

Summary

1. Definition
2. Application Domains
3. Reference Architecture



Definition

Web Information Systems

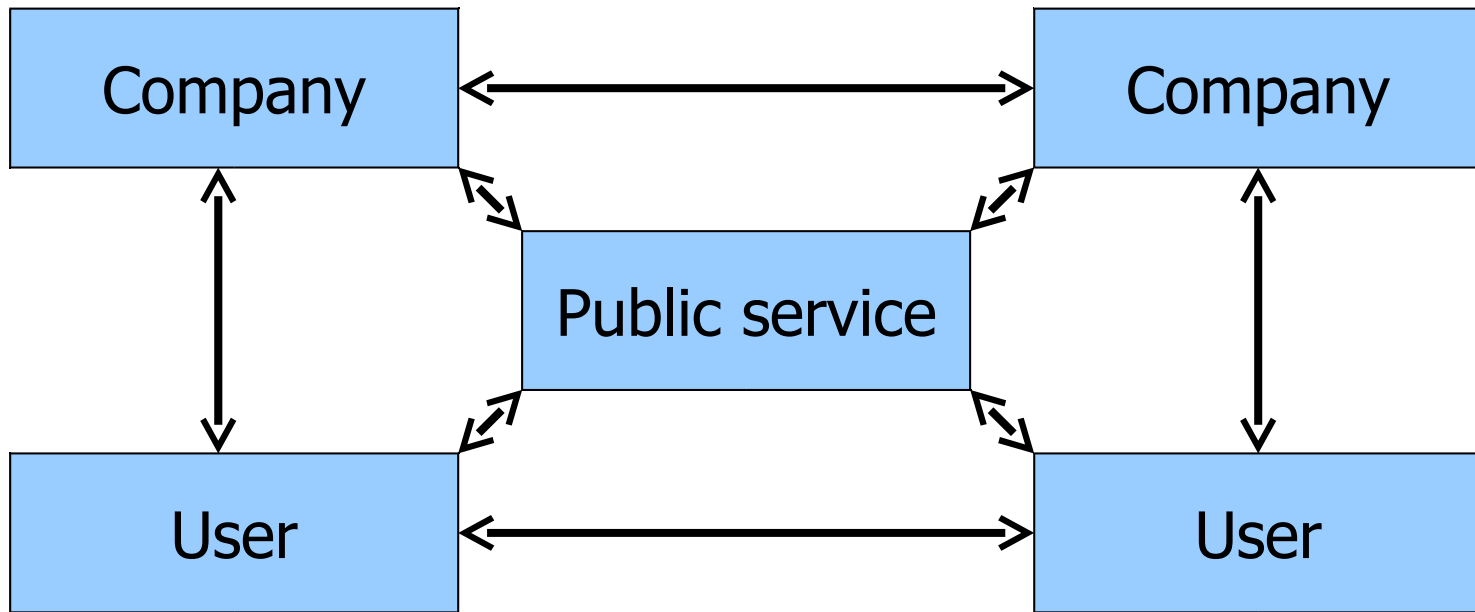
Definition

- ▶ **Web Information System (WIS)**
 - ▶ Communication between computers and hosts takes place in the Internet or through a Virtual Private Network (VPN) based on the internet standards
 - ▶ Access to information and services is supported by program that manage the user interface, known as browser



Cap. 3
Pag. 93

Actors



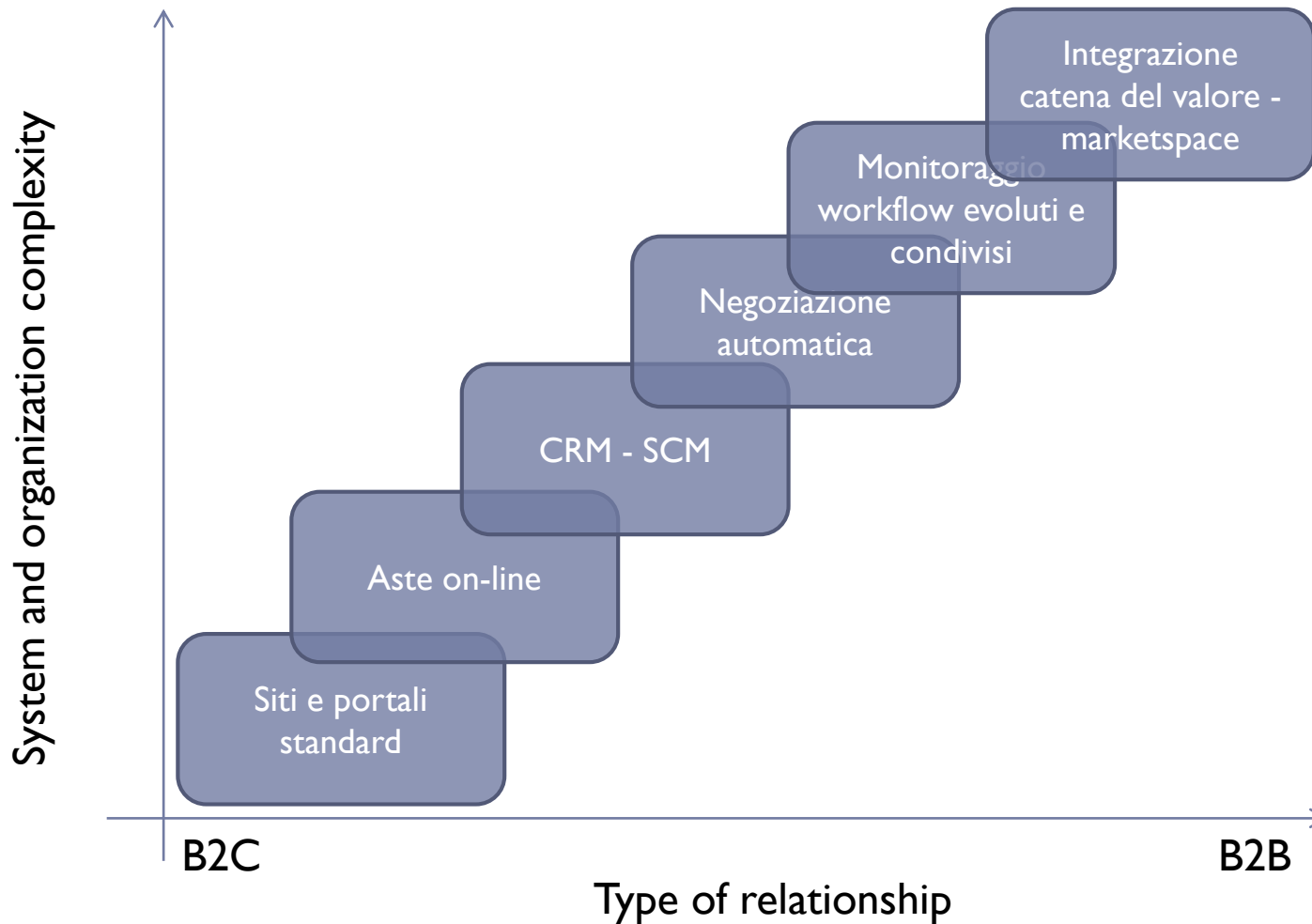
Collaboration models

- ▶ **B2B (business to business):** collaboration among companies
- ▶ **B2C (business to consumer):** on-line shops
- ▶ **C2C (consumer to consumer):** auctions, buy-sell notices
- ▶ **Government to business :** on-line taxes, services to companies
- ▶ **Government to citizens :** on-line taxes

Examples

- ▶ On-line shops of consumer goods
- ▶ On-line auctions
- ▶ Thematic portal (links, user community, latest news)
- ▶ Distribution of components or raw materials
- ▶ Services (bank, finance, insurance, travel, consultancy, ...)
- ▶ Publications (newspapers, encyclopedias, press agencies, ...)

A possible taxonomy



Levels of complexity

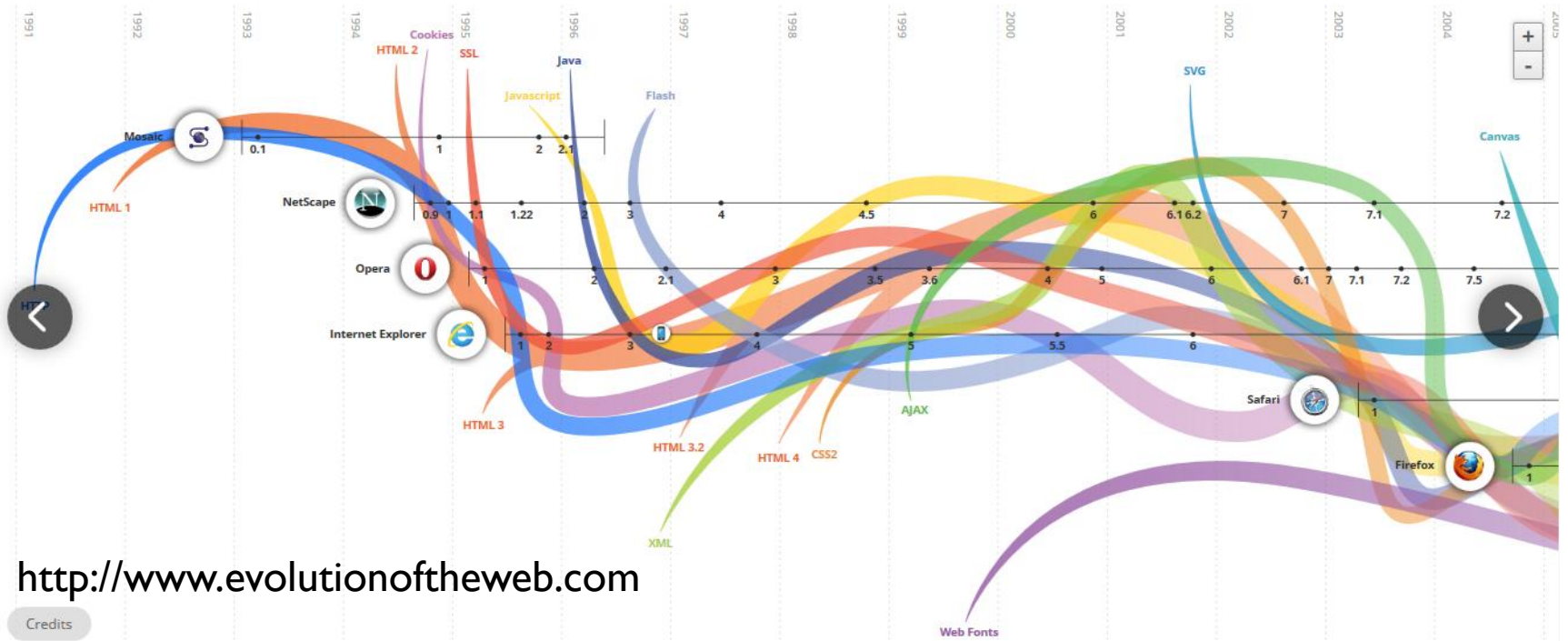
- ▶ **Informative sites**
 - ▶ Who we are / Products / Services / Contacts
 - ▶ Newsletter, Journal, Blog, ...
- ▶ **Ordering sites**
 - ▶ Product selection, configuration, purchase
- ▶ **Management systems**
 - ▶ CRM, SCM, ERP, MRP, ...
- ▶ **Autonomous systems**
 - ▶ Negotiation, transaction, monitoring
- ▶ **Portals, marketplace, marketSPACE**
 - ▶ Aggregation of several related companies/products



Reference Architecture

Web Information Systems

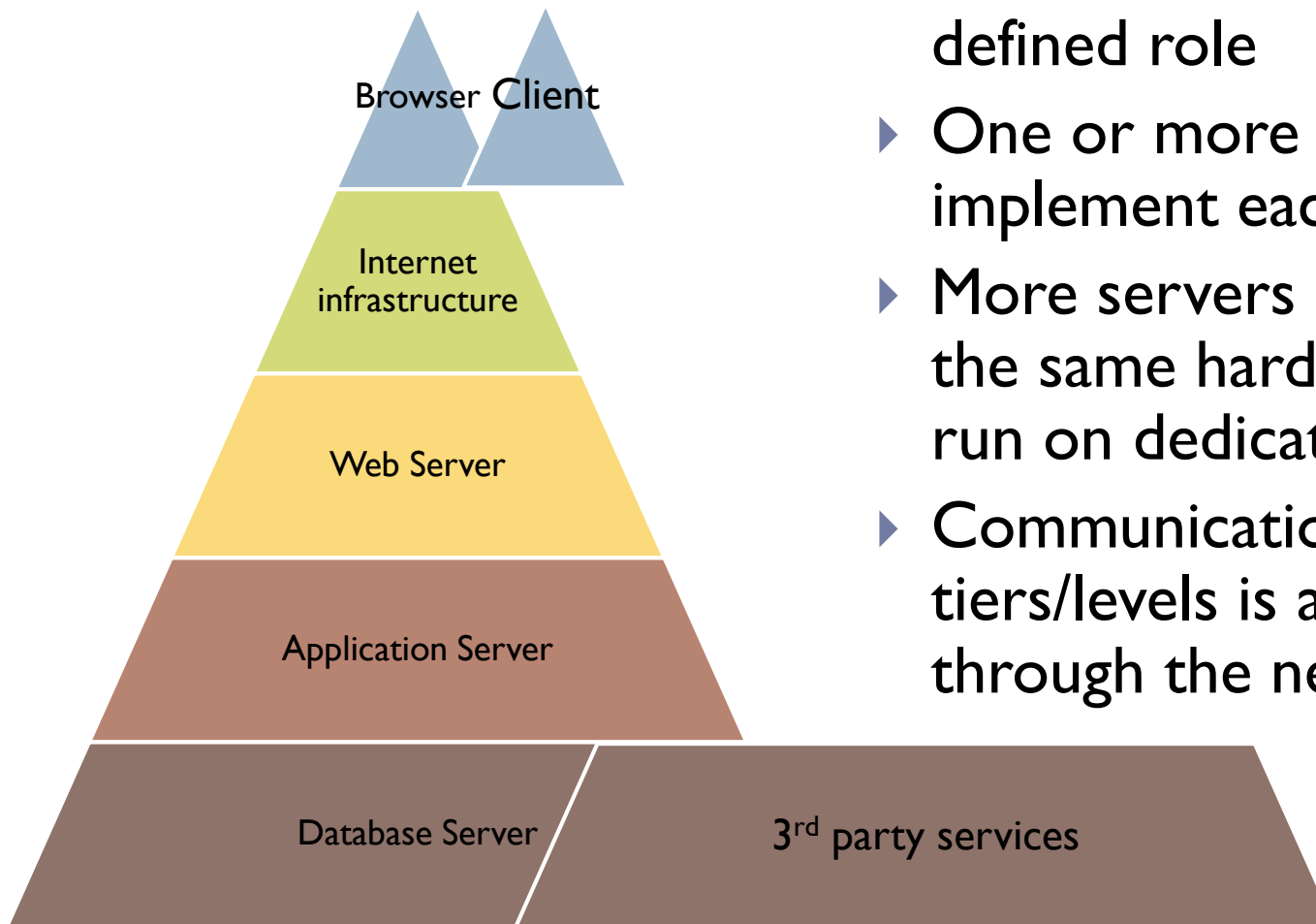
Evolution of web architectures



<http://www.evolutionoftheweb.com>

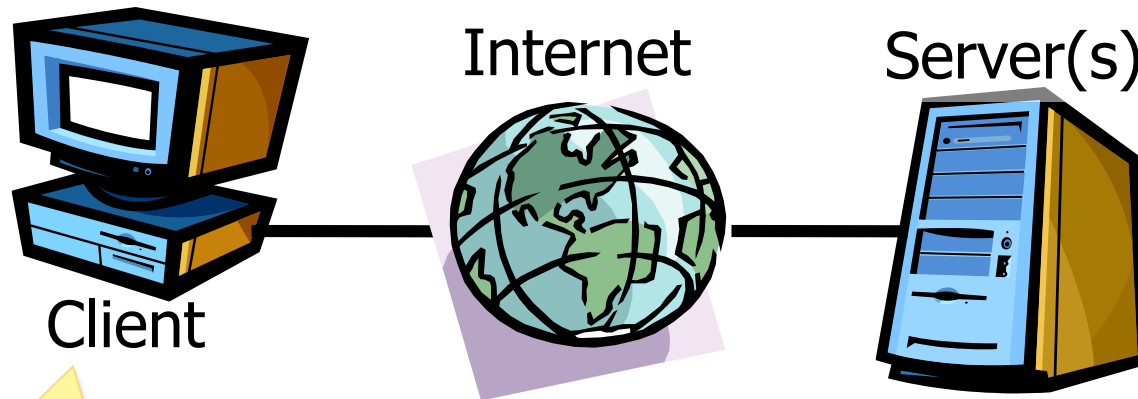
Credits

N-tier (N-level) architecture



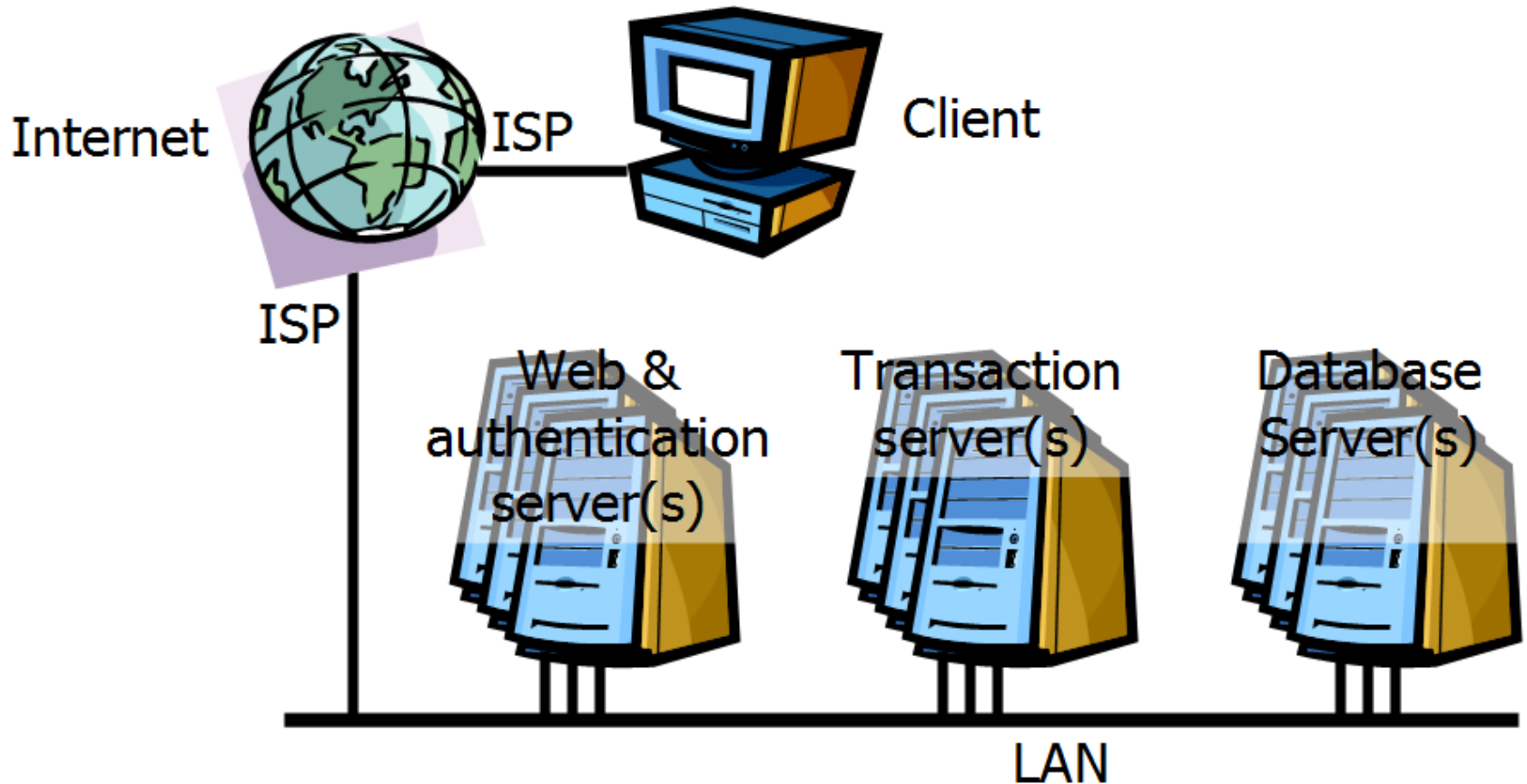
- ▶ Each level/tier has a well defined role
- ▶ One or more servers implement each tier/layer
- ▶ More servers can share the same hardware or can run on dedicated devices
- ▶ Communication between tiers/levels is achieved through the network

General Architecture



- Historically, a web browser
- Also:
 - Mobile app
 - Desktop app
 - Other server application

General Architecture



Components

- ▶ One or more connections to the Internet by means of an Internet Service Provider (ISP).
- ▶ One or more servers implementing each tier/level of the architecture.
- ▶ One or more physical networks for interconnecting the servers.
- ▶ One or more network devices (router, firewall, switch) which implement communication and security policies.

Definition

- ▶ “Server” may be defined as:

- ▶ Logical definition:

- A **process** that runs on a host that relays information to a **client** upon the client sending it a **request**.

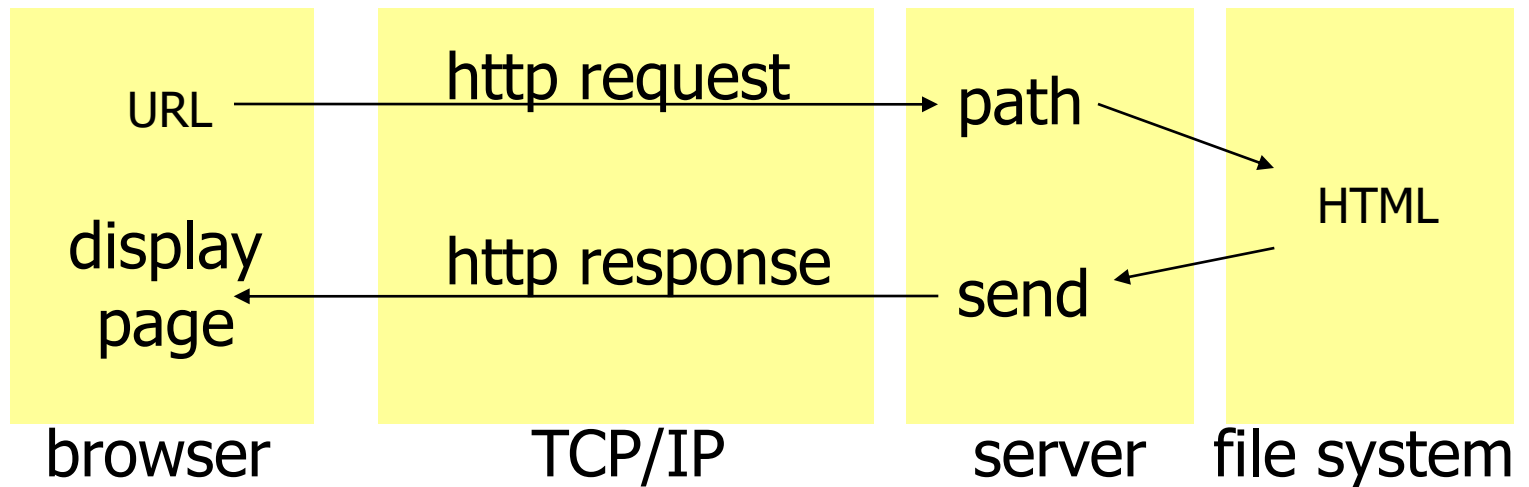
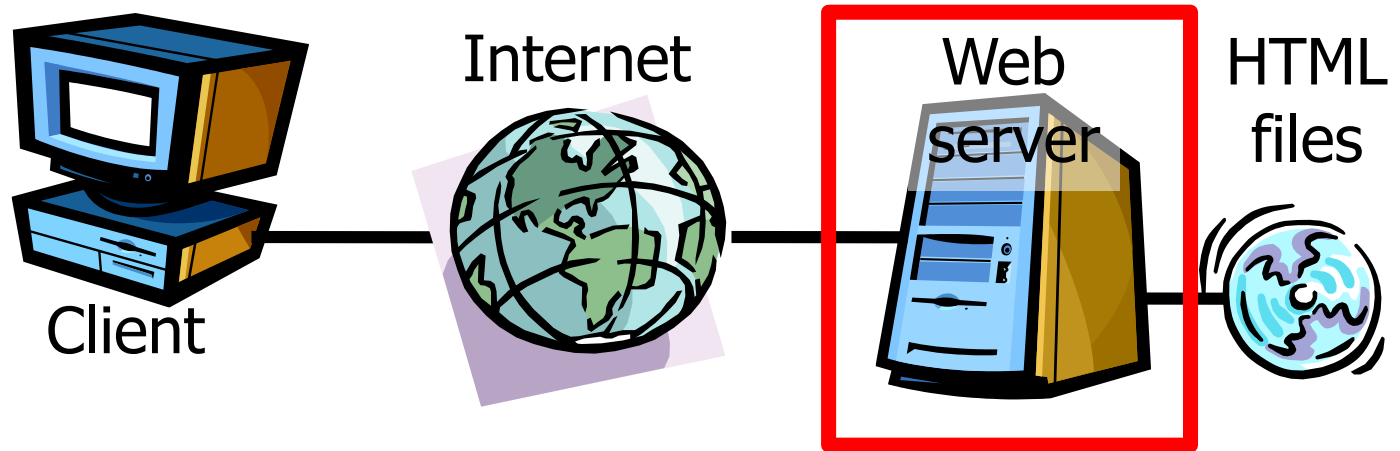
- ▶ Physical definition:

- A **host computer** on a network that holds information (eg, Web sites) and responds to requests for information

Web server

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

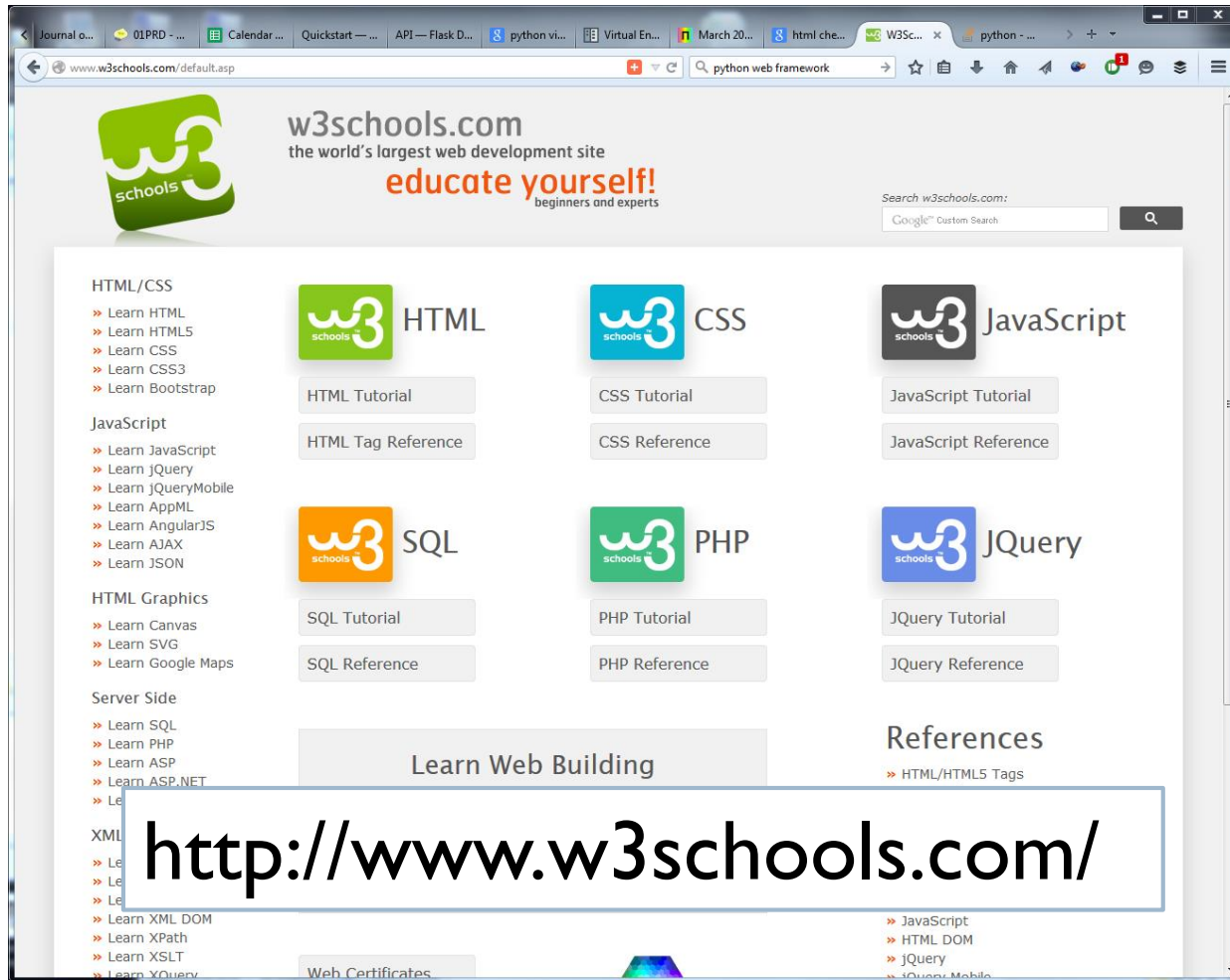
Example



Adopted standards

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

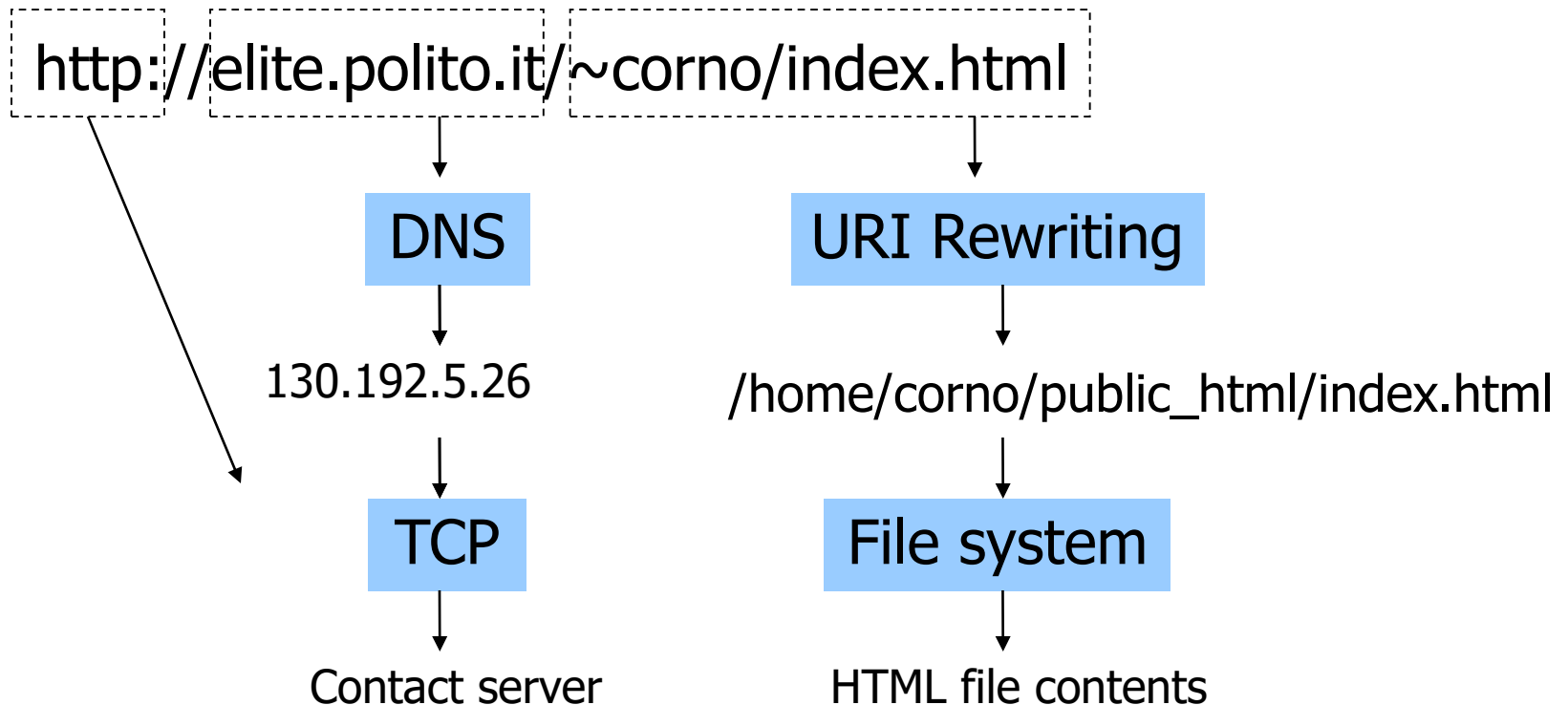
HTML in 5 minutes



URL

RFC 2396

<http://www.w3.org/Addressing/>



URI Basics

- ▶ **http://www.sadev.co.za/users/1/contact**
Scheme Hostname Query
Scheme Hostname Query
- ▶ **http://www.sadev.co.za?user=1&action=contact**
Scheme Userinfo Hostname Port
- ▶ **http://rob:pass@bbd.co.za:8044**
Scheme Hostname Query Fragment
- ▶ **https://bbd.co.za/index.html#about**

HTTP protocol

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

```
GET /~corno/index.html HTTP/1.0
Accept: text/html
Accept: image/gif
User-Agent: FireChrome SuperBrowser 9.45
```

```
HTTP/1.0 200 OK
Date: Monday, 01-Jan-2001 00:00:00 GMT
Server: Apache 1.3.0
MIME-Version: 1.0
Last-Modified: 31-Dec-2000
Content-type: text/html
Content-length: 3021

<HTML> . . .
```


Browser developer tools

The image displays two screenshots of a web browser showing the e-Lite website. The top screenshot shows the website's main content area with three featured items: "SEMINARIO: INDICATORI QUANTITATIVI PER LA VALUTAZIONE DEI PROCESSI", "PUBLICATION: DESIGN RECOMMENDATIONS FOR SMART ENERGY MONITORING", and "PRESENTATIONS AT ACM CHI 2015". The bottom screenshot shows the same website with the browser's developer tools open. The Network tab is active, showing a list of resources. The selected resource is "elite.polito.it", and the Headers section is expanded, displaying the following information:

```
General
Remote Address: 130.192.5.26:80
Request URL: http://elite.polito.it/
Request Method: GET
Status Code: 200 OK

Response Headers
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Connection: Keep-Alive
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
Date: Wed, 08 Apr 2015 13:47:35 GMT
Expires: Mon, 1 Jan 2001 00:00:00 GMT
Keep-Alive: timeout=15, max=100
Last-Modified: Wed, 08 Apr 2015 13:47:35 GMT
P3P: CP="NOI ADM DEV PSAI COM NAV OUR OTRP STP IND DEM"
Pragma: no-cache
Server: Apache/2.4.6 (Linux/SUSE)
Transfer-Encoding: chunked
X-Powered-By: PHP/5.4.20
```

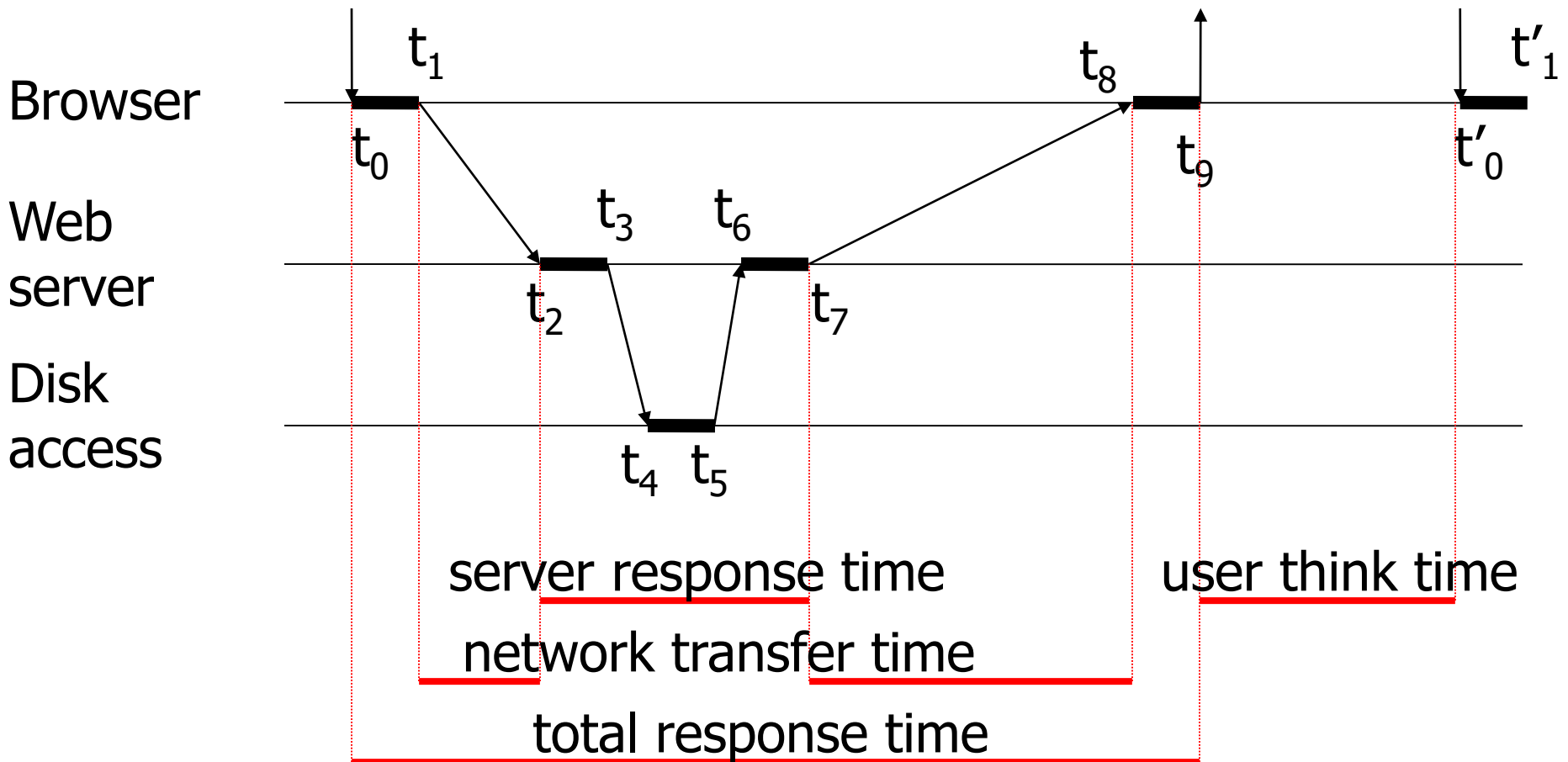
Performance measures

- ▶ 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

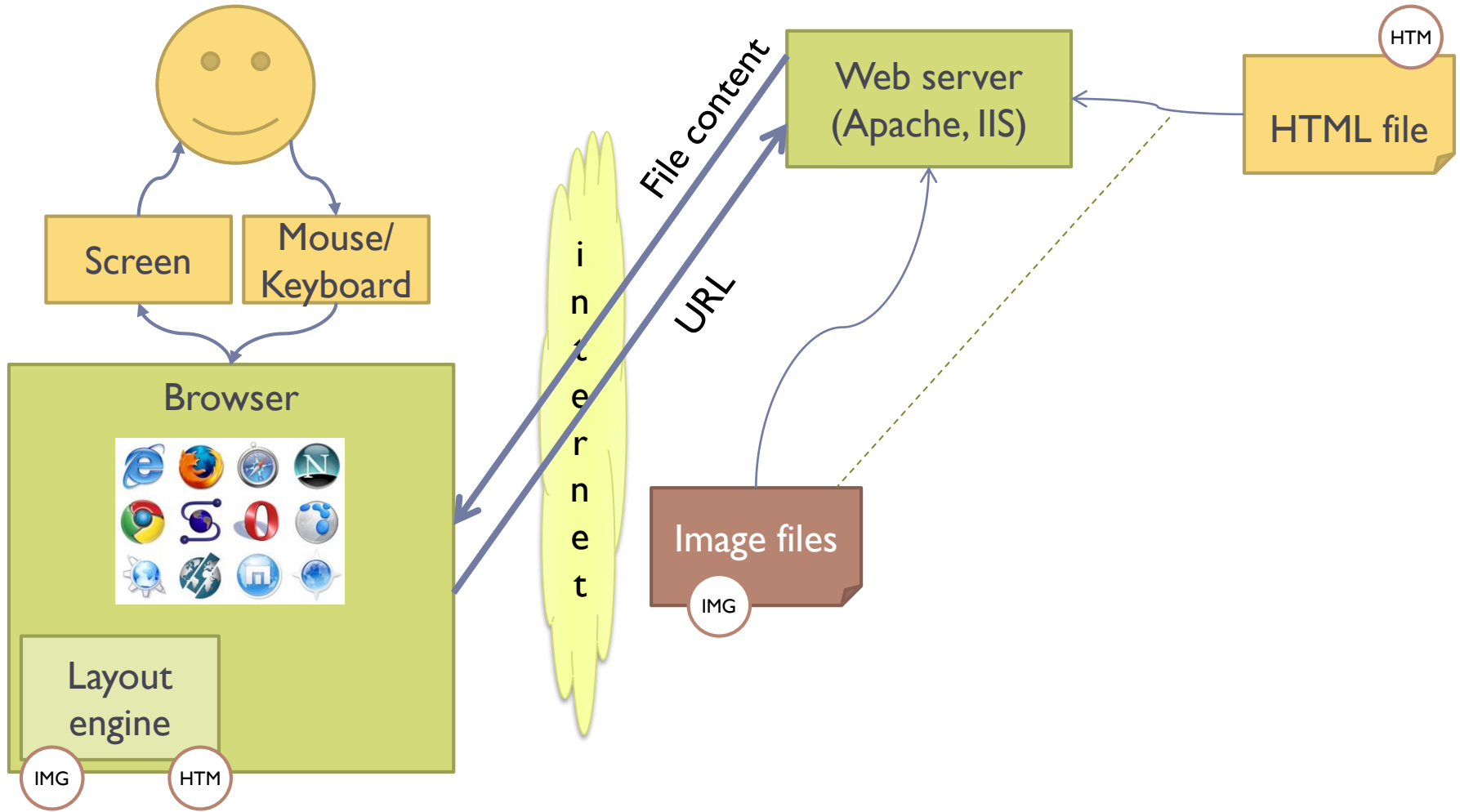
Delay time

- ▶ $T = \text{Latency} + \text{Size}_{\text{HTML}} / \text{Throughput}$
- ▶ 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: 0,1 s
 - ▶ $\text{Size}_{\text{HTML}}$: 100kBytes
 - ▶ Throughput: 800kBytes/s
 - ▶ $T = 0,1\text{s} + 100\text{KBytes} / 800\text{KBytes/s} = 0,225\text{s}$

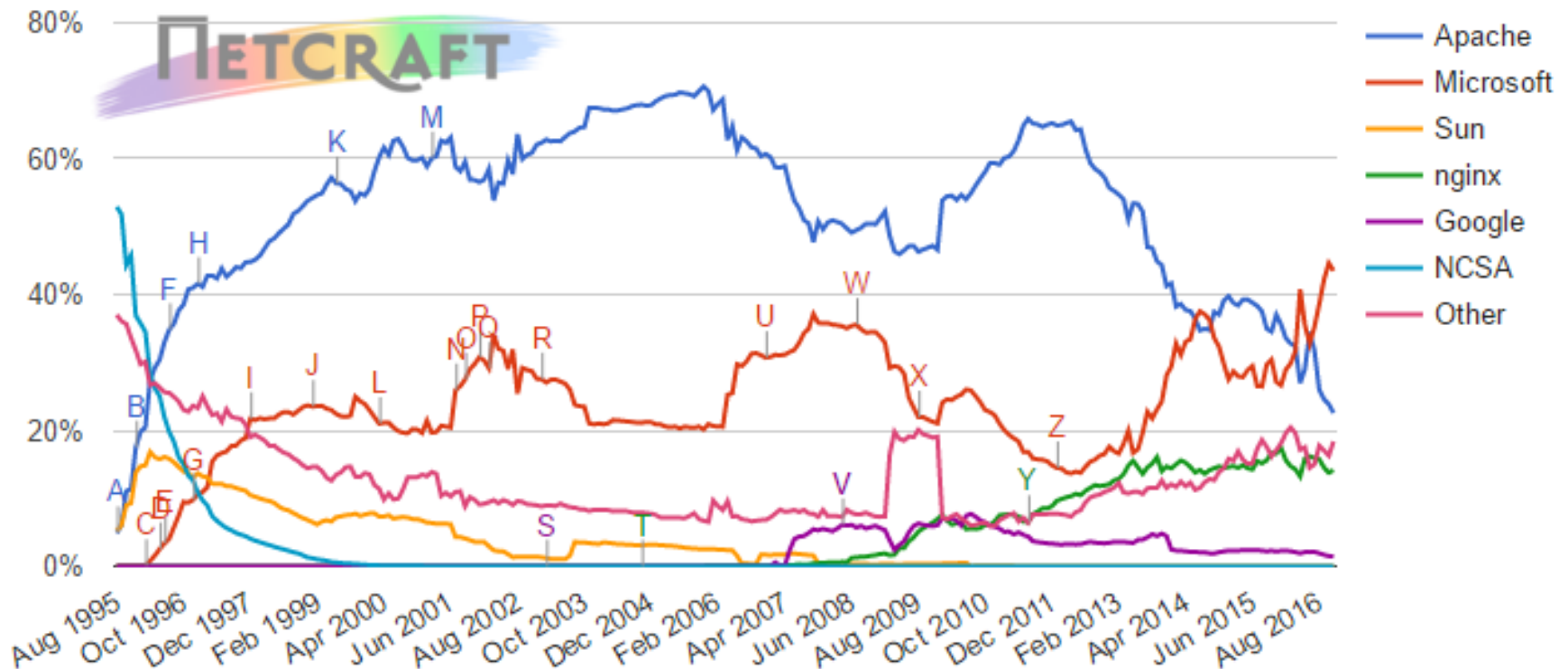
Static web transaction



General web architecture



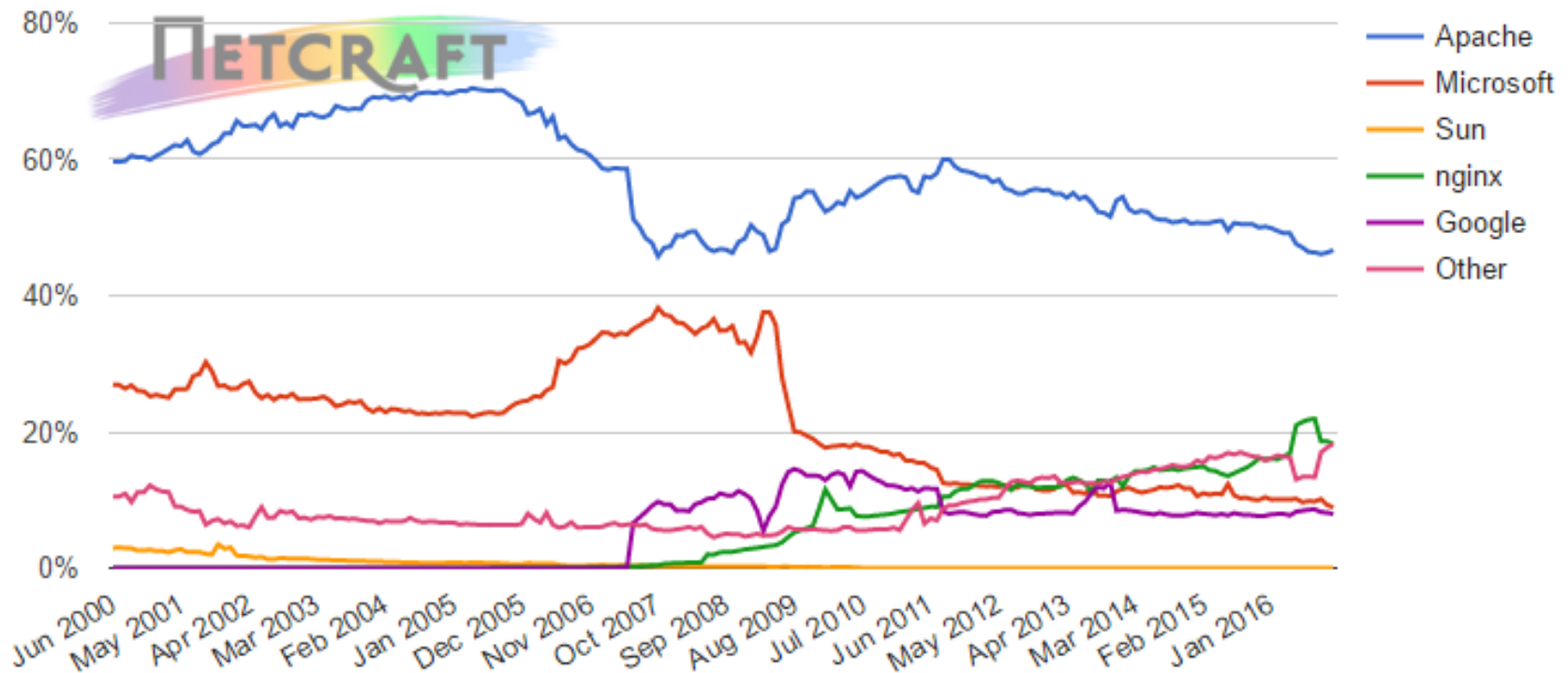
Market share of “all” sites



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

<https://news.netcraft.com/archives/2016/11/22/november-2016-web-server-survey.html>

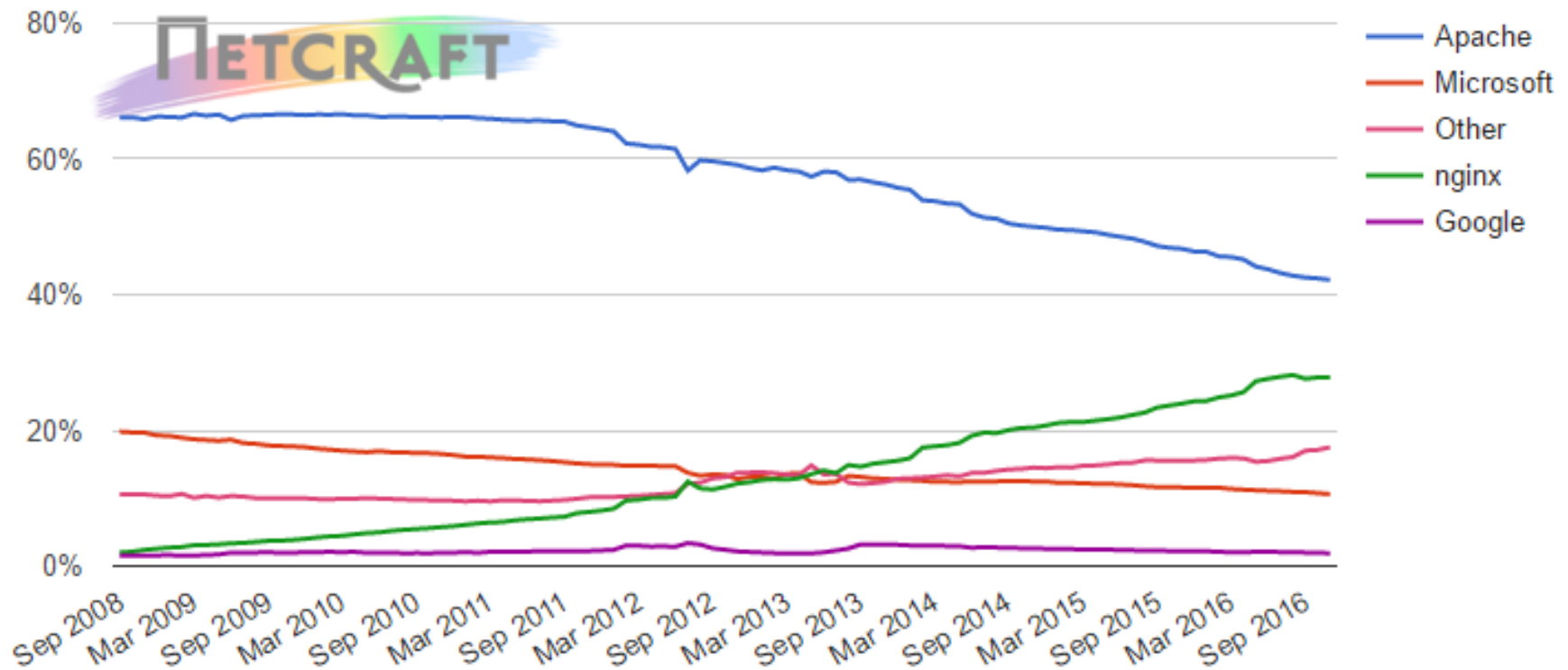
Market share of **active** sites



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

<https://news.netcraft.com/archives/2016/11/22/november-2016-web-server-survey.html>

Market share of top million **busiest** sites



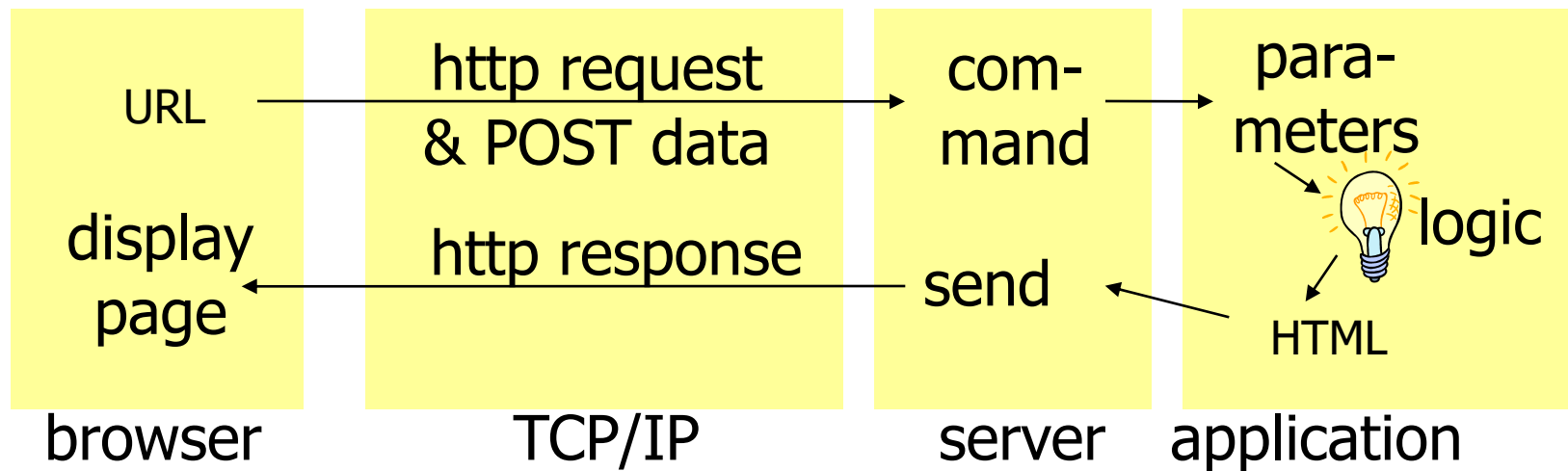
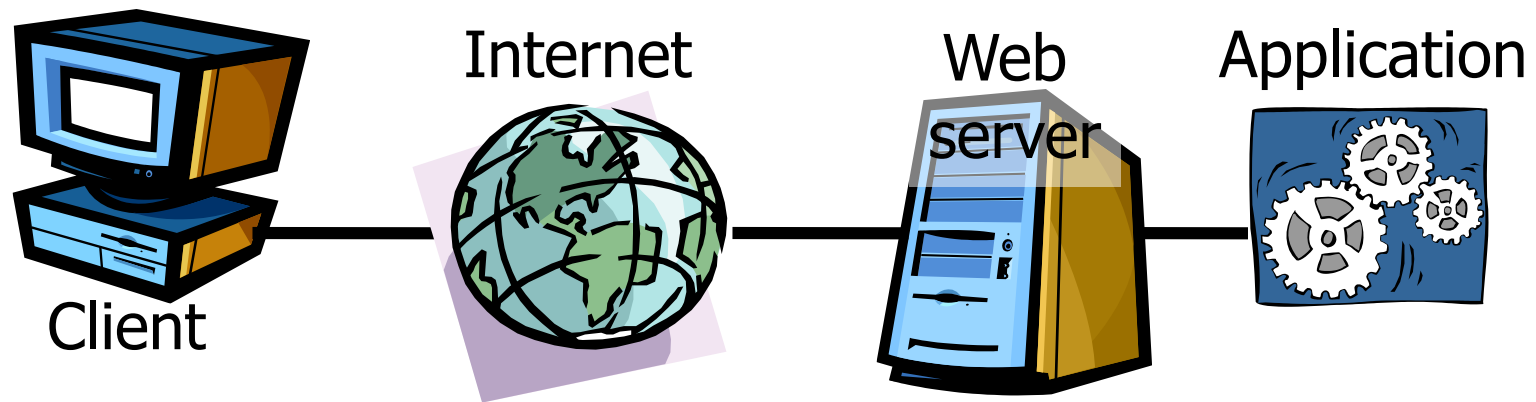
Source: <http://news.netcraft.com/>

<https://news.netcraft.com/archives/2016/11/22/november-2016-web-server-survey.html>

Application server

- ▶ Dynamic page 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

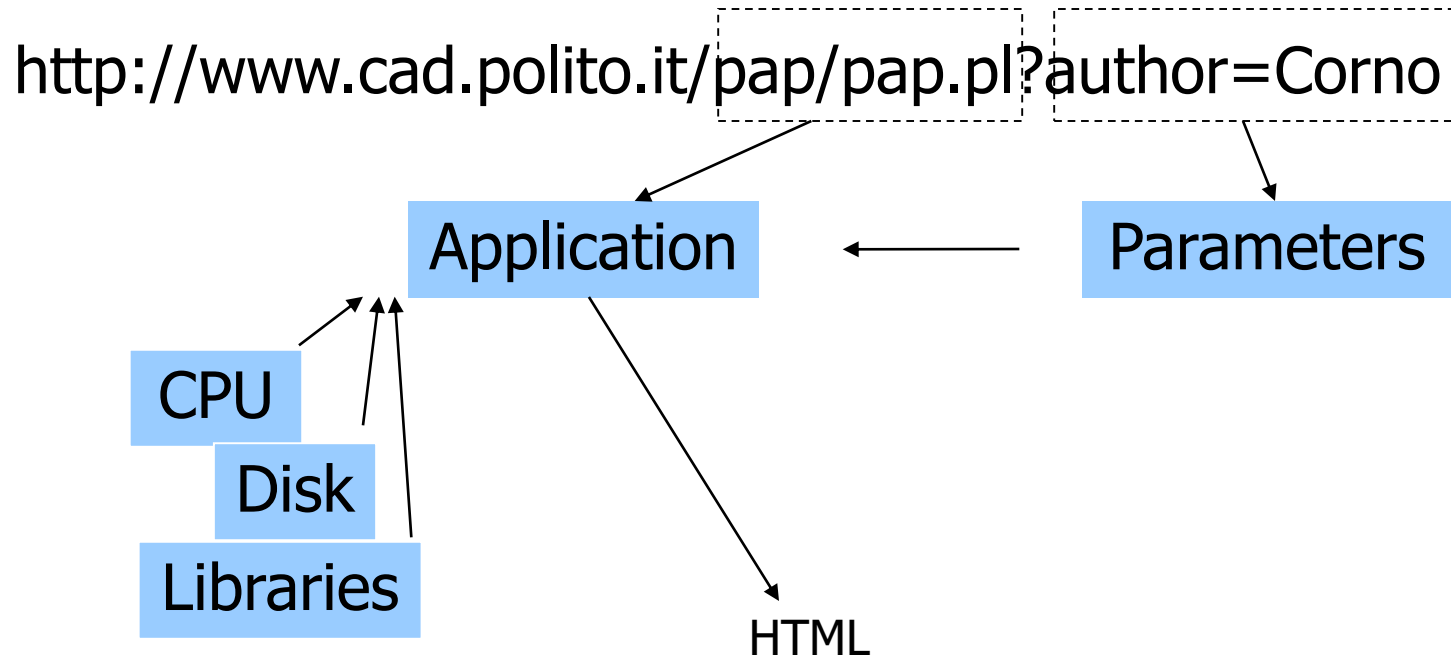
Dynamic web transaction



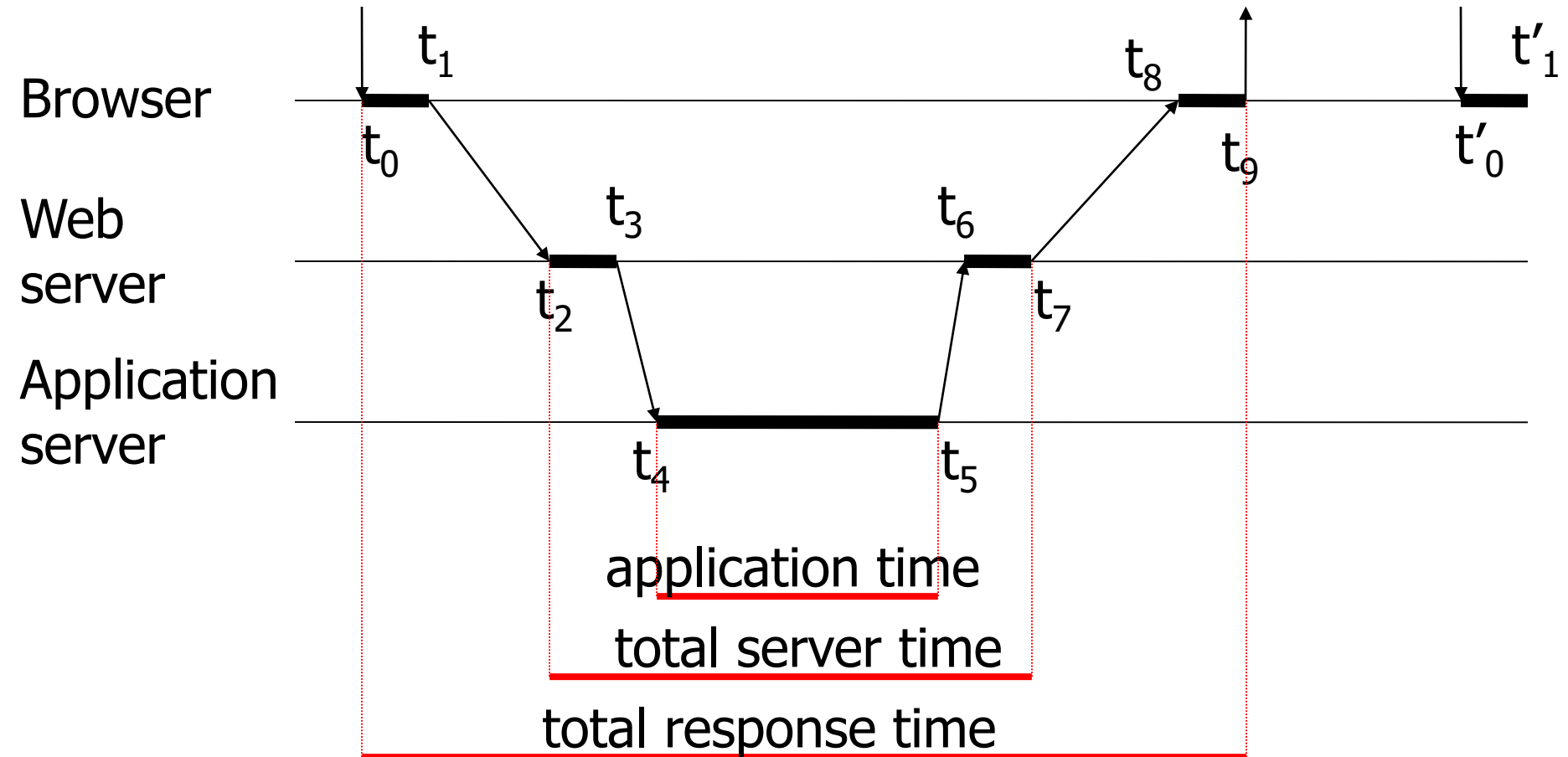
Adopted standards

- ▶ HTTP-POST for sending user-specified data
- ▶ CGI (common gateway interface), ISAPI (internet information server application programming interface), server-side script, java servlet for integrating application logic into web servers
- ▶ ASP (active server pages), PHP, PERL as new languages for application development

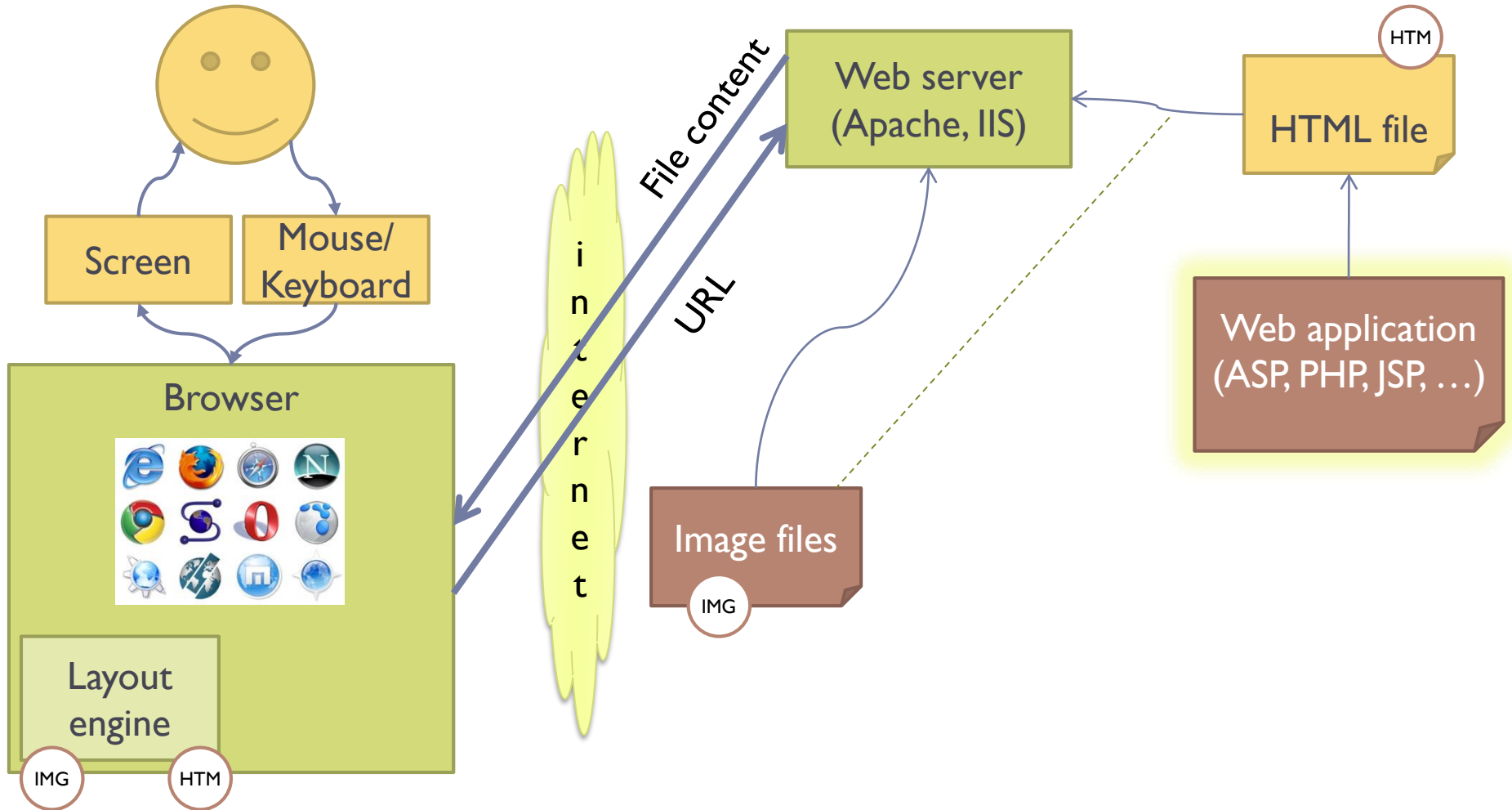
URL (HTTP GET)



Dynamic web transaction



General web architecture

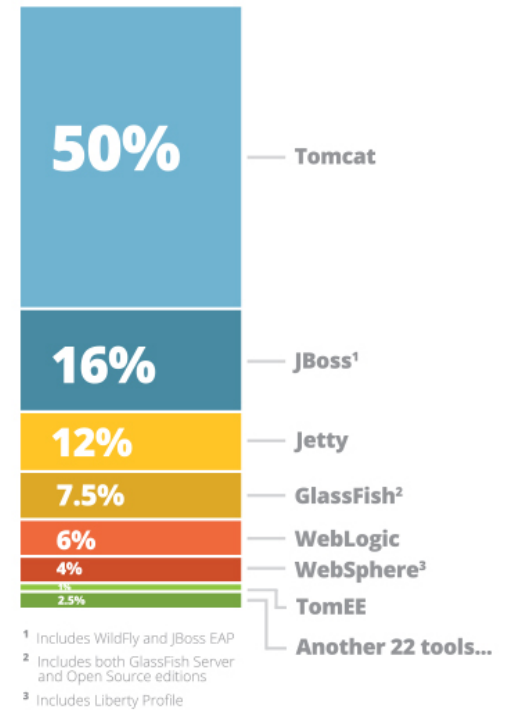


Application Servers



https://en.wikipedia.org/wiki/Website#Dynamic_website

App Server most often used*



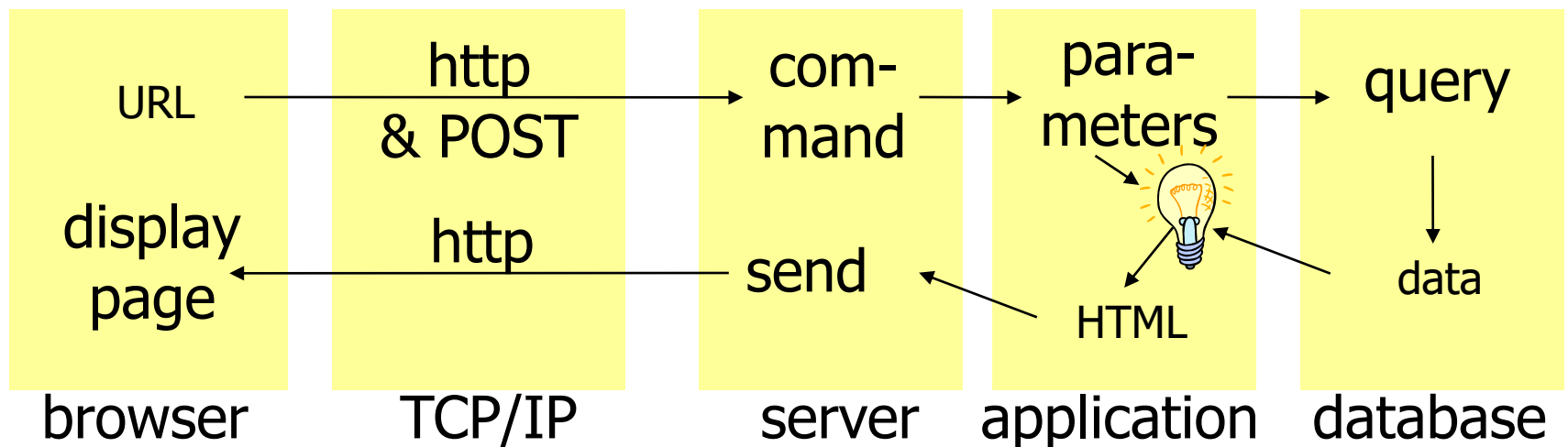
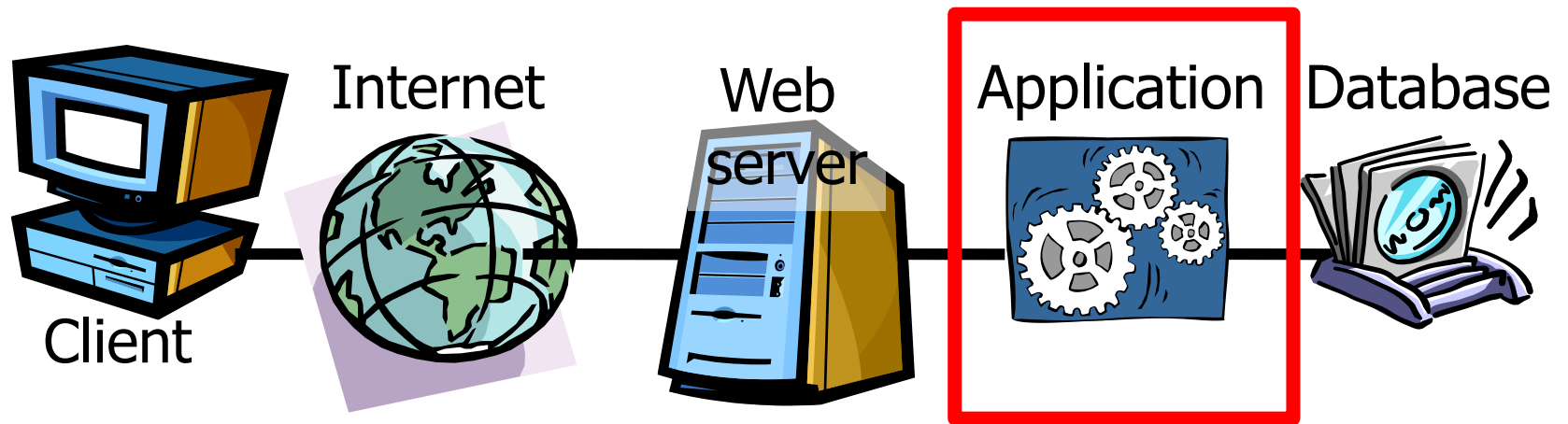
REBELLABS

* The results were normalized to exclude non-users

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



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

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"];";
```

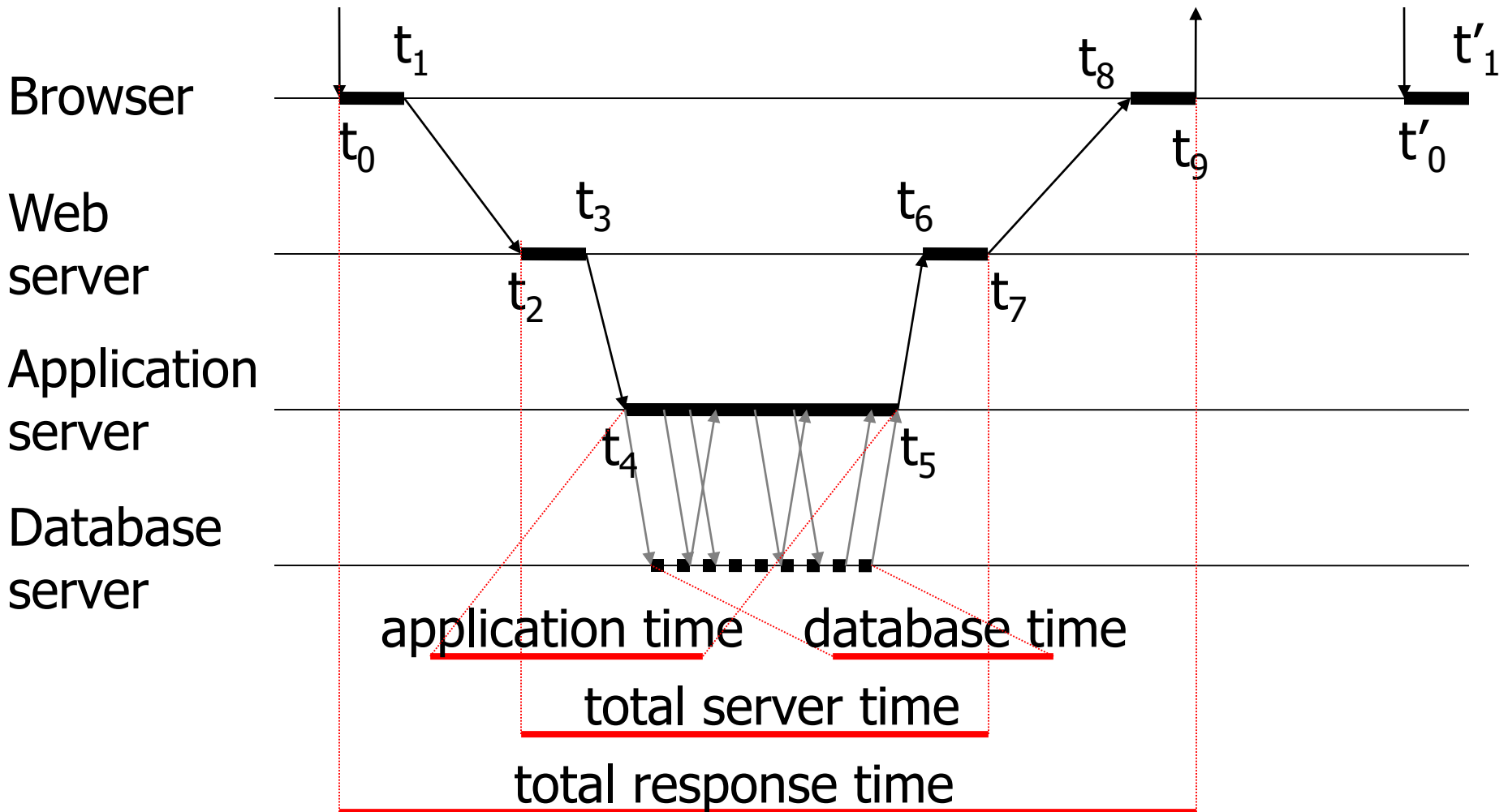
The query is sent to the db-server and a rowset containing the results is returned

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

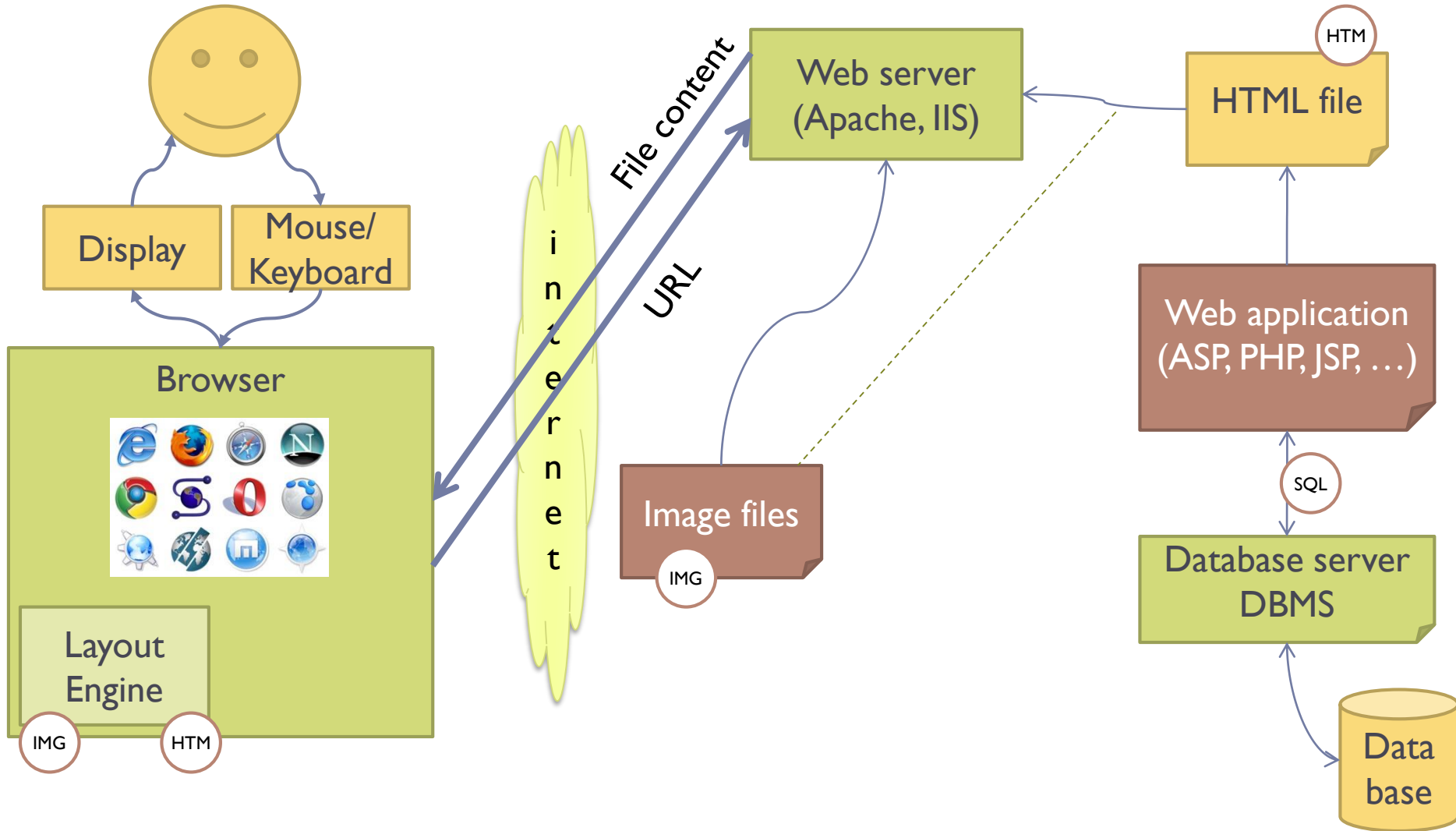
```
▶ while($row = mysql_fetch_row($rowset))
▶ {
▶ //elaborate data
▶ }
▶ ?>
```

The application elaborates the data

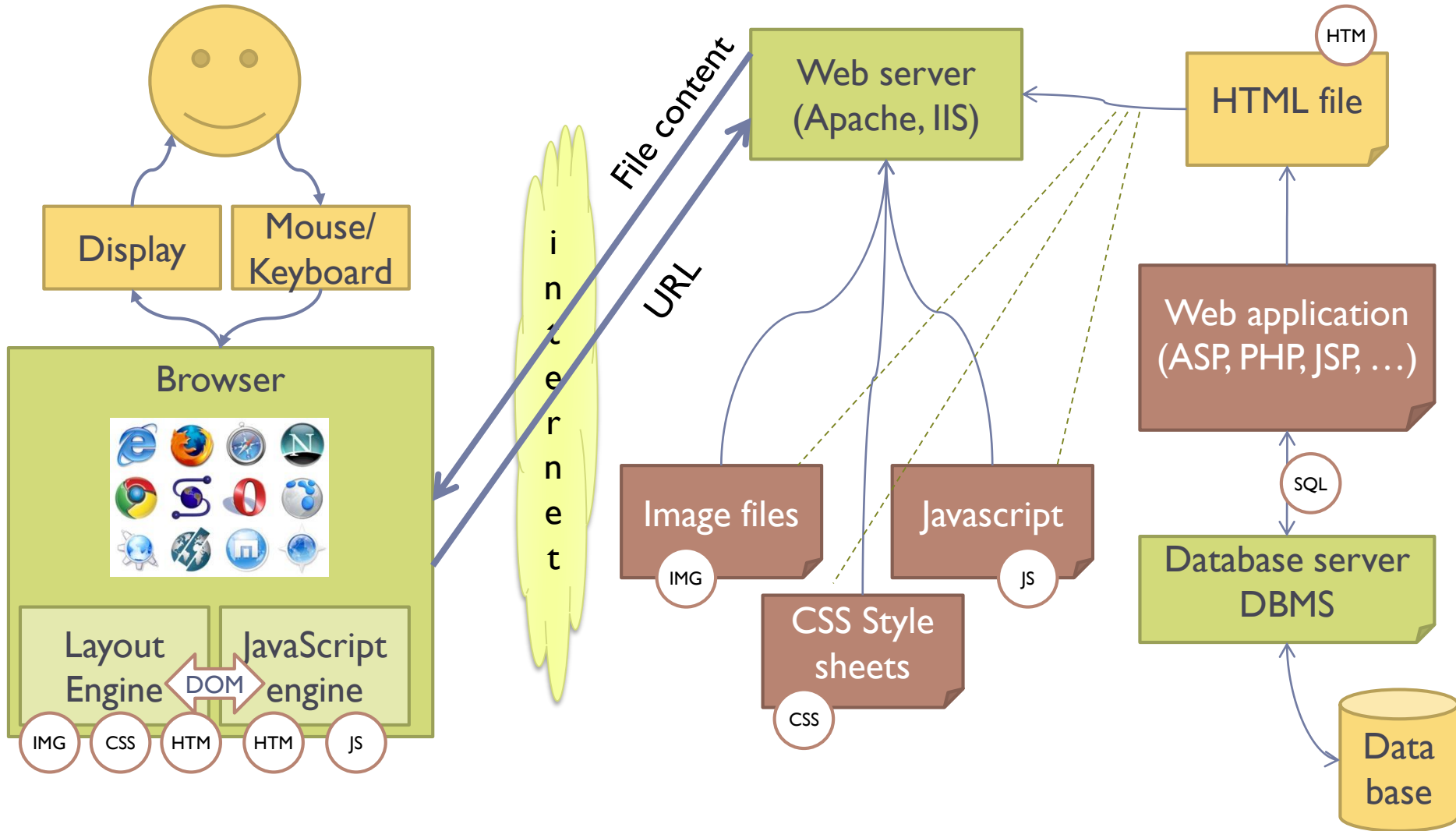
Database-driven transaction



General web architecture



General web architecture



Client-side, server-side, databases

Programming languages used in most popular websites*

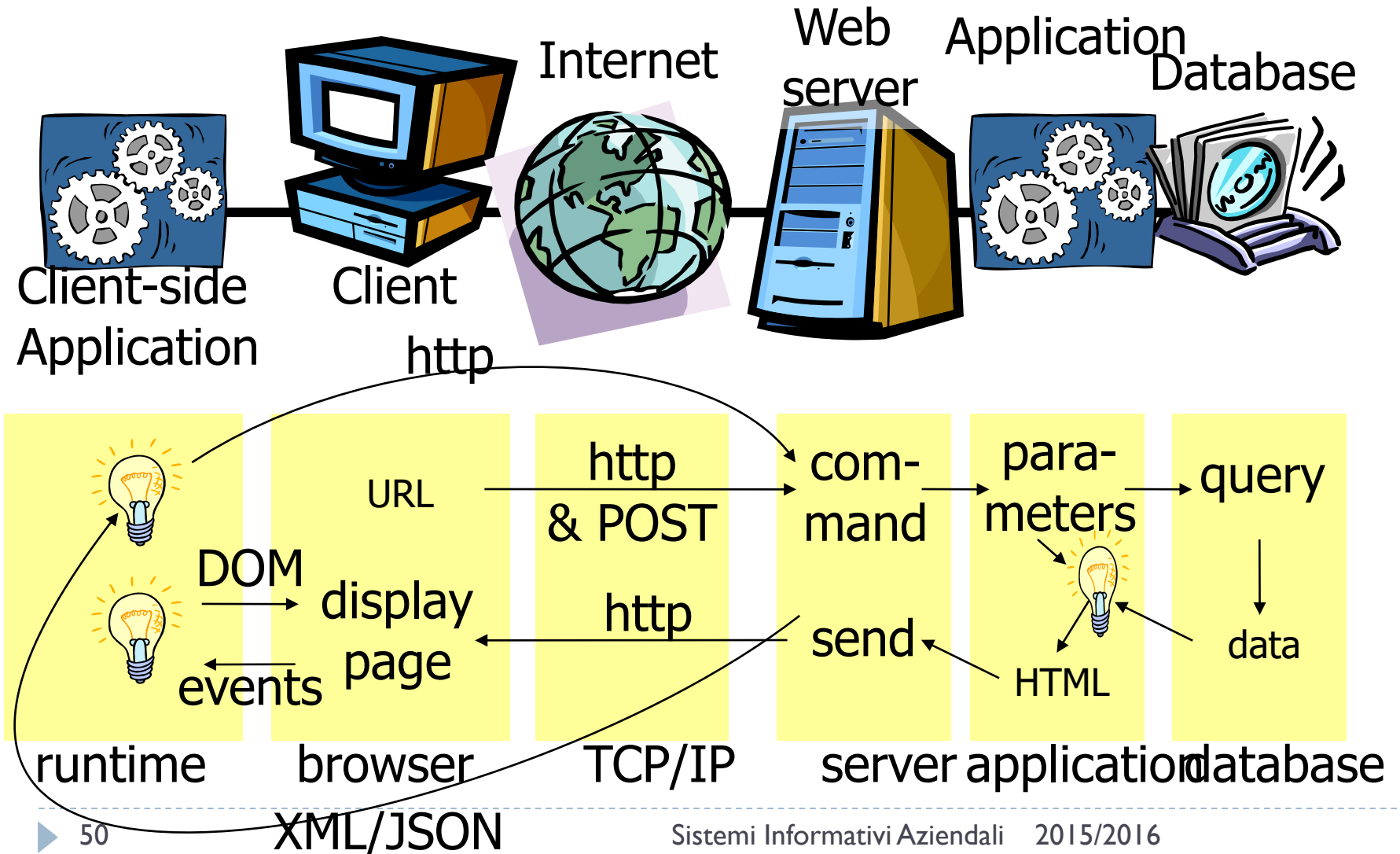
Websites	Popularity (unique visitors per month) ^[1]	Front-end (Client-side)	Back-end (Server-side)	Database
Google.com ^[2]	1,600,000,000	JavaScript	C, C++, Go, ^[3] Java, Python	BigTable, ^[4] MariaDB ^[5]
Facebook.com	1,100,000,000	JavaScript	Hack, PHP (HHVM), Python, C++, Java, Erlang, D, ^[6] Xhp, ^[7] Haskell ^[8]	MariaDB, MySQL, ^[9] HBase Cassandra ^[10]
YouTube.com	1,100,000,000	JavaScript	C, C++, Python, Java, ^[11] Go ^[12]	BigTable, MariaDB ^{[5][13]}
Yahoo	750,000,000	JavaScript	PHP	MySQL, PostgreSQL ^[14]
Amazon.com	500,000,000	JavaScript	Java, C++, Perl ^[16]	Oracle Database ^[17]
Wikipedia.org	475,000,000	JavaScript	PHP, Hack	MySQL ^[citation needed] , MariaDB ^[18]
Twitter.com	290,000,000	JavaScript	C++, Java, Scala, Ruby on Rails ^[19]	MySQL ^[20]
Bing	285,000,000	JavaScript	ASP.NET	Microsoft SQL Server
eBay.com	285,000,000	JavaScript	Java, ^[21] JavaScript ^[22]	Oracle Database
MSN.com	280,000,000	JavaScript	ASP.NET	Microsoft SQL Server
Microsoft	270,000,000	JavaScript	ASP.NET	Microsoft SQL Server
Linkedin.com	260,000,000	JavaScript	Java, JavaScript, ^[23] Scala	Voldemort ^[24]
Pinterest	250,000,000	JavaScript	Django (a Python framework), ^[25] Erlang	MySQL, Redis ^[26]
WordPress.com	240,000,000	JavaScript	PHP, JavaScript ^[27] (Node.js)	MariaDB, MySQL

https://en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites

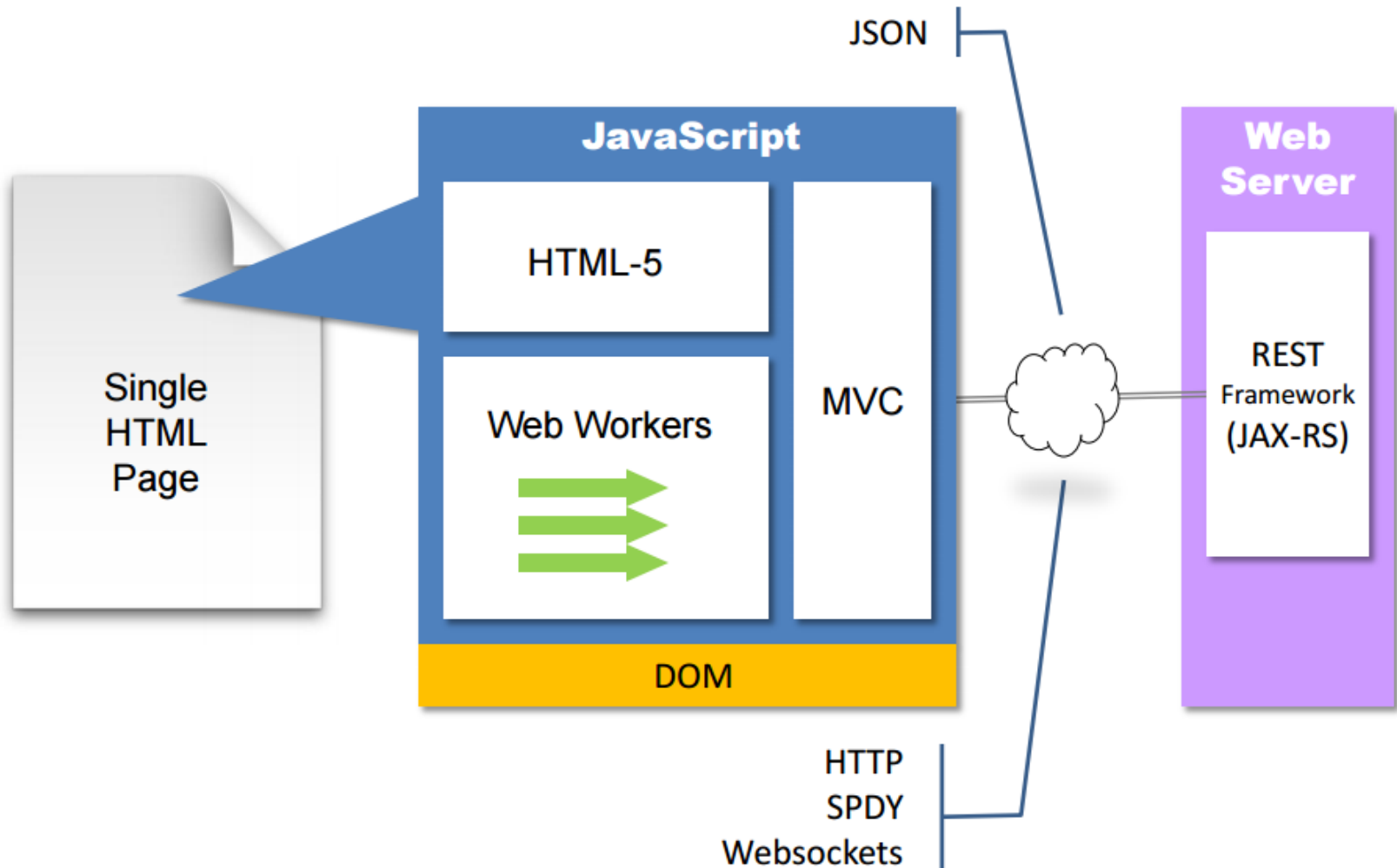
Web 2.0

- ▶ 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)

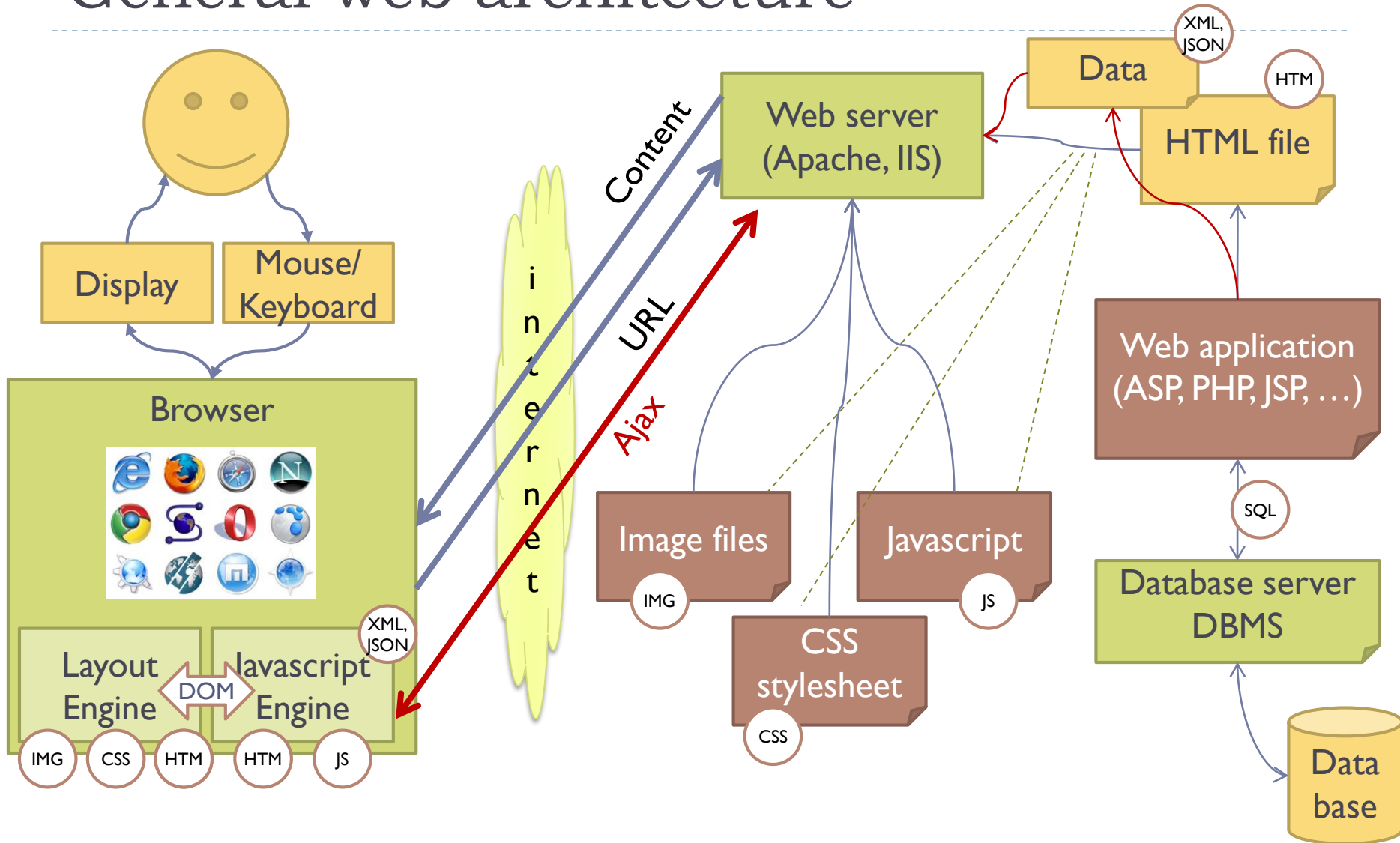
Rich-Client Asynchronous Transactions



Single Page Applications (SPA)



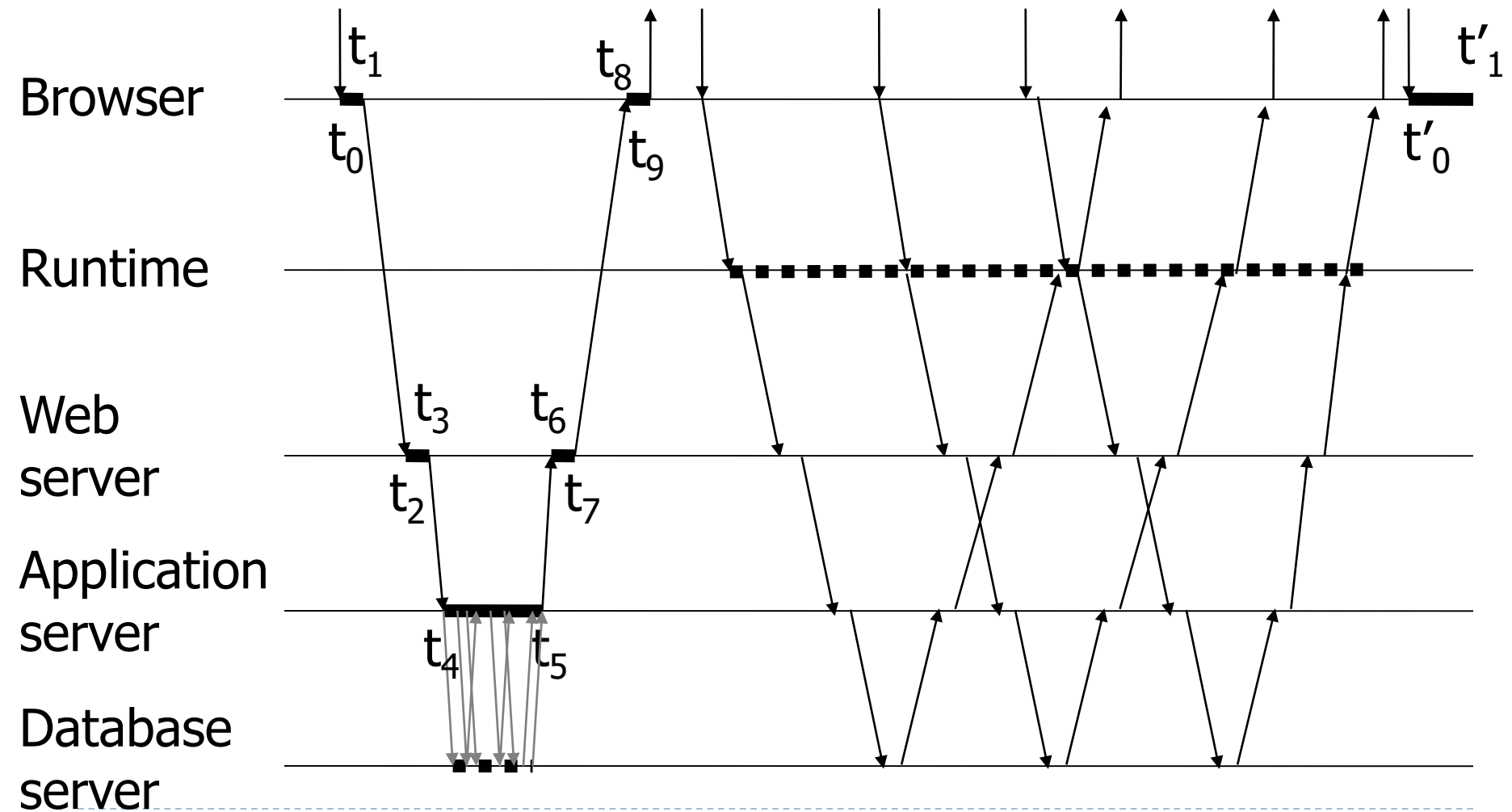
General web architecture



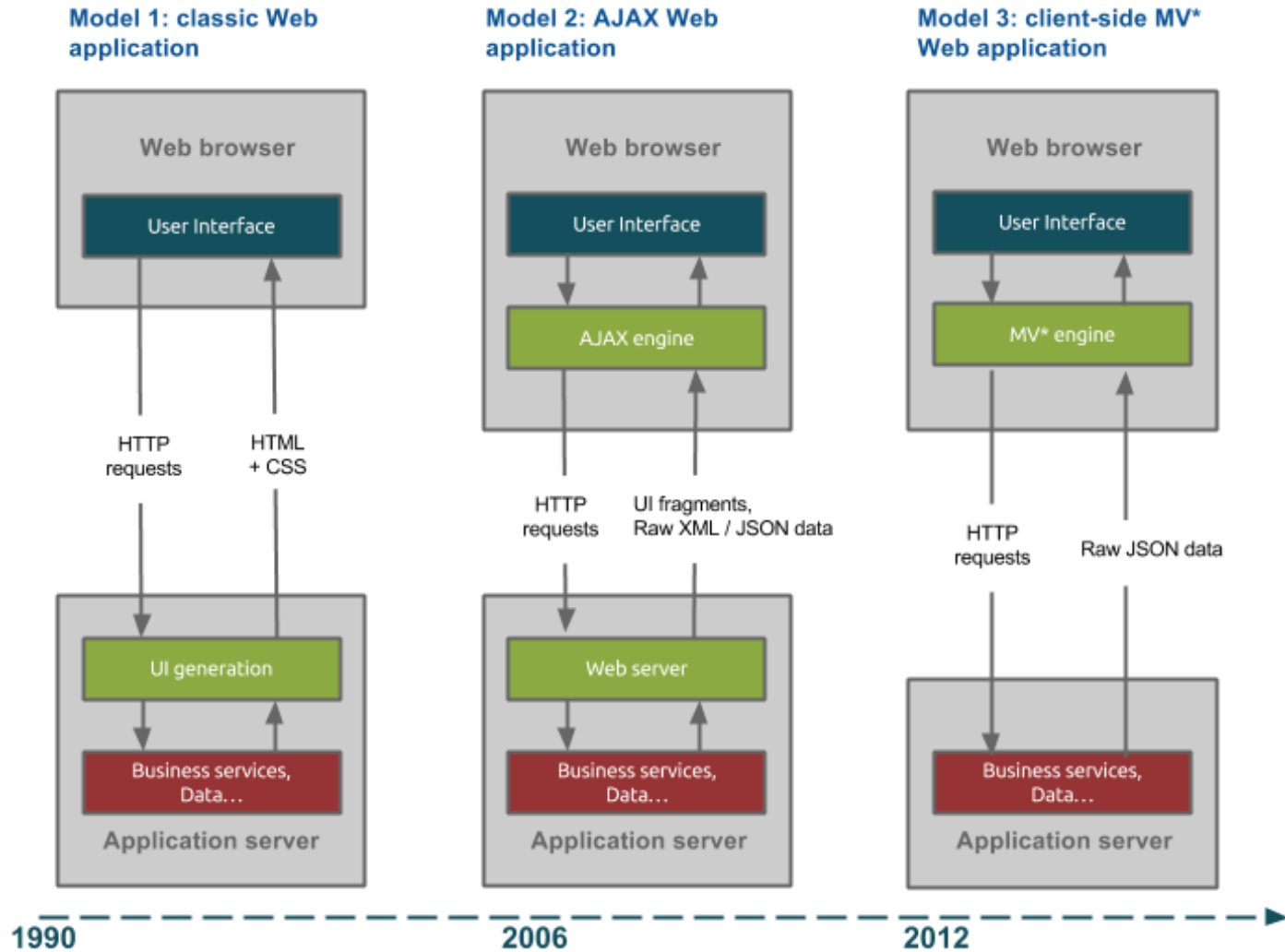
Adopted standards

- ▶ **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**

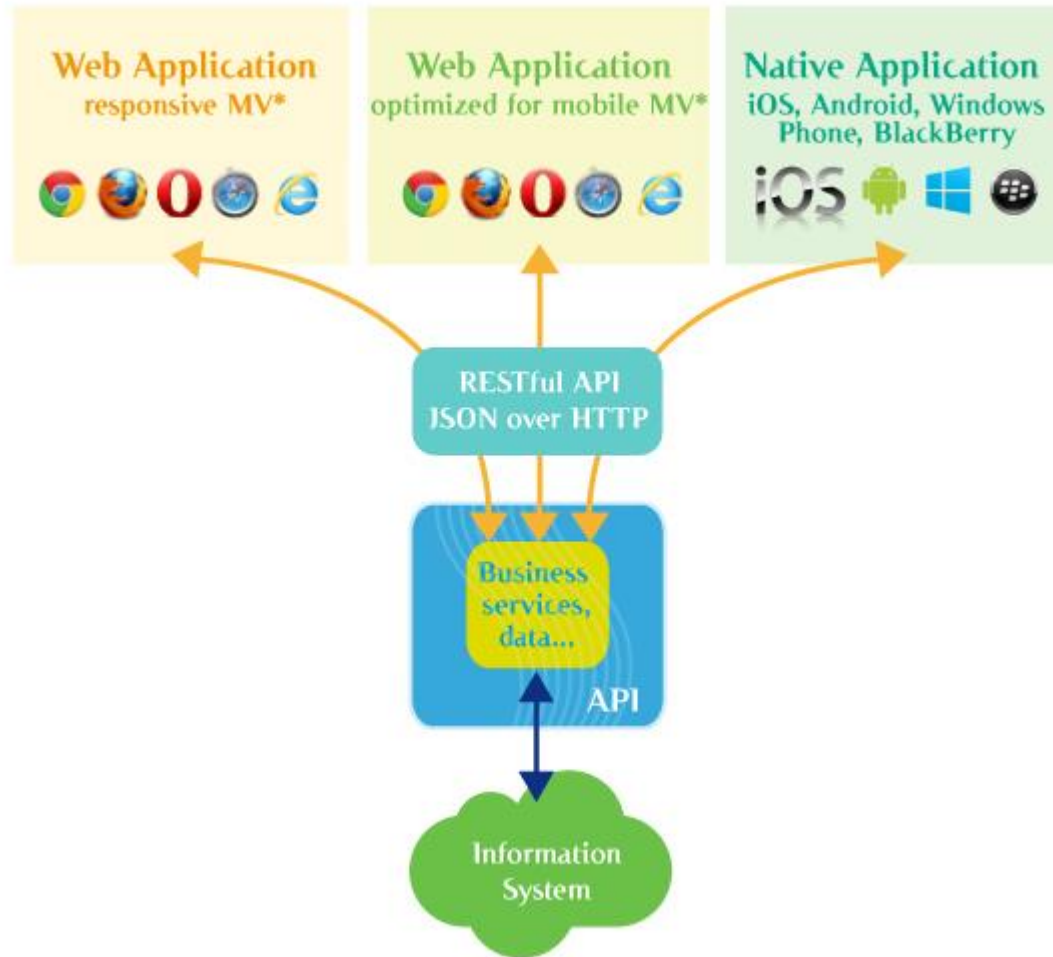
Rich-client transaction



Web application architectures



Supporting mobile development

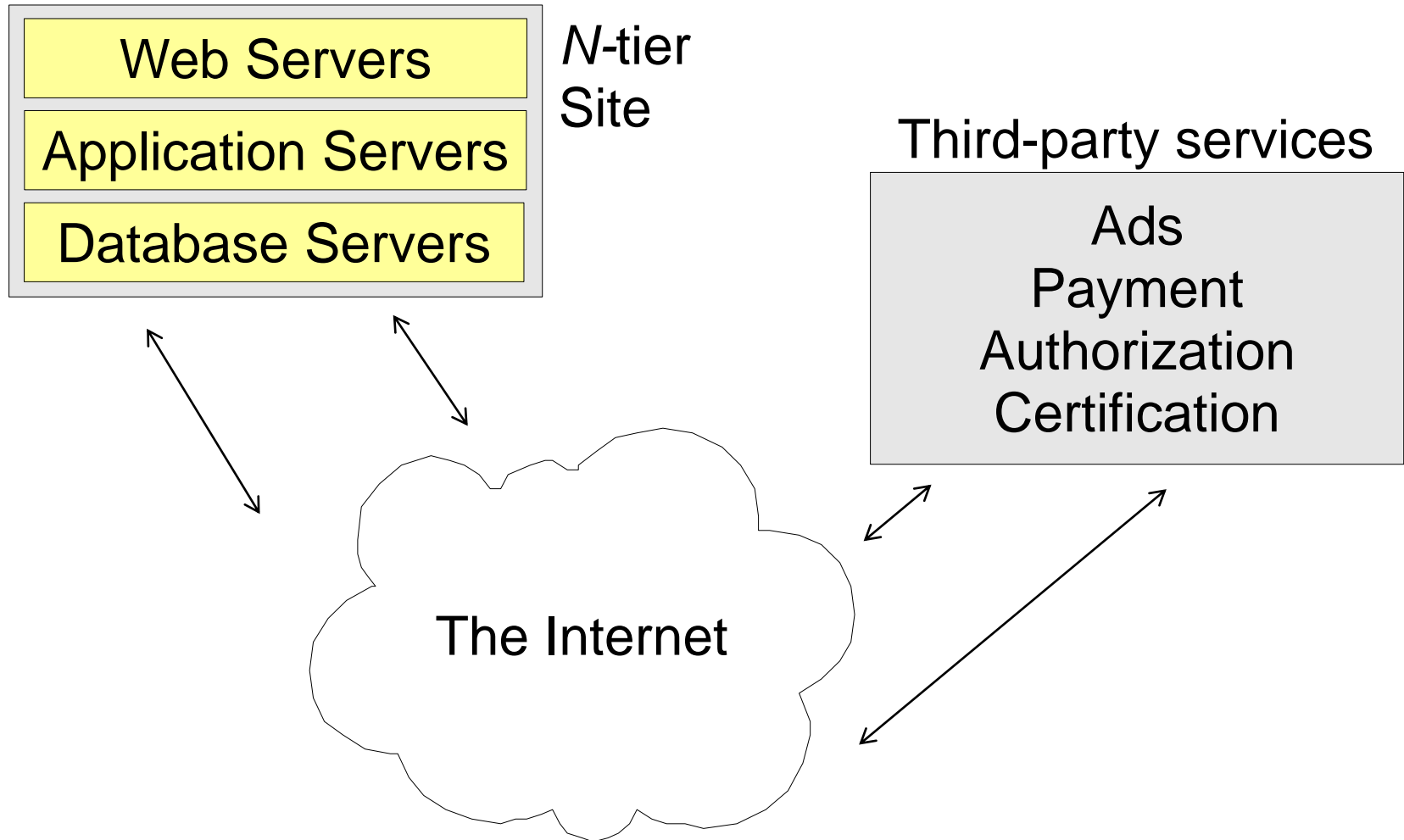


Challenges for Enterprise Systems

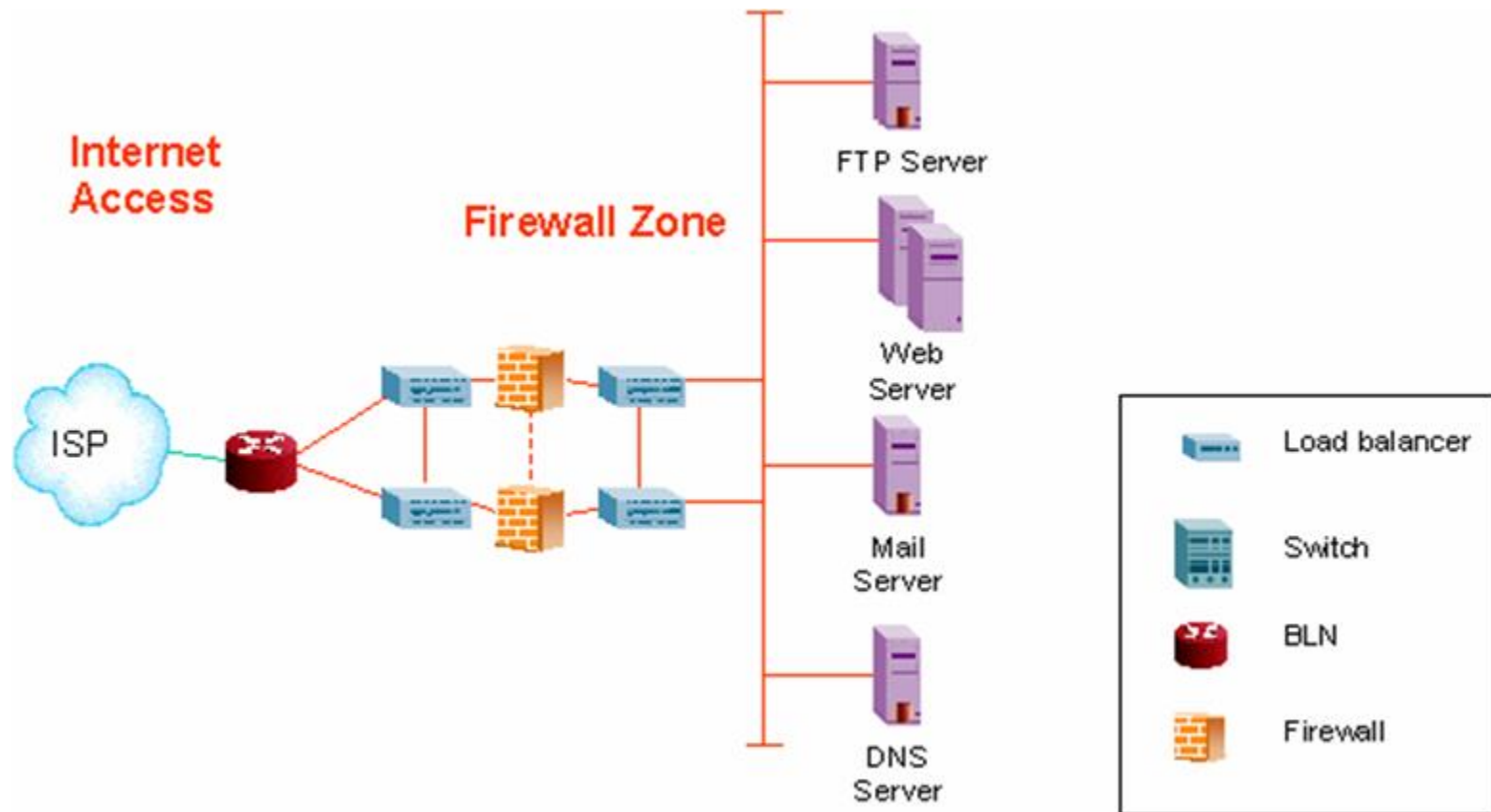
- ▶ The users
- ▶ Functionality
- ▶ Flexibility
- ▶ Portability
- ▶ Reliability
- ▶ Security
- ▶ Integrity
- ▶ Maintenance
- ▶ Performance
- ▶ Scalability
- ▶ Costs
- ▶ Maintenance
- ▶ Development times
- ▶ Interactions with existing systems
- ▶ Interactions with the “physical” world



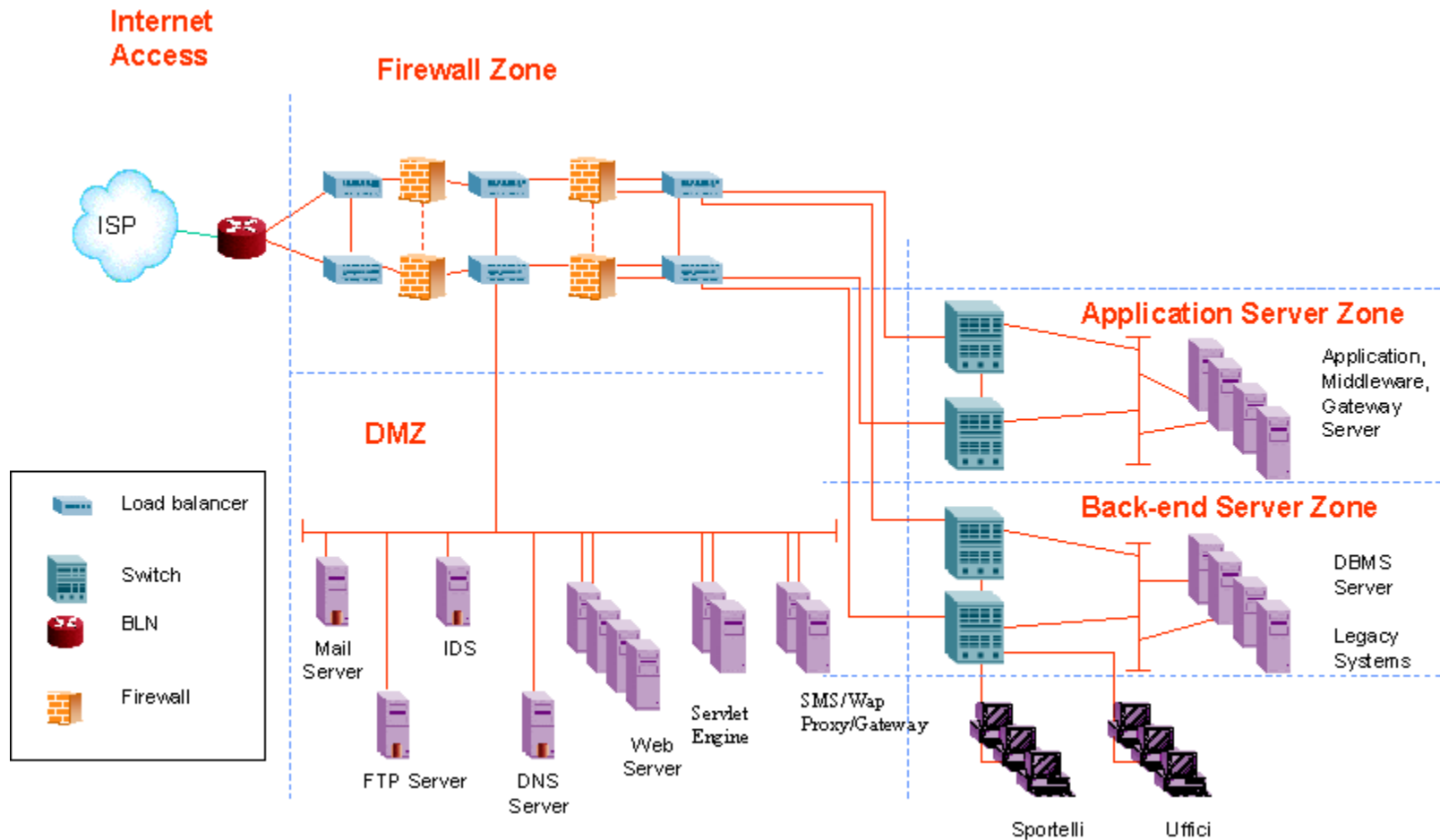
E-business architectures



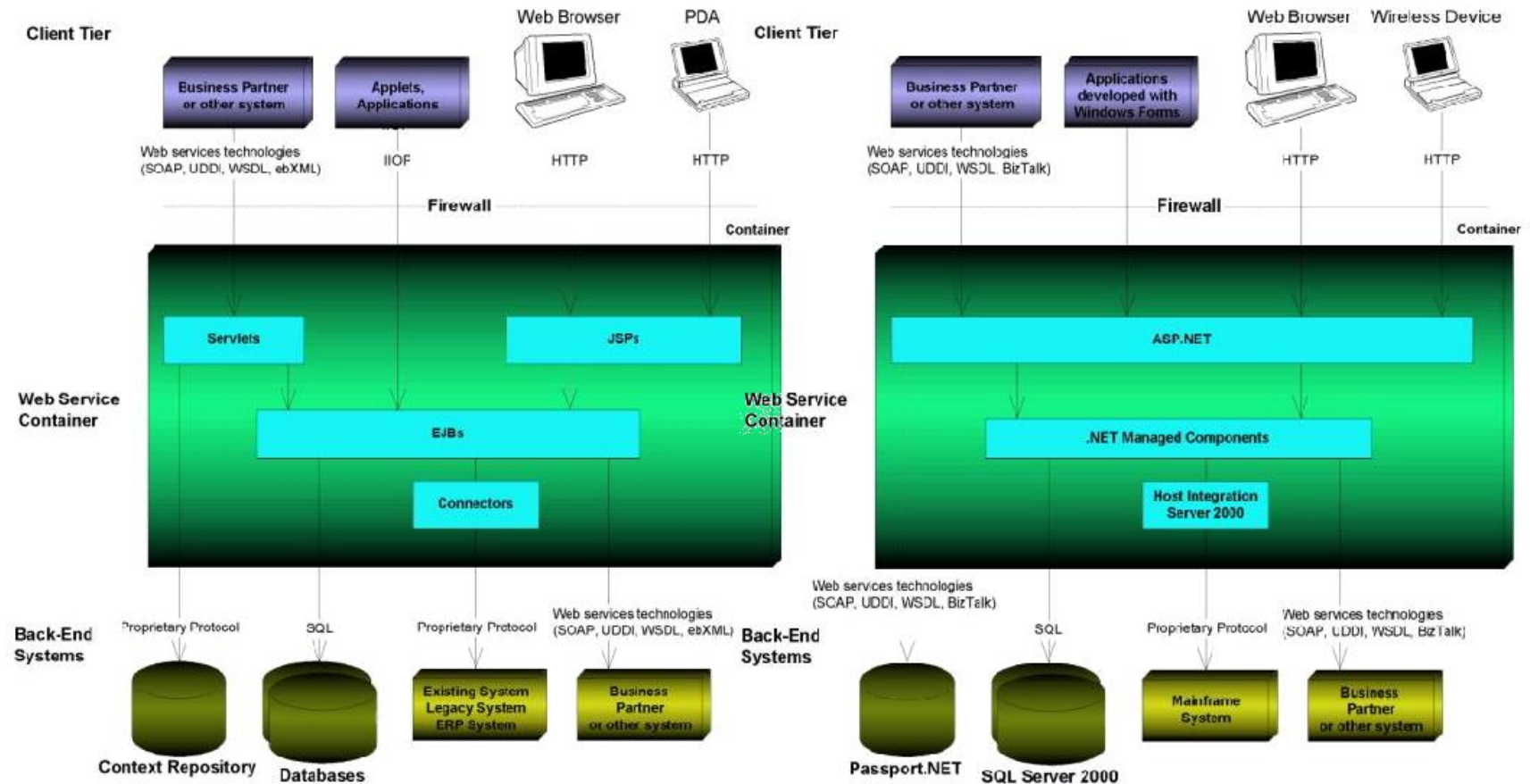
Informative site – complete



Ordering site – typical structure



Legacy systems are always there...








Interacting with other suppliers...

- ▶ **Application Server needs to require services available on an external host**
 - ▶ Ordering services (e.g. payment)
 - ▶ Informative services (e.g. stock quotes)
 - ▶ Security services (e.g. authentication)
- ▶ **A web page contains sections originating from different sites**
 - ▶ “Application” approach, sections interact and share data (*mashup*)

Licenza d'uso



- ▶ Queste diapositive sono distribuite con licenza Creative Commons “Attribuzione - Non commerciale - Condividi allo stesso modo 2.5 Italia (CC BY-NC-SA 2.5)”
- ▶ Sei libero:
 - ▶ di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera 
 - ▶ di modificare quest'opera 
- ▶ Alle seguenti condizioni:
 - ▶ **Attribuzione** — Devi attribuire la paternità dell'opera agli autori originali e in modo tale da non suggerire che essi avallino te o il modo in cui tu usi l'opera. 
 - ▶ **Non commerciale** — Non puoi usare quest'opera per fini commerciali. 
 - ▶ **Condividi allo stesso modo** — Se alteri o trasformi quest'opera, o se la usi per crearne un'altra, puoi distribuire l'opera risultante solo con una licenza identica o equivalente a questa. 
- ▶ <http://creativecommons.org/licenses/by-nc-sa/2.5/it/>