Introduction to the course

Prerequirements

Very good knowedge of OOP – **ESSENTIAL: MUST HAVE !**

- **What is a Class?**
- **What is an Object?**
- **What is Inheritance?**
- **What is an abstract class?**
- **What is an instance variable (member variable) ?**
- **What is a method (member function)?**
- **What is polymorphism?**

Prerequirements

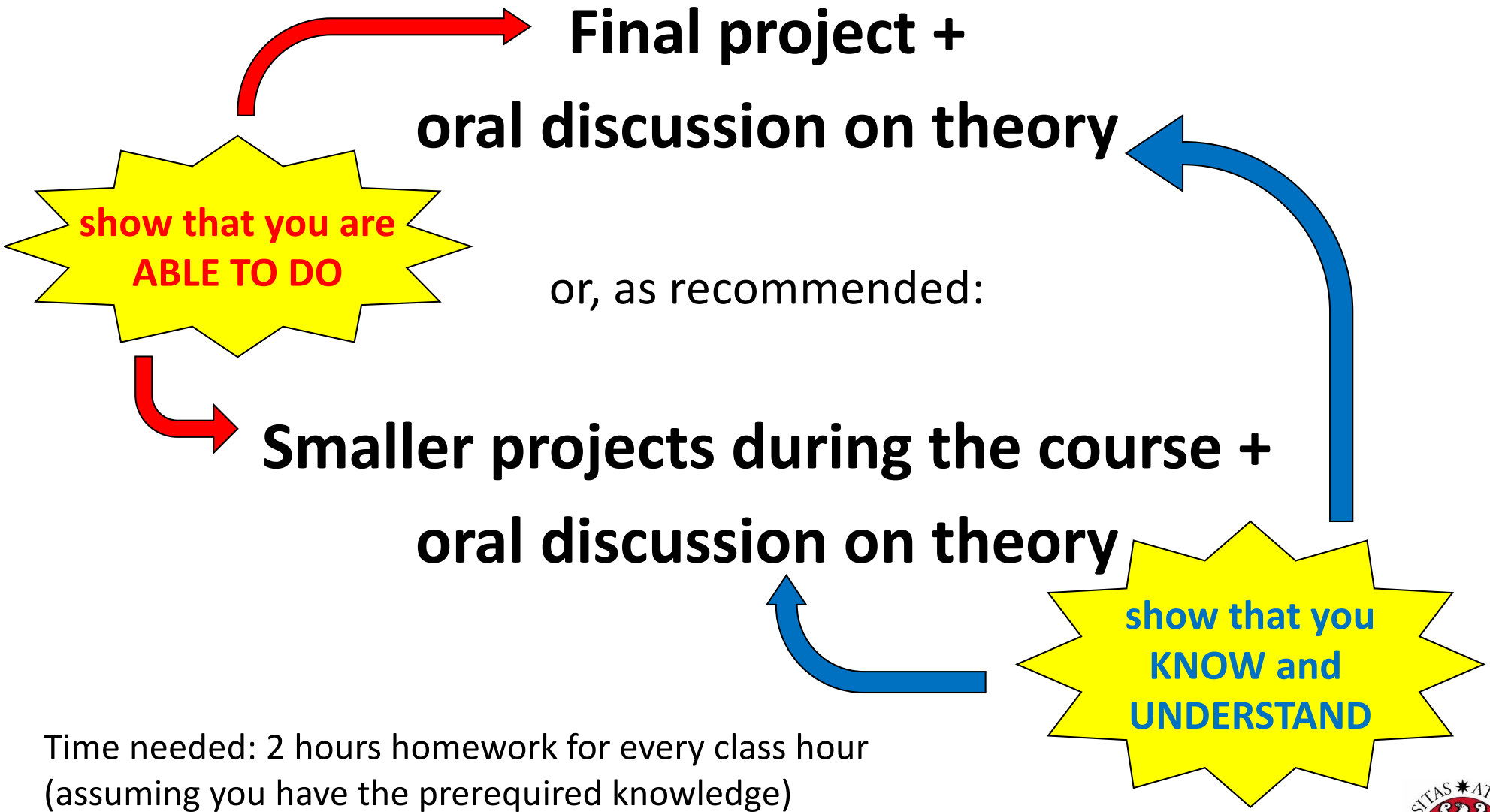
- Good knowedge of Java
- Basic knowedge of Networks
- Good knowedge of Databases (SQL)
- Basic knowedge of HTML – only advanced topics will be discussed
- Basic knowedge of CSS –a quick catch-up will be given
- Basic knowedge of Web servers (cookies, sessions – a quick catch-up will be given)

Readings to (partially) catch up:

- HTML:
  - http://www.cellbiol.com/bioinformatics_web_development/chapter-3-your-first-web-page-learning-html-and-css/
  - https://www.tutorialspoint.com/html/index.htm

- Java: https://docs.oracle.com/javase/tutorial/
  - sections: Basics, up to "Collections" included
- Networking: http://www.cellbiol.com/bioinformatics_web_development/chapter-1-internet-networks-and-tcp-ip/
  - sections: 1.1 to 1.3 included
- Databases: https://www.tutorialspoint.com/dbms/index.htm
- SQL: https://www.w3schools.com/sql/

Exam



Time needed: 2 hours homework for every class hour
(assuming you have the prerequired knowledge)

Our first tool: IntelliJ Idea (Ultimate)

Free educational licence for students.

See:

https://www.jetbrains.com/community/education/#students

Download, install and get familiar with it.

- installation guide at https://www.jetbrains.com/help/idea/installation-guide.html#standalone
- Overview of UI: https://www.jetbrains.com/help/idea/guided-tour-around-the-user-interface.html
- Create you first Java App: https://www.jetbrains.com/help/idea/creating-and-running-your-first-java-application.html

(Next tools will be Apache WS (LAMP/MAMP/WAMP), Tomcat, Firefly.)

Resources:

Web site:

https://tinyurl.com/5wp749s7

or

https://latemar.science.unitn.it

STUDENTS ARE REQUIRED TO ENROL IN THE COURSE ON MOODLE

https://didatticaonline.unitn.it/dol/course/view.php?id=32812

On moodle there is a forum for questions – students are encouraged to respond, helping their peers (support activity will be positively evaluated)

Introduction to the Web Architectures course

Q

What is the difference between the Web and the Internet?

What is the Internet?

- "The Internet is a global system of interconnected computer networks that interchange data by packet switching using the standardized Internet Protocol Suite (TCP/IP)."

- Thus, the Internet is a network of networks, defined by the TPC/IP standards (such as FTP, Telnet, SMTP…).

What is the web?

The Wold Wide Web is an **information system** where documents and other web resources are identified by **Uniform Resource Locators** (URLs, such as https://www.example.com/), which may be interlinked by **hypertext**, and are **accessible over the Internet**.

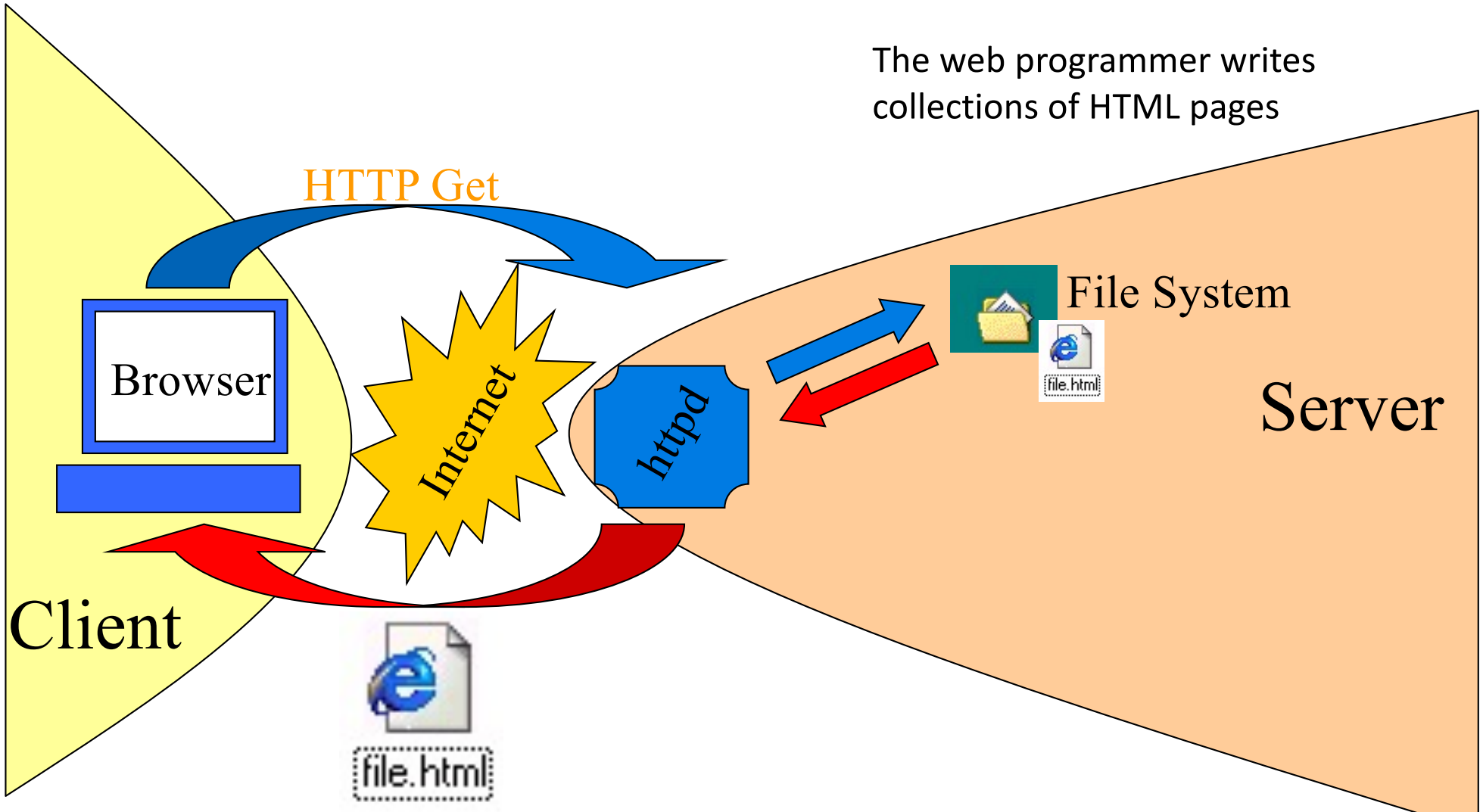
The resources of the WWW are

- described via the **Hypertext Markup Language** (HTML)
- transferred via the **Hypertext Transfer Protocol** (HTTP)
- accessed by users by a software app called a web browser
- published by a software app called a web server.

Q

How does the web work ?

The original web architecture: static pages



Initial idea: get (static) interlinked documents

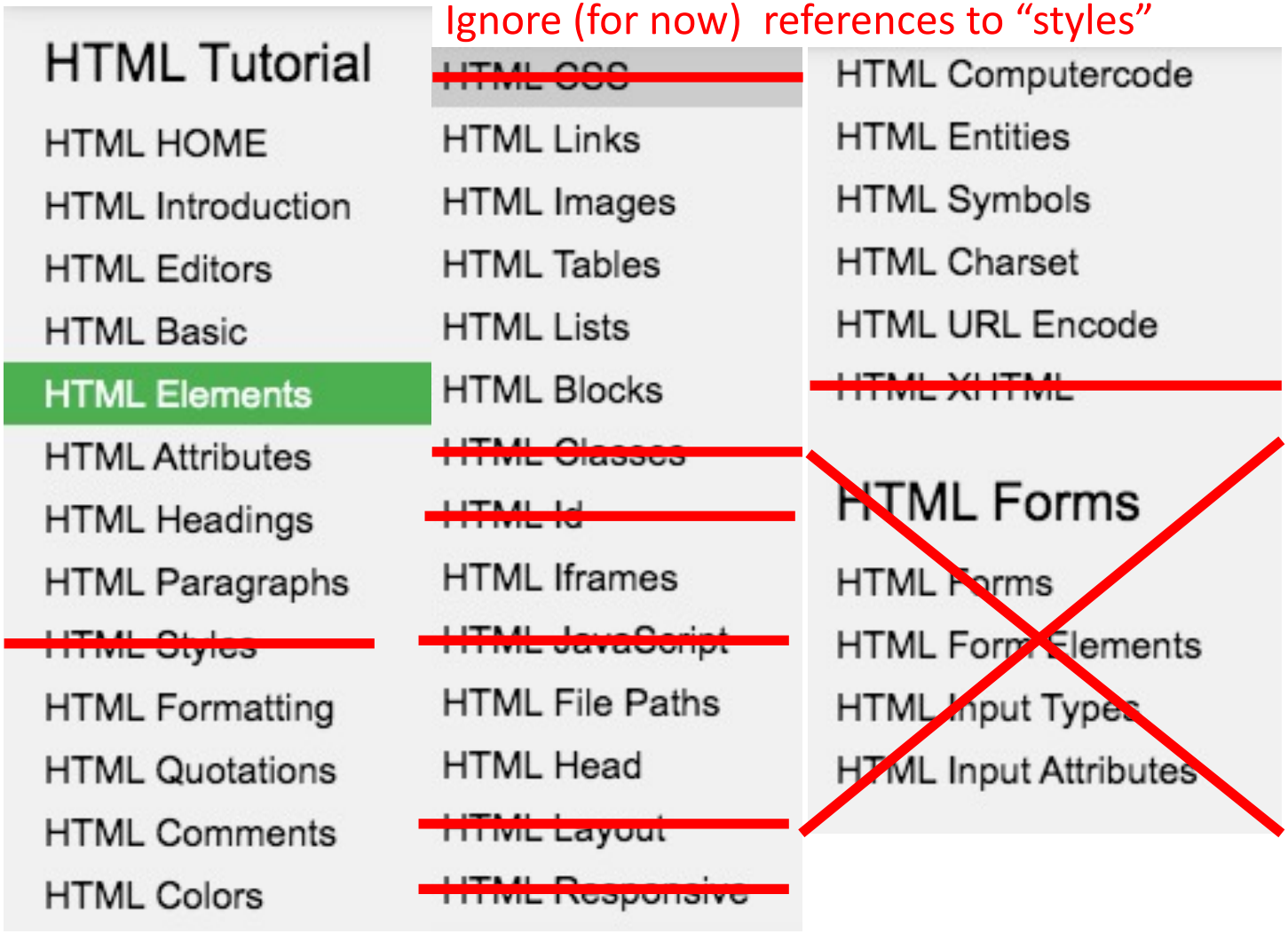
What am I supposed to already know about HTML ?

Basic HTML

http://www.cellbiol.com/bioinformatics_web_development/chapter-3-your-first-web-page-learning-html-and-css/

https://www.tutorialspoint.com/html/index.htm

Ignore (for now) references to "styles"



https://www.w3schools.com/html/default.asp

Q

What is a Server ?

Client and Server

from "Computer Networking: A Top-Down Approach", Kurose-Ross:

- The computers and other devices connected to the Internet are often referred to as end systems (**hosts**).
- Hosts are sometimes further divided into two categories: **clients** and **servers**.

BAD DEFINITION

- Informally, clients tend to be desktop and mobile PCs, smartphones, and so on, whereas servers tend to be more powerful machines that store and distribute Web pages, stream video, relay e-mail, and so on.

Servers ?

Server = Waiter, Client = Customer

Client and Server

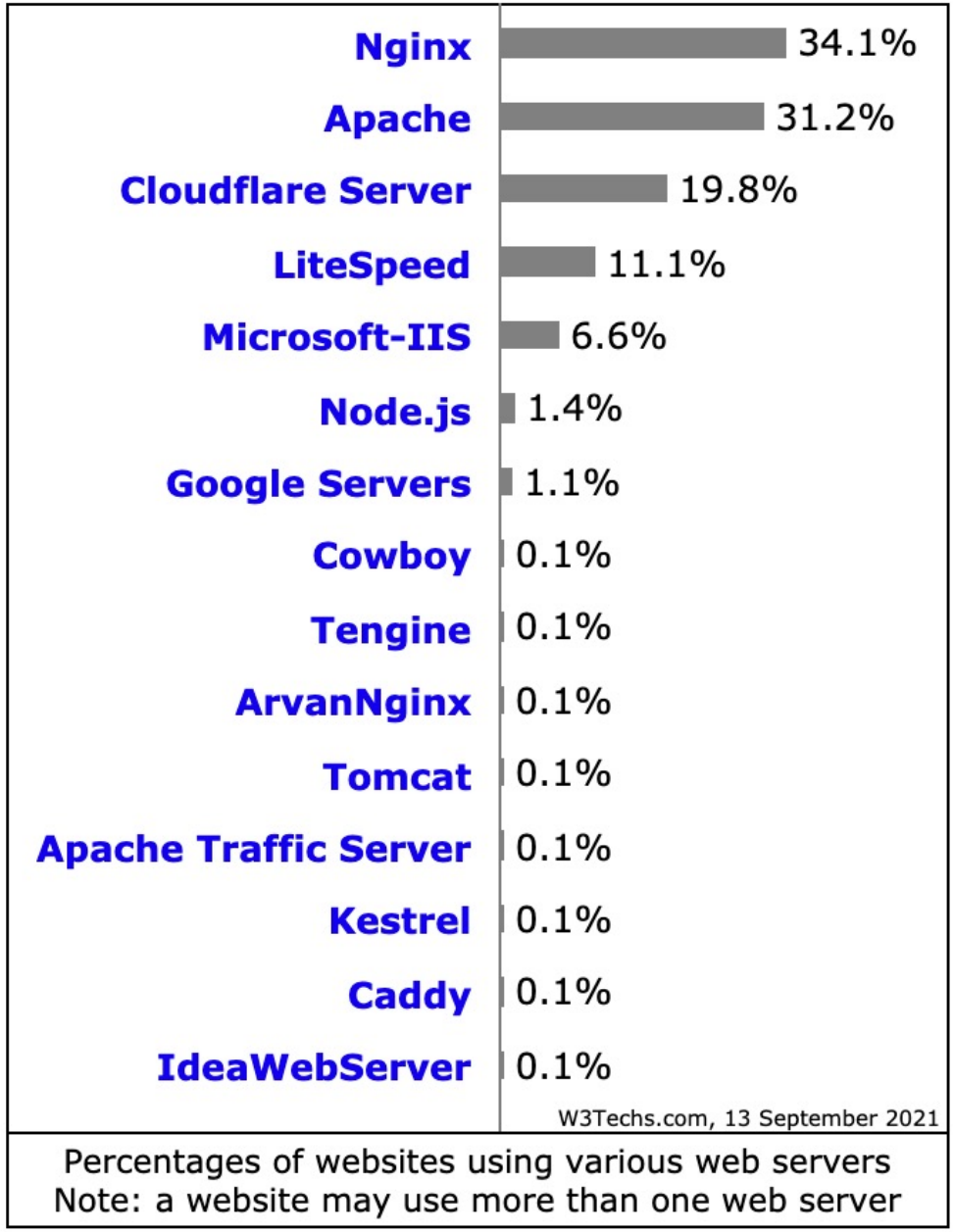
A better definition:

- A Server is a machine that opens a SocketServer connection, and waits for incoming calls (in order to provide a service)

- A Client is a machine the starts a connection (opening a Socket to the server) and requests a service.

Server and client are SOFTWARE roles,
not HARDWARE concept!

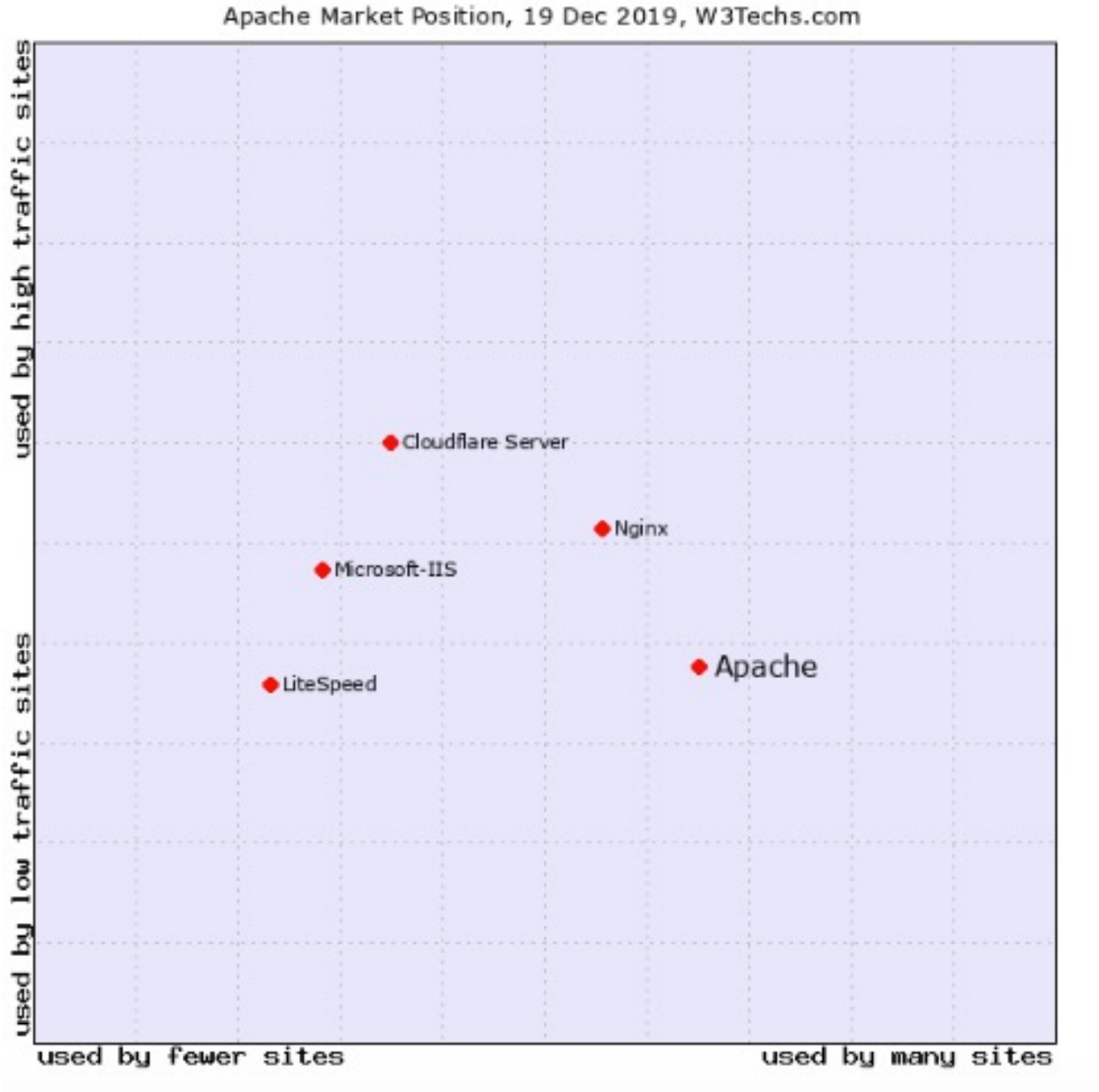
Servers on the web



Percentages of websites using various web servers
Note: a website may use more than one web server

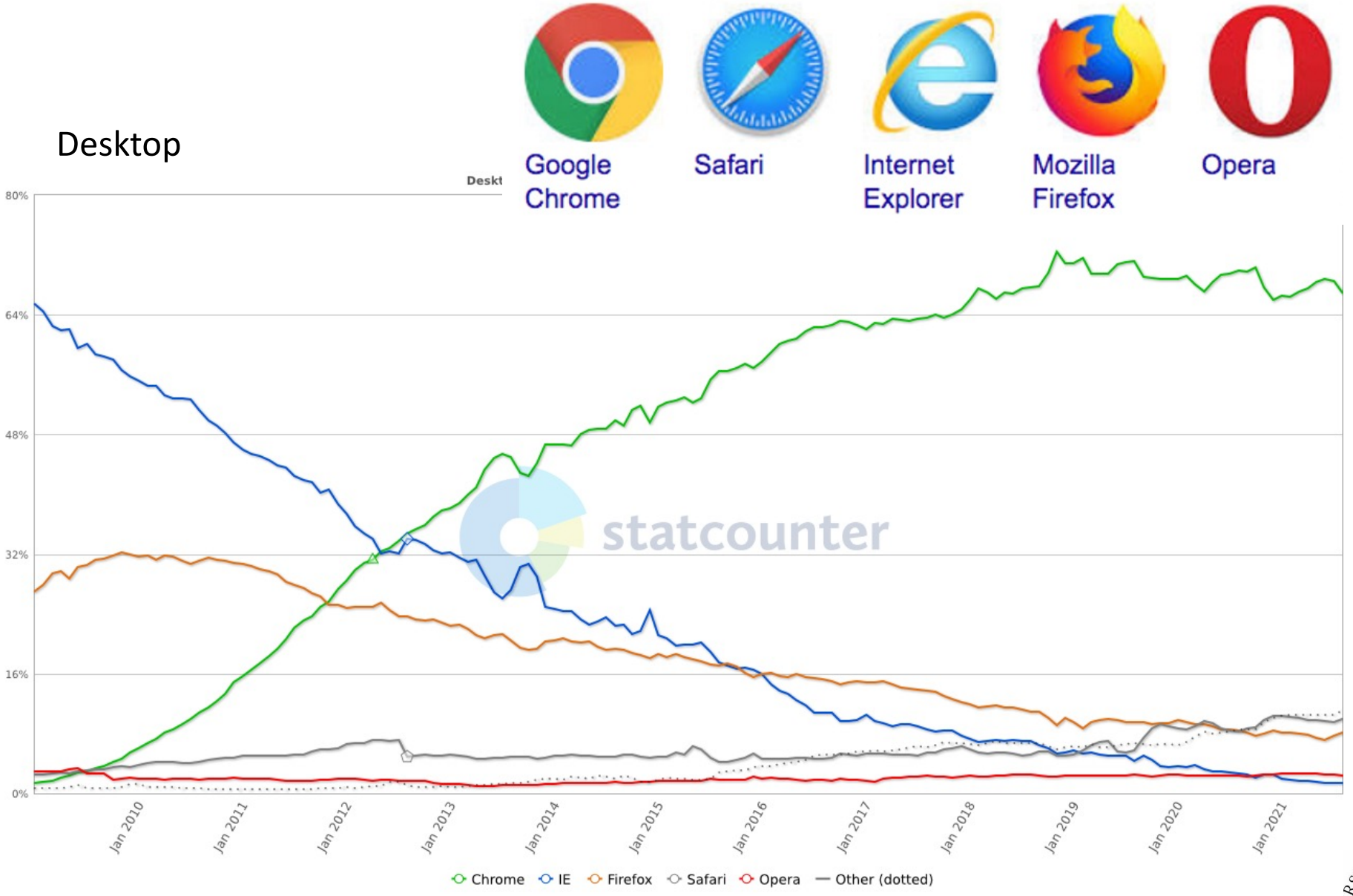
We will mostly refer to servers on Unix Systems.

Servers on the web



Clients on the Web: Browsers

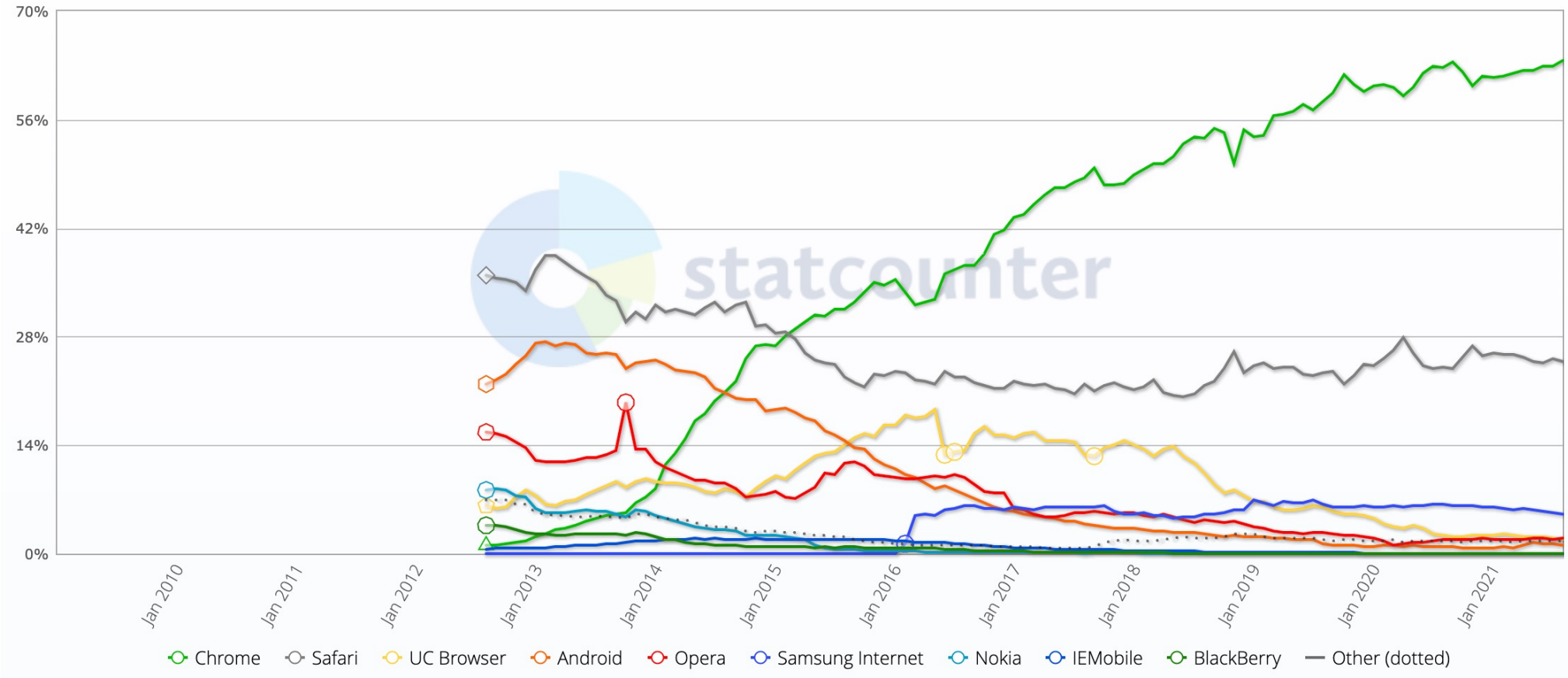
Desktop



Clients on the Web: Browsers

https://gs.statcounter.com/browser-market-share

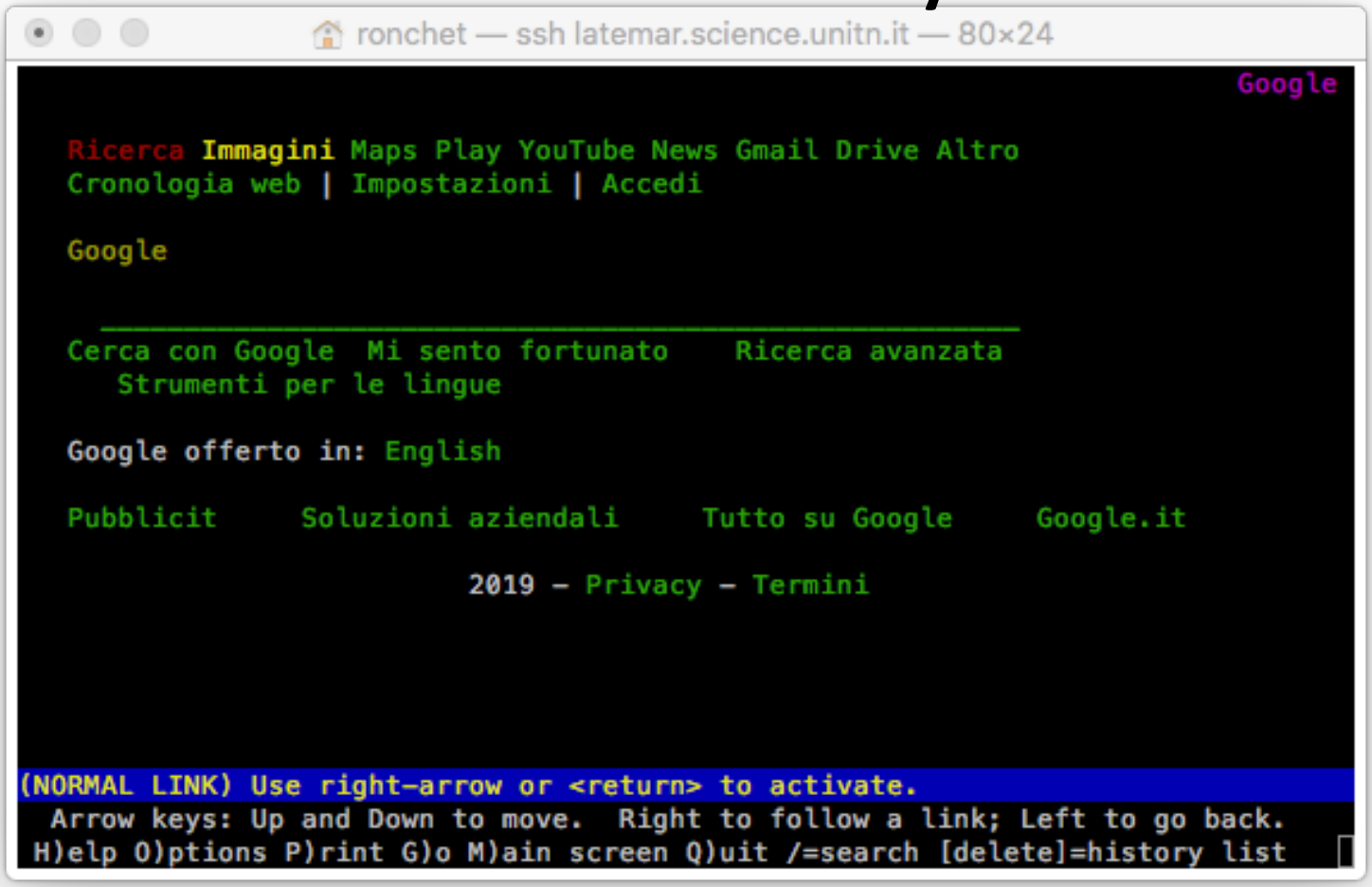
Tablet + Mobile



Clients on the Web: Browsers

Could also be something else!

Lynx – text browser



https://invisible-island.net/lynx/lynx.html
https://macappstore.org/lynx/

sudo apt install lynx

Clients on the Web: Browsers

Could also be something else!

WebView in JavaFX

Q

Which languages are most used on the web ?

Back-end (Server-side) languages in most popular websites

Websites	C#	C	C++	D	Erlang	Go	Hack	Java	JavaScript	Perl	PHP	Python	Ruby	Scala	Xhp
Google.com	No	Yes	Yes	No	No	Yes	No	Yes	No	No	Yes	Yes	No	No	No
YouTube.com	No	Yes	Yes	No	No	Yes	No	Yes	No	No	No	Yes	No	No	No
Facebook.com	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No	No	Yes
Yahoo	No	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Amazon.com	No	No	Yes	No	No	No	No	Yes	No	Yes	No	No	No	No	No
Twitter.com	No	No	Yes	No	No	No	No	Yes	No	No	No	No	Yes	Yes	No
eBay.com	No	No	No	No	No	No	No	Yes	Yes	No	No	No	No	Yes	No
Linkedin.com	No	No	No	No	No	No	No	Yes	Yes	No	No	No	No	Yes	No
Wikipedia.org	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No	No
Bing	Yes	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No
MSN.com	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Pinterest	No	No	No	No	Yes	No	No	No	No	No	No	Yes	No	No	No
WordPress.com	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No	No

7 8 5 5

Front-end (client-side) languages in most popular websites

Websites	Popularity (unique visitors per month)	Front-end	Notes
Google.com	1,600,000,000	JavaScript, TypeScript	The most used search engine in the world
Facebook.com	1,100,000,000	JavaScript	The most visited social networking site
YouTube.com	1,100,000,000	JavaScript	The most visited video sharing site
Yahoo	750,000,000	JavaScript	
Amazon.com	500,000,000	JavaScript	Popular internet shopping site
Wikipedia.org	475,000,000	JavaScript	"MediaWiki" is programmed in PHP; free online encyclopedia
Twitter.com	290,000,000	JavaScript	Popular social network.
Bing	285,000,000	JavaScript	Search engine from Microsoft.
eBay.com	285,000,000	JavaScript	Online auction house.
MSN.com	280,000,000	JavaScript	An email client, for simple use. Previously known as "messenger", not to be confused with Facebook's messaging platform.
Linkedin.com	260,000,000	JavaScript	World's largest professional network.
Pinterest	250,000,000	JavaScript	Search engine for ideas.
WordPress.com	240,000,000	JavaScript	Website manager software.

Why is Web project development different?

Web development challenges

- The functionality of a web application is distributed, with part running on the server near the application's data and part running in the client(browser) near the application users.

- Web applications must integrate a complex collection of diverse technologies, (HTML, JavaScript, HTTP, SSL and one or more server side languages such as PHP, Java, Python or Ruby). Typically different languages are used for server and client side programming.

- Web Applications must support variety of browsers (and devices!) with feature sets that are not still 100% compatible.

Web development challenges

- Web applications support multiple users with (what appears to be) a single server. Developers must deal with concurrency and scalability. Popular web applications must handle 100 – 1000× the workload of traditional applications.

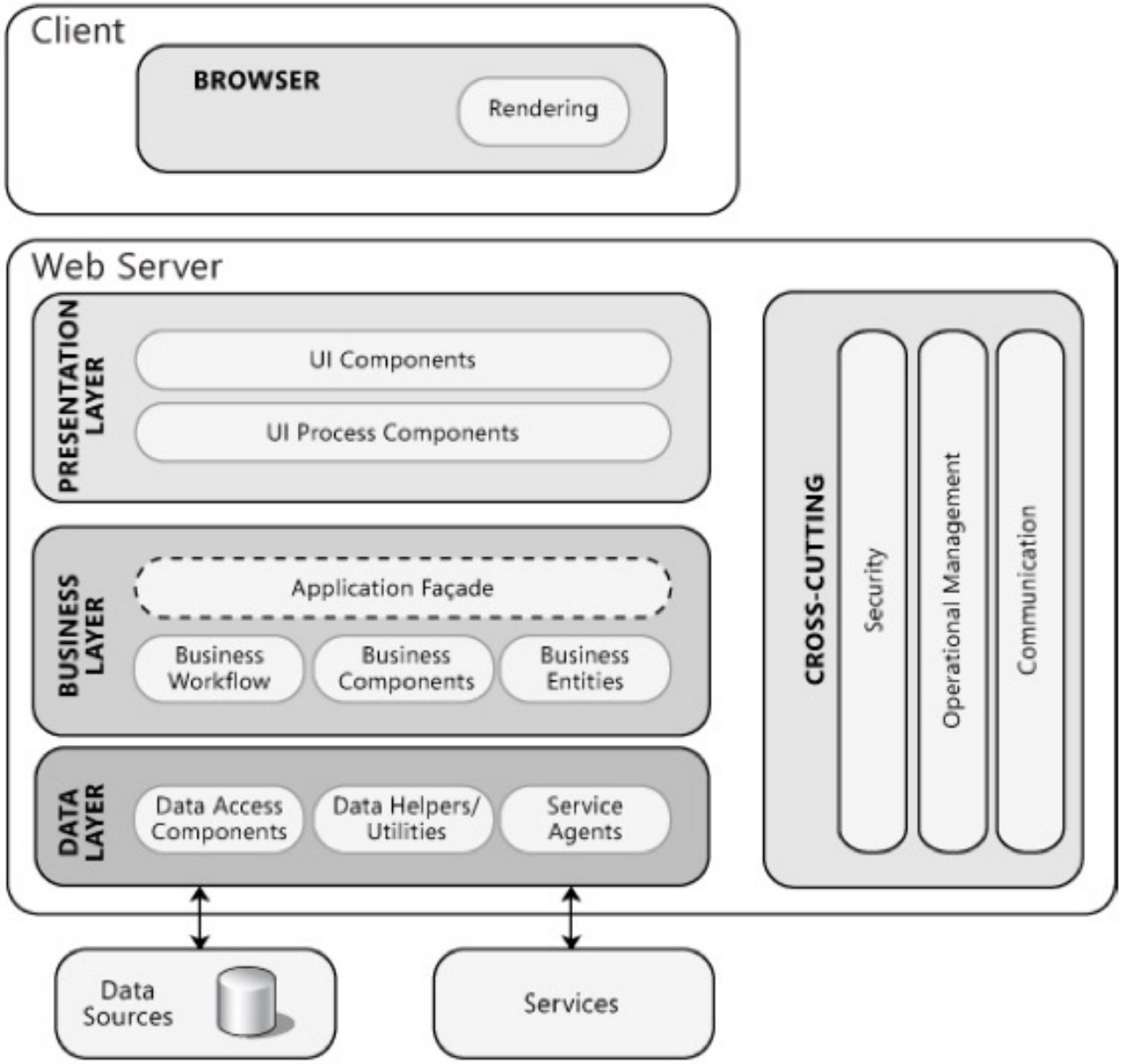
- Web applications must be highly customizable. In the pre-Web GUI world a uniform look and feel was encouraged, but web applications strive for unique appearance and behaviors. This makes it more difficult to create reusable components.

- The public accessibility of web applications introduces a variety of security and privacy issues. It is easy for unaware developers to create security loopholes such as SQL injection attacks.

Q

What are the steps on a Web project development ?

(Traditional) Logical Web Architecture



Data Layer

abstracts the logic required to access the data source.

- Using a separate data layer makes the application easier to configure, maintain and hides the details of the data sources from other layers of the application.
- Entity objects can be populated by this layer or can be used to update the data sources.
- Data Transfer Objects(DTOs) are used to pass the data between layers.

Business Layer

implements the business logic of the application and long-running work-flows.

- A separate business layer:
  - improves the maintainability and testability of an application
  - allows to centralize and reuse common business logic functions.
- Business entities that represent the real world data are used to pass data between components.

Presentation layer

- **displays the user interface(UI)**
- **facilitates the user interaction**

user interaction logic is decoupled from the UI components.

The presentation layer consists of

- server-side components: prepares the HTML
- client side component: executes scripts and displays the HTML sent by the server side component.

Cross-cutting components

- **Security** for implementing functionality related with authentication, authorization, and validation.
- Operational Management for **exception handling** policies, **logging**, **performance counters**, **configuration**, and **tracing** tasks.
- Communication to **facilitate the communication with other services and applications**.

Web development

- Model the domain.
- Define the Database, and database queries to persist and fetch data from the database
    - Create server-side code for accessing the database
    - Implement server-side code for any business logic
    - Implement an API that can be used to provide data for the presentation over an HTTP connection
    - Write integration tests for the API
- Start writing frontend code in JavaScript or similar technology
- Write client-side services to fetch data from the backend API
- Write client-side components to display the data on a web page.
- Build the page and style it
- Write some automated end-to-end tests for the web page.