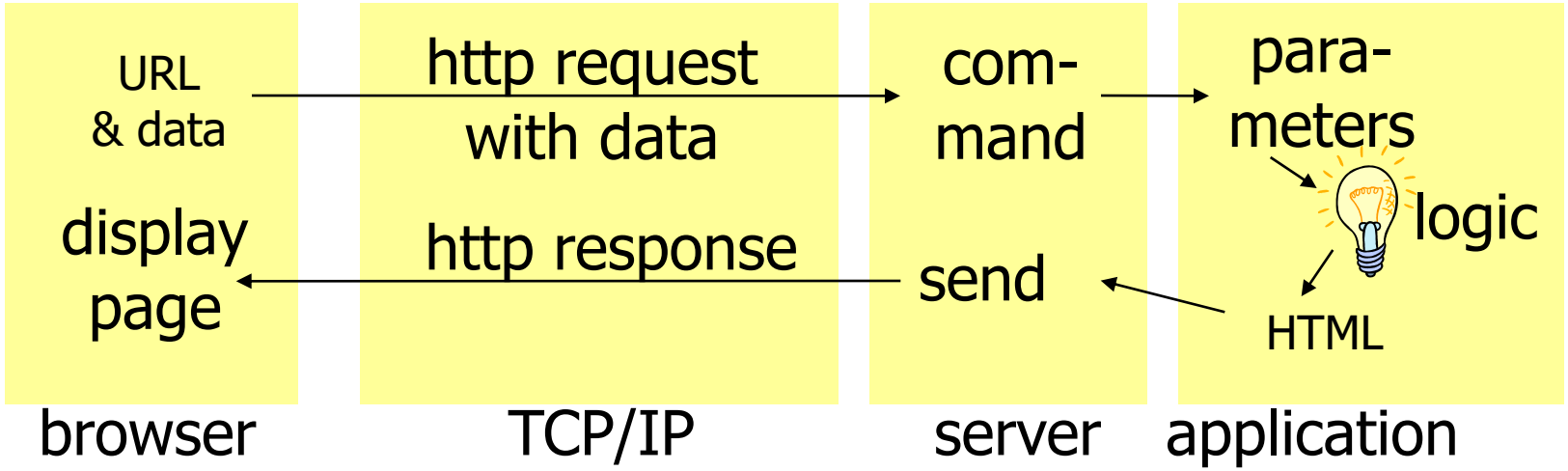
http://news.netcraft.com/archives/2015/03/19/march-2015-web-server-survey.html

Application server

- Dynamic page generation and content generation
- Manages the site business logic
- It’s the middle tier between the client browser and the data residing on a database
- Implements the session mechanisms
- Different technologies and architectures are available

Dynamic web transaction

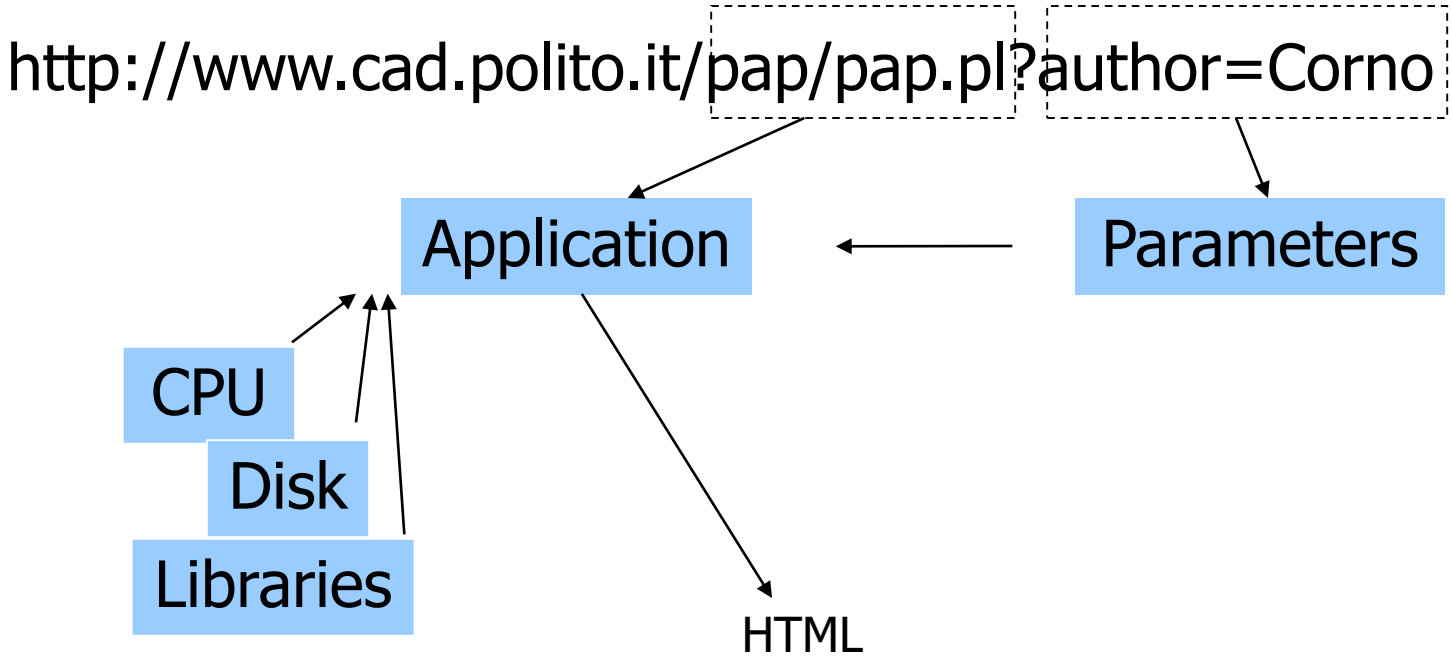




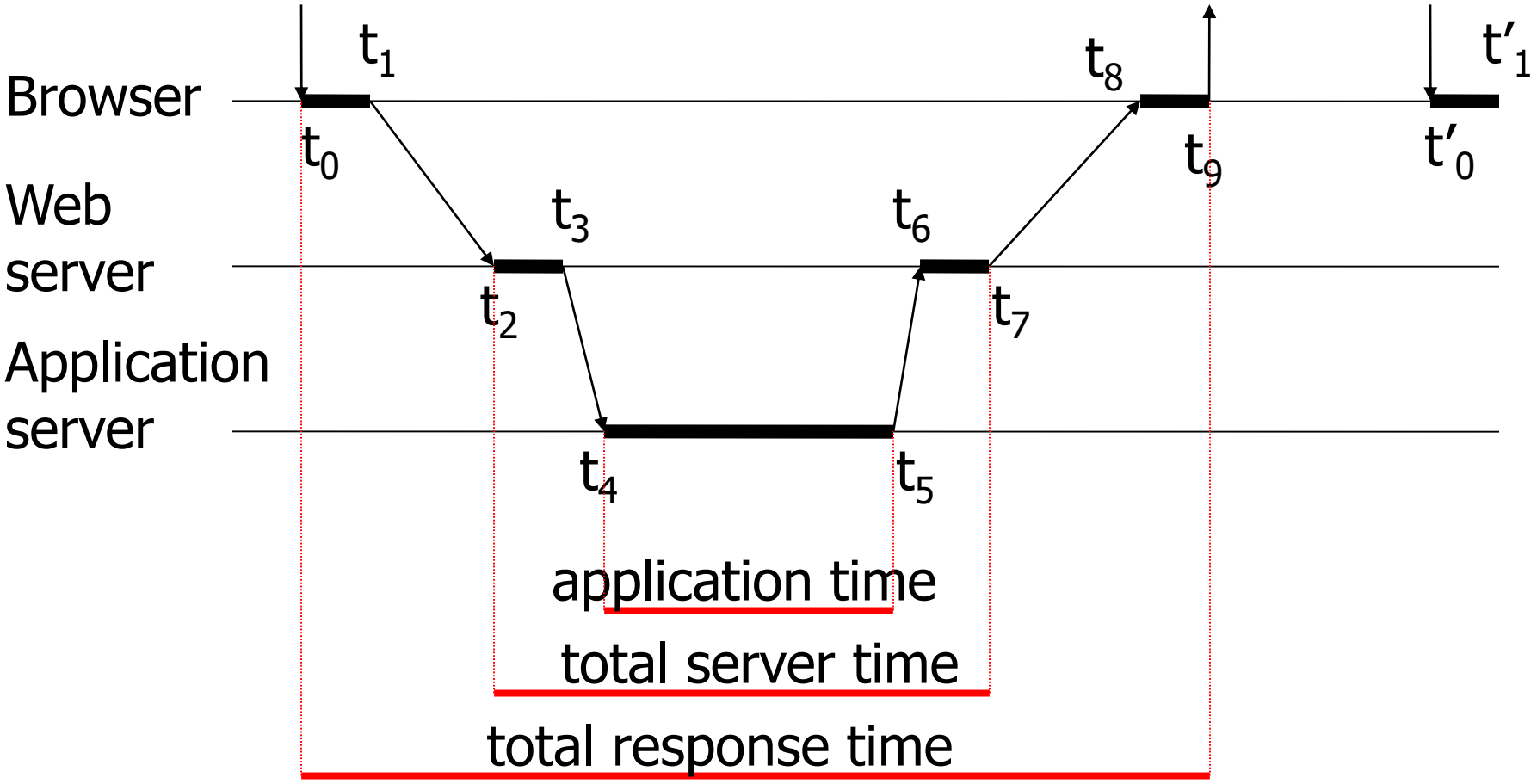
Adopted standards

- HTTP-POST for sending user-specified data
  - In addition to URL-encoding in GET requests
- Technologies for integrating application logic into web servers
  - Obsolete: CGI (common gateway interface), ISAPI (internet information server application programming interface), server-side script
  - java servlets
  - ASP (active server pages), JSP, PHP, PERL, Python as new languages for application development

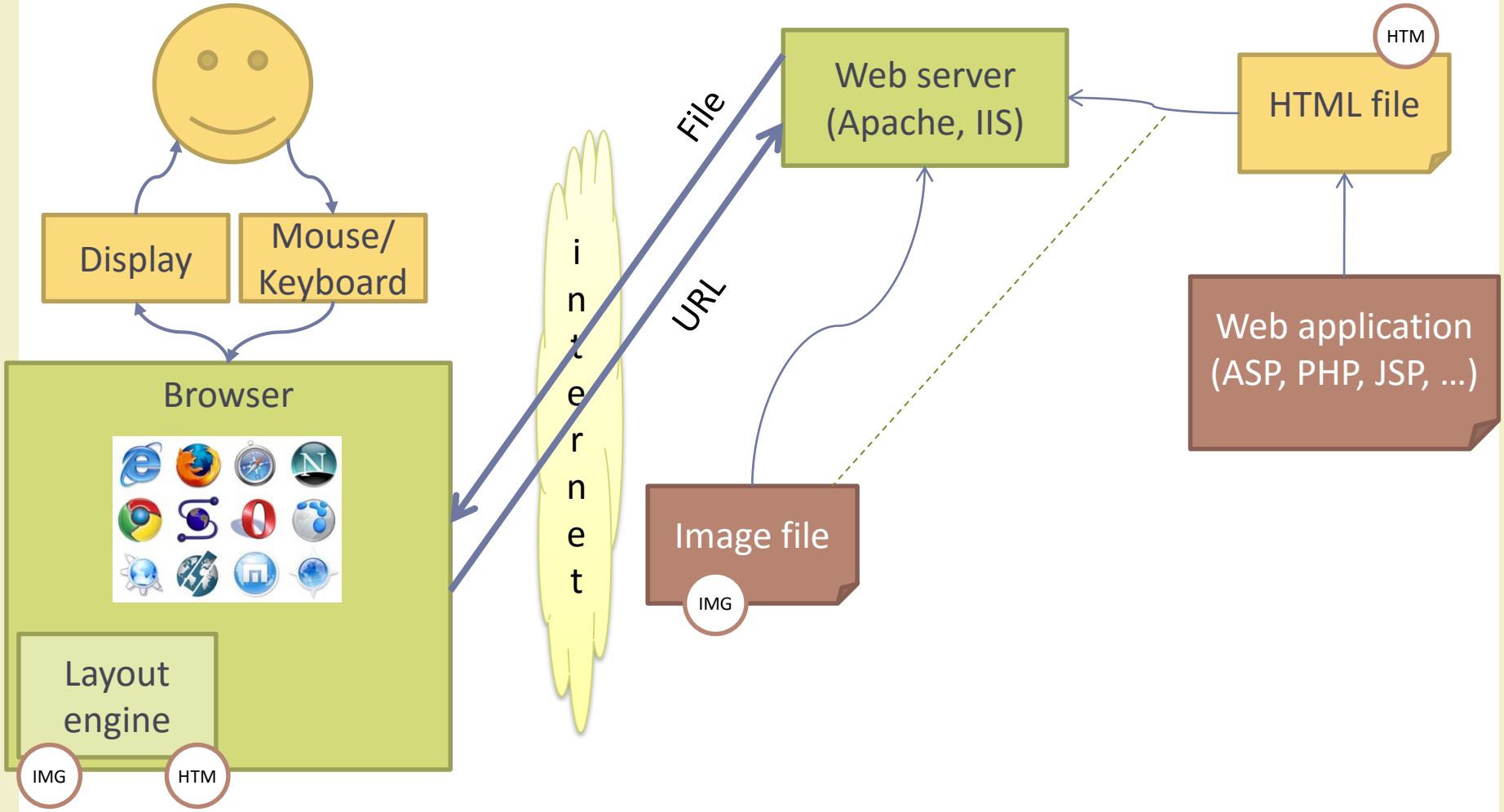
URL (HTTP GET)


http://www.cad.polito.it/pap/pap.pl?author=Corno
Application
Parameters
CPU
Disk
Libraries
HTML

Dynamic web transaction



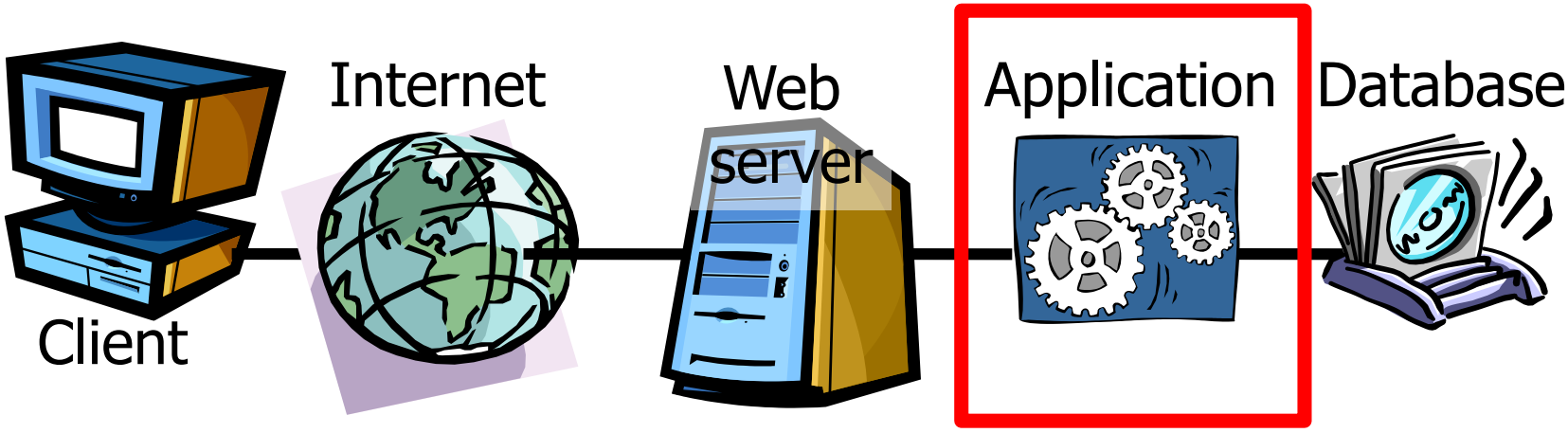
General Architecture

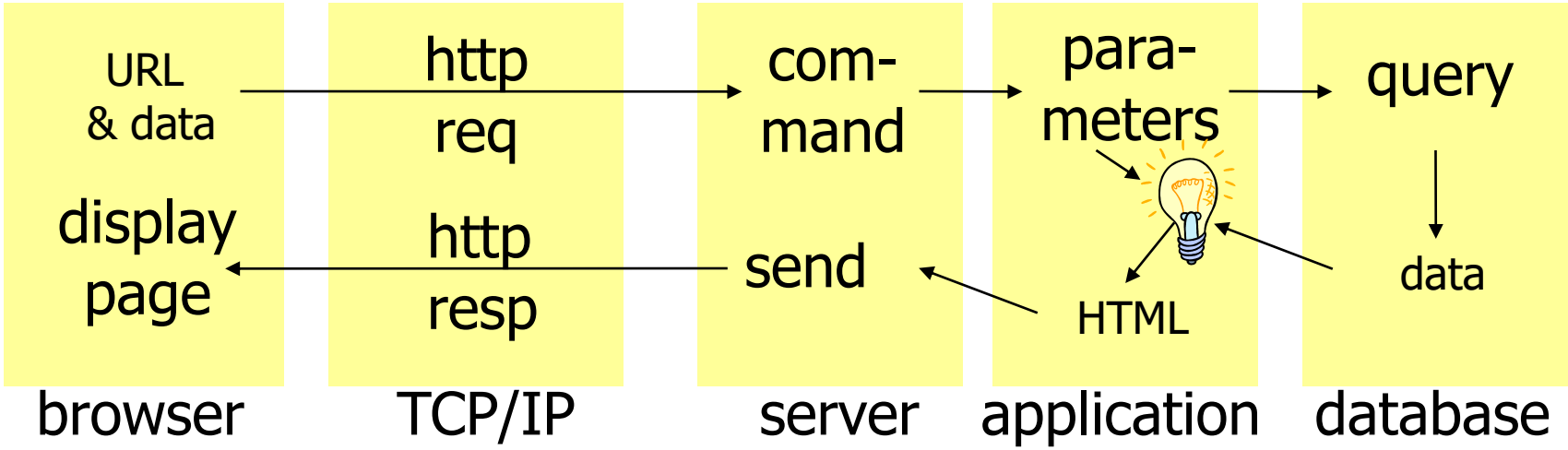


Database server

- Stores the data on which the application server works.
- Executes the queries issued by the application server:
  - Updates the stored data
  - Inserts new data
  - Provides back query results
- The most frequent/complex queries can be implemented internally as stored procedures (pre-compiled queries with parameters)

Example


Internet
Web server
Application
Database
Client


URL & data
http req
com-mand
para-meters
query
display page
http resp
send
data
HTML
browser
TCP/IP
server
application
database

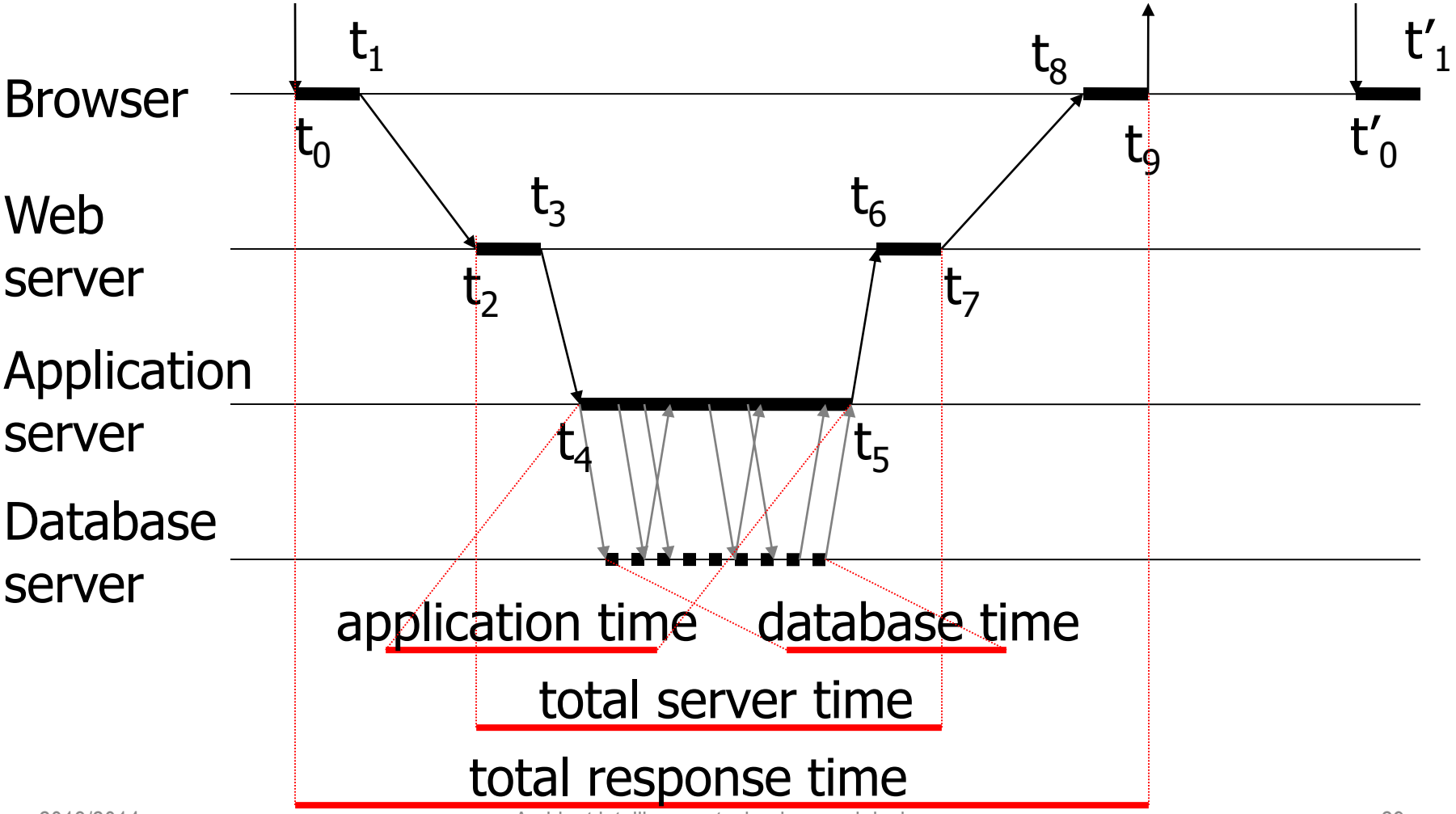
Adopted standards

- Cookies for storing the state of a session
- Java, JavaScript, ActiveX, Flash to program the user interface on the browser
- SQL (structured query language), ODBC (open database connectivity) to access data bases

Database server

- Queries are almost always in SQL
  - SELECT * FROM table;
  - ….
- Often adopts the relational database model
  - Other models can be used
    - Object model
    - Triple model
- The most advanced/complete solutions are called Transaction servers

Database-driven transaction



Example (PHP)

The application composes the query

```
<?php
$query = “SELECT doc_id FROM key_doc_index, keywords WHERE
key_doc_index.key_id = keywords.id AND keywords.key =
$_REQUEST[“query”];”;
```

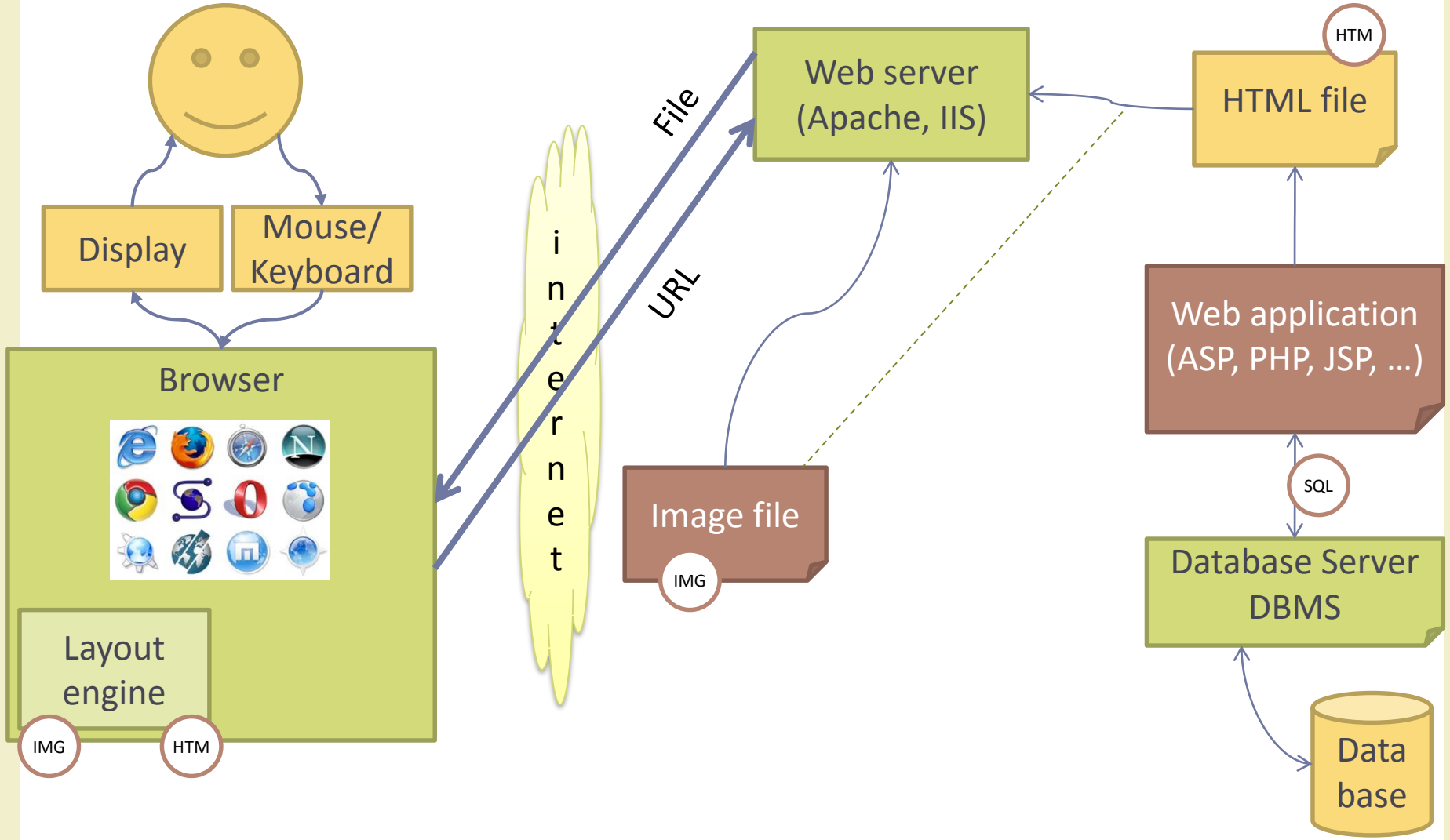
The query is sent to the db-server and a rowset containing the results is returned

```
$rowset = mysql_query($query);
```

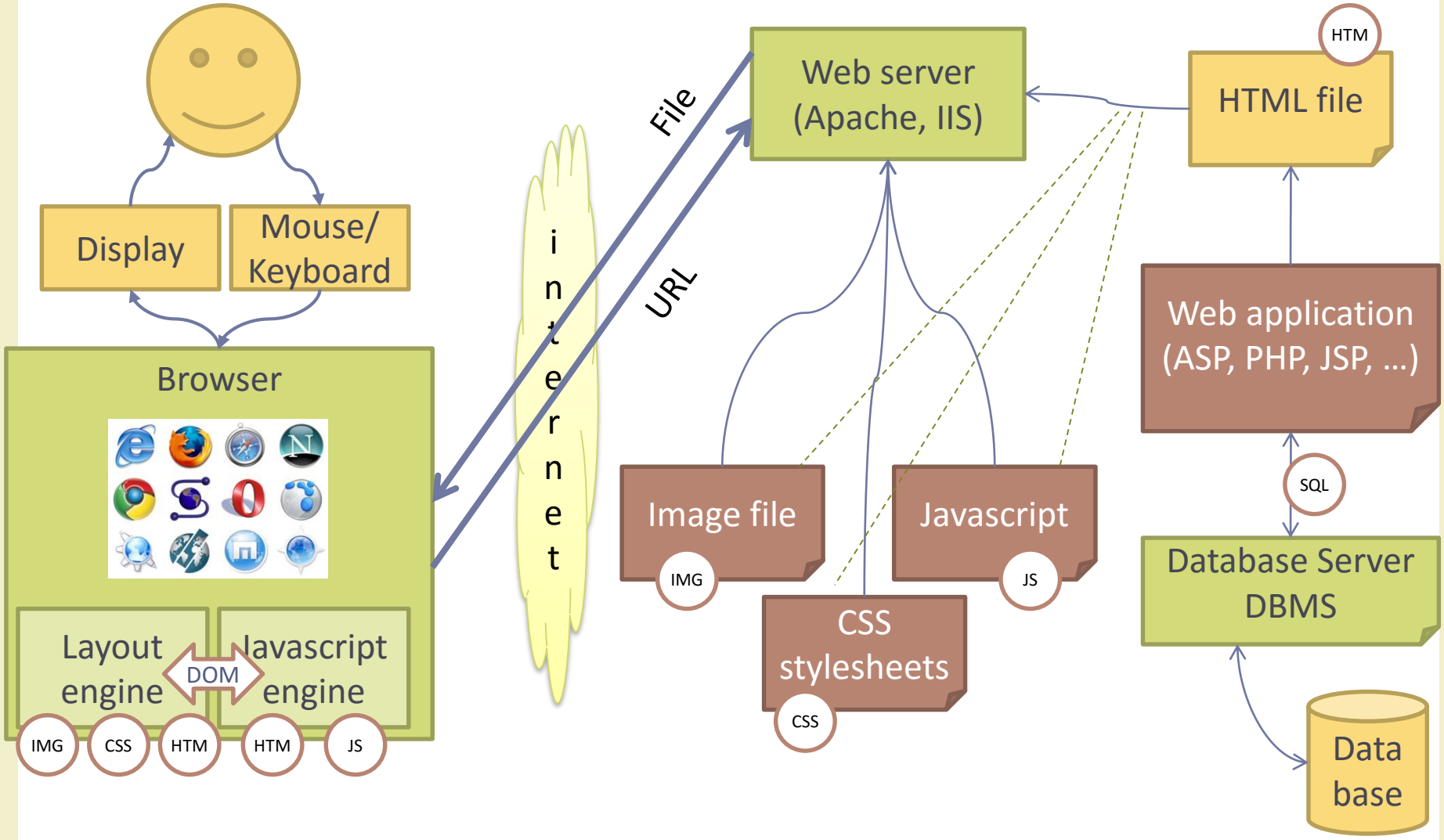
```
while($row = mysql_fetch_row($rowset))
{
//elaborate data
}
?>
```

The application elaborates the data

General Architecture



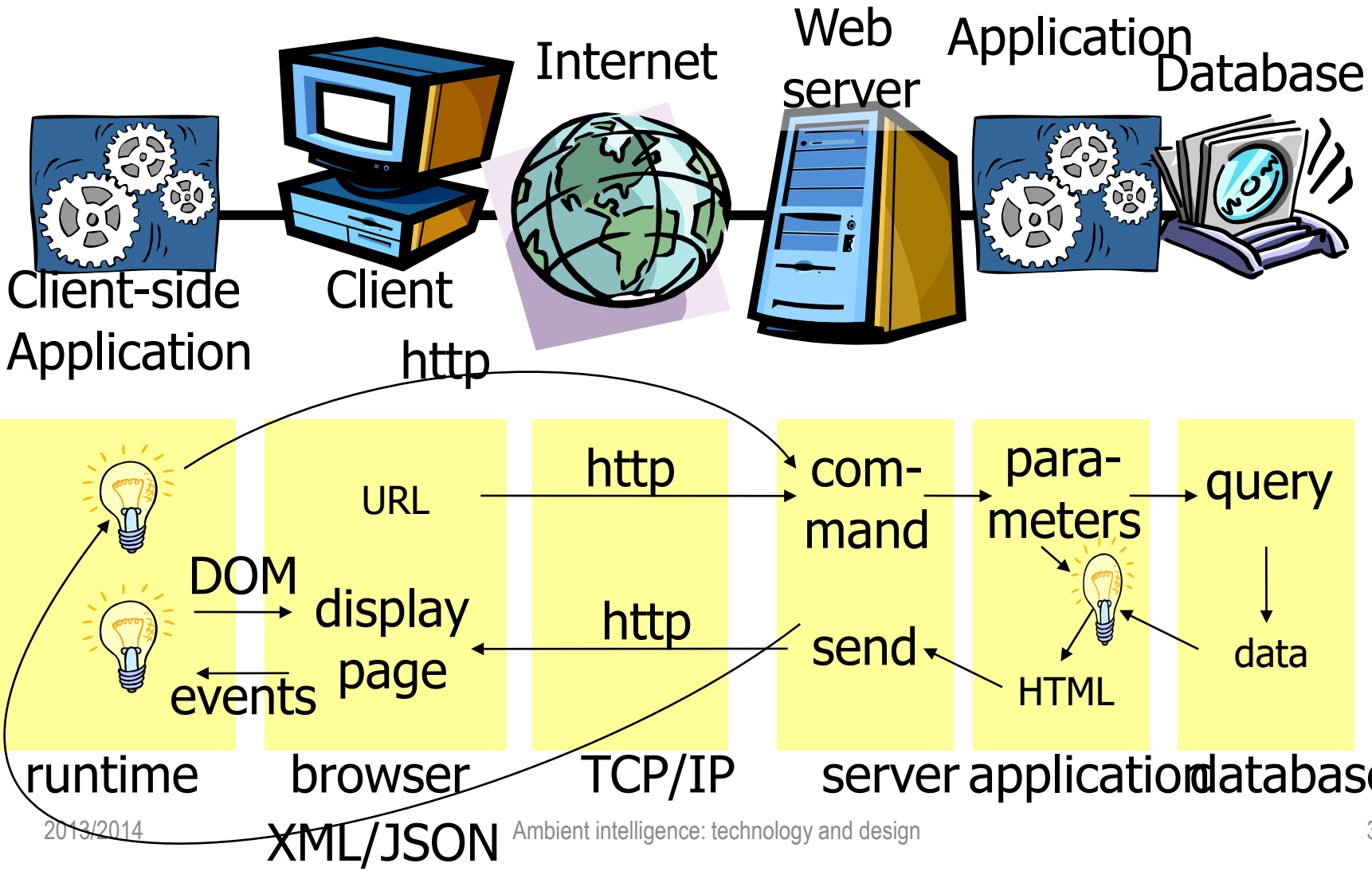
General Architecture



Web 2.0

- Web applications support social interaction models
- Peer exchange and user-contributed content instead of rigid publisher/reader pattern
  - Online communities
- Rich, dynamic, interactive user interfaces
- Integration of contents across web sites (mashups)

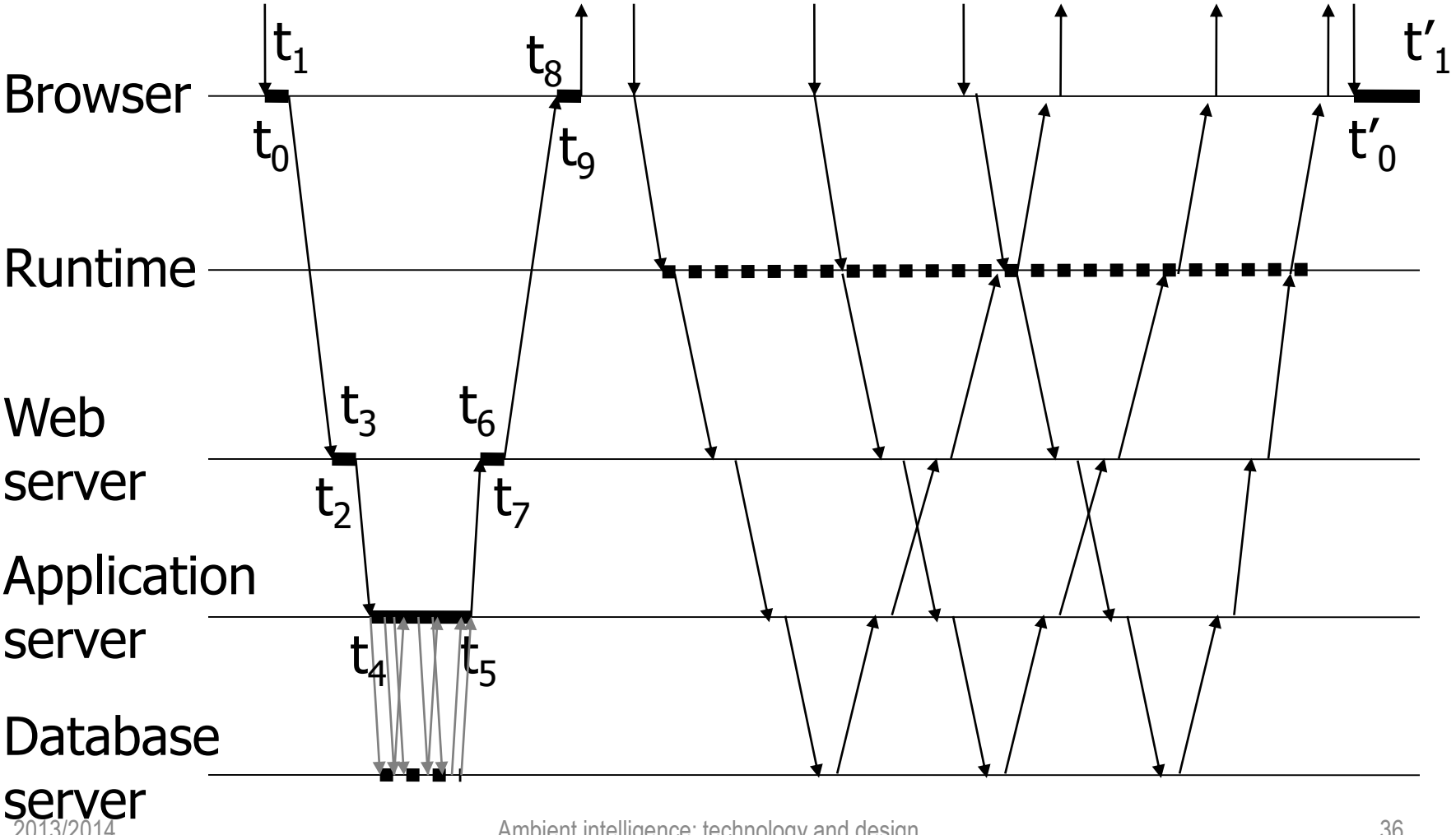
Rich-Client Asynchronous Transactions



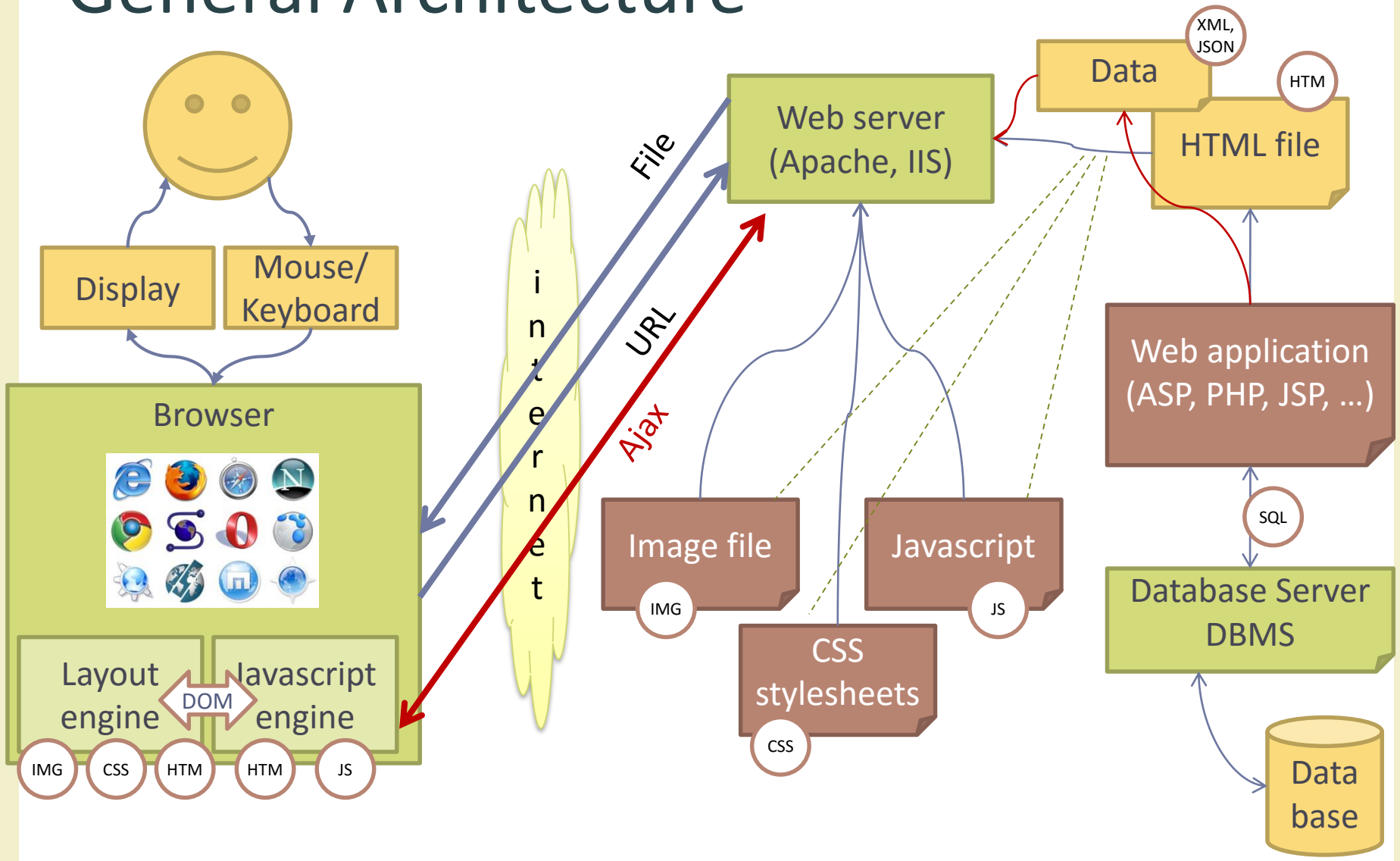
Adopted standards

- Dynamic HTML: DOM, Javascript, CSS
  - JavaScript, Flash to handle a runtime environment on the browser
  - DOM (XHTML Document Object Model) to allow on-the fly modification of the web page
  - CSS 2.1 to modify attribute and handle objects
- AJAX: Asynchronous Javascript and XML
  - XMLHttpRequest for asynchronous communication to the server
  - Data transfer formats: JSON, XML, RDF, RSS, Atom, FOAF, ...
- Mash-up technology

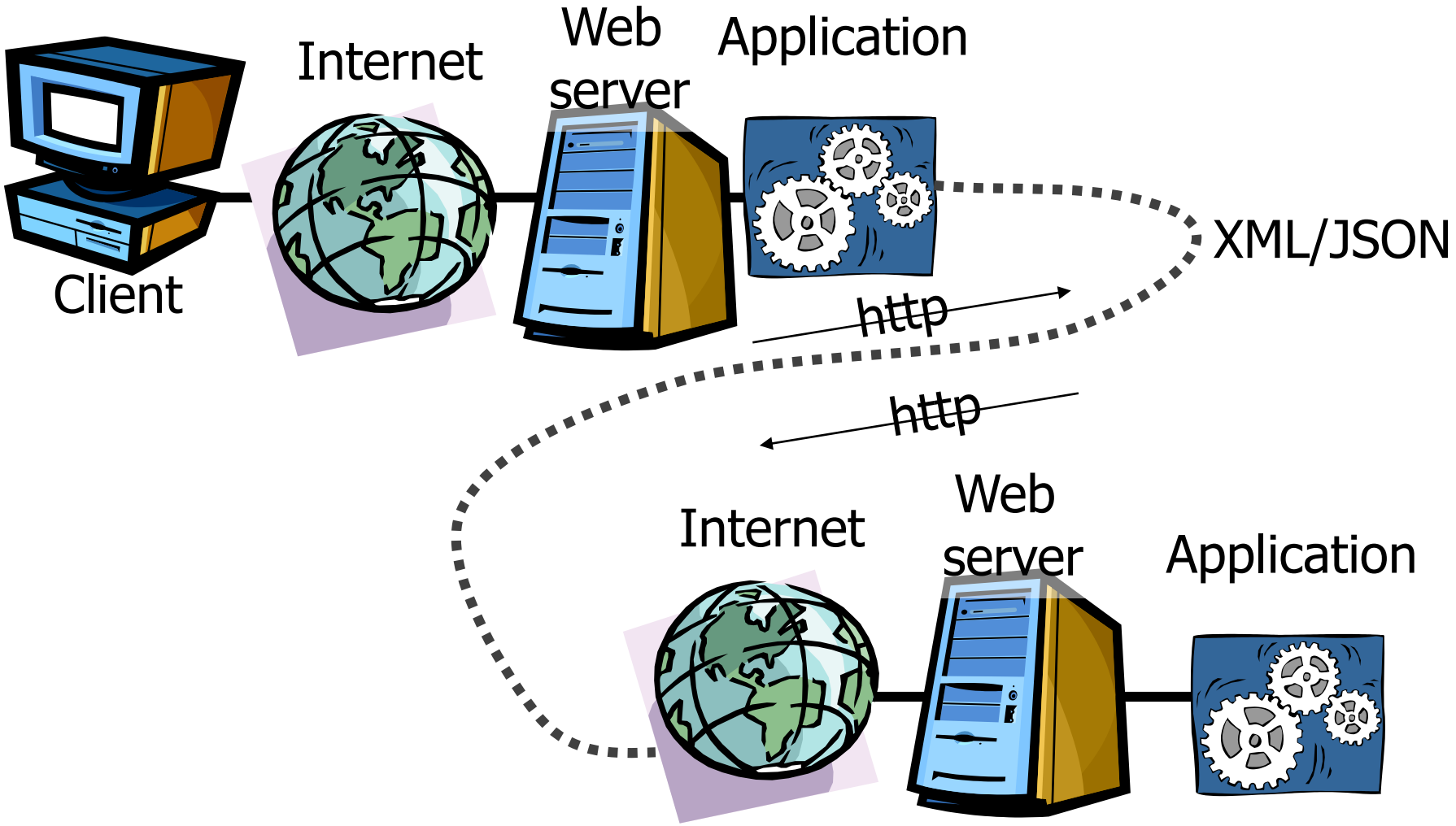
Rich-client transaction



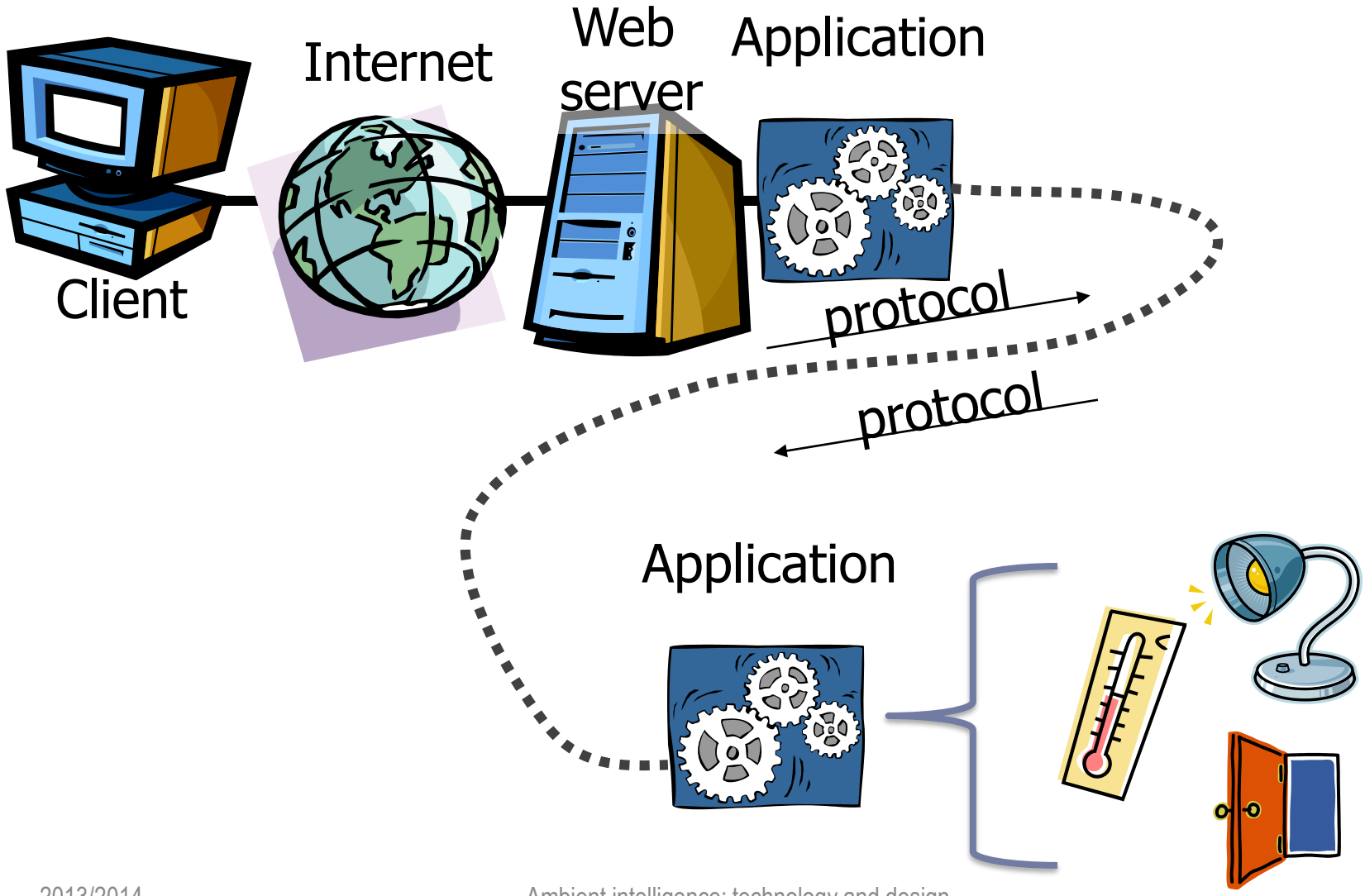
General Architecture



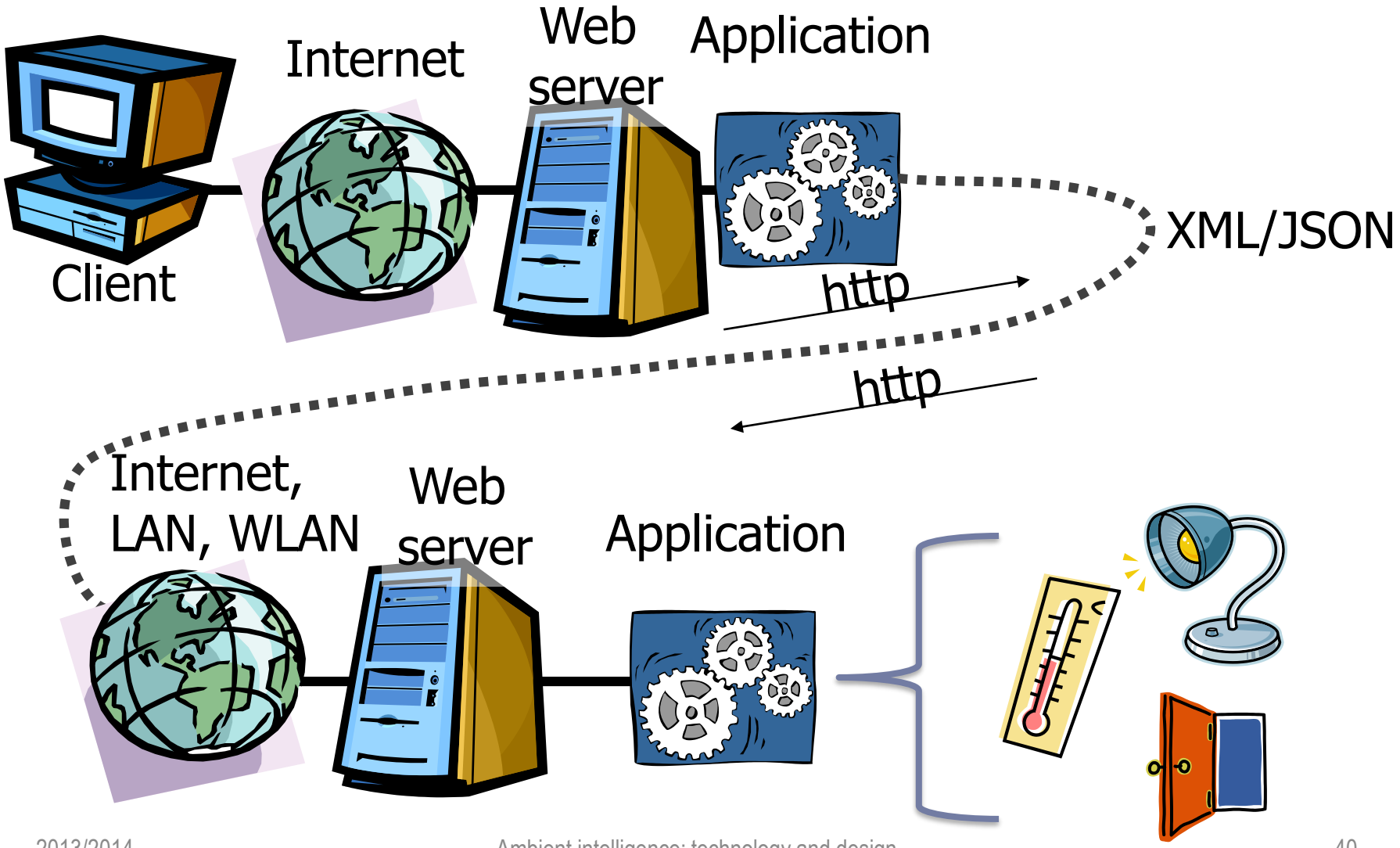
Distributed transactions



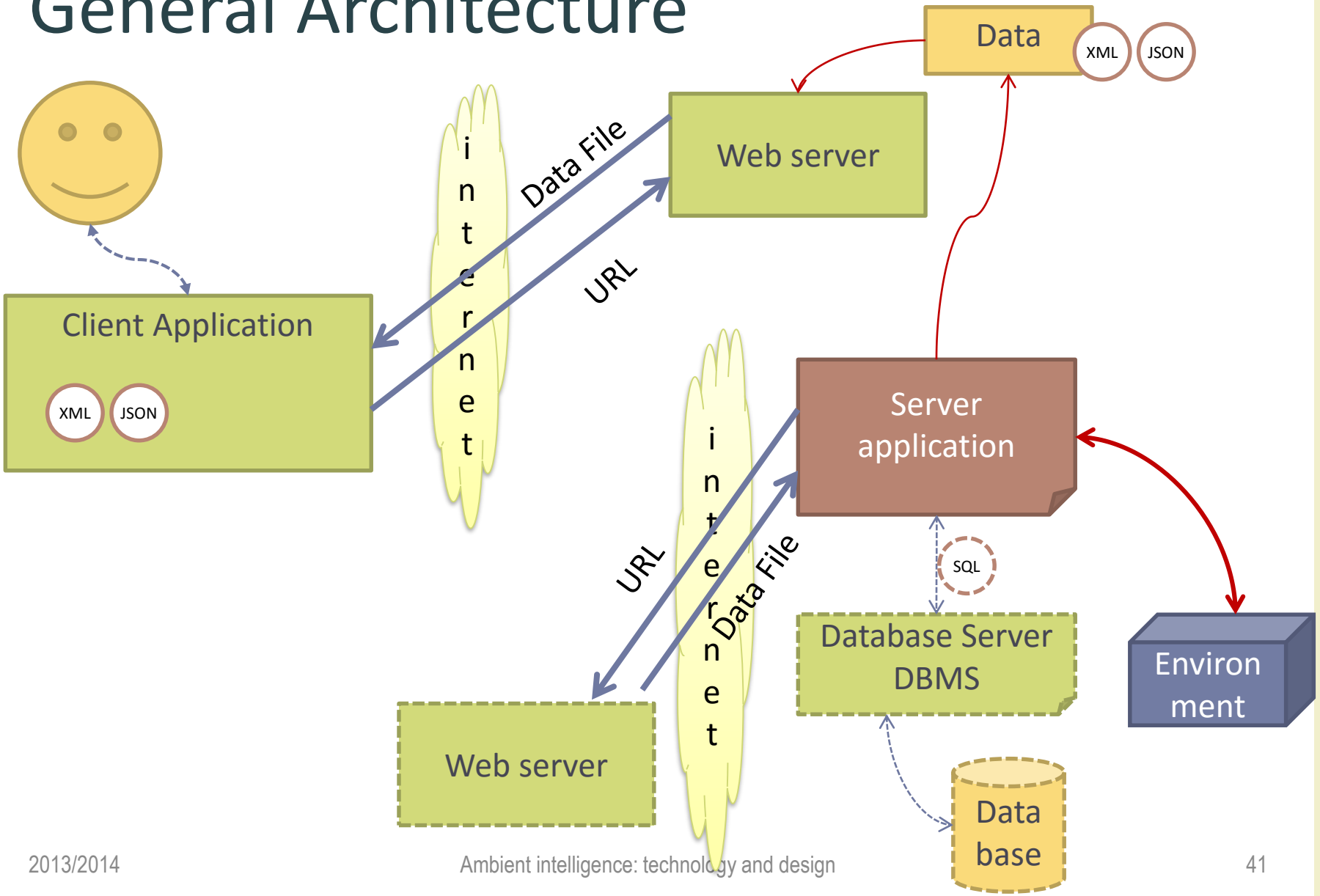
AmI control


Client
Internet
Web server
Application
protocol
protocol
Application

AmI control via http



General Architecture



License

- These slides are distributed under a Creative Commons license “Attribution – NonCommercial – ShareAlike (CC BY-NC-SA) 3.0”
- You are free to:
  - Share — copy and redistribute the material in any medium or format
  - Adapt — remix, transform, and build upon the material
  - The licensor cannot revoke these freedoms as long as you follow the license terms.
- Under the following terms:
  - Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  - NonCommercial — You may not use the material for commercial purposes.
  - ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.
- http://creativecommons.org/licenses/by-nc-sa/3.0/