



Web-based Transactions

Fulvio Corno



Evolution

The world wide web, if we analyze the involved technologies, evolved through 3 main phases:

1. Static and informative Web
2. Dynamic and interactive Web
3. Integrated and application-oriented Web

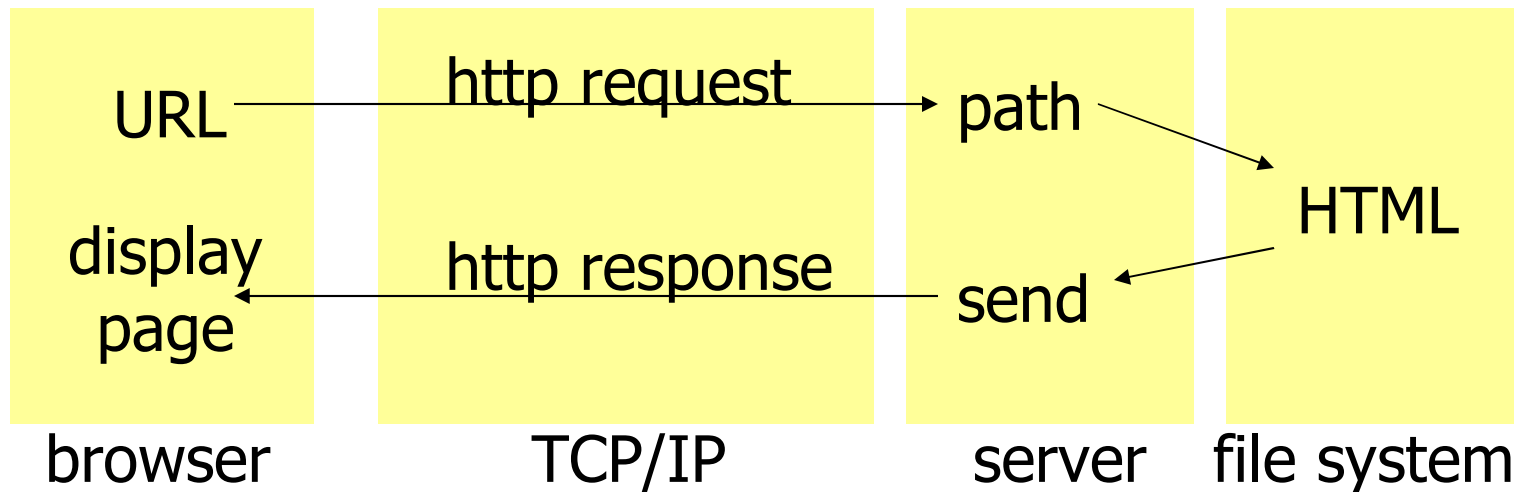
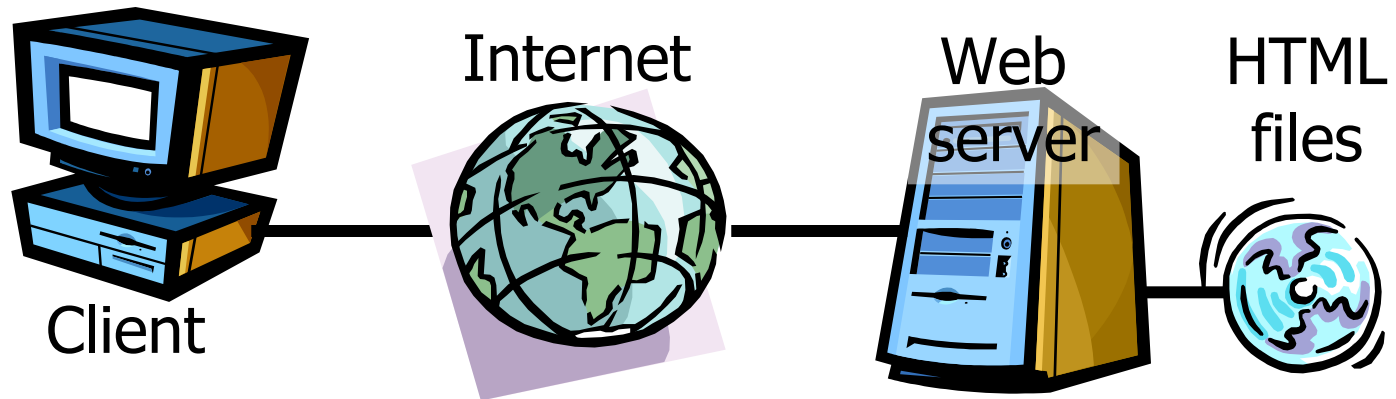
E-commerce and e-business are a particular application domain of Phase 3.



Summary

- Static Transactions
- Dynamic Transactions
- Database-driven Transactions
- Rich-Client Asynchronous Transactions

Static web transaction





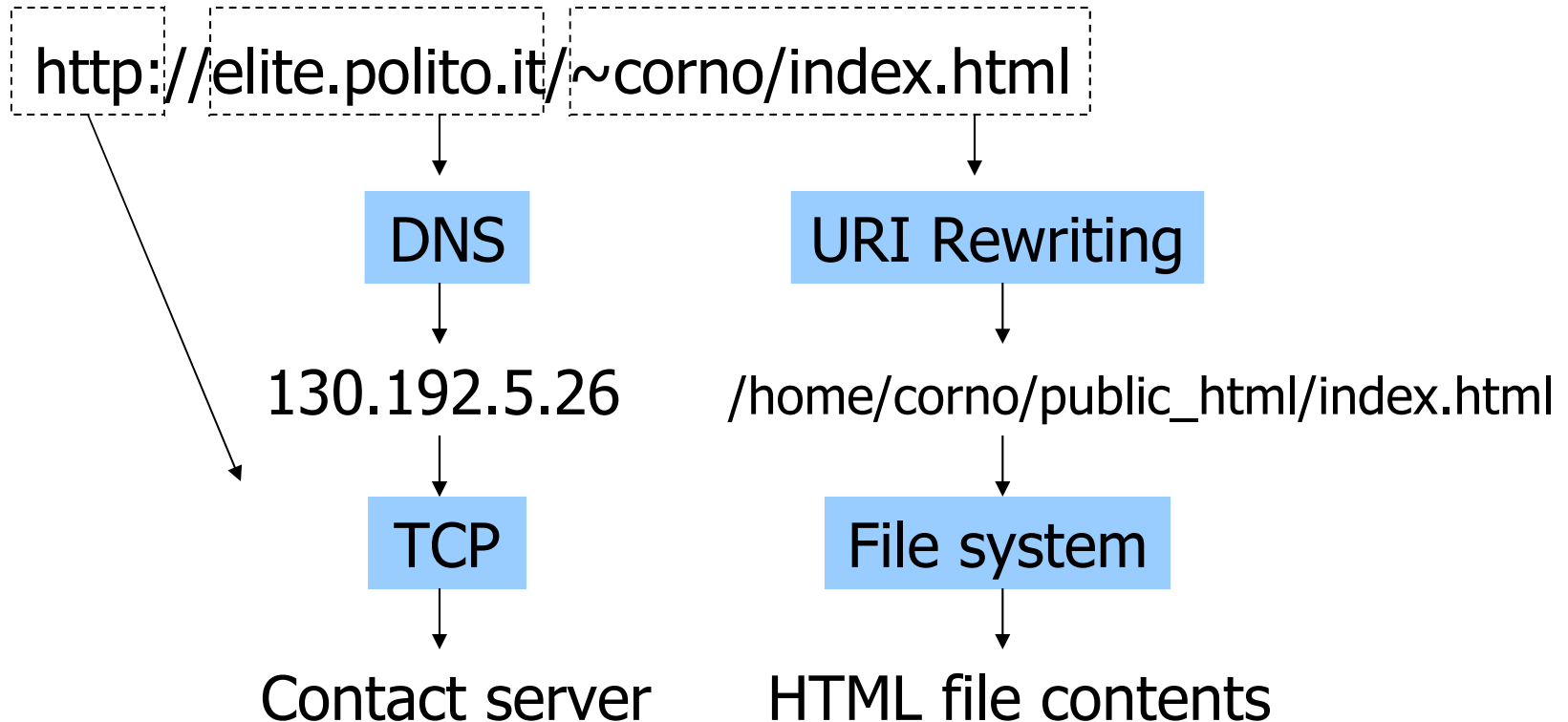
Adopted standards

- **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

URL

RFC 2396

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





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: CornoSoft 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

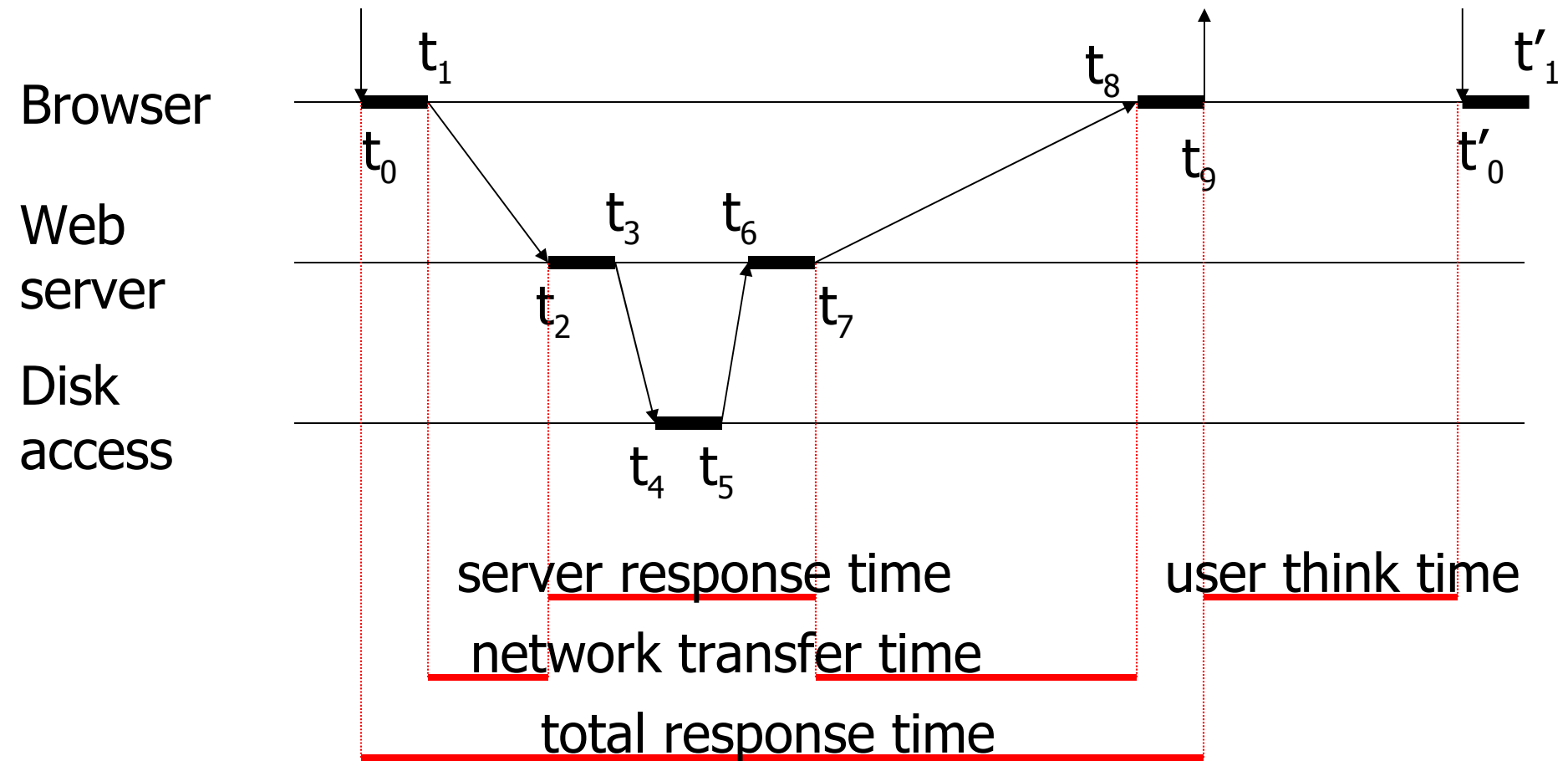
Last-Modified: 31-Dec-2000

Content-type: text/html

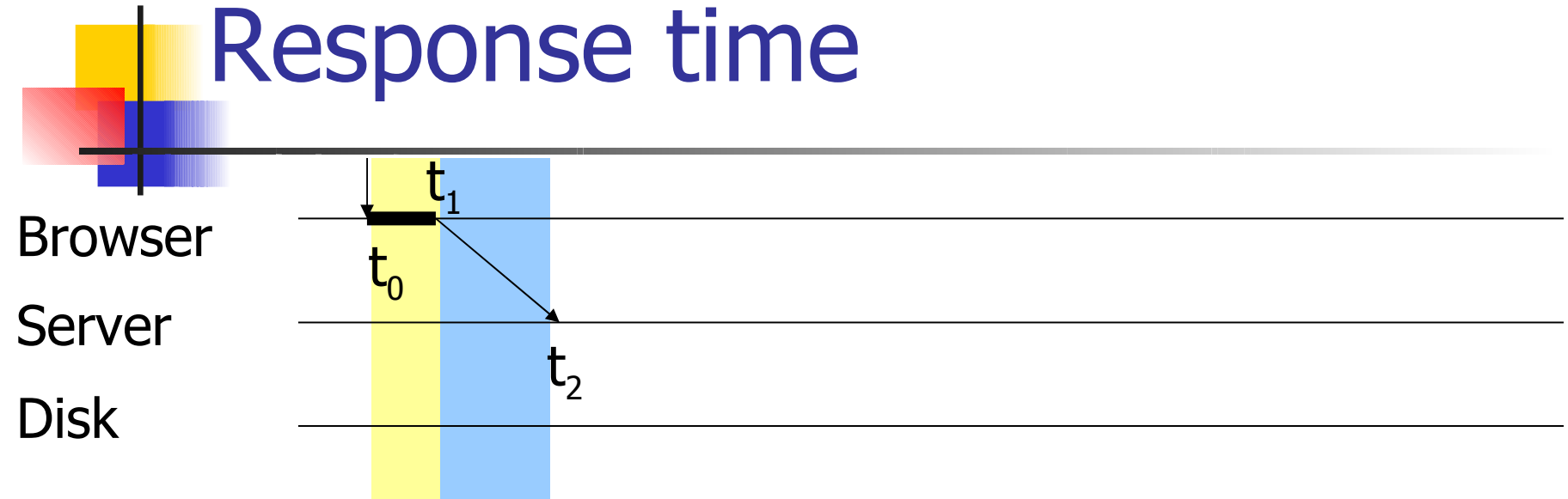
Content-length: 3021

<HTML> . . .

Static web transaction

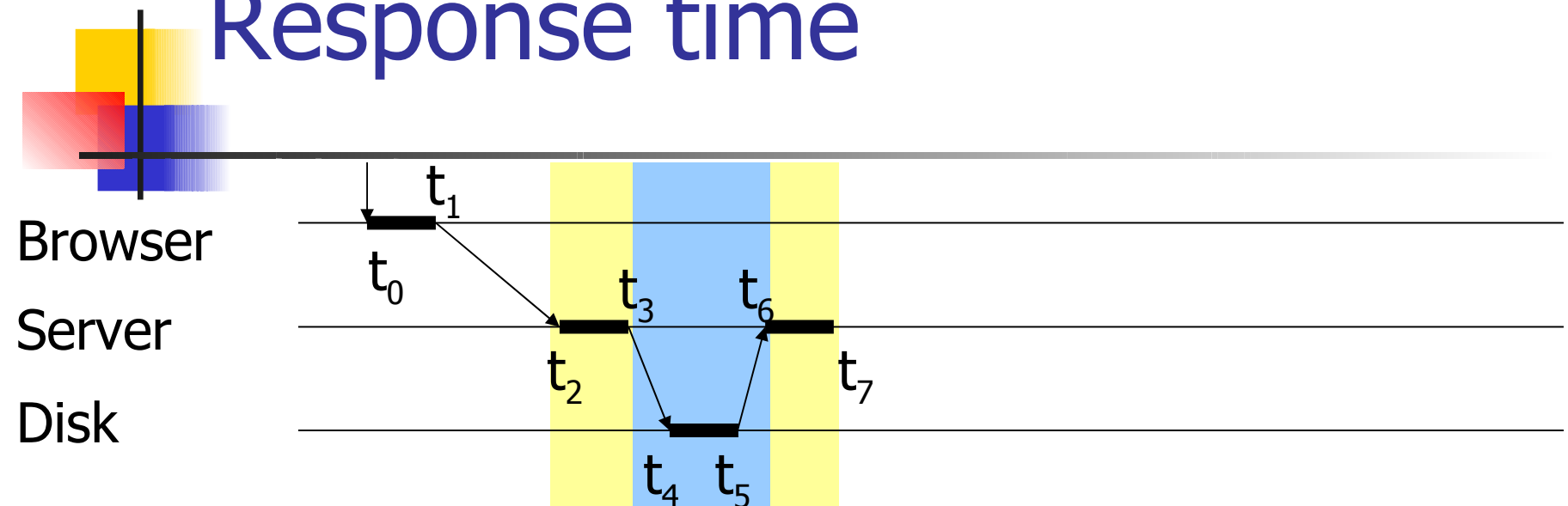


Response time



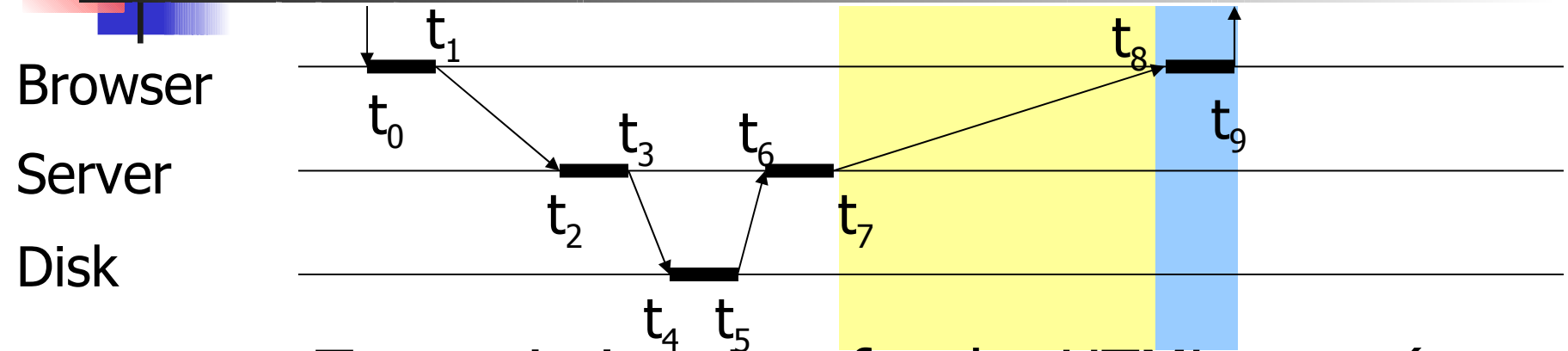
- Browser elaboration time ($t_0 - t_1$): **power** of the users' computer
- Transmission time for http request ($t_1 - t_2$): **latency** of Internet connection

Response time



- Server elaboration time (t_2-t_3 , t_6-t_7): power of the server and number of simultaneous accesses
- Disk access time (t_3-t_6): I/O architecture del server and number of simultaneous accesses

Response time



- Transmission time for the HTML page (t_7 - t_8): **size** of the page and of the images, **latency**, available **bandwidth**
- Visualization time of the page (t_8 - t_9): **power** of the users' computer and **complexity** of the page



Examples

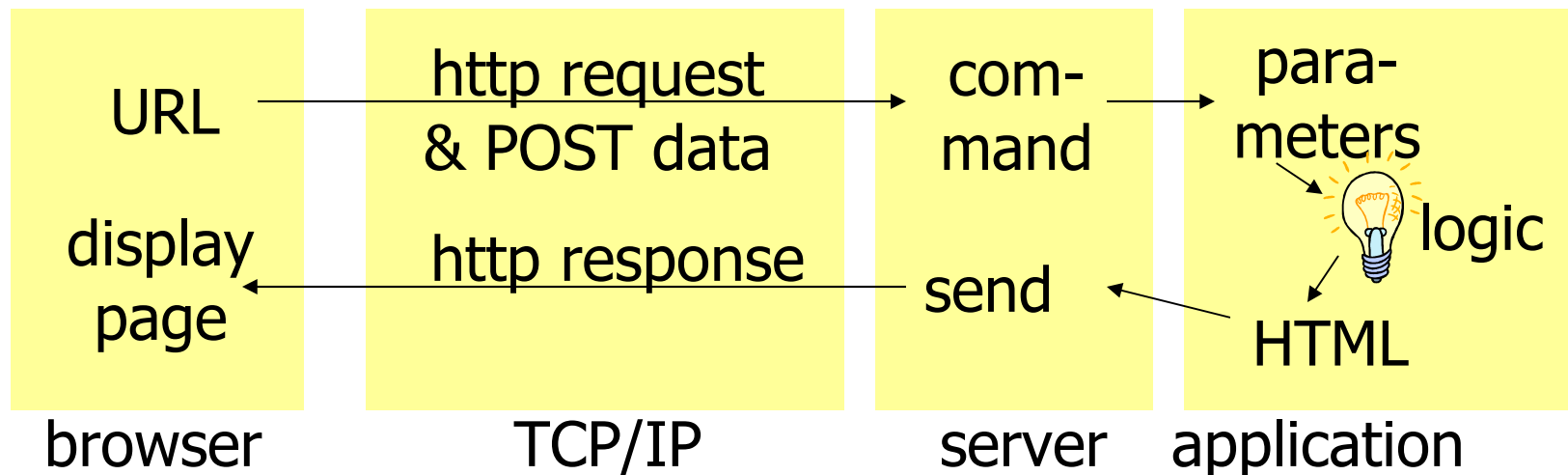
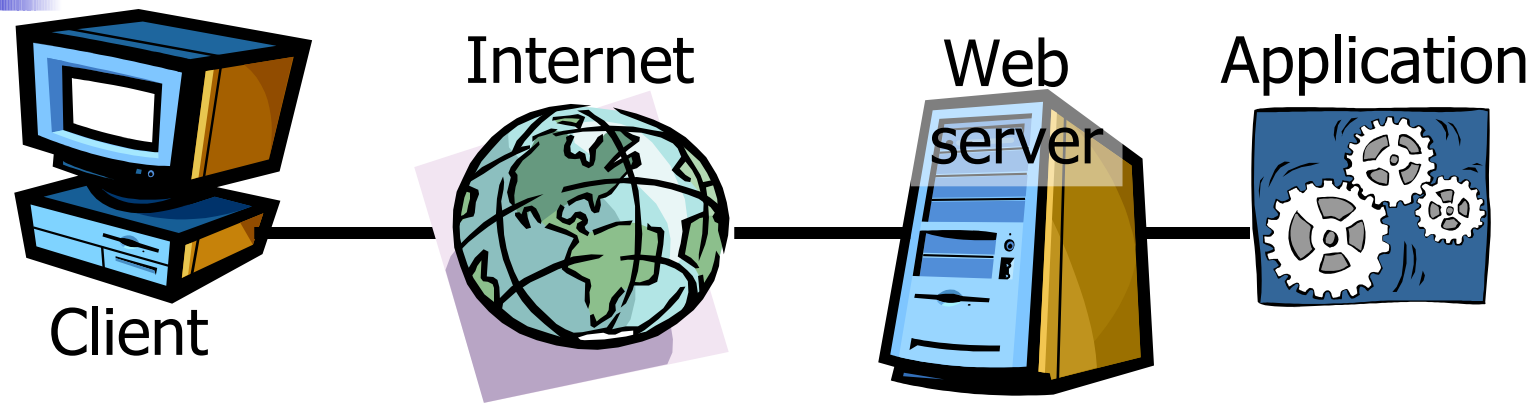
www.w3c.org



Summary

- Static Transactions
- Dynamic Transactions
- Database-driven Transactions
- Rich-Client Asynchronous Transactions

Dynamic web transaction



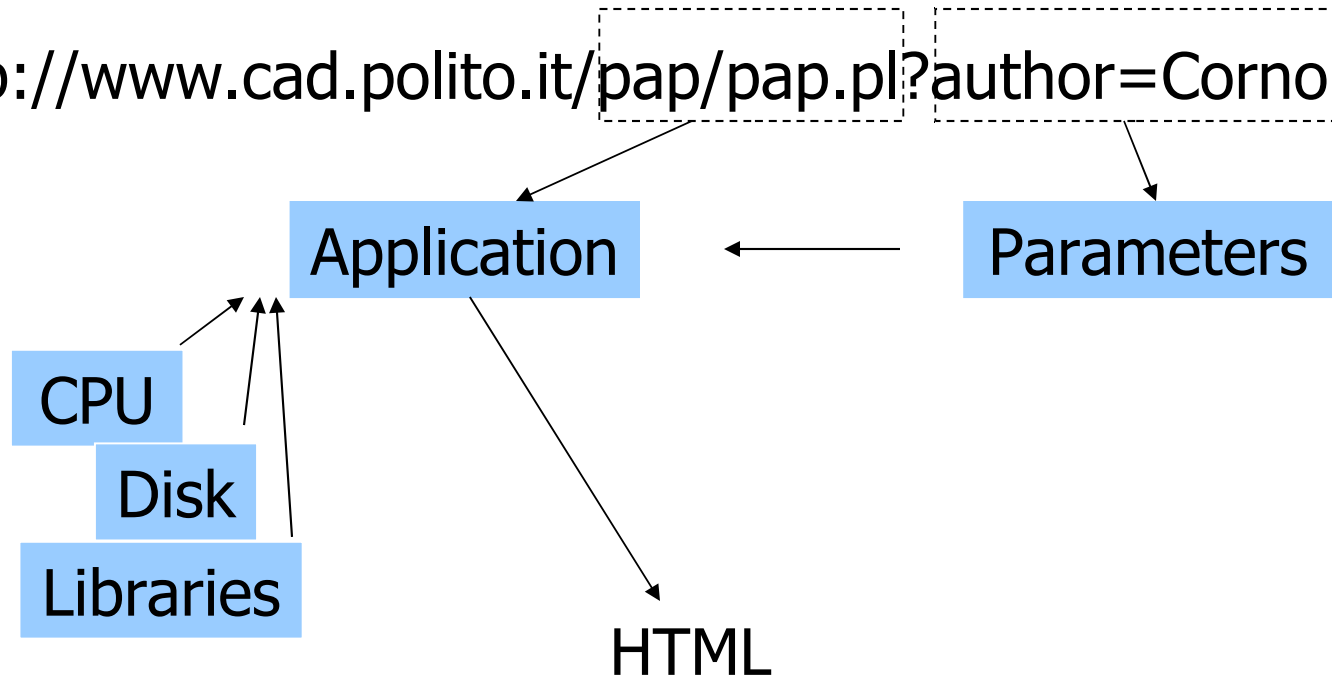


Adopted standards

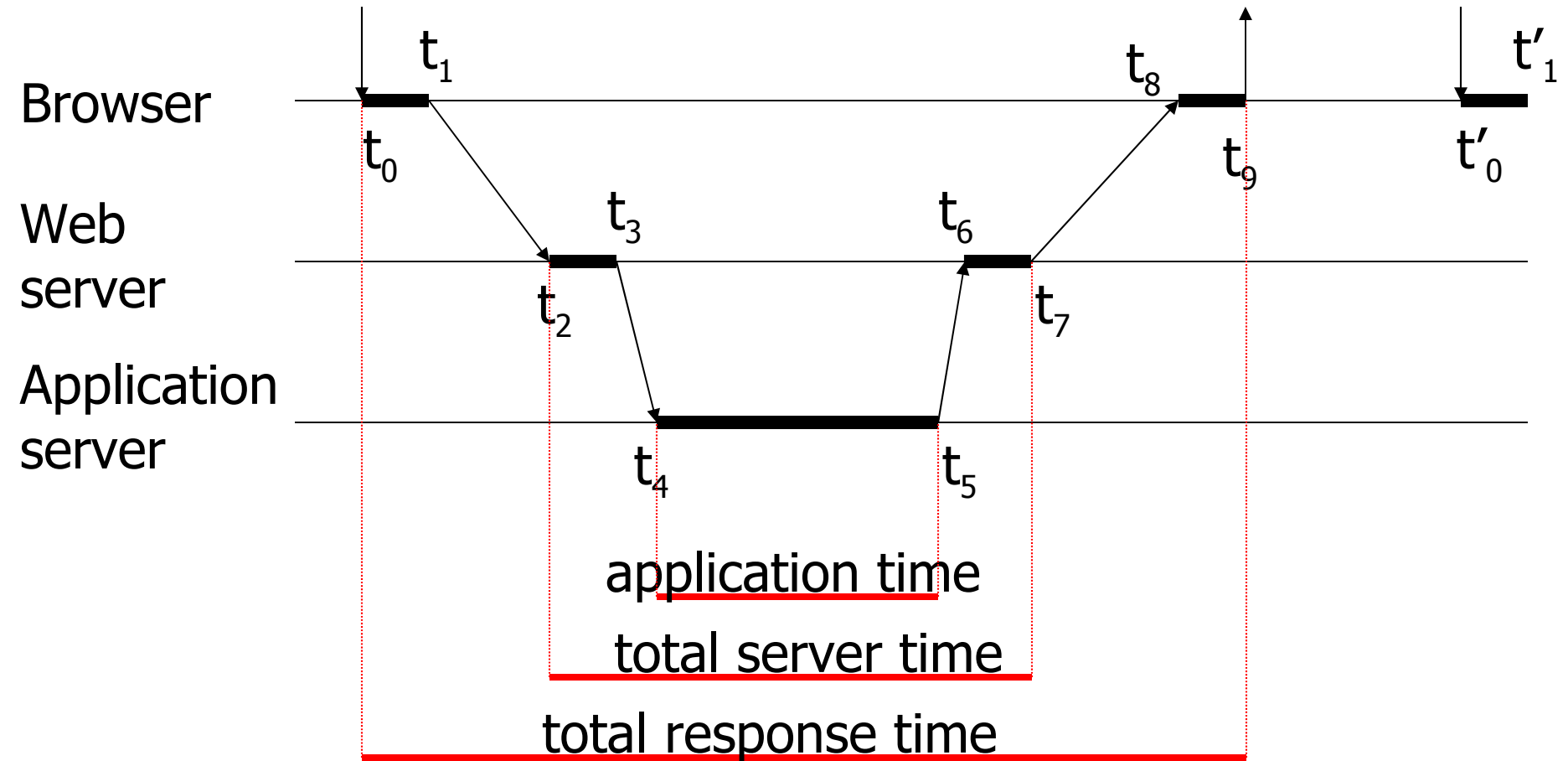
- **HTTP-POST** for sending user-specified data
- **CGI** (common gateway interface), **ISAPI** (internet information server application programming interface), **server-side script**, java **servlet** for integrating application logic into web servers
- **ASP** (active server pages), **PHP**, **PERL**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

URL (HTTP GET)

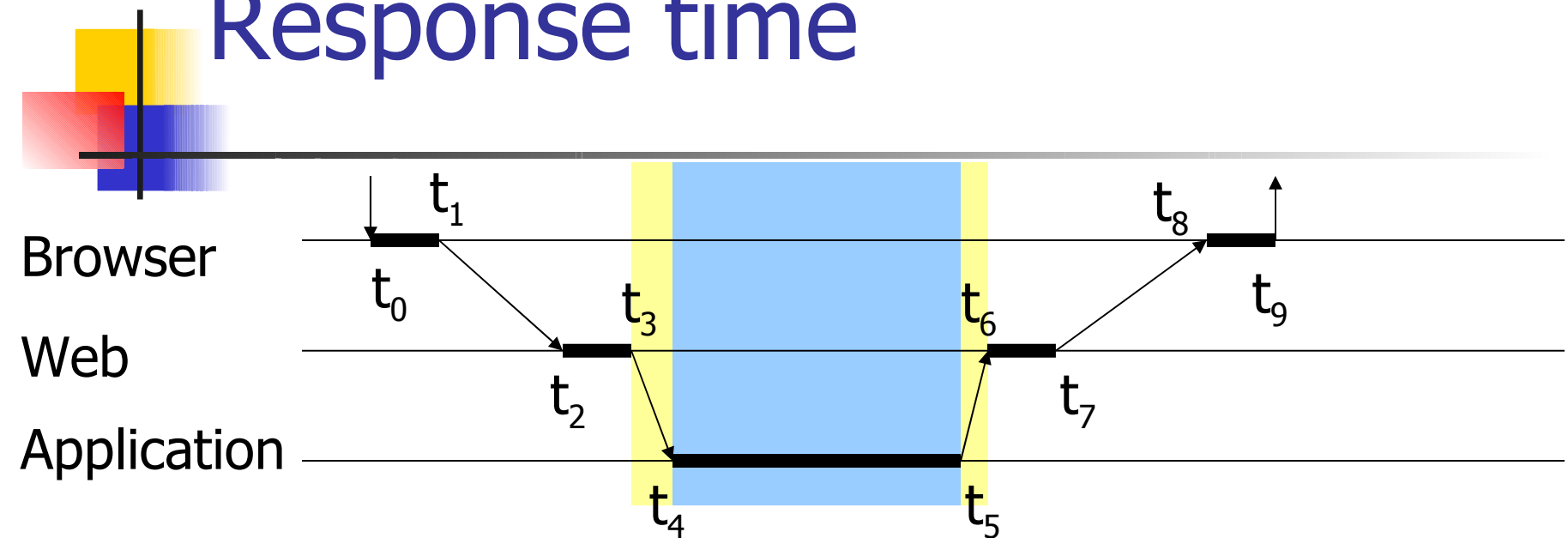
http://www.cad.polito.it/pap/pap.pl?author=Corno



Dynamic web transaction



Response time



- Communication time between web server and application server ($t_3 - t_4$, $t_5 - t_6$): speed of the **local network**
- Execution time of the application ($t_3 - t_6$): **server power**, number of **accesses**, **complexity** of the application



Examples

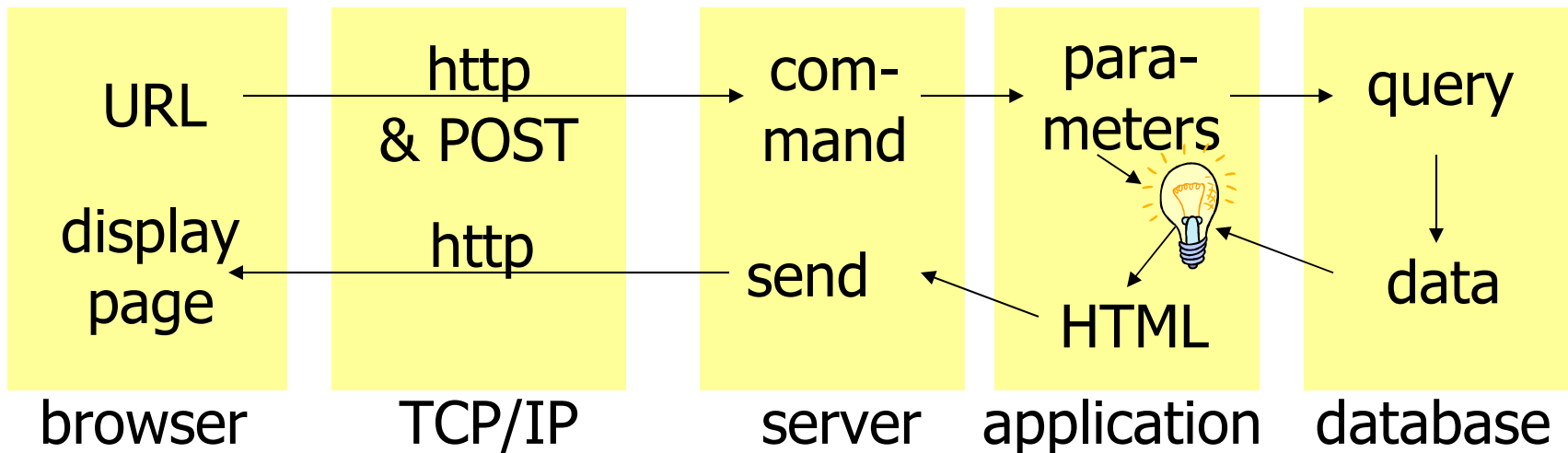
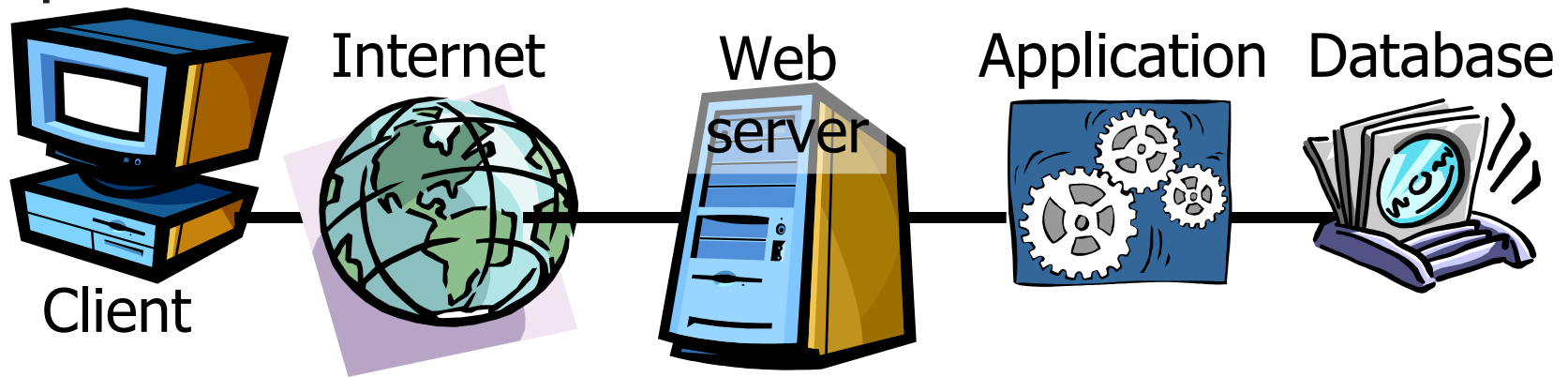
- www.sun.com
- www.lastampa.it
- www.altavista.com



Summary

- Static Transactions
- Dynamic Transactions
- Database-driven Transactions
- Rich-Client Asynchronous Transactions

Database-driven transactions



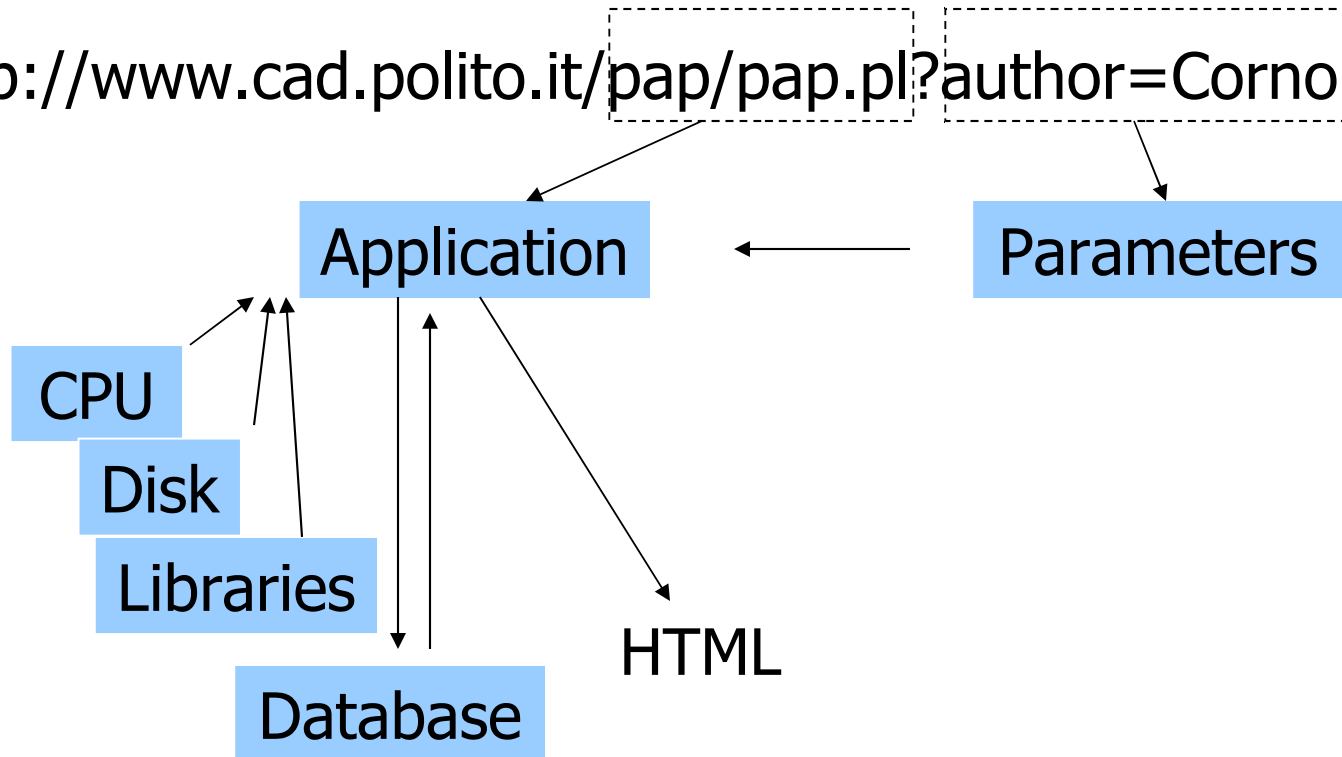


Adopted standards

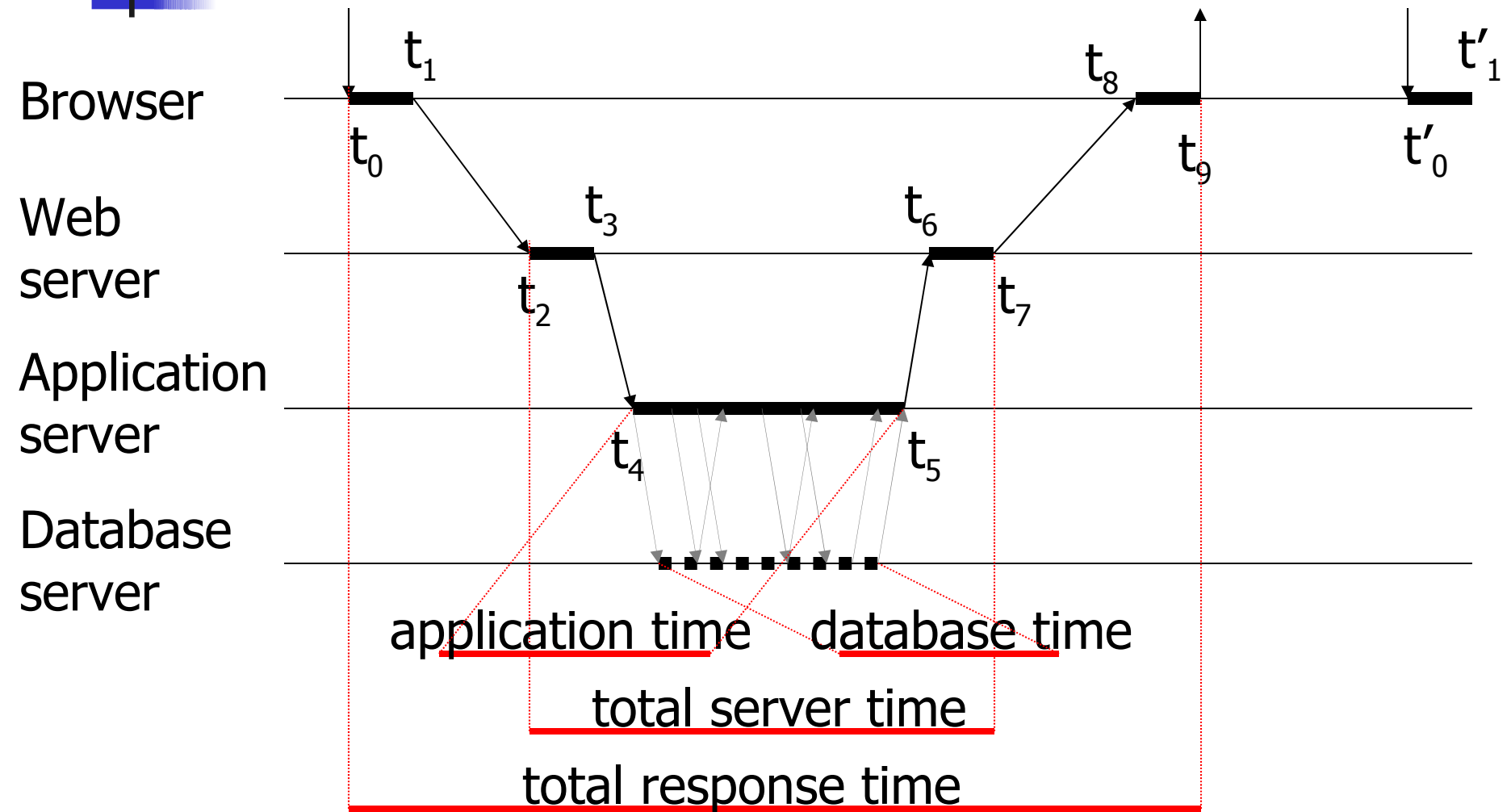
- **Cookies** for storing the state of a session
- **Java, JavaScript, ActiveX, Flash** to program the user interface on the browser
- **SQL** (structured query language), **ODBC** (open database connectivity) to access data bases

URL (HTTP GET)

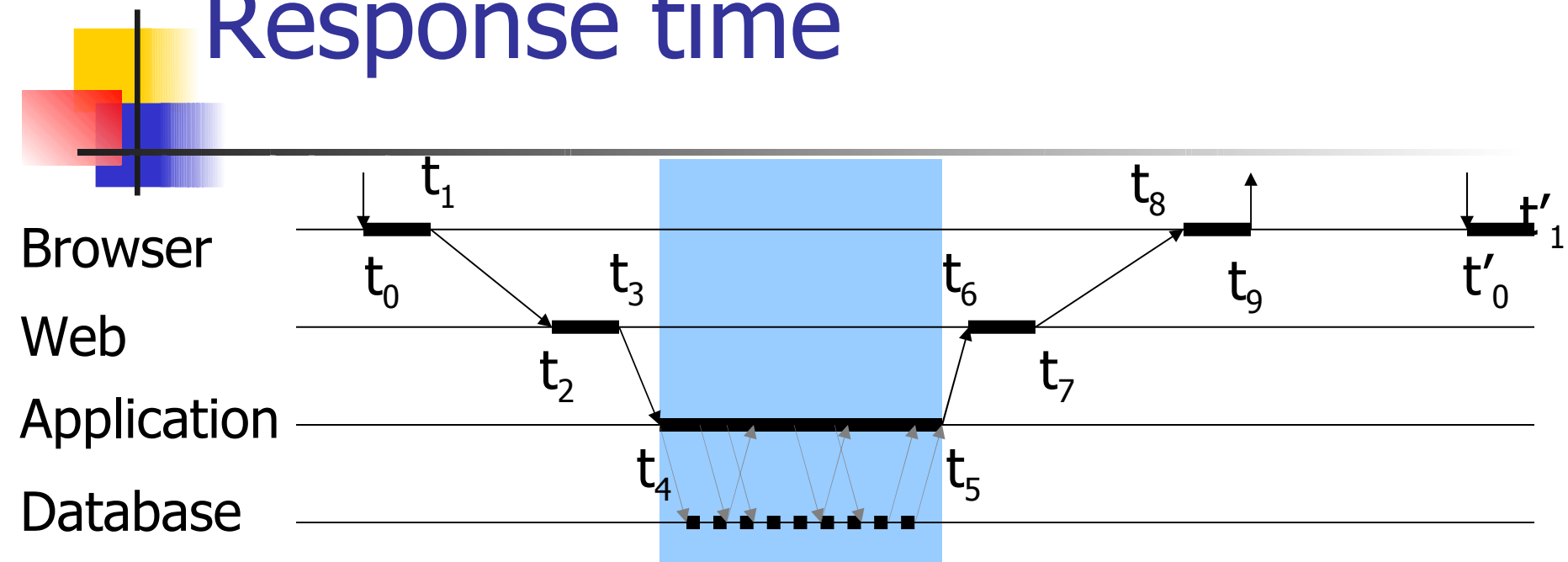
http://www.cad.polito.it/pap/pap.pl?author=Corno



Database-driven transaction



Response time



- Application: server power, number of accesses, complexity of application
- Database: number of queries, complexity of queries, size of the database
- Communications: local network



Examples

- www.amazon.com



Summary

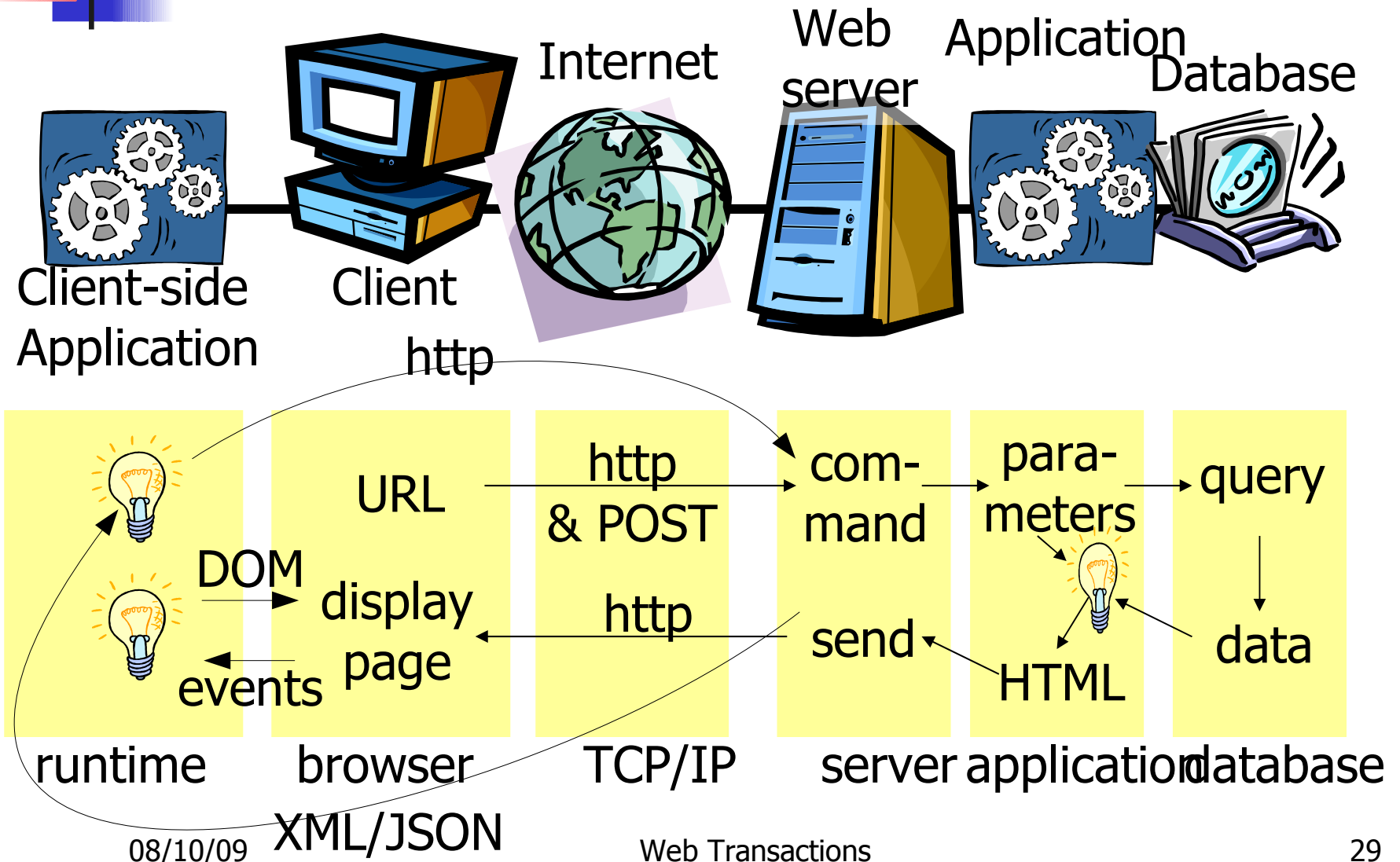
- Static Transactions
- Dynamic Transactions
- Database-driven Transactions
- Rich-Client Asynchronous Transactions



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**, ...

Rich-client transaction

