Web Applications

Databases and Web Applications Laboratory (LBAW)
Bachelor in Informatics Engineering and Computation (L.EIC)

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Current Status

➔ Plan: https://web.fe.up.pt/~ssn/wiki/teach/lbaw

  ➔ 6th week of classes;

  ➔ Lecture: start architecture specification plus prototype development (EAP);

  ➔ Labs: finish database specification (EBD = A4 + A5 + A6).

  ➔ Next week: EBD delivery.

➔ Monitor sessions: Wednesday, at 15h, online

  ➔ Previous sessions (recordings available): Git and GitFlow; PostgreSQL setup and use;

  ➔ This week: triggers, indexes, database population (A6).

Outline for Today

➔ LBAW Survey Results

➔ Web Applications

  ➔ History

  ➔ Technologies

  ➔ Architectures

➔ Architecture Specification and Prototype (EAP) component

  ➔ A7: Web Resources Specification

LBAW Artifacts

1. Requirements & UI

A1: Project Presentation

A2: Actors and User Stories

A3: Information Architecture

2. Database Specification

A4: Conceptual Data Model

A5: Relational Schema

A6: Indexes and Triggers

3. Web Architecture & Prototype

A7: Web Resources Specification

A8: Vertical Prototype
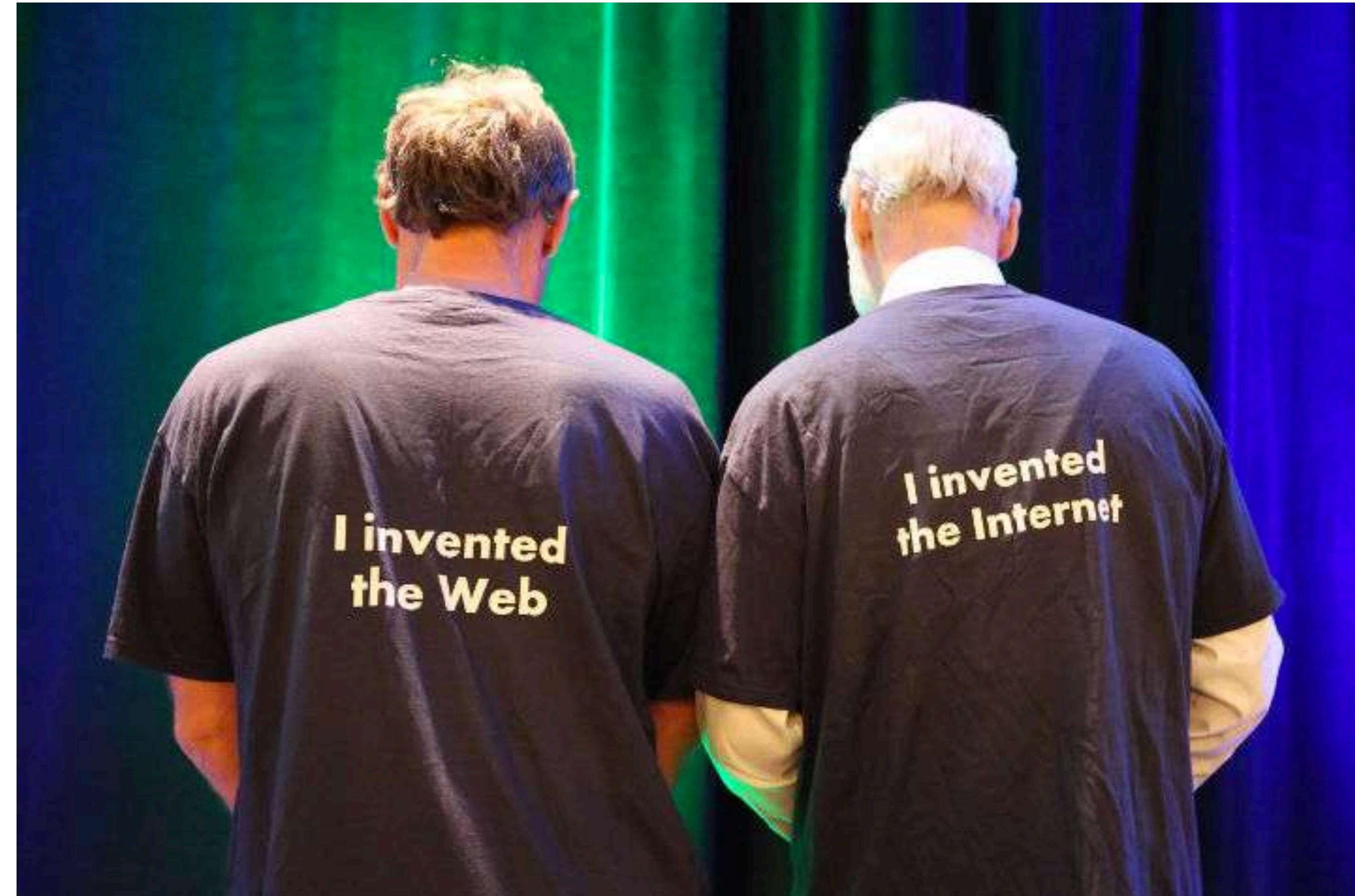
4. Product and Presentation

A9: Product

A10: Presentation and Discussion

Introduction

Who is who?

Who is who?

Web Applications

- What are Web Applications?

- A **Web Application**, or web app, is a software system, based on web standards and technologies, that is accessible through a web browser.

- Web applications changed significantly over time due to technological advancements, namely asynchronous interactions (AJAX), JavaScript developments, the new HTML5 standards, and the broad adoption of mobile devices.

- Examples: GMail, Google Search, Facebook, SIGARRA.

- Why web applications? Advantages and disadvantages?

Pros of Web Apps

- Platform independence.

- Easier updates & bug fixes.

- Only one version of the application.

- Access from anywhere.

- Reduced "piracy".

- No installation hurdles.

- Developers can measure user interaction in real-time.
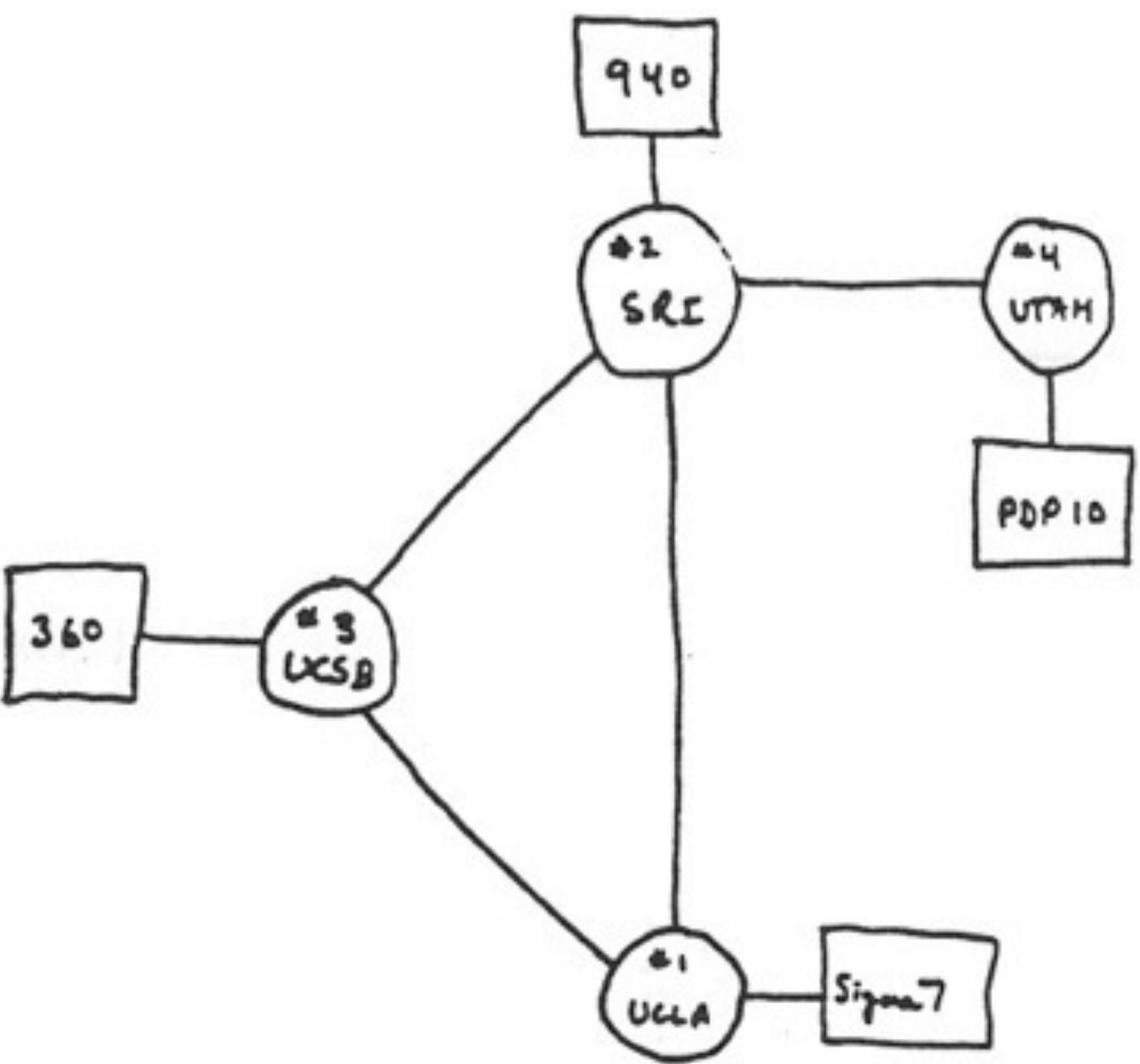
Cons of Web Apps

- Depends on network connectivity.

- Less sophisticated UIs.

- Limited hardware access.

- Reduced OS integration (e.g. drag&drop).

- Need to address browser versions.

- Harder to debug.

- Higher security risks.

- Infrastructure costs.

The Internet

Internet Origins
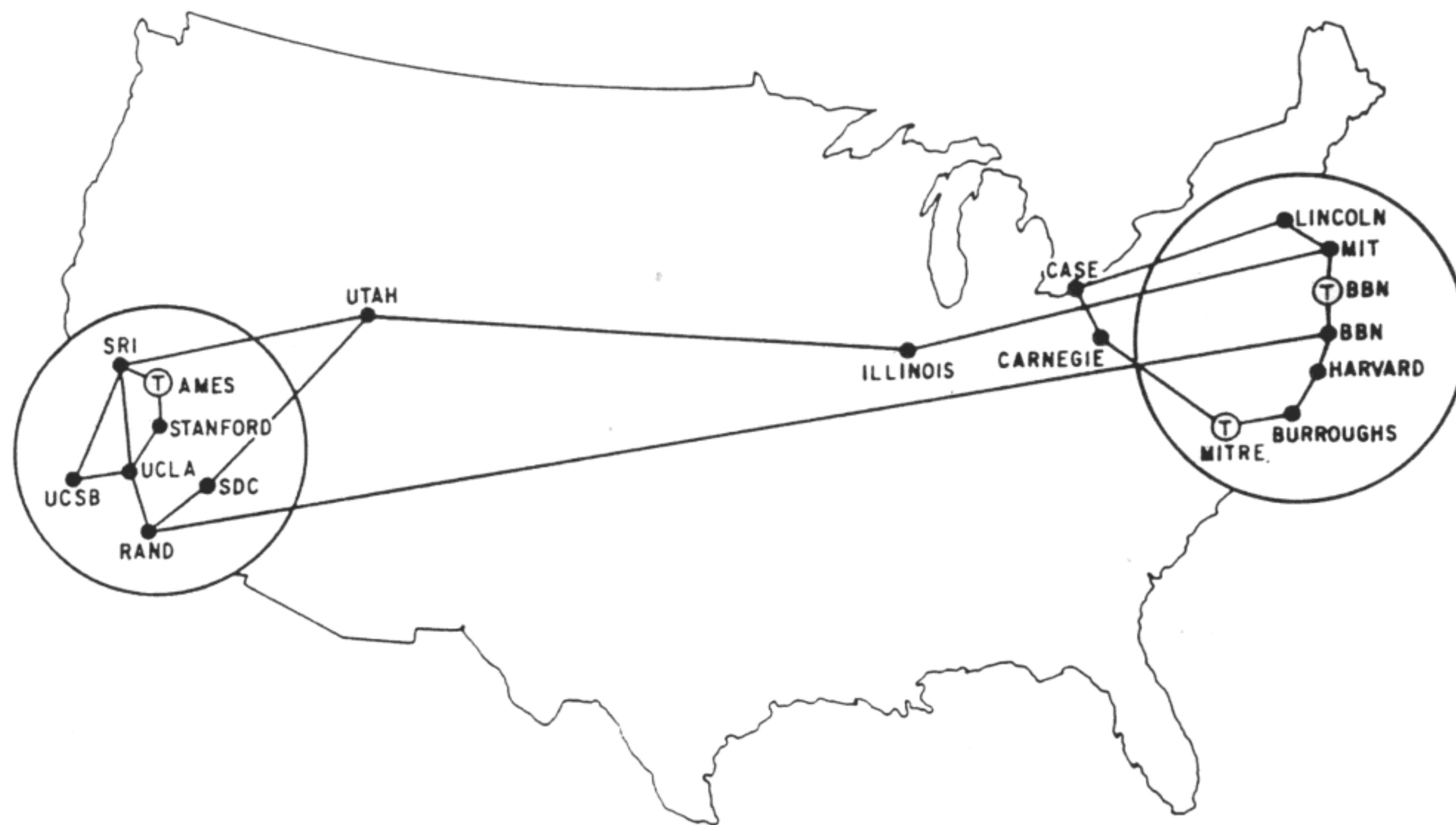
- Started as a project of the USA Department of Defense (DoD) Advanced Research Projects Agency (ARPA) in 1958.

- Developed the idea of "networked computers" which led to the creation of the ARPANET in 1967. The core challenge was how to connect separate physical networks without requiring permanent links between them.

- Other networks appeared worldwide during the same period. E.g. JANET, X.25, USENET, CompuServe.

THE ARPA NETWORK

DEC 1969
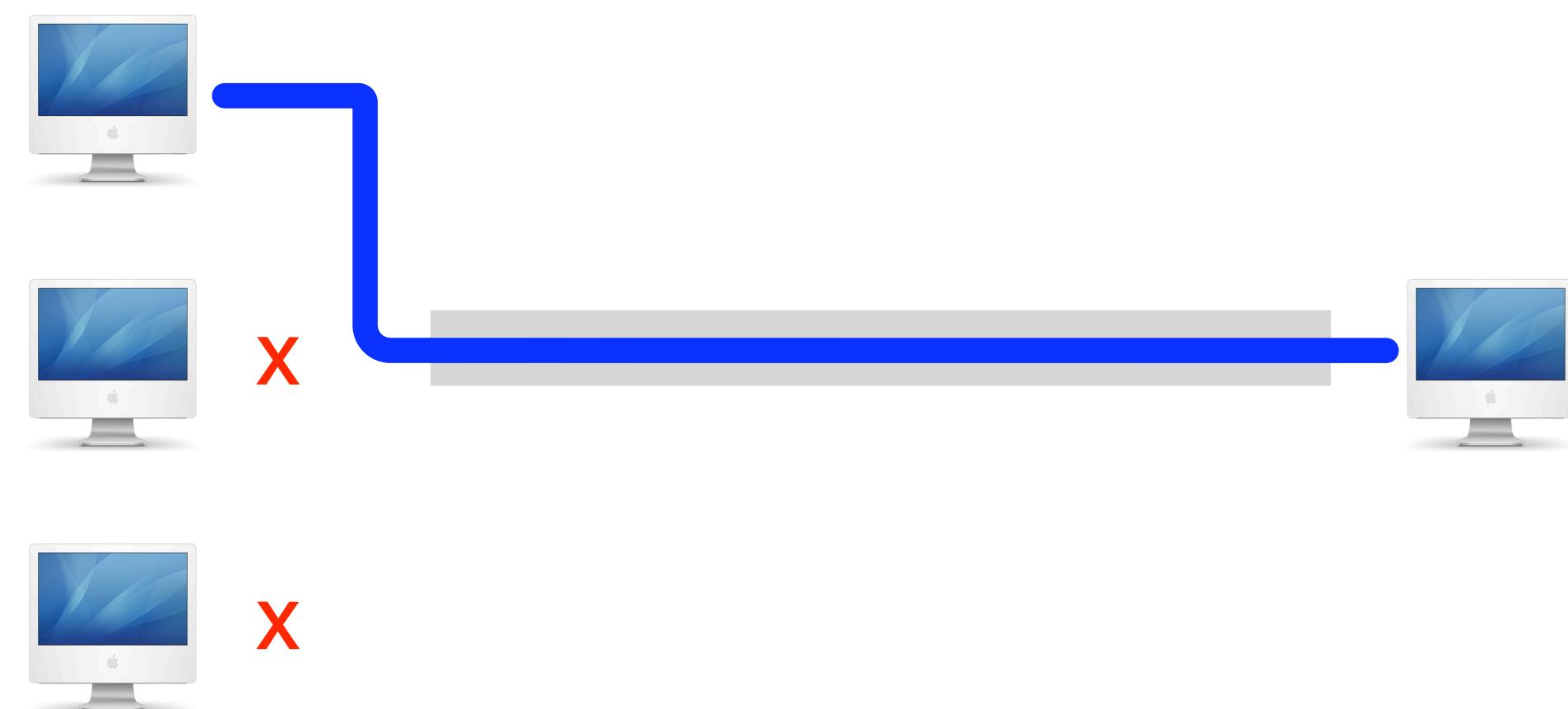
4 NODES
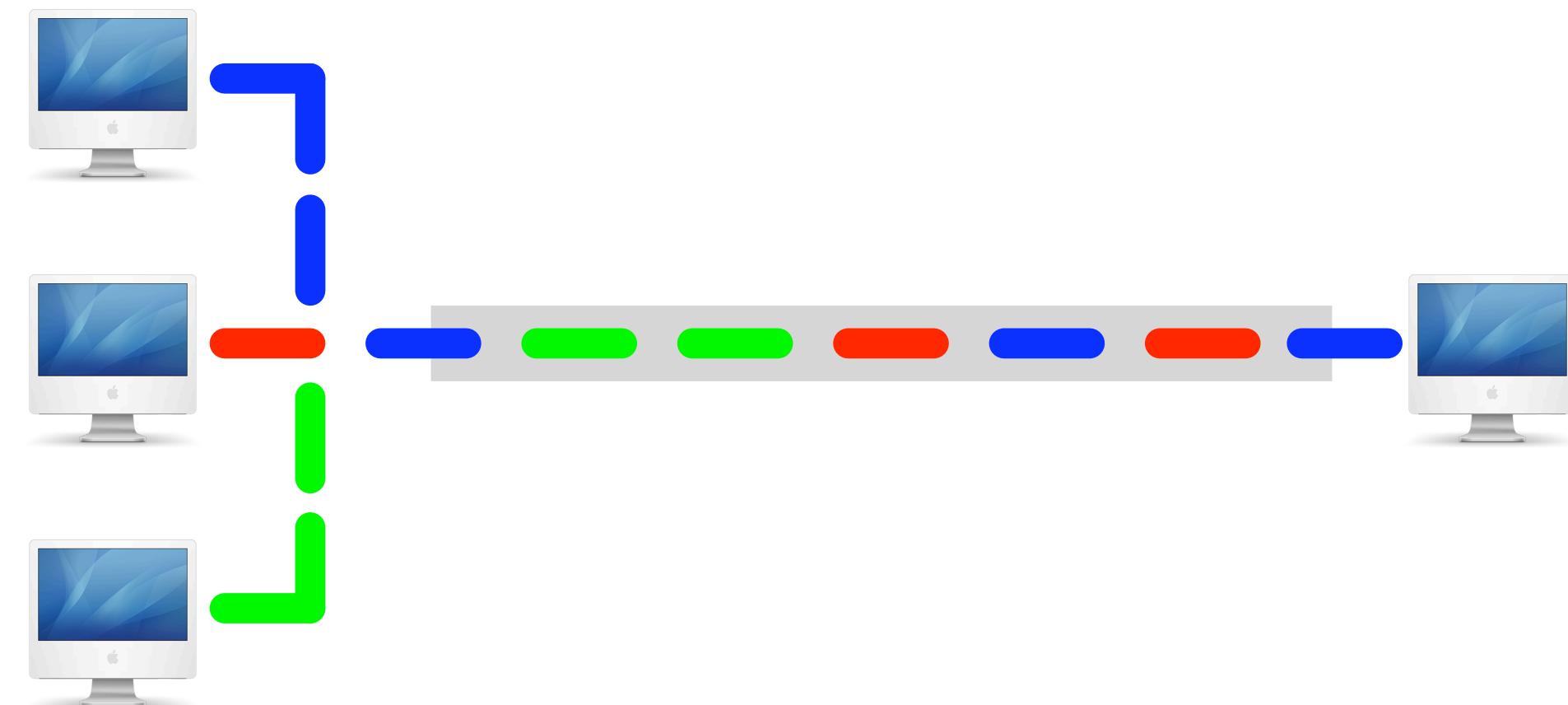
MAP 4   September 1971

32

Packet-Switching

- In the late 1960s, computers were expensive and rare. Communication networks were used to make these resources available to wider audiences.

- One of the first challenges faced was how to connect separate physical networks without dedicated links.

- Implementing point-to-point connections does not scale. The only option was to share the available links to optimize resource usage.

- **Packet switching** is a method where data is divided in small chunks and sent out separately.

- With packet switching it is possible to have multiple connections over the same link, i.e. share the communication medium.

Continuous transmission

Packet-based transmission

Internetworking

- The proliferation of different networking technologies and protocols became a problem when trying to connect different networks.

- No network technology is ideal for all scenarios (e.g. Ethernet, Wireless, DSL), thus different technologies will always co-exist. Each network technology was becoming an island, isolated from all others.

- To overcome this problem, and achieve a homogeneous service across heterogeneous networks, both **hardware** and **software** were combined.

- Hardware: routers.

- Software: protocols, specifically TCP/IP.
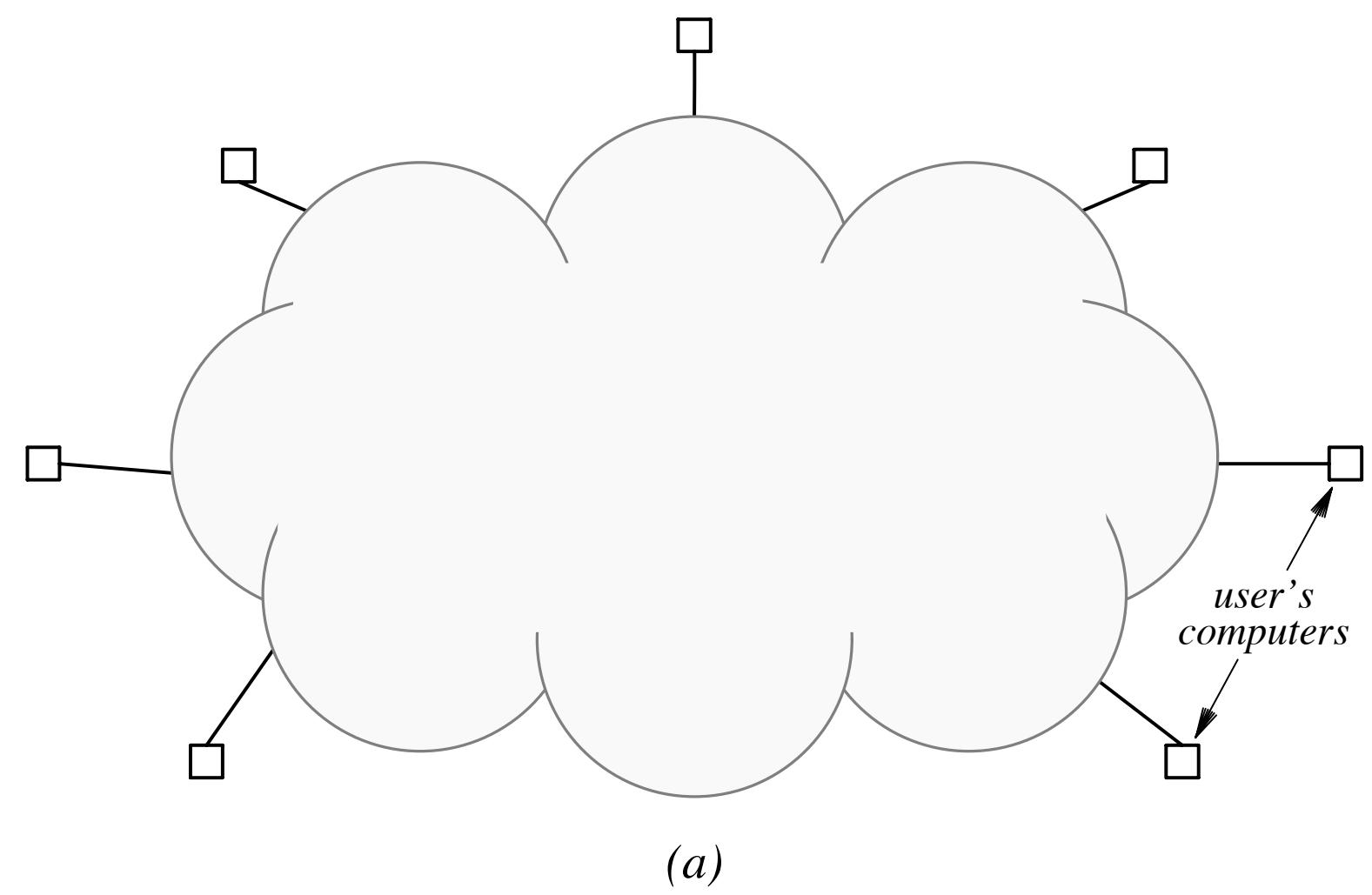
Hardware — Routers

- The router is the core hardware equipment used to connect networks using different physical technologies. There is a vast number of router types.

- An internet (note the lowercase) is a set of networks connected by routers.
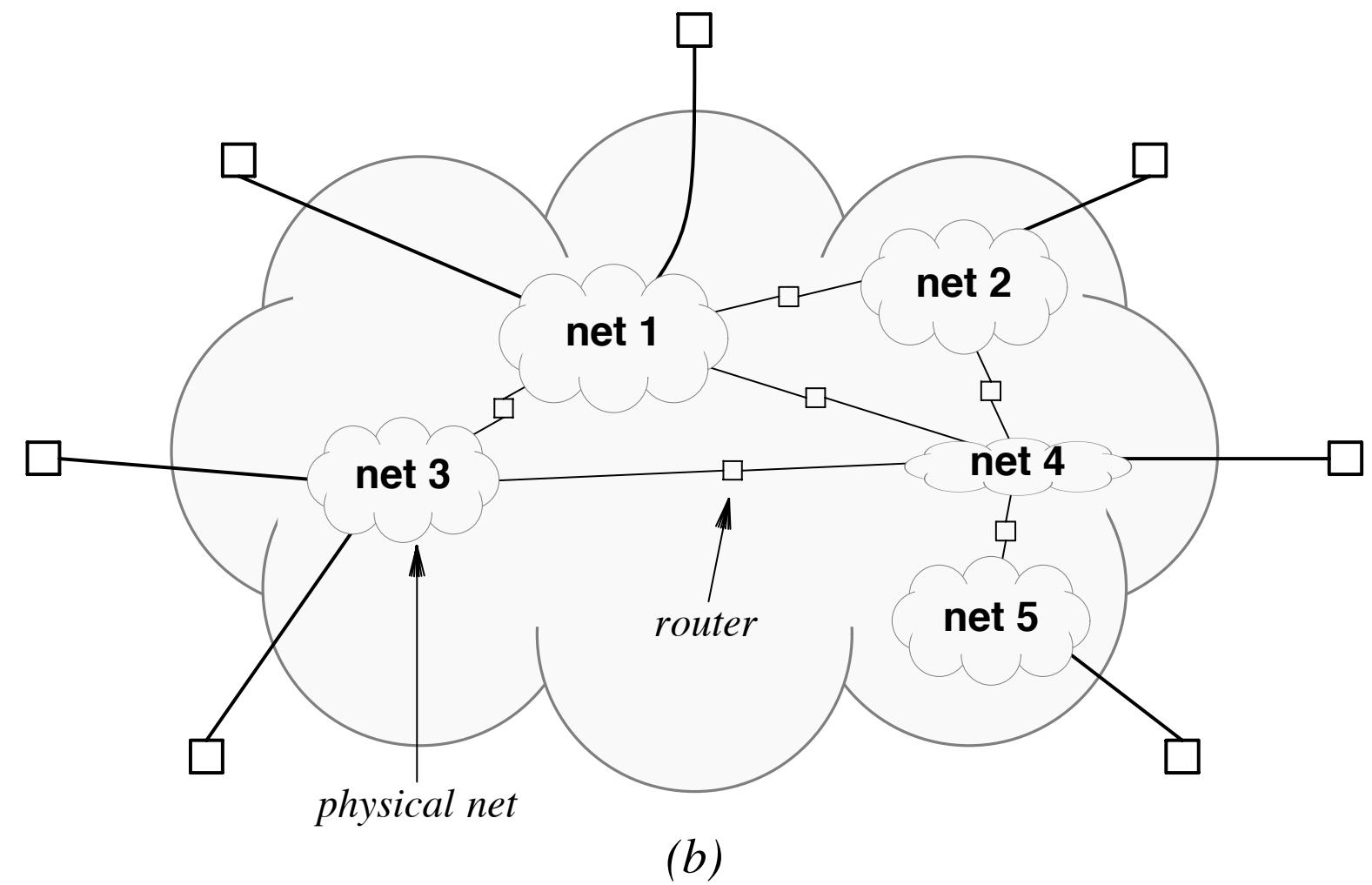
Software — Protocols

- To connect different networks we need communication protocols.

- These protocols establish message formats and message exchanging rules.

- The most important protocols for connecting different networks are called the Internet Protocols or TCP/IP Protocols.

- These protocols were developed in the 1970s and approved as standards in the 1980s.

(a)
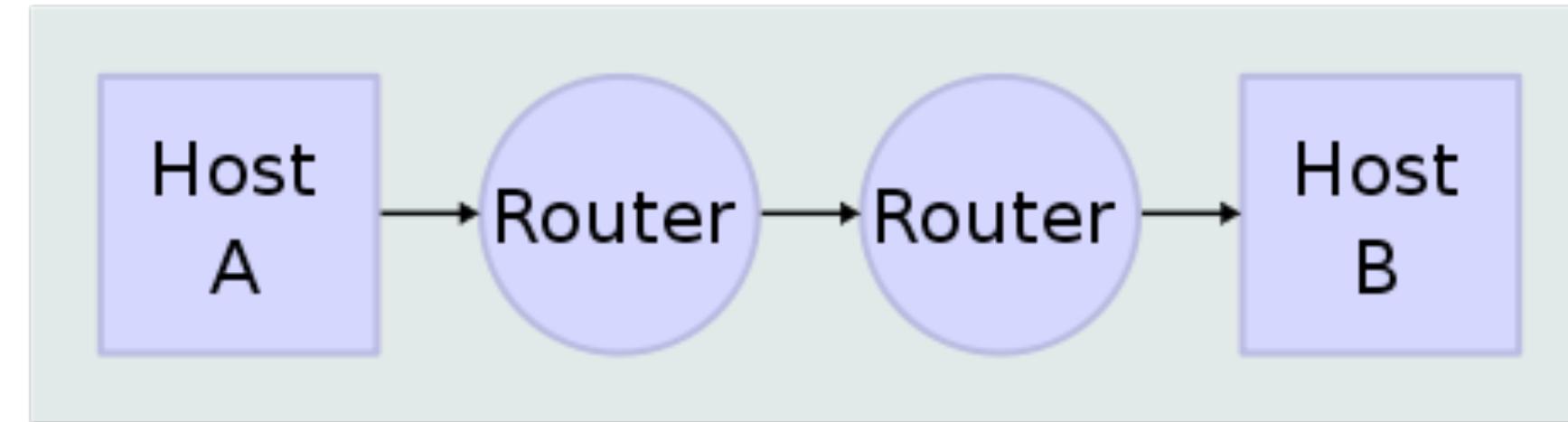
An internet is a virtual network because in reality it is built by combining many physical networks.

(b)
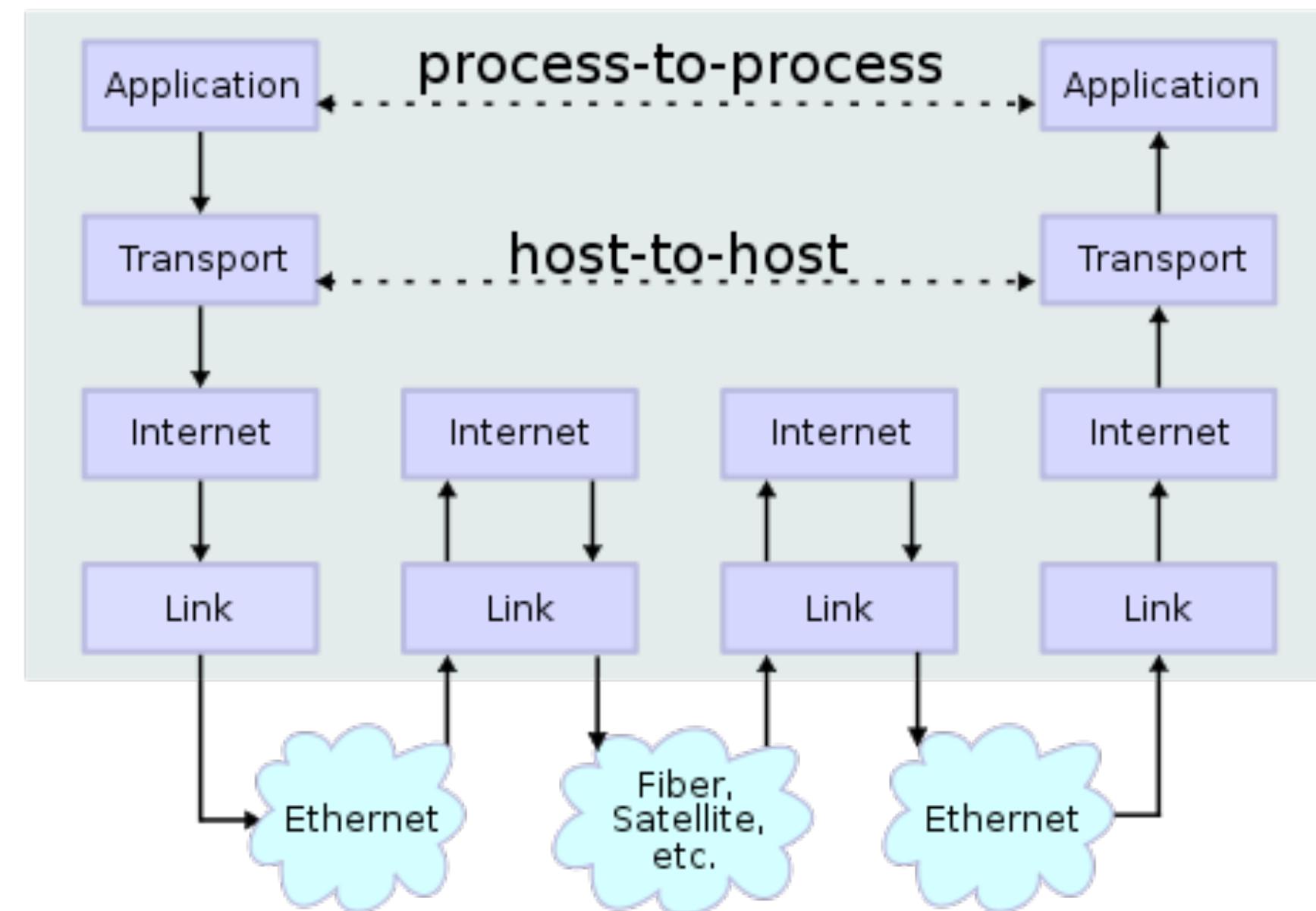
An internet is a network of networks, not a network of computers.

Network Topology

Host A → Router → Router → Host B

Data Flow

Application			Application
	process-to-process		
Transport			Transport
	host-to-host		
Internet	Internet	Internet	Internet
Link	Link	Link	Link

Ethernet   Fiber, Satellite, etc.   Ethernet

Internet Protocols

| BGP | FTP | HTTP | SMTP | TELNET | | SNMP |

| TCP | UDP |

| ICMP | IGMP | OSPF | RSVP |

| IP |
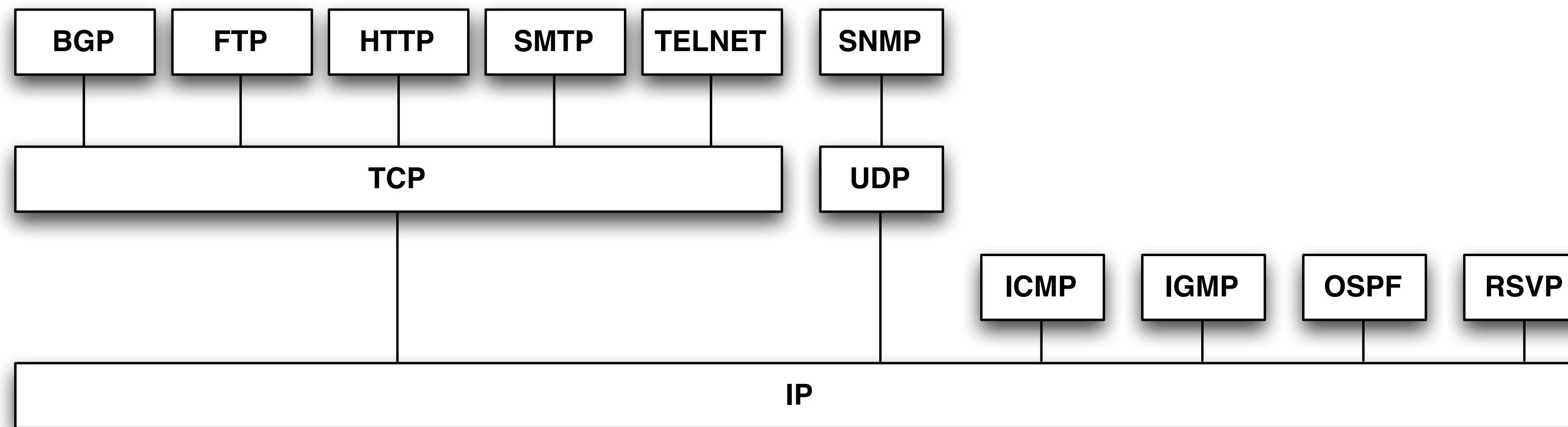
BGP = Border Gateway Protocol

FTP = File Transfer Protocol

HTTP = HyperText Transfer Protocol

ICMP = Internet Control Message Protocol

IGMP = Internet Group Management Protocol

IP = Internet Protocol

OSPF = Open Shortest Path First

RSVP = Resource ReSerVation Protocol

SMTP = Simple Mail Transfer Protocol

SNMP = Simple Network Management Protocol

TCP = Transmission Control Protocol

UDP = User Datagram Protocol

Internet Protocol (IP)

- The main function of the Internet Protocol is to offer a virtual network, hiding the underlying physical networks.

- It offers two fundamental services:

  - Addressing system (IP addresses).

  - Datagram structure (packets).

Transmission Control Protocol (TCP)

- The Transmission Control Protocol offers a reliable and ordered delivery of packets between applications in different computers.

- Handles problems not addressed in the lower layers: packet duplication and loss, packer ordering, communication delays, among others.

- Supports important applications such as the WWW, e-mail, FTP, etc.

Internet Services

- DNS — Maps IP addresses to symbolic names.

- SMTP — Handles electronic messages (e-mail).

- SFTP — Offers a mechanism for secure file transfers between computers.

- WWW (HTTP) — A hypertext-based distributed information system.

- …

Domain Name System (DNS)

- The Domain Name System is an application layer service of the Internet.

- Translates human-readable symbolic names to numeric addresses (IP).

- Symbolic names are organized hierarchically.
  The right-most element is the top-level domain (TLD).

- There are different groups of TLDs, namely country TLDs and generic TLDs.

- Country TLDs (ccTLD): .pt, .es, .uk, .usa, .fr, .it, .ao, .fi, .io, .tv ...

- Generic TLDs (gTLD): .com, .net, .org, .name, .biz ...

Generic TLDs

- Generic top-level domains (TLDs) are maintained by the Internet Assigned Numbers Authority (IANA), a department of the Internet Corporation for Assigned Names and Numbers (ICANN), a nonprofit private USA organization.

- Initial list of general purpose domains (1984): com, edu, gov, mil, org and net.

- Recent gTLDs: .biz, .info, .jobs, .mobi, .name, .guru, .today, .xyz …
  https://www.gandi.net/en/tlds

- In 2012, ICANN decided to open the creation of new gTLDs through a process where organizations can propose new strings.
  There is some debate around this expansion.

- There are currently more than 1,500 TLDs.

.pt Domain

- .pt is the country code top-level domain (ccTLD) for Portugal.

- The .pt ccTLD is currently managed by Associação DNS.PT, which replaced the Fundação para a Computação Científica Nacional (FCCN) as the domain name registry in Portugal.

- FCCN kept very strict rules during the first years. Multiple adjustments have been made during the years. FCCN also tried to promote the .com.pt subdomain as a more flexible solution.

- In March 2019 there were approximately 350,000 active .pt domain names. More than a one million registered. https://www.dns.pt/estatisticas

WHOIS Protocol

- WHOIS is a query/response protocol used to query databases that contain information about internet resources, such as domain names or IP addresses.

- The protocol presents the information in a human-readable format.

- It is the *de facto* standard for querying domain name information.

https://www.dns.pt/pt/ferramentas/whois/detalhes/?site=up&tld=.pt

https://reports.internic.net/cgi/whois?whois_nic=facebook.com&type=domain

% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%   To receive output for a database update, use the "-B" flag.

% Information related to '193.137.48.0 - 193.137.55.255'

% Abuse contact for '193.137.48.0 - 193.137.55.255' is 'report@cert.pt'

```
inetnum:        193.137.48.0 - 193.137.55.255
netname:        PTUNET-UP-6
descr:          Universidade do Porto
descr:          Porto
country:        PT
admin-c:        LMR2-RIPE
tech-c:         FC1899-RIPE
tech-c:         FL514-RIPE
tech-c:         RC14028-RIPE
tech-c:         RR9044-RIPE
status:         ASSIGNED PA
remarks:        SERVIP-UP
remarks:        created 19951220
mnt-by:         AS1930-MNT
mnt-lower:      AS1930-MNT
source:         RIPE # Filtered

person:         Fernando Correia
address:        Universidade do Porto - Reitoria
address:        Praca Gomes Teixeira
address:        4099-002 Porto
address:        PORTUGAL
phone:          +351 220408000
fax-no:         +351 220408186
nic-hdl:        FC1899-RIPE
mnt-by:         AS1930-MNT
source:         RIPE # Filtered
```

The RIPE database contains registration details of IP addresses.

Domain Name Registrars

- Domains are reserved and managed by accredited organizations according to the rules of each specific TLD.

- Domain names are rented for a limited period (e.g. 1 year).
  At the end of the registration period, the owner can decide to renew it or not.

- Different TLDs have different registration prices, some examples:
  .com, .net, .org (~8€/year), .pt (~15€/year), .io (~90€/year), .biz (~15€/year).

Extensions	Creation	Renewal	Transfer

Find your own domain name　　　　　　　**Bulk registration**　🔍

	Creation	Renewal	Transfer
.com	€15.42	€15.99	€11.73
.pt	€20.44	€20.44	Free
PROMO **.site**	~~€13.53~~ €1.22	€35.74	€35.74
PROMO **.net**	~~€20.91~~ €10.46	€20.91	€17.22
.io	€39.36	€39.36	€36.90
.fr	€14.76	€14.76	€14.76
PROMO **.club**	~~€17.50~~ €4.92	€17.50	€14.83
.app	€24.48	€24.48	€20.74
PROMO **.online**	~~€14.76~~ €7.37	€52.96	€44.88
.page	€16.45	€16.45	€13.94
.museum	€82.15	€82.15	€69.62

Want to see prices for our 700+ extensions?　　**See more**　→

52
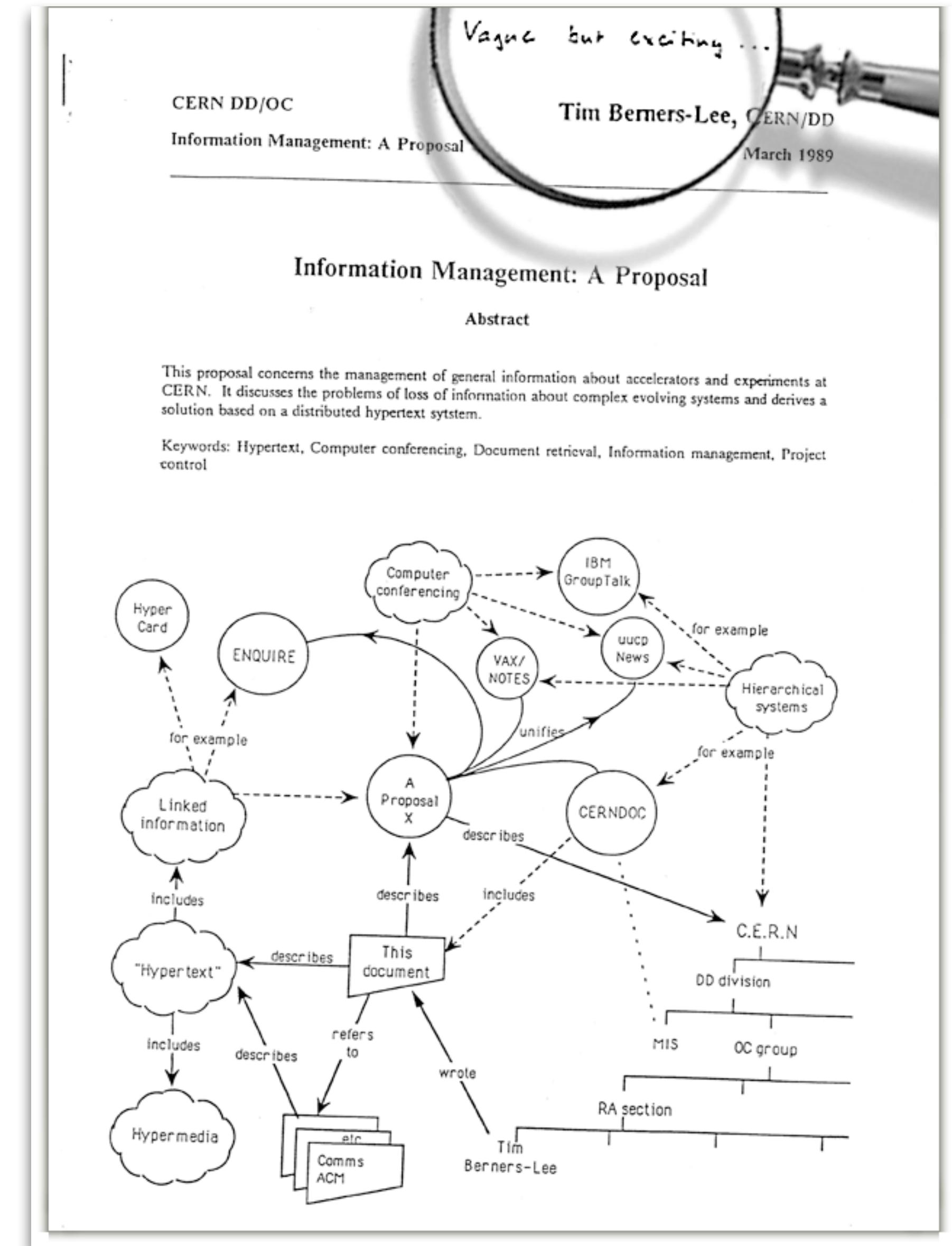
Registo e renovação de domínios

Domínio	1 ano	2 anos	3 anos	5 anos
.pt	19,95 €	–	59,00 €	98,00 €
.com.pt, .org.pt*	19,95 €	–	59,00 €	98,00 €
.com, .net, .biz	19,95 €	39,50 €	59,00 €	98,00 €
.org	19,95 €	39,50 €	59,00 €	98,00 €
.info	19,95 €	39,50 €	59,00 €	98,00 €
.eu	19,95 €	39,50 €	59,00 €	98,00 €
.me	22,95 €	45,90 €	68,85 €	114,75 €
.tv	32,95 €	65,90 €	–	164,75 €
.co.uk	19,95 €	39,50 €	59,00 €	98,00 €
.co	27,95 €	55,90 €	83,85 €	139,75 €
.es	19,95 €	39,50 €	59,00 €	98,00 €
.xxx	90,00 €	156,00 €	234,00 €	385,00 €

The World Wide Web

WWW Origins

- The World Wide Web was invented in 1989 at the European Council for Nuclear Research (CERN), Europe.

- It was a joint work by Tim Berners-Lee and Robert Cailliau to share and link information of various kinds, where the user could browse at will.

- Basically, a distributed information system over the Internet, designed to facilitate content sharing across different computer systems and technologies.

- Initial proposal "WorldWideWeb", or simply WWW or W3.

- "Information Management: A Proposal", May 1989 http://www.w3.org/History/1989/proposal.html

- 30 years in 2019. See: https://web30.web.cern.ch

The original proposal: "Vague but exciting…"

Initial Success

- The WorldWideWeb source code was released into the public domain in 1993. The software that was open-sourced included a simple client, a server, and a common library. The protocols were also released royalty-free.

- The royalty-free license was a key factor in the initial success of the WWW when compared to similar alternatives, e.g. WAIS, Gopher, ARCHIE.

- The NCSA released Mosaic, a software program that was a combined WWW browser and Gopher client.

- Mosaic's popularity was determinant to the growth of the World Wide Web. Mosaic introduced significant innovations at the graphical interface level, namely the integration of text and images in a single page.

ORGANISATION EUROPEENNE POUR LA RECHERCHE NUCLEAIRE

CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

STATEMENT CONCERNING CERN W3 SOFTWARE RELEASE INTO PUBLIC DOMAIN

TO WHOM IT MAY CONCERN

Introduction

The World Wide Web, hereafter referred to as W3, is a global computer networked information system.

The W3 project provides a collaborative information system independent of hardware and software platform, and physical location. The project spans technical design notes, documentation, news, discussion, educational material, personal notes, publicity, bulletin boards, live status information and numerical data as a uniform continuum, seamlessly intergated with similar information in other disciplines.

The information is presented to the user as a web of interlinked documents .

Acces to information through W3 is:

- via a hypertext model;
- network based, world wide;
- information format independent;
- highly platform/operating system independent;
- scalable from local notes to distributed data bases.

Webs can be independent, subsets or supersets of each other. They can be local, regional or worldwide. The documents available on a web may reside on any computer supported by that web.

...

2.

Declaration

The following CERN software is hereby put into the public domain:

- W 3 basic ("line-mode") client
- W 3 basic server
- W 3 library of common code.

CERN's intention in this is to further compatibility, common practices, and standards in networking and computer supported collaboration. This does not constitute a precedent to be applied to any other CERN copyright software.

CERN relinquishes all intellectual property rights to this code, both source and binary form and permission is granted for anyone to use, duplicate, modify and redistribute it.

CERN provides absolutely NO WARRANTY OF ANY KIND with respect to this software. The entire risk as to the quality and performance of this software is with the user. IN NO EVENT WILL CERN BE LIABLE TO ANYONE FOR ANY DAMAGES ARISING OUT THE USE OF THIS SOFTWARE, INCLUDING, WITHOUT LIMITATION, DAMAGES RESULTING FROM LOST DATA OR LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES.

Geneva, 30 April 1993

W. Hoogland
Director of Research

H. Weber
Director of Administration
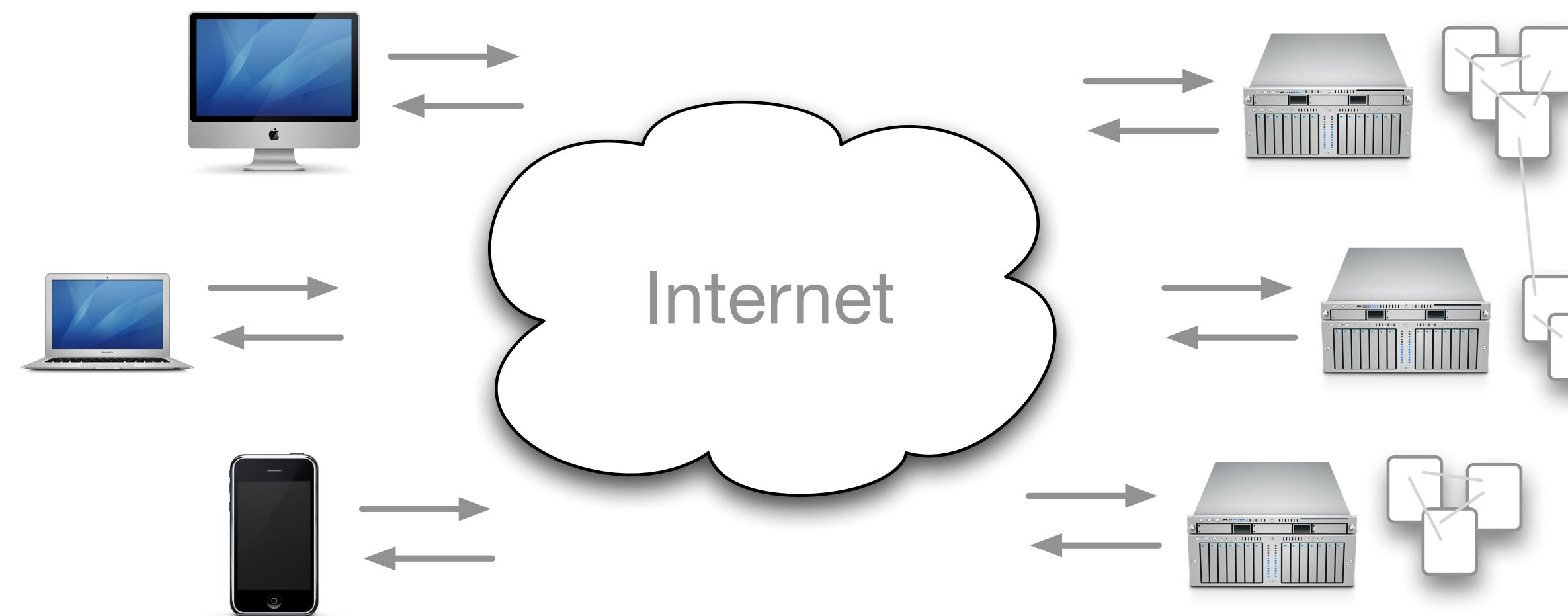
opie certifiée conforme

ait à Genève le 03-05-93

DIVISION DES FINANCES
CERN
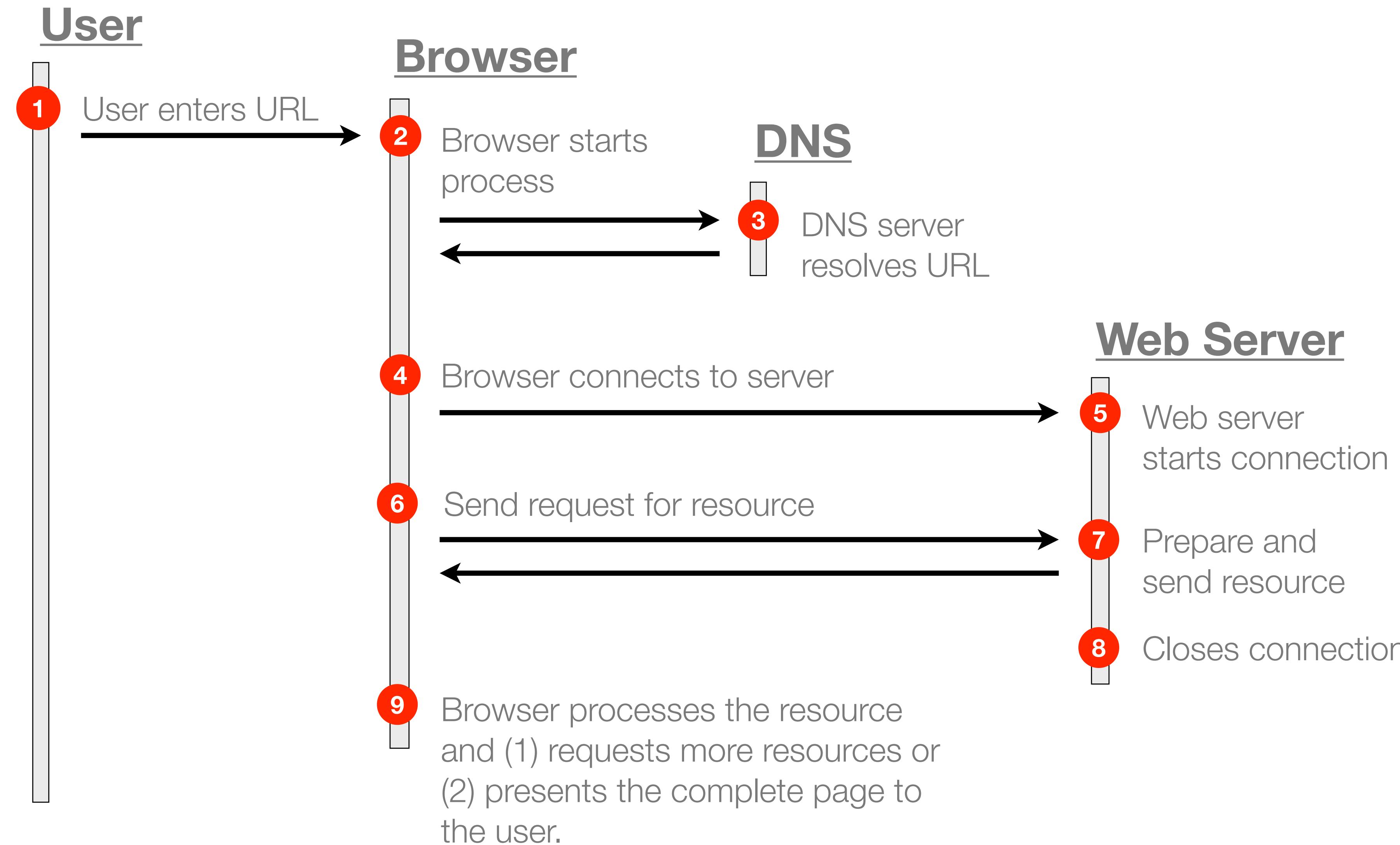
April 1993, CERN puts the WWW in the public domain

WWW Architecture

- The Web's architecture follows a standard **client-server model**, where servers provide a function and clients initiate requests for those services.

- **Servers** are machines that are running server applications waiting for requests from clients. Each server can simultaneously serve multiple clients.

- **Clients** are typically web browsers that initiate the communication session with servers. Interactions are simple, one request results in one response.

Internet
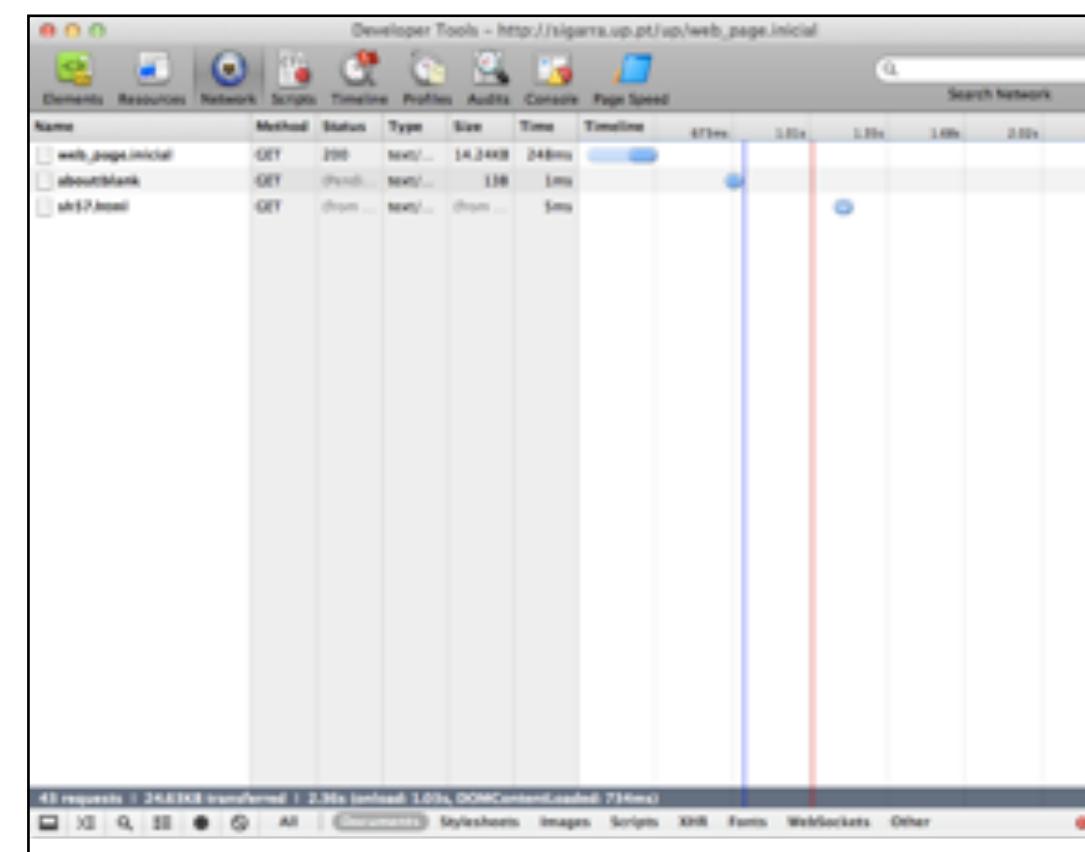
Client Server Interaction

User

Browser

DNS

Web Server

1 User enters URL

2 Browser starts process

3 DNS server resolves URL

4 Browser connects to server

5 Web server starts connection

6 Send request for resource

7 Prepare and send resource

8 Closes connection

9 Browser processes the resource and (1) requests more resources or (2) presents the complete page to the user.
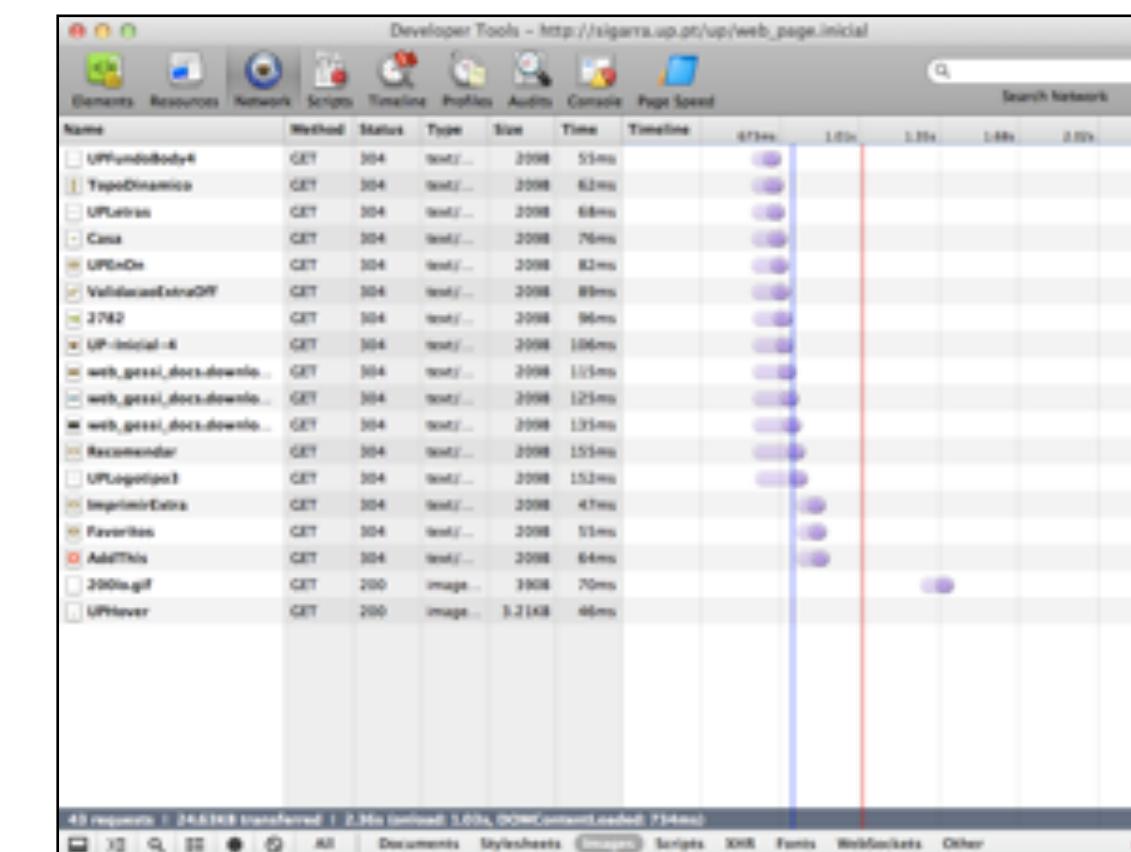
Requesting a Web Page
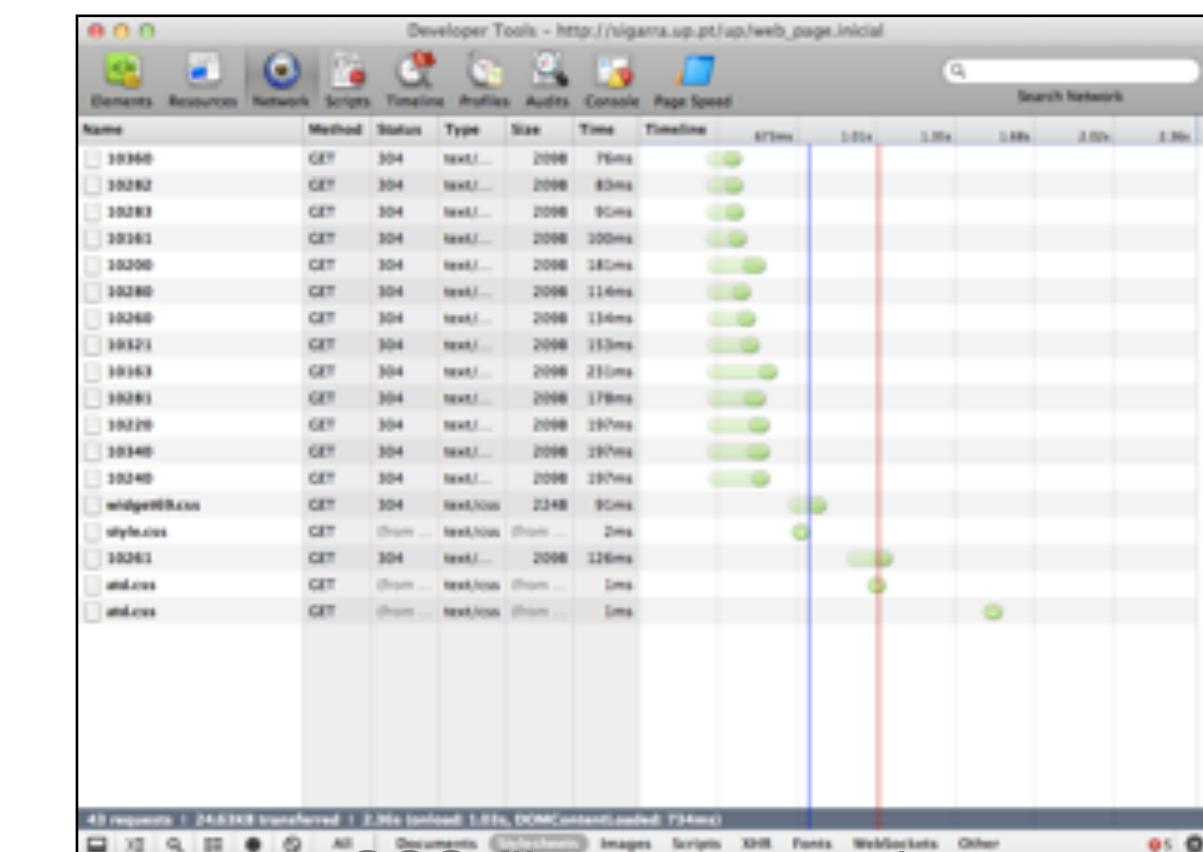
- Web pages combine text, images, and other resources.

- Web browsers issue multiple requests when preparing a single web page, one request for each individual resource (e.g. HTML document, CSS files, images, JavaScript files, etc).

Base documents requested.

Images requested.
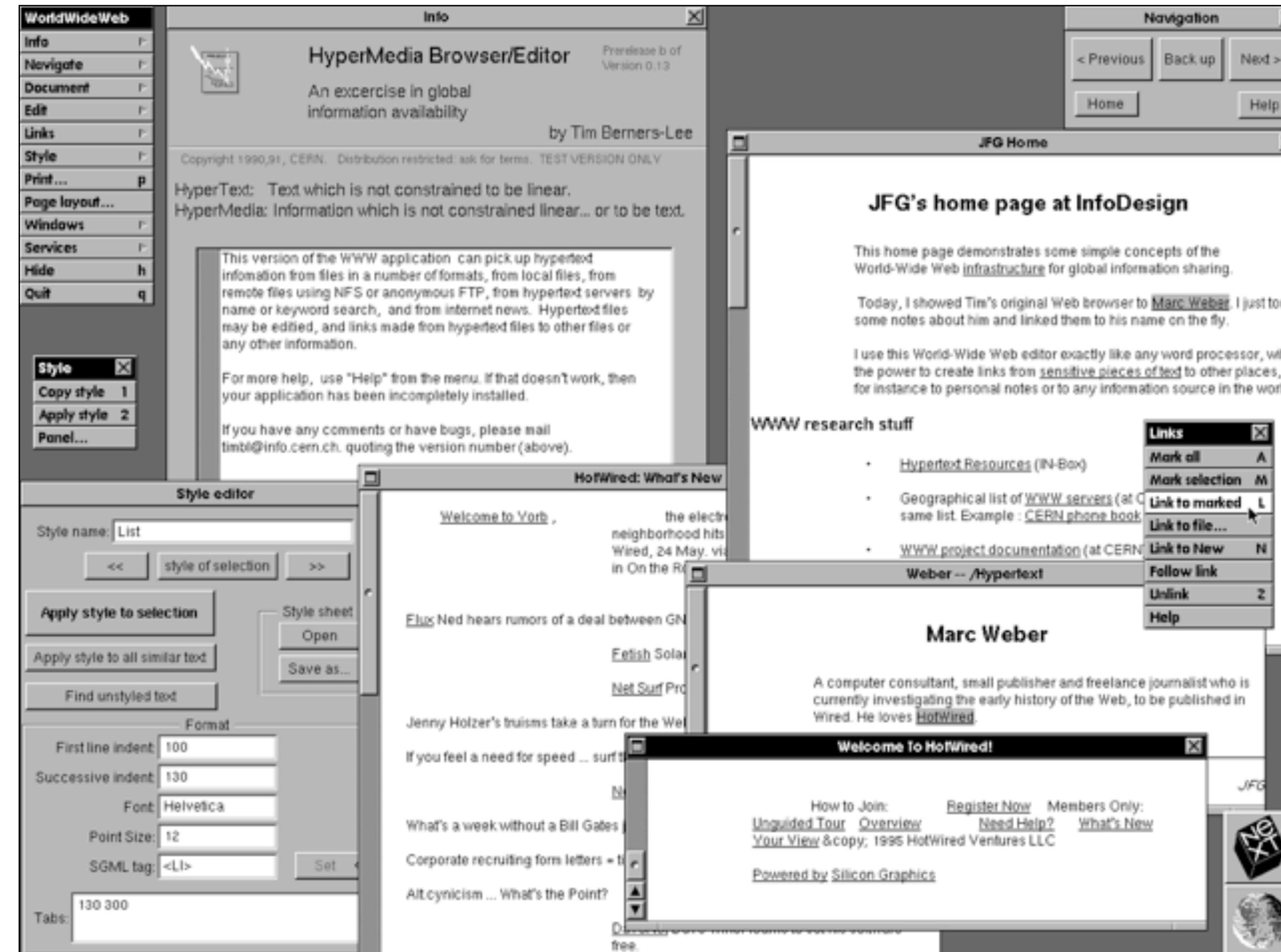
CSS files requested.

Web Clients

- World Wide Web client applications, commonly known as web browsers, are software applications capable of retrieving, presenting and transversing information resources available on the World Wide Web.

- Web browsers communicate with web servers using the HTTP protocol.

- Web browsers are increasingly sophisticated.

- The first web browser was called WorldWideWeb and was bundled with the releases of the WorldWideWeb system.

- Mosaic, developed at NCSA, was the first popular browser. It was the first to integrate text and images in a single page.

WorldWideWeb (later Nexus) Browser

WorldWideWeb (1990), developed by Tim Berners-Lee, was the first web browser.

Mosaic
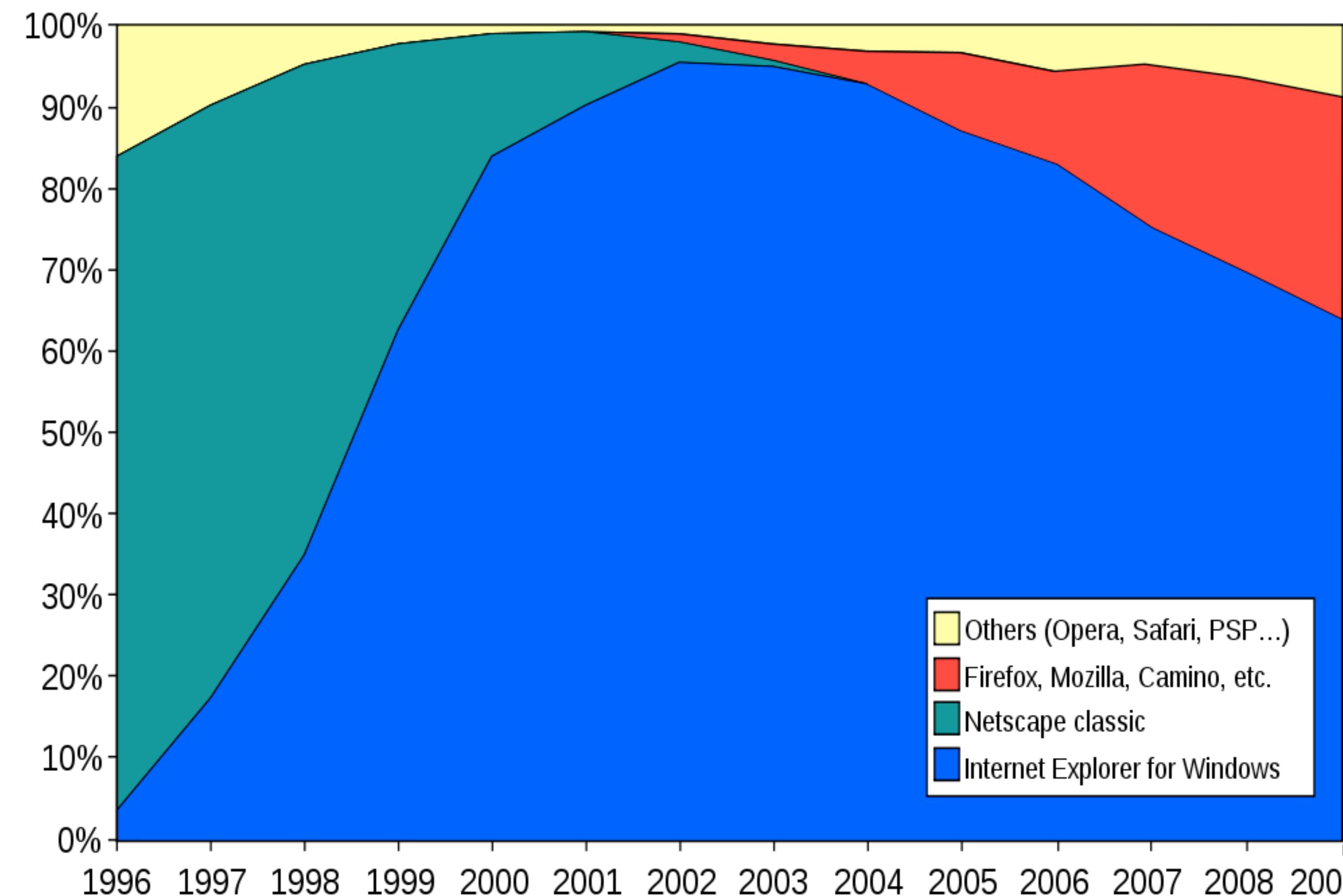
Mosaic was the first mainstream web browser (1993).
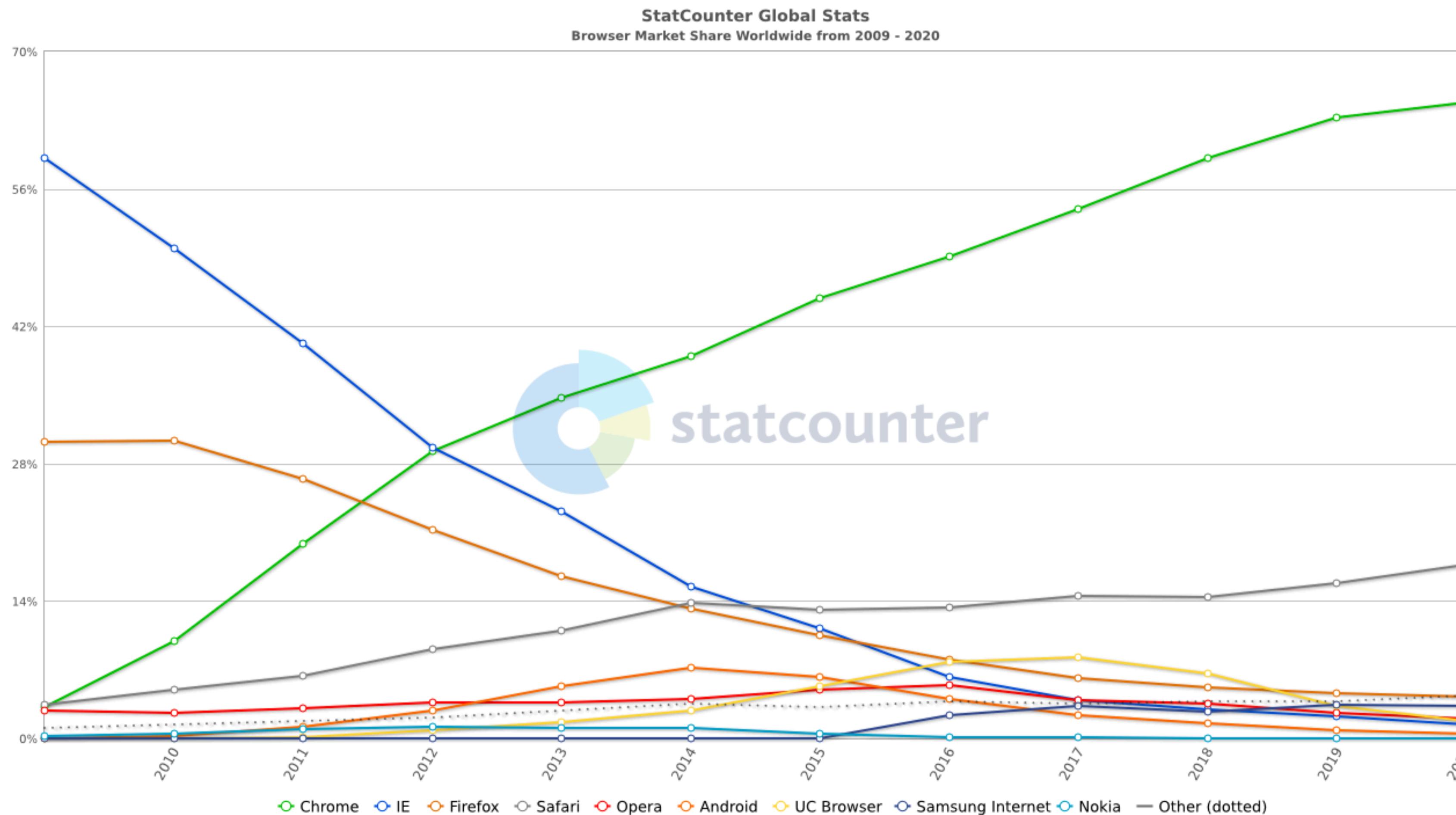
Mosaic

- Marc Andreessen and Jim Clark left NCSA to start Netscape Communications in 1994. Later that year, Netscape version 1 was released.

- NCSA licensed Mosaic technology to Microsoft to form the basis of Internet Explorer. Version 1 was released in 1995.

- The "Browser Wars" started for the dominance of the web browser market.

- Internet Explorer had a major advantage — it was bundled with every copy of Windows. This latter led to the USA vs. Microsoft case on monopoly abuse.
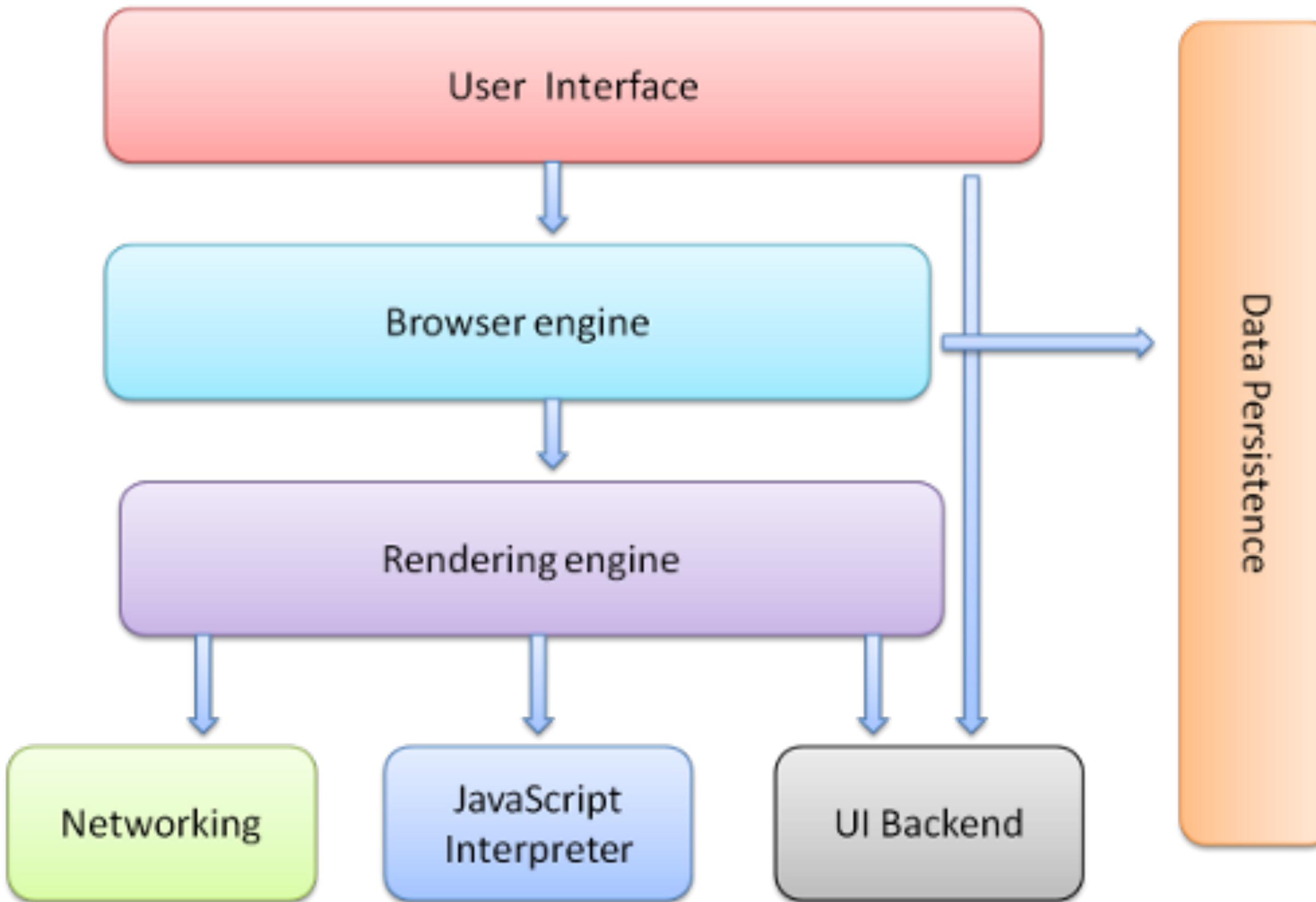
Source: Wikipedia - "Browser Wars"

Source: Wikipedia - "Usage share of web browsers"

Browser's High Level Structure

- User Interface - browser controls
- Browser engine - mapping
- Rendering engine - HTML & CSS
- Networking - network calls
- JS Interpreter - execute javascript
- UI Backend - drawing widgets
- Data Persistence - saves data

How Browsers Work: Behind the scenes of modern web browsers (2011)
http://www.html5rocks.com/en/tutorials/internals/howbrowserswork/

Notable Layout Engines

- **Trident** — Developed by Microsoft for use in Internet Explorer.

- **Gecko** — Developed by the Mozilla Foundation, used in Firefox, Camino.

- **WebKit** — A fork of KHTML developed by Apple, used in Safari.

- **Blink** — A fork of WebKit developed by Google, used in Chrome, Opera and Edge.

- **Presto** — Developed by Opera Software, used in Opera (until 2013).

- **EdgeHTML** — New Microsoft rendering engine launched in 2015. RIP December, 2018.

"Comparison of Layout Engines", Wikipedia
https://en.wikipedia.org/wiki/Comparison_of_layout_engines
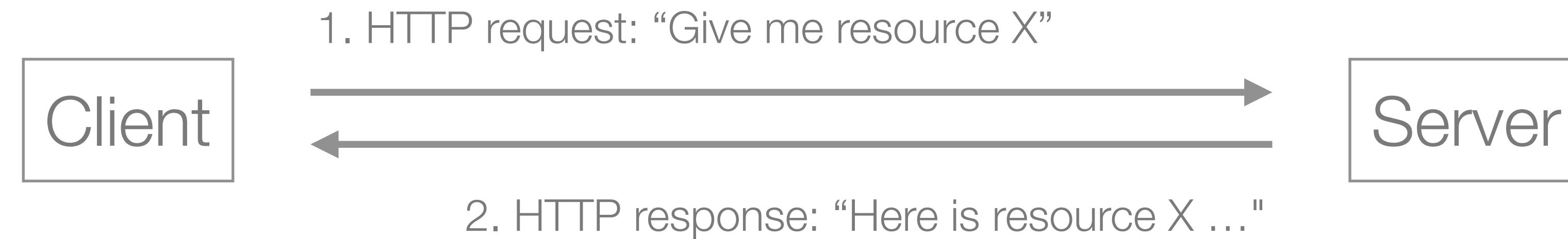
"Timeline of web browsers", Wikipedia
https://en.wikipedia.org/wiki/Timeline_of_web_browsers
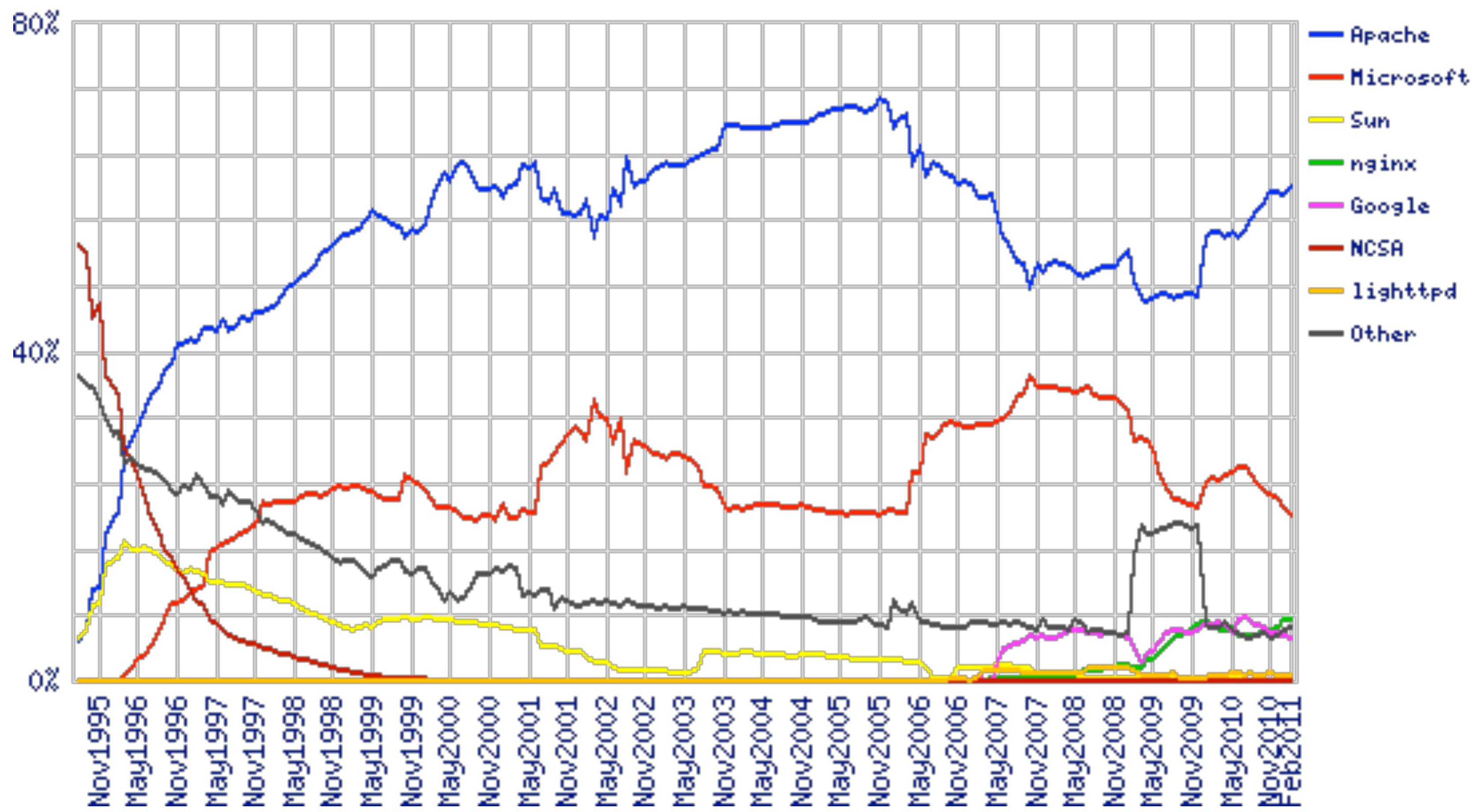
Web Servers

- A web server is a program whose primary function is to deliver resources on clients' requests. Only acts when requests arrive.

- Web servers handle multiple web clients simultaneously. Servers and clients communicate using the HTTP protocol.

1. HTTP request: "Give me resource X"

| Client | → | Server |

2. HTTP response: "Here is resource X …"
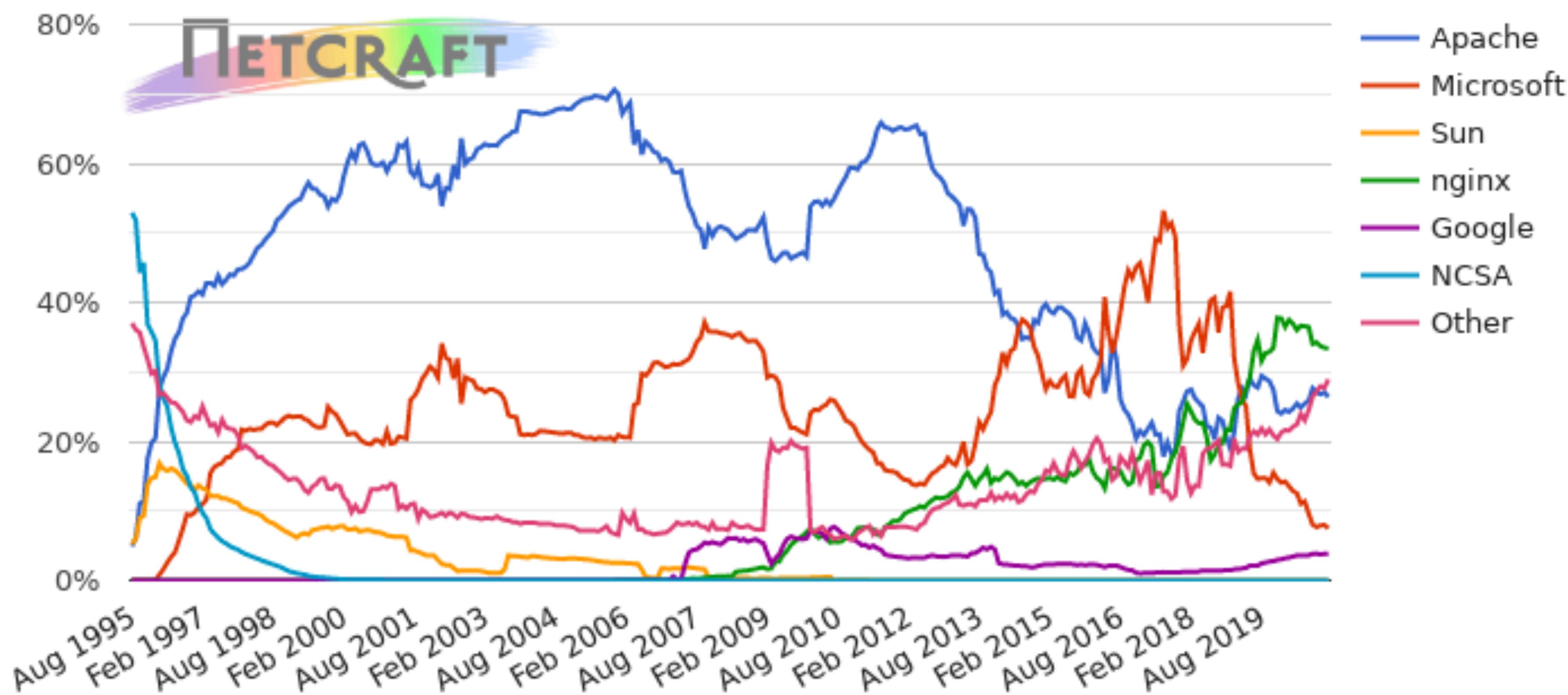
- The most common web servers are the Apache HTTP Server and Microsoft's Internet Information Server (IIS). Lightweight alternatives: nginx, lighttpd.

- Typically, different web servers coexist in a production environment.

Source: Netcraft

Web server developers: Market share of all sites

Source: Netcraft

72

WWW's Core Technologies

- The Web is supported by three core technologies:

  - **URL: Uniform Resource Locator**

    Used to identify the resources available on the web.

  - **HTTP: HyperText Transfer Protocol**

    Used to specify how clients communicate with servers.

  - **HTML: HyperText Markup Language**

    Used to represent and interlink documents on the web.

URL: Uniform Resource Locator

- A URL establishes a unique address for a World Wide Web resource, e.g. pages, images, etc. URLs are used to locate web resources.

- Syntax (simplified): protocol://machine:port/directory/file.type

  - Protocol, e.g. http://, ftp://, file://

  - Machine, e.g. www.up.pt or 193.137.55.13

  - Port, e.g. 80 (the default), 1000, 20

  - Resource path, the directory path to the file

- Example: http://www.up.pt:80/sobre/index.html

HTTP: HyperText Transfer Protocol

- The HTTP protocol defines how web client communicate with web servers to access web resources.

- HTTP was developed as a joint work of IETF and W3C.

- It is a request-response protocol, i.e. client issues a request and waits for the server to respond. Timeouts can occur if servers take too long.

- HTTP is a stateless protocol, i.e. each request is treated as an independent transaction. This results in a simpler design, but requires additional information to be send in each request.

HTTP: Request Methods

- HTTP supports several request commands, called HTTP methods. A total of nine methods are defined in the HTTP standard. GET and POST are the most commonly used by web browsers.

  - GET — Requests the resource from the server. Idempotent operation.

  - HEAD — Requests only the headers (without the content).

  - POST — Submits data to be processed to the identified resource.

  - PUT — Uploads data into the specified resource.

  - DELETE — Deletes the specified resource.

HTTP: Status Codes

- All HTTP responses include a numeric status code, indicating if the request succeeded or if other actions are required. Codes are organized in five classes of responses.

  - 1xx — Informational

  - 2xx — Success (e.g. 200 OK, 201 Created)

  - 3xx — Redirection (e.g. 301 Moved Permanently)

  - 4xx — Client Error (e.g. 404 Not Found, 403 Forbidden)

  - 5xx — Server Error (e.g. 500 Internal Server Error)

HTTP is Stateless

- Web servers do not keep any information about clients.
  Each request is isolated from all others.

- **State must be maintained by web applications.**

- How can we implement a stateful user experience over a stateless protocol (e.g. shopping cart, authenticated access)? Two options:

  - Cookies — client-side pieces of data generated by the server and attached to each HTTP request.

  - Sessions — server-side files with unique identifiers (session IDs), these can be passed in URLs or Cookies.

HyperText Markup Language — HTML
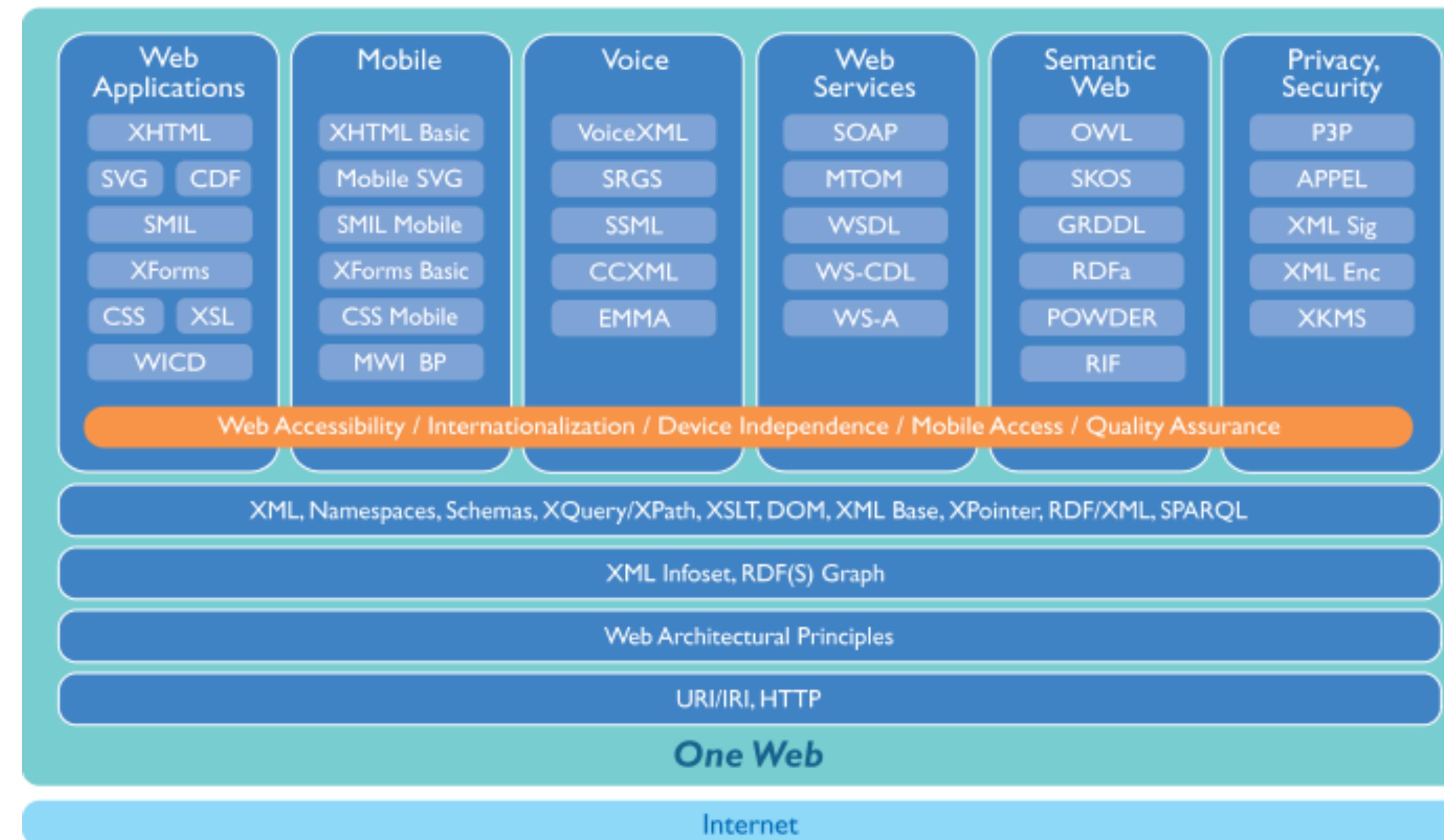
- The HyperText Markup Language (HTML) is used to define the content and structure of hypertext documents.

- First standard was published in 1995 — HTML 2.0.

- HTML 4.01 was published in 1999.

- XHTML was a reformulation of HTML as XML. W3C tried to "force" authors to write well-formed code. Later abandoned due to low adoption by web developers.

- HTML5 is the latest major revision to HTML.

W3C

- The World Wide Web Consortium was founded in 1994 by Tim Berners-Lee to standardize the protocols and technologies used to build the web.

- The W3C is an international standards organization, composed by member organizations and full time staff, that develops technical specifications and guidelines for the web.

- Mission: "Led the Web to its full potential".

- W3C Process: (1) members propose new technologies or ideas; (2) working groups are formed; (3) recommendations are developed and approved by consensus.

- The W3C does not enforce their recommendations.

W3C Standards

W3C standards are many and in different areas, from technical specifications to guidelines.

W3C technology stack (circa 2004).
http://www.w3.org/Consortium/techstack-desc.html
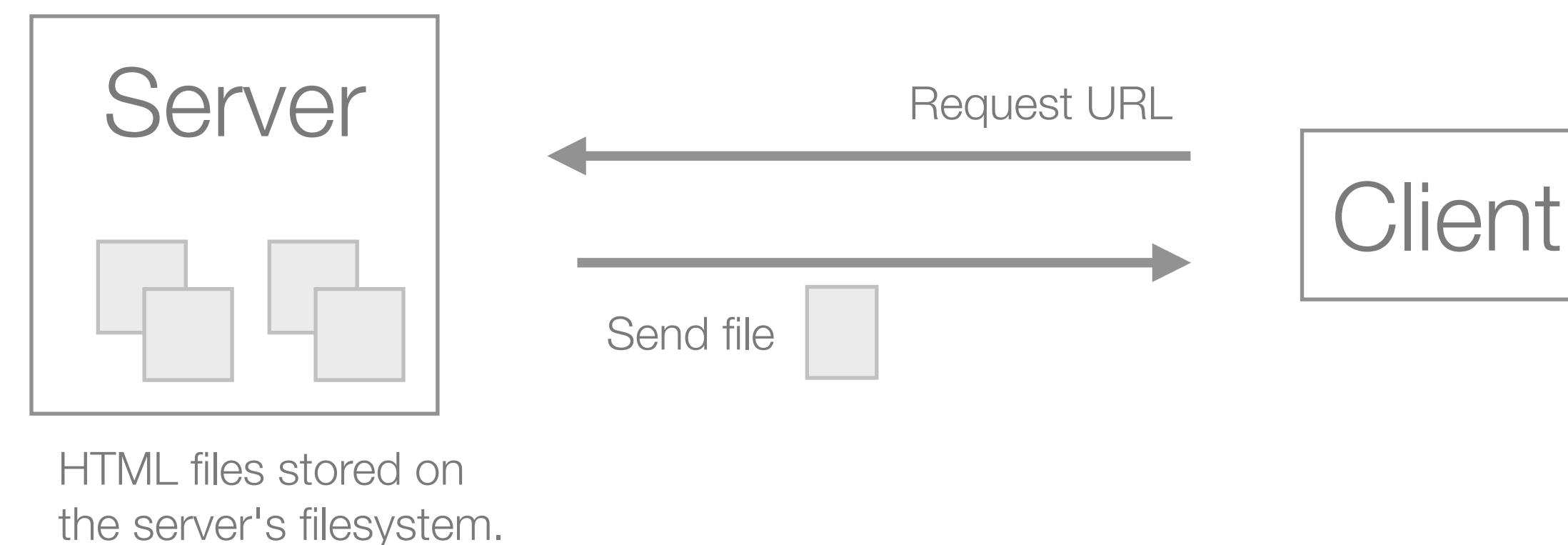
W3C Process

- People generate interest in a particular topic (e.g., web services). For instance, Members express interest in the form of Member Submissions, and the Team monitors work inside and outside of W3C for signs of interest.

- When there is enough interest in a topic, the Director announces the development of a proposal for a new Activity or Working Group charter, depending on the breadth of the topic of interest.

- There are three types of Working Group participants: Member representatives, Invited Experts, and Team representatives. Team representatives both contribute to the technical work and help ensure the group's proper integration with the rest of W3C.

- Working Groups generally create specifications and guidelines that undergo cycles of revision and review as they advance to W3C Recommendation status.

Web Applications
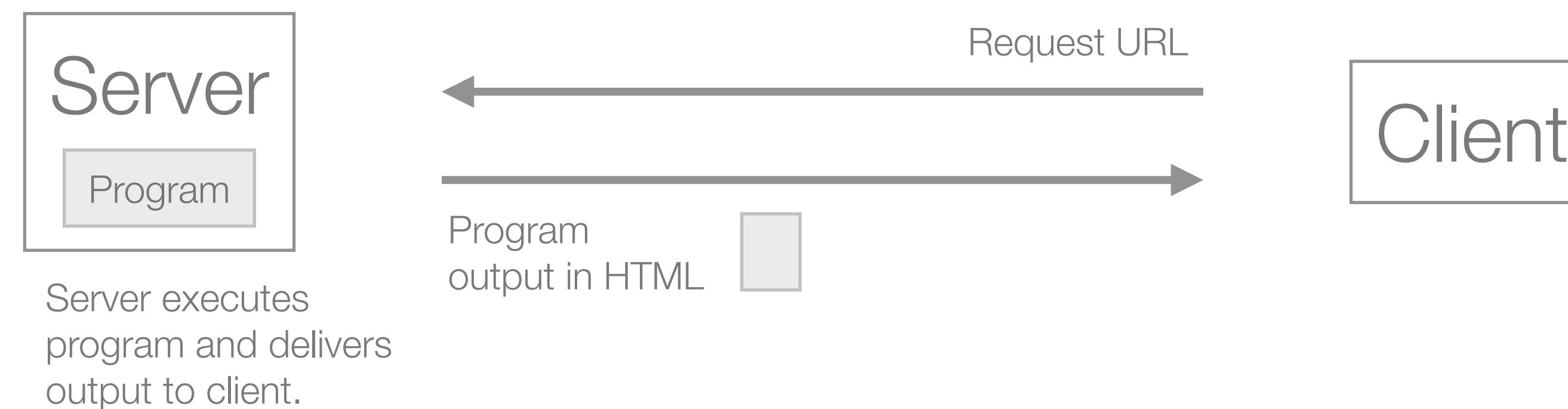
Static Web Pages (early 1990s)

- In the early days of the web, most pages were static files served directly from the filesystem. **Pages are constructed at design time**.

- Developing web pages involved few technologies, mostly just HTML. Learning by example, using the "view source" option, was an important method for knowledge dissemination.

Server

Request URL

Client

Send file

HTML files stored on
the server's filesystem.

Dynamic Web Pages (mid 1990s)

- Dynamic web pages emerged in the mid 1990s. Instead of serving static files from the filesystem, software applications produce web pages when requested.
  Pages are constructed at run time, when "called" by the browser.

- The Common Gateway Interface (CGI) is a specification that defines how web requests and responses interact with an application program.

- There are several alternatives: Apache modules, IIS plug-ins, FastCGI, WSGI.

Request URL

Server

Program

Server executes
program and delivers
output to client.

Program
output in HTML

Client
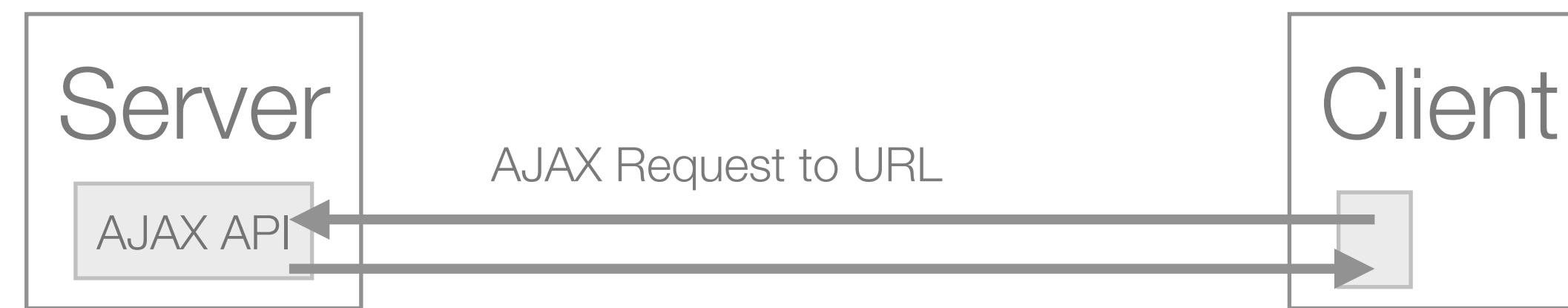
Dynamic Web Sites (1990s …)

- Sites emerged as a collection of multiple and coherent web pages. Instead of treating each page independently, multiple web pages were handled by shared functions and libraries.

- Provide a stateful experience to the user (e.g. shopping cart).

- Typically a common data layer was implemented across pages.

- Libraries and frameworks to address repetitive and common tasks.

- Richer user interfaces (e.g. JavaScript).

Web Applications (mid 2000s ...)

- Strong developments in client-side technologies and methodologies led to richer and interactive user interfaces.

- AJAX enabled the creation of asynchronous web applications. No need to reload full pages on each user interaction.

- Web documents became applications itself, i.e. code runs on the document.

- Rise of new web frameworks — Ruby on Rails, Django, etc.

- Wide adoption of HTML5.

- A new interaction paradigm with mobile devices.

AJAX

Server

AJAX API

AJAX Request to URL

Response in HTML,
JSON, etc.

Client

Requests are made to the server, **from the document using JavaScript**, without loading a new document.

The HMTL document is dynamically altered depending on the answer and using JavaScript.

Code Execution

	Server		Client
Static web pages		*No code execution*	
Dynamic web pages	Code Execution		
Web page applications	Code Execution	+	Code Execution

The Three-Tier Architecture

- Web applications are typically structured in three tiers, corresponding to three core aspects of a web system: presentation, business logic and data access.

- Main advantages of this approach: separation of concerns, maintainability.
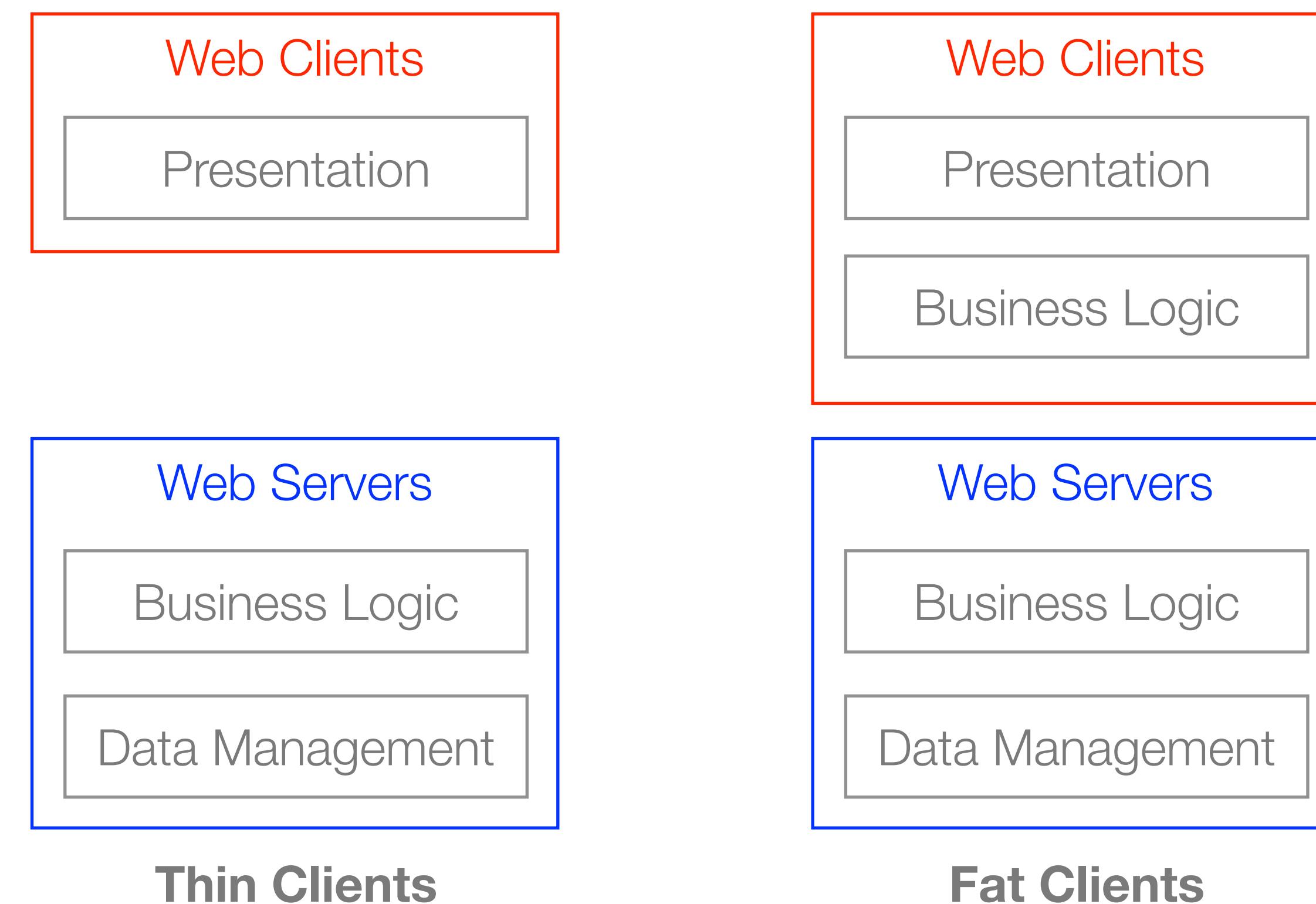
Presentation

Business Logic
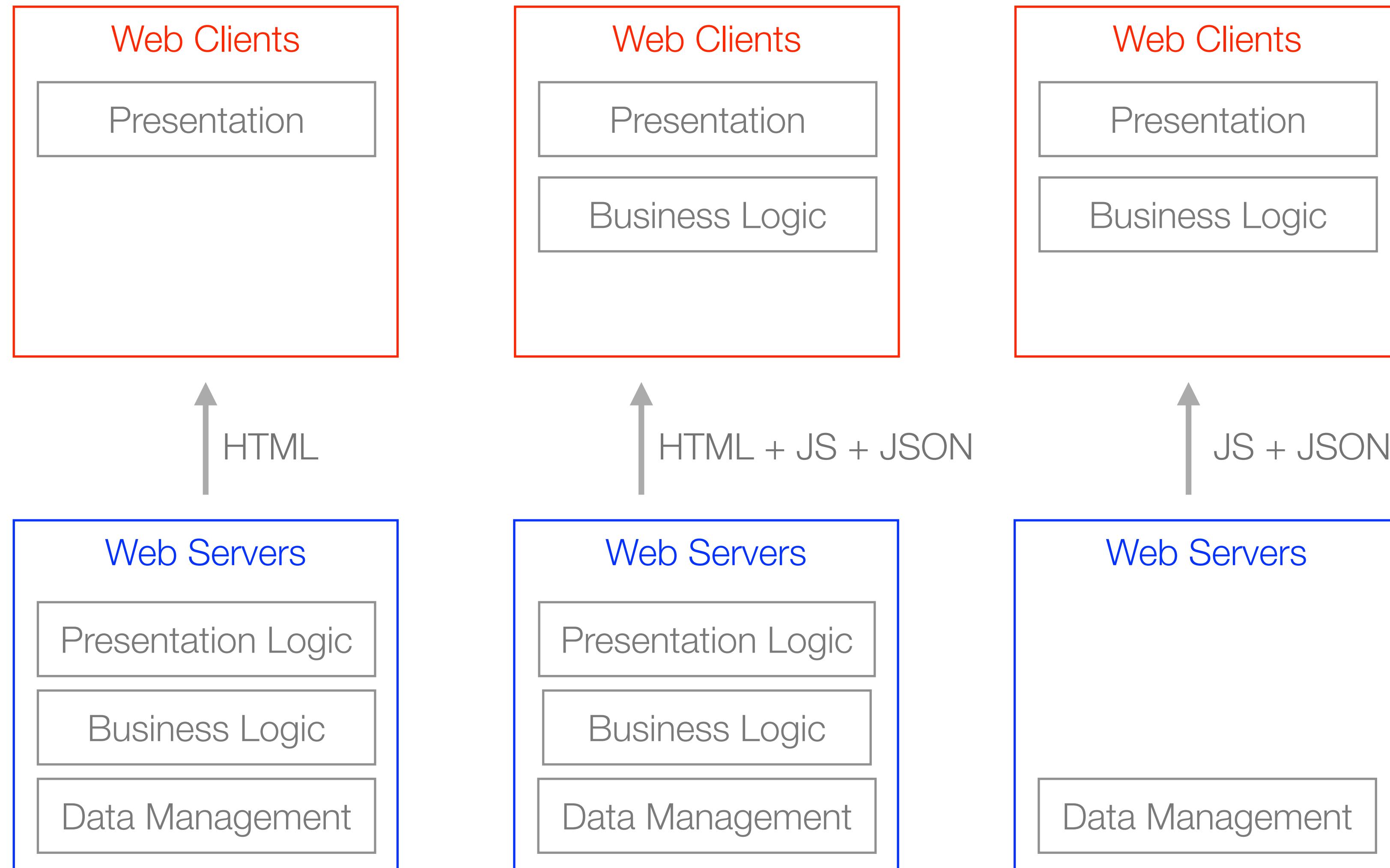
Data Management

Architecture of Web Applications

- Over time, the architecture of web applications has changed significantly. In the beginning, web clients were only used to present the user interface. Recently, application logic has moved partially to the clients.

Thin Clients

Fat Clients
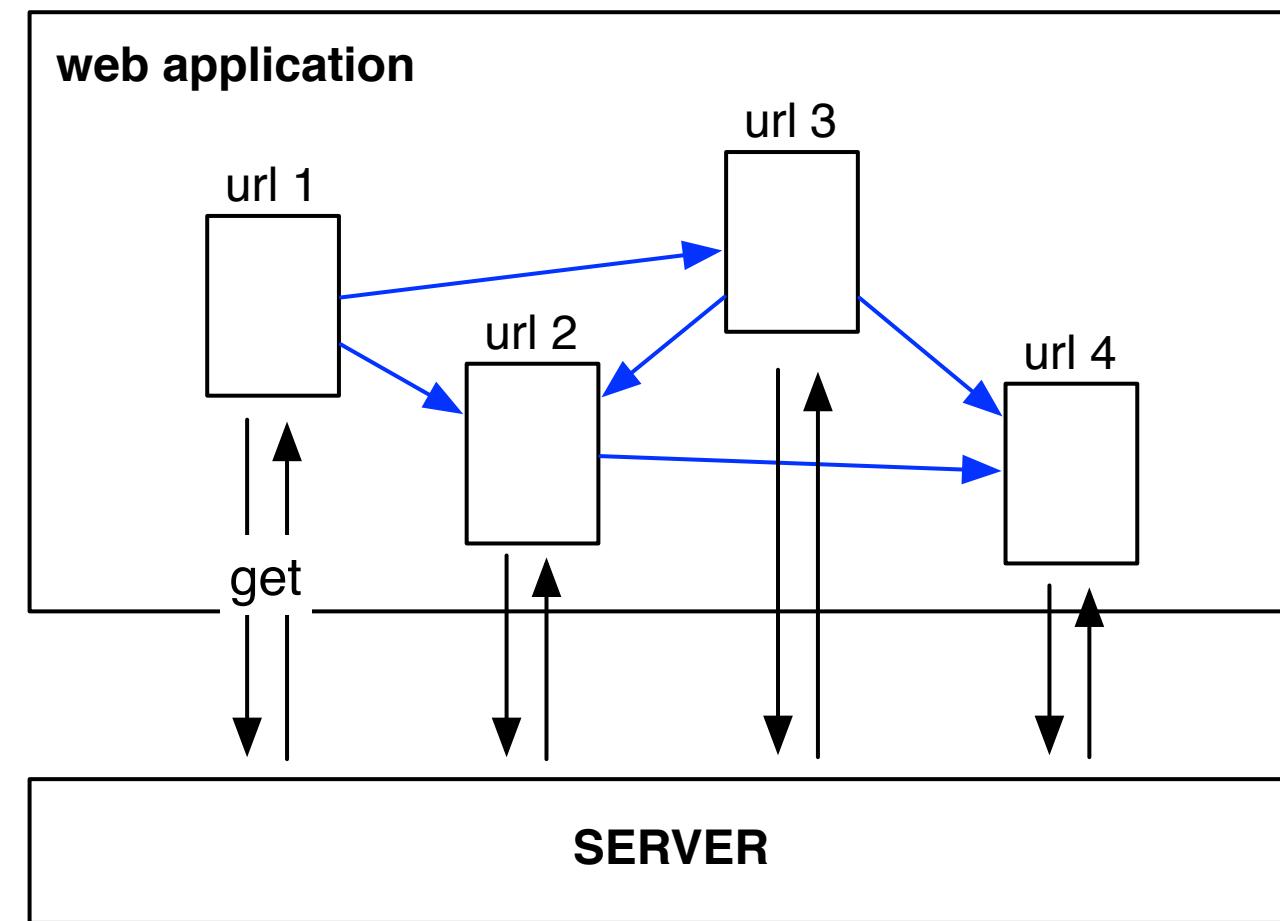
Different Architectural Options

Web Clients	Web Clients	Web Clients
Presentation	Presentation	Presentation
	Business Logic	Business Logic

↑ HTML ↑ HTML + JS + JSON ↑ JS + JSON

Web Servers	Web Servers	Web Servers
Presentation Logic	Presentation Logic	
Business Logic	Business Logic	
Data Management	Data Management	Data Management

Multi-Page Web Applications
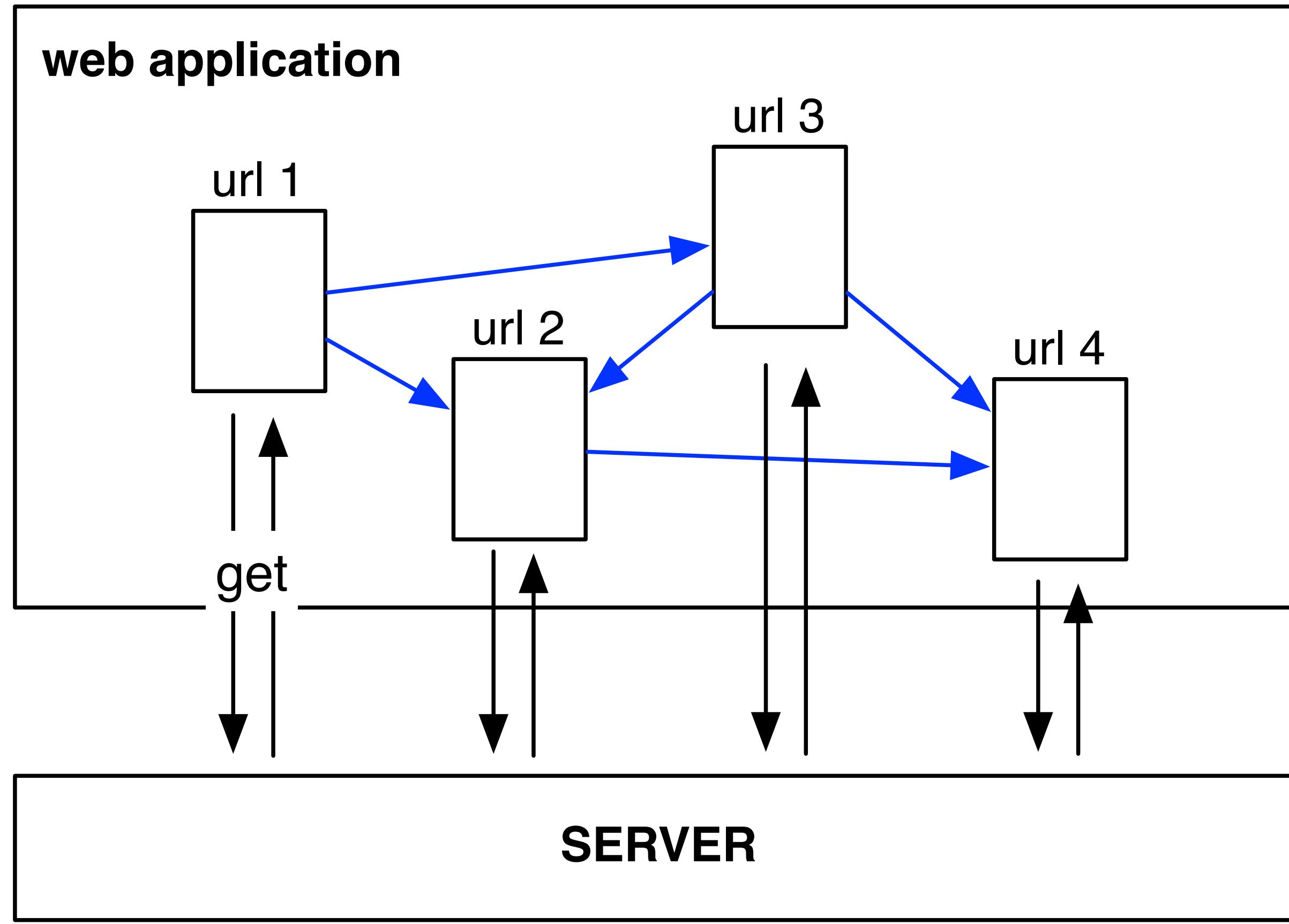
- The web application is implemented as a collection of multiple web pages. The user interacts with the application navigating through these pages. Each page is prepared on the server and only presentation details are sent to the browser.

Each page corresponds to a HTTP GET request. Application logic is all maintained on the server.

E.g. Amazon, SIGARRA
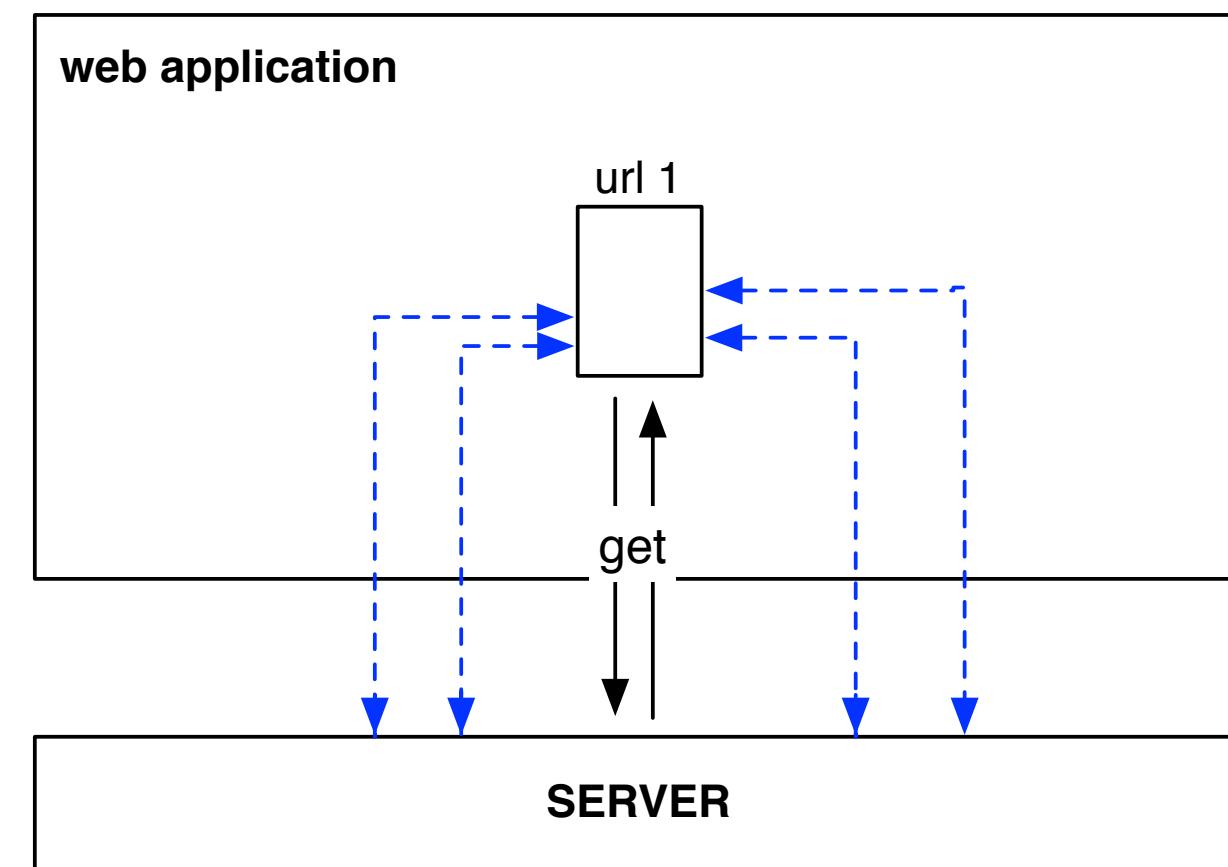
- Advantages: REST style, client independent, consistency across browsers, broad technological ecosystem, application logic is kept on the server.

- Disadvantages: slow performance and responsiveness, fragmented code, no way to deliver updates to an open web page.

Each page corresponds to a HTTP GET request. Application logic is maintained on the server.

Single-Page Web Applications

- The web application is implemented as a single web page. All necessary resources are loaded or dynamically added to the page (e.g. using Ajax). Application logic is pushed to the client — fat client architecture.

web application

url 1

get
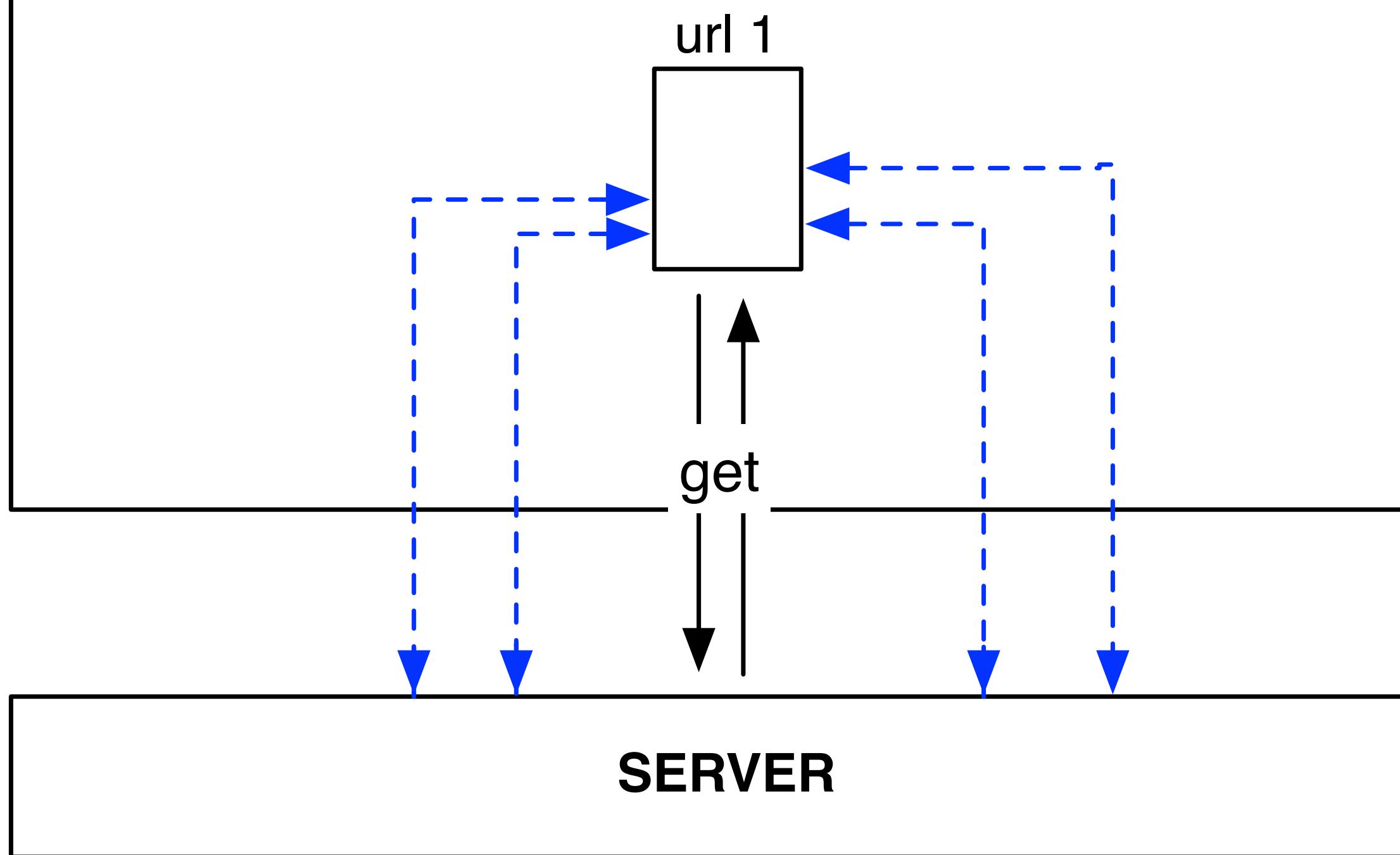
SERVER

An initial page is loaded using HTTP GET. All other resources are loaded dynamically following user interactions.

E.g. Slack

- <u>Advantages</u>: improved user experience, reduced bandwidth consumption, decoupled client and server, reusable server interfaces, reusable client code.

- <u>Disadvantages</u>: JavaScript required, breaks browser history, increased browser dependency (versions, features, performance), no REST, difficult to crawl and index.

web application

url 1

get

SERVER
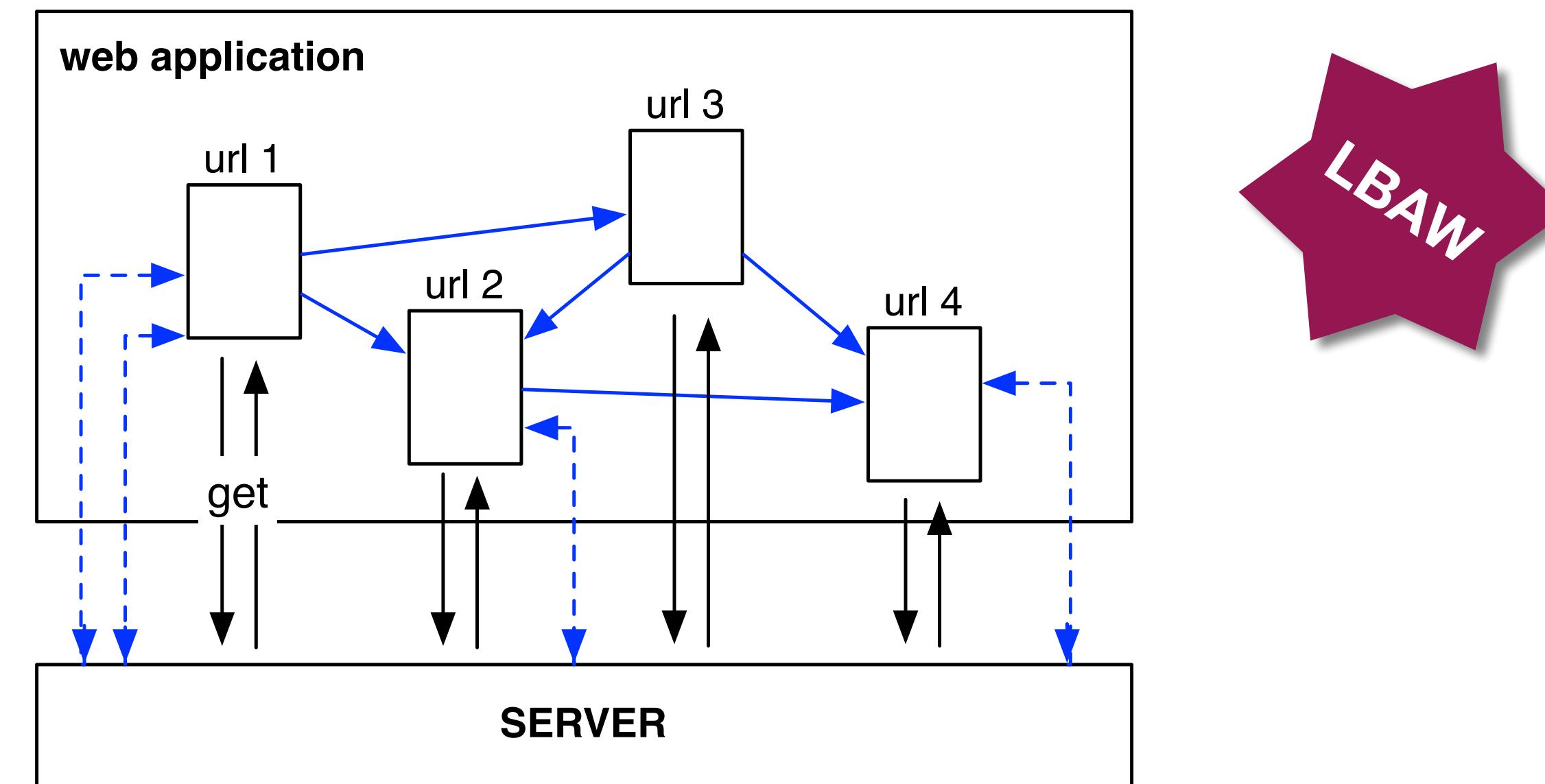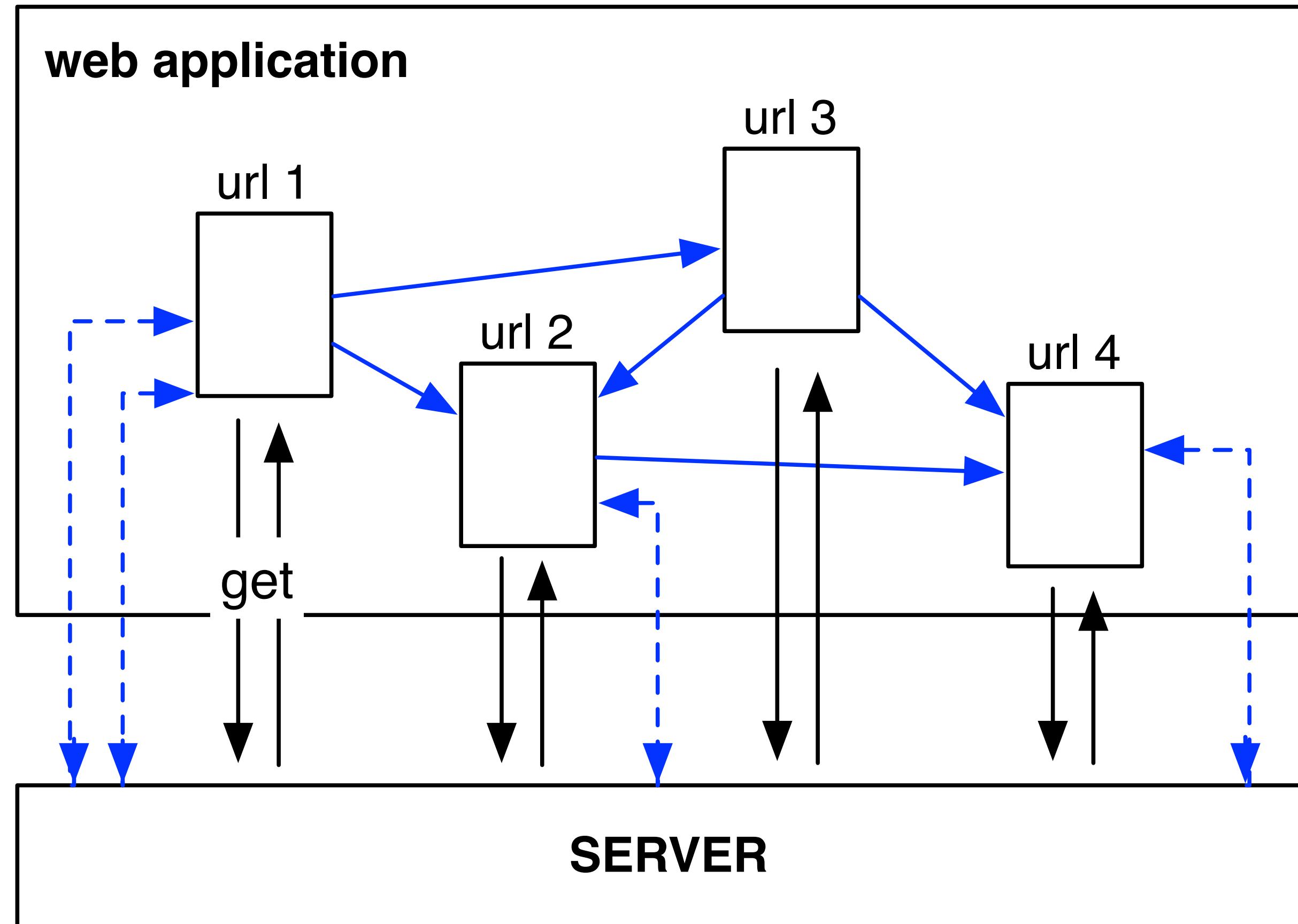
An initial page is loaded using HTTP GET, including
the initial HTML document plus JavaScript code.

All other resources are loaded dynamically following
user interactions.

Mixed Approach

- The two architectural styles can be mixed to combine the benefits of both.

- Use different web pages to setup the core structure of the application.

- Define data APIs to provide updates to the web pages and support different devices.

- Use asynchronous requests to improve performance and user experience.

Main application resources have a URL.
User interaction is improved by using
asynchronous calls to data APIs.

Web Architecture Summary

- Criteria to consider in choosing an architecture:

  - **Usability**

    User friendly? Instant updates possible? Browser history?

  - **Search / Share**

    Search engine friendly?

  - **Linking**

    Mapping between URLs and views? 1-to-1?

  - **Performance**

    Consistent across platforms? Optimized content loading? Client effort?

  - **Productivity**

    Developers background? Modularity?

  - **Testing**

    Modularity? What to test? Client dependency?

- **Choosing an Architecture == Choosing the Challenges**
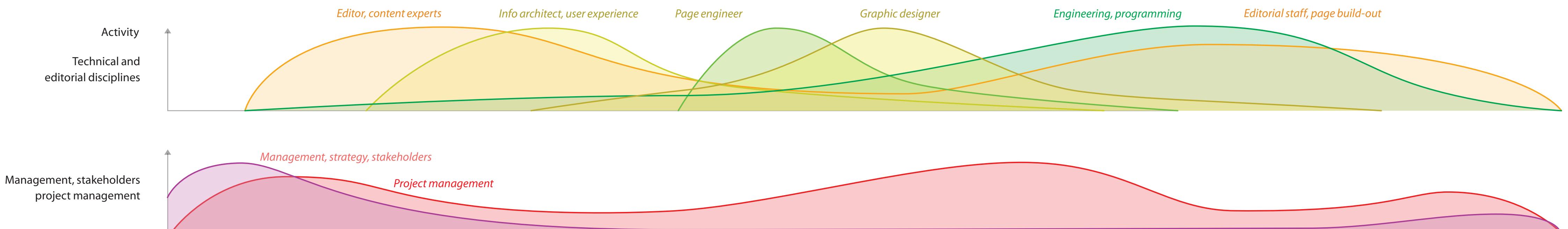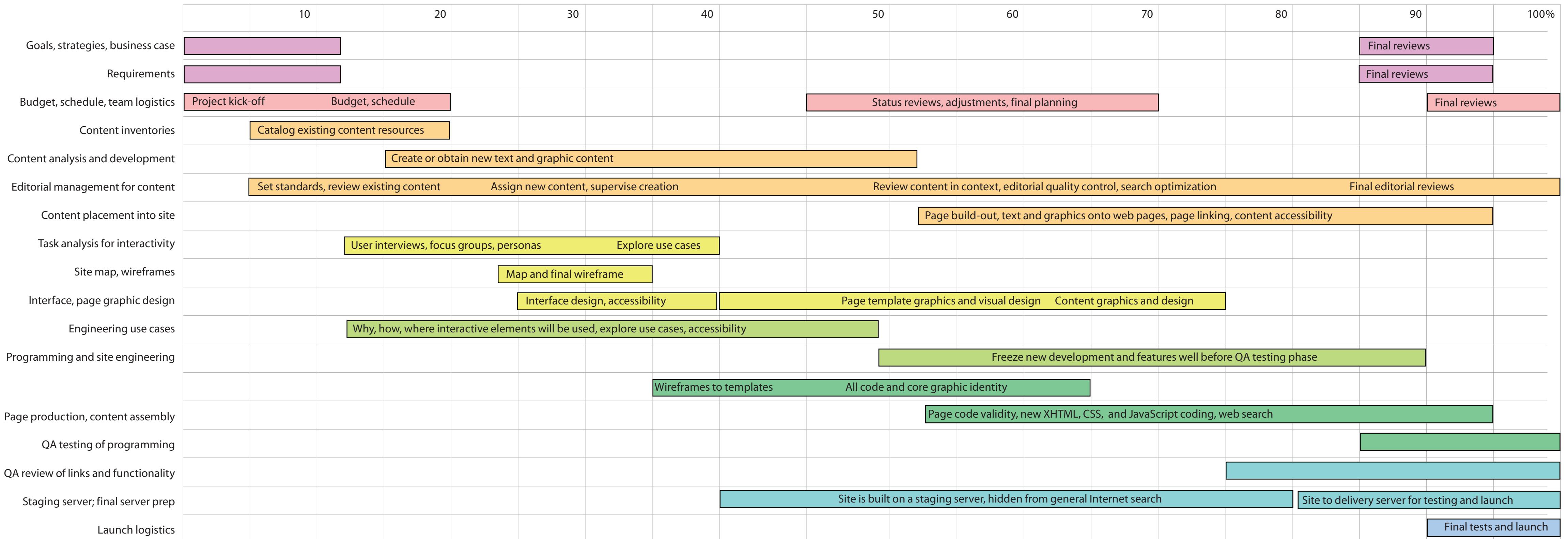
Rendering on the Web
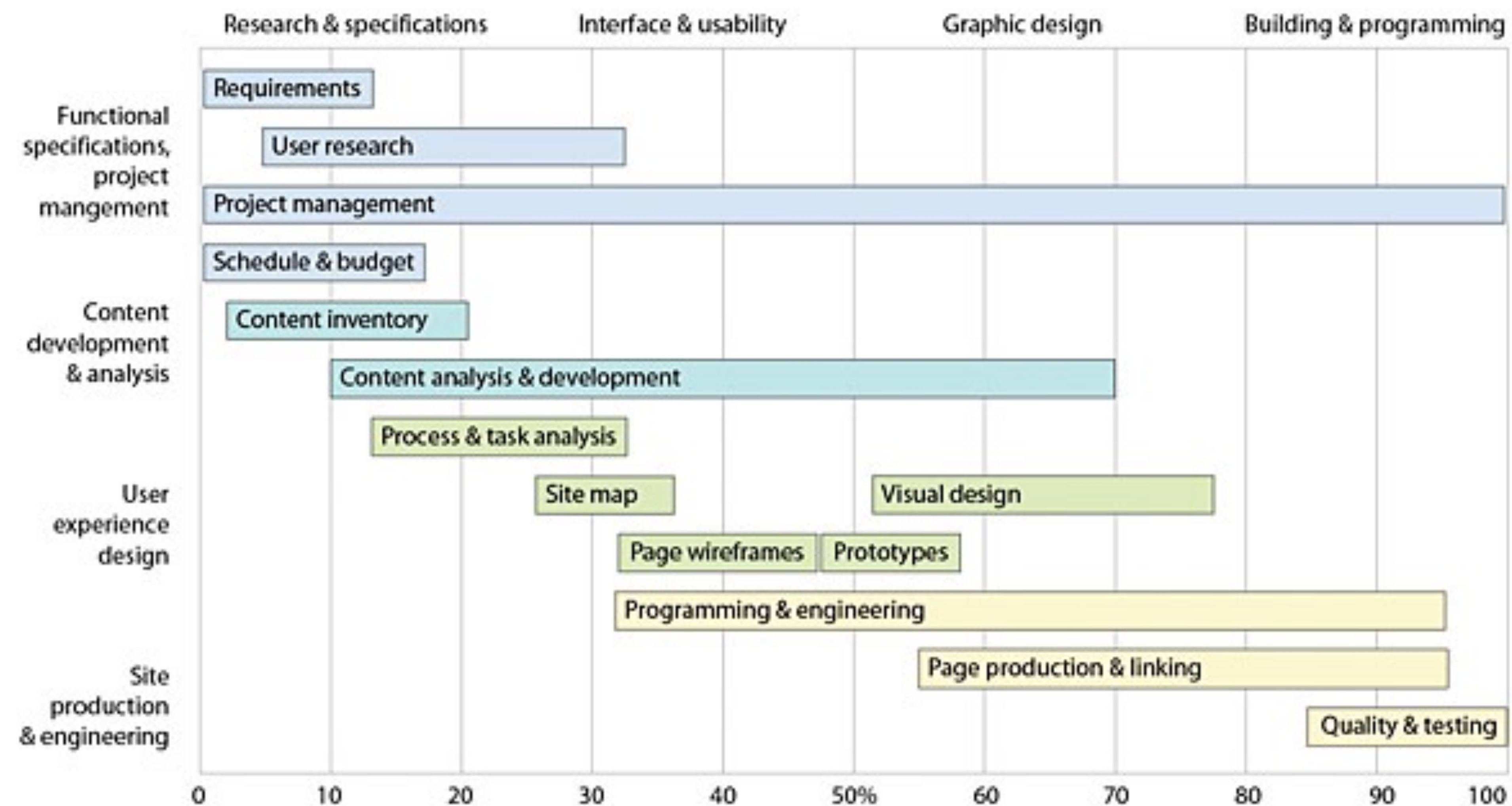
- **Exclusive server-side rendering**

  - Static – HTML corresponds to pre-built files on the server

  - Dynamic – HTML results from applications executed (on request) on the server

- **Server and client-side**

  - Server-side rendering with hydration
    HTML is dynamically built on the server with updates on the client (logic mostly on the server)

  - Client-side rendering with server pre-rendering
    HTML initial version is prepared on the server but the client then takes over (logic mostly on the client)

- **Exclusive client-side rendering**

  - Full client-side rendering
    Only a minimal skeleton is served from the server, all the application logic and rendering is done on the client

Discussion on Architectural Options

- Personal web site

- Personal web site with comments

- Amazon

- SIGARRA

- Slack

Web Site Development Process

| | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100% |

Goals, strategies, business case — Final reviews

Requirements — Final reviews

Budget, schedule, team logistics — Project kick-off · Budget, schedule · Status reviews, adjustments, final planning · Final reviews

Content inventories — Catalog existing content resources

Content analysis and development — Create or obtain new text and graphic content

Editorial management for content — Set standards, review existing content · Assign new content, supervise creation · Review content in context, editorial quality control, search optimization · Final editorial reviews

Content placement into site — Page build-out, text and graphics onto web pages, page linking, content accessibility

Task analysis for interactivity — User interviews, focus groups, personas · Explore use cases

Site map, wireframes — Map and final wireframe

Interface, page graphic design — Interface design, accessibility · Page template graphics and visual design · Content graphics and design

Engineering use cases — Why, how, where interactive elements will be used, explore use cases, accessibility

Programming and site engineering — Freeze new development and features well before QA testing phase · Wireframes to templates · All code and core graphic identity

Page production, content assembly — Page code validity, new XHTML, CSS, and JavaScript coding, web search

QA testing of programming

QA review of links and functionality

Staging server; final server prep — Site is built on a staging server, hidden from general Internet search · Site to delivery server for testing and launch

Launch logistics — Final tests and launch

Activity / Technical and editorial disciplines:
Editor, content experts · Info architect, user experience · Page engineer · Graphic designer · Engineering, programming · Editorial staff, page build-out

Management, stakeholders project management:
Management, strategy, stakeholders · Project management

Figure copyright 2009 Patrick J. Lynch and Sarah Horton. All rights reserved.   From the book *Web Style Guide: Basic Design Principles for Creating Web Sites*, 3rd ed.   Yale University Press.   www.webstyleguide.com

103

Project phases (shown across top): Research & specifications | Interface & usability | Graphic design | Building & programming

Functional specifications, project mangement
- Requirements
- User research
- Project management
- Schedule & budget

Content development & analysis
- Content inventory
- Content analysis & development
- Process & task analysis

User experience design
- Site map
- Visual design
- Page wireframes
- Prototypes
- Programming & engineering

Site production & engineering
- Page production & linking
- Quality & testing

Timeline axis: 0, 10, 20, 30, 40, 50%, 60, 70, 80, 90, 100

Project production Gantt chart

Content strategy and inventory

Personas and audience research

Information architecture

User experience design

Editorial and content development

Analysis and research on similar sites

Wireframes and page engineering

Graphic design and templates

Site assembly and application testing

Site testing final editing

Project scope and schedule

Web development

Business goals and budget resources

Web technology planning

Process analysis and use cases

Start with a broad range of input from users, sponsors, and stakeholders

Finished site

Final integration

Site construction

Information design

Strategy, architecture, and project management

Finish with a tight, focused team

Iterative design

Iteration in the build phase is deadly to projects

0%   20%   40%   60%   80%   100%

Start   Finish

106

Web Technologies

The Web Stack of Technologies (1)

- **Networking** – DNS, TCP/IP.

- **Web Protocols** – HTTP, REST.

- **Data Formats** – HTML, JSON, XML.

- **Web Servers** – Apache, IIS, nginx, lighttpd, node.js.

- **Data Storage** – SQL, NOSQL, cache system, file system.

- **Server-side Programming** – Languages, libraries, frameworks.

- **Content Management Systems** – Reusable building blocks (e.g. Wordpress).

- **Library and framework management**

- **Building and packaging** – How to distribute web products: hosting, CDNs.

Based on: http://radar.oreilly.com/2015/03/full-stack-tensions-on-the-web.html
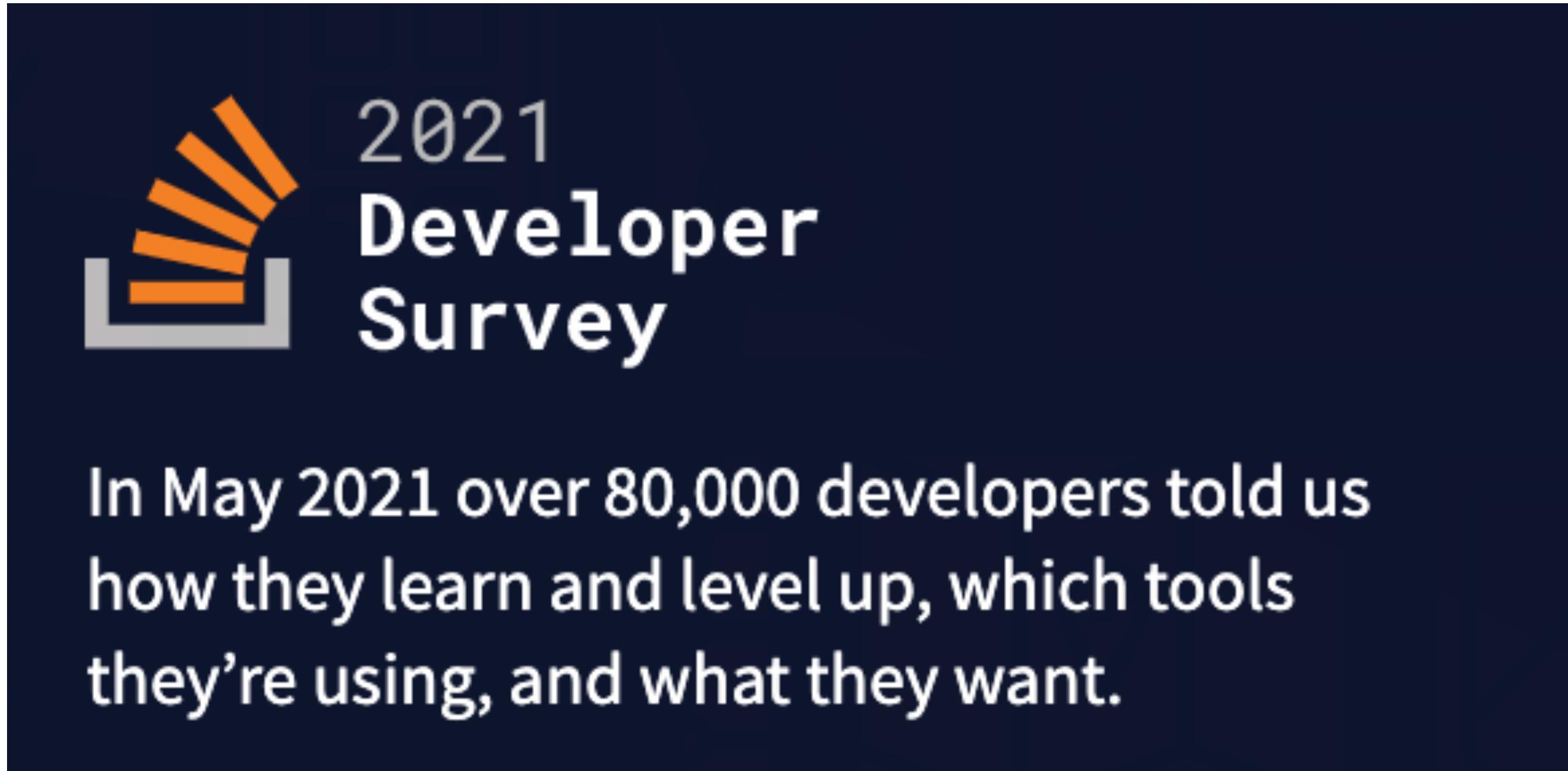
The Web Stack of Technologies (2)

- **Device Capabilities** – Which features are supported.

- **Template Design** – Reusable, modular solutions.

- **Content Presentation** – HTML, CSS.

- **Interface Programming** – Also HTML and CSS but a lot of JavaScript.

- **Performance** – How to improve response speed.

- **Resilience** – How to make reliable systems.

- **Security** – Critical and complex in web systems.

- **Navigation Design** – How to organize content and navigation.

- **Graphics** – Images, SVG, Canvas.

Based on: http://radar.oreilly.com/2015/03/full-stack-tensions-on-the-web.html

Web Technologies

- **The modern web technology stack is huge**.

- There are many different options (at different tiers) for the same task.

- No one is a "web expert".

- But knowing about the full web stack is important for collaboration.

- Web development is organized in two major areas:
 **client-side** and **server-side** development.

Stack Overflow Developer Survey 2021

- https://insights.stackoverflow.com/survey/2021
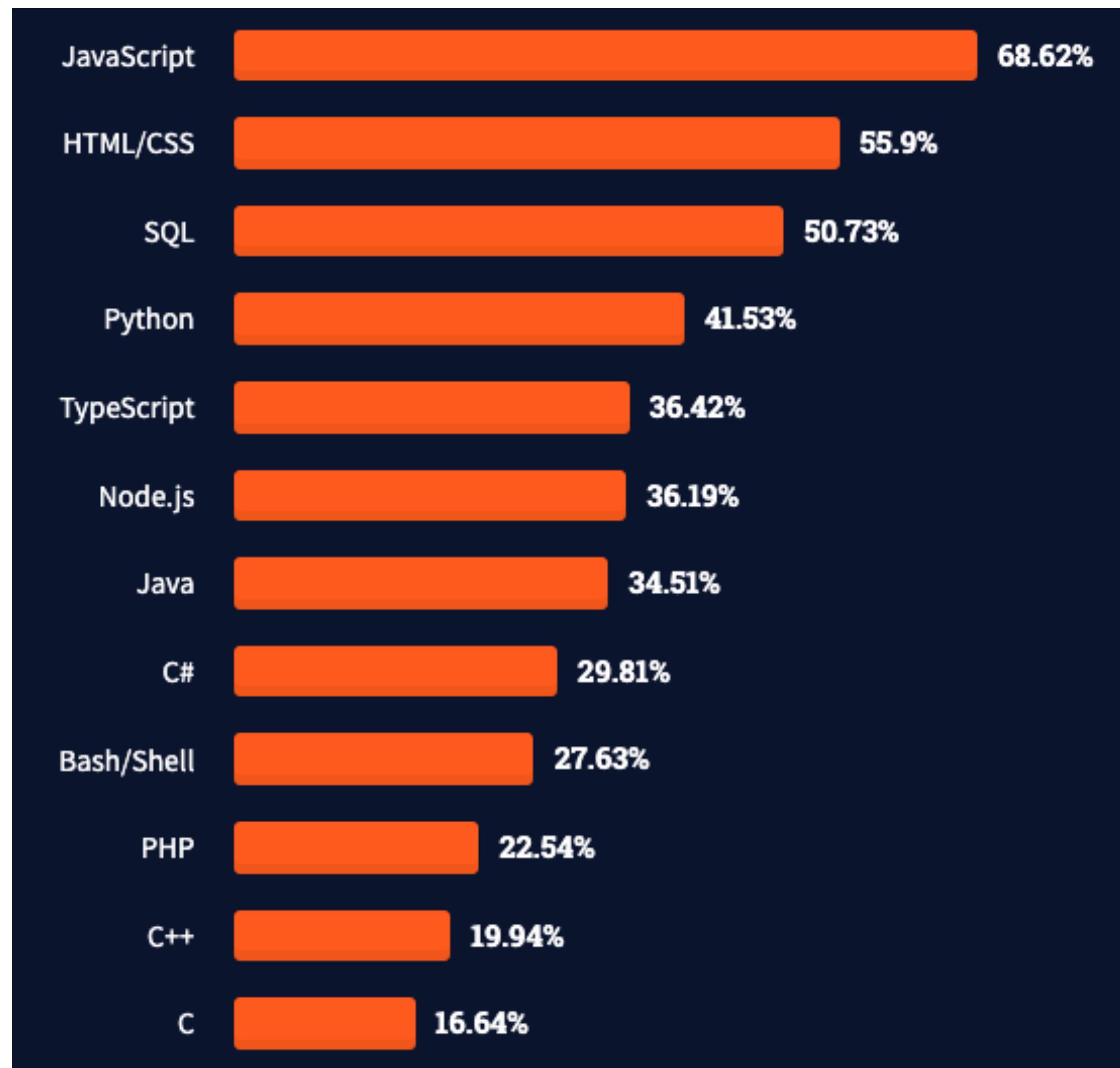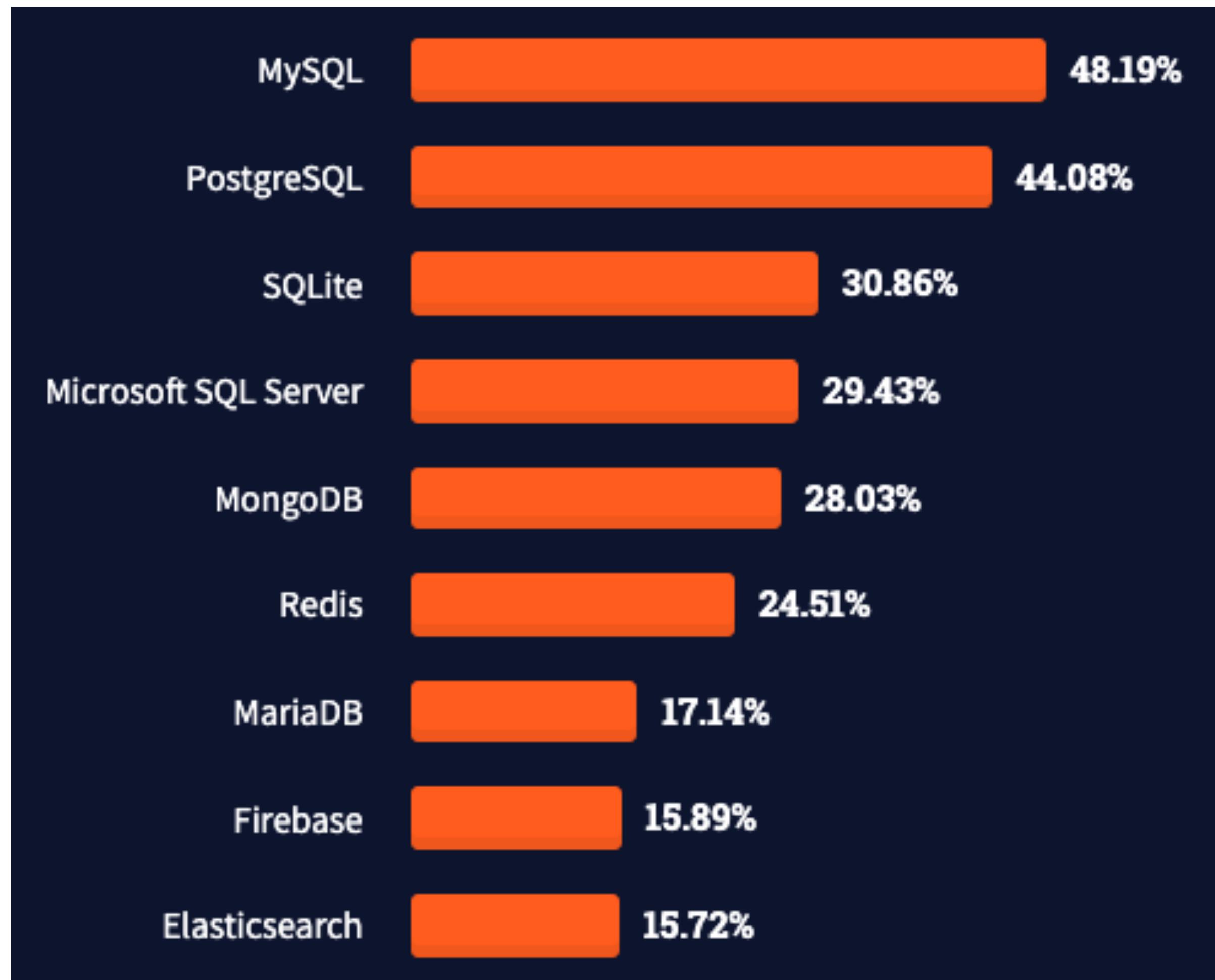
# Developer Roles
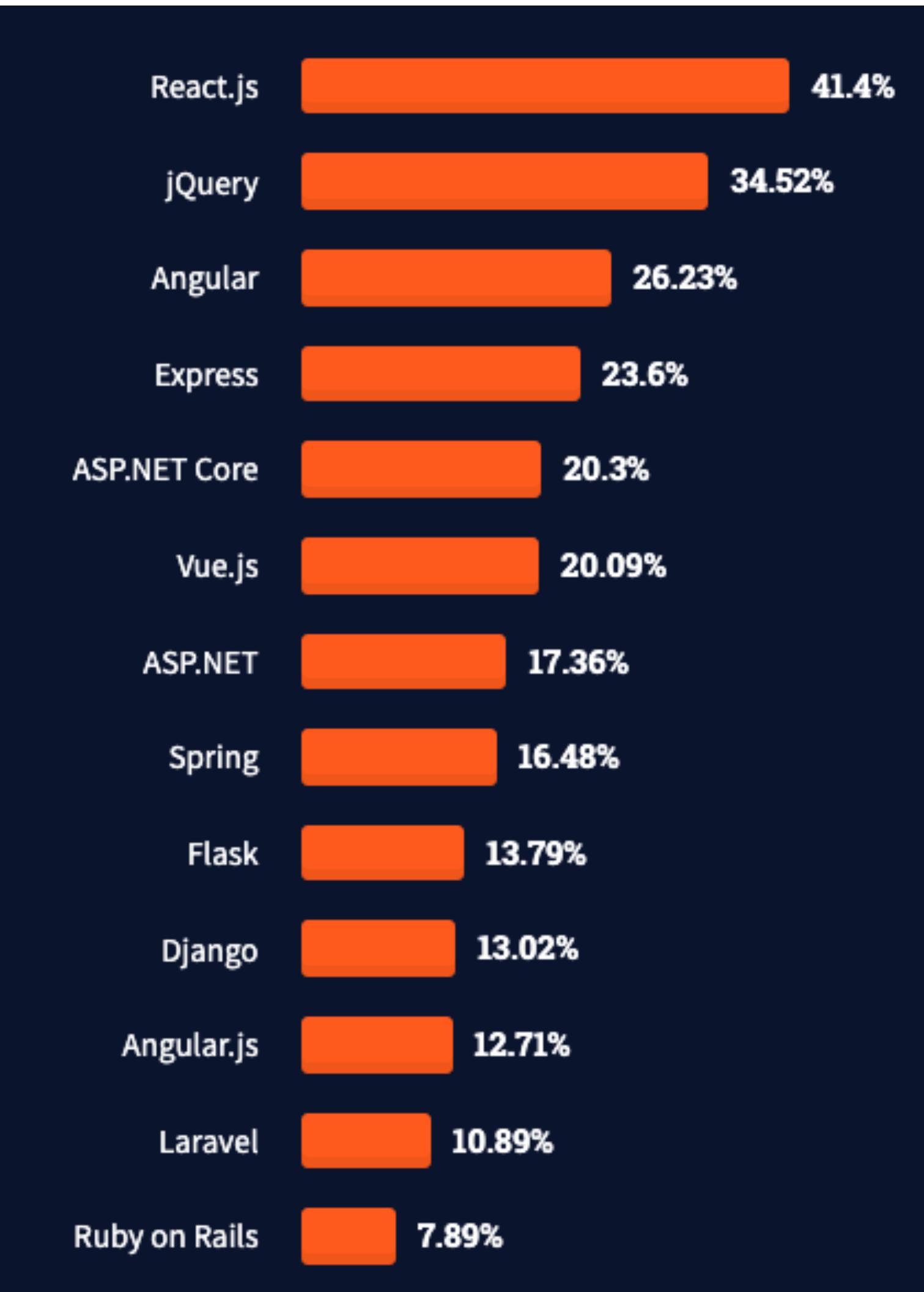
Popular Technologies
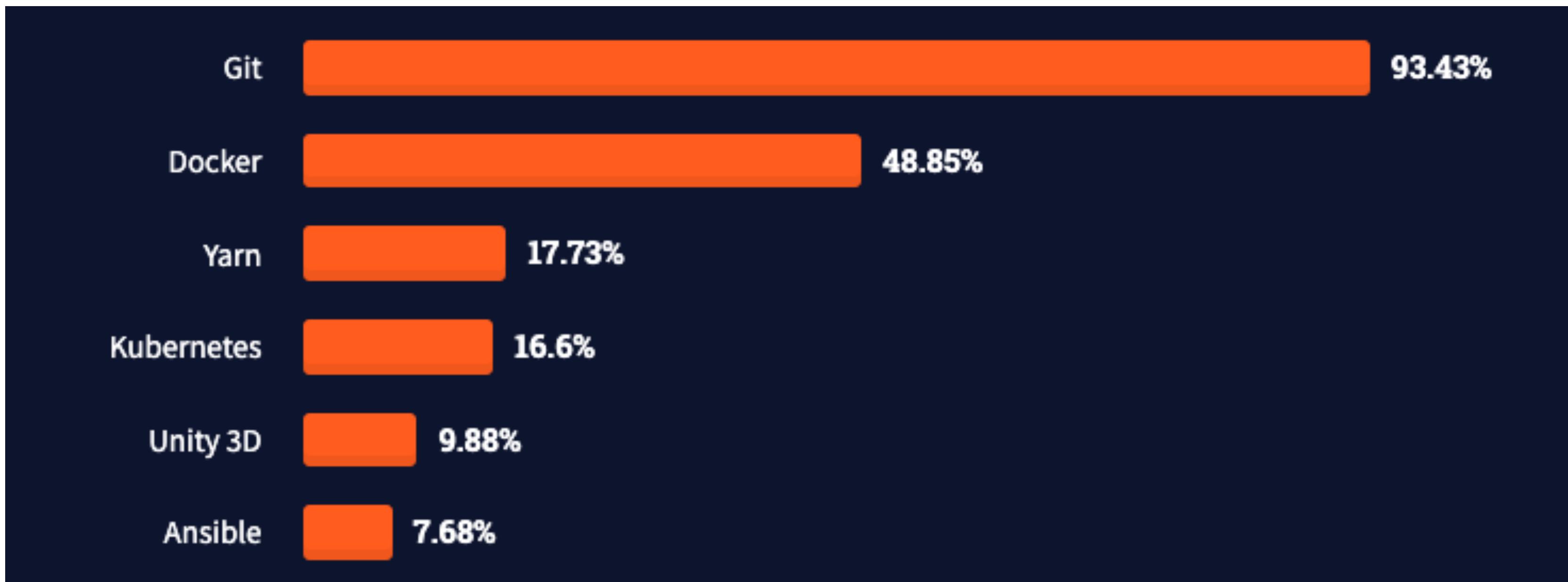
Database Technologies

Web Frameworks

Other Tools

Server-Side Technologies

- There are many options in server-side technologies. In theory as many as programming languages. Most popular options: PHP, Java, Python, Ruby, Perl, C#, Go. Also JavaScript with node.js.

- Server-side technologies also includes data management solutions, e.g. filesystem, database management systems. Most popular options: MySQL, PostgreSQL, Oracle, SQL Server, SQLite.

- **In LBAW we will use PHP, Laravel framework, and PostgreSQL.**

Client-Side Technologies

- Client-side technologies run on the web client.

- Three main technologies: HTML, CSS and JavaScript. HTML and CSS are declarative languages, while JavaScript is a full-fledged programming language.

- There is an increasing number of libraries and *boilerplates* to support client-side web development.

- **In LBAW we will use HTML, CSS, JavaScript and AJAX.**

- Also notes on web performance, security, accessibility and usability.

Further Reading

- "As We May Think"
  Vannevar Bush. Atlantic Monthly, 1945

- Computer Networks and Internets
  Douglas Comer. Prentice Hall, 2004

- HTTP: The Definitive Guide
  David Gourley, Brian Totty. O'Reilly, 2002

- Opera Web Standards Curriculum
  http://dev.opera.com/articles/wsc/

- The Modern Web: Multi-Device Web Development with HTML5, CSS2 and JavaScript
  Peter Gasston. No Starch Press, 2013
  http://modernwebbook.com/