ICT for Organizations

Web Applications: Architectures and Development

Prof. ssa Romina Eramo

Università degli Studi di Teramo

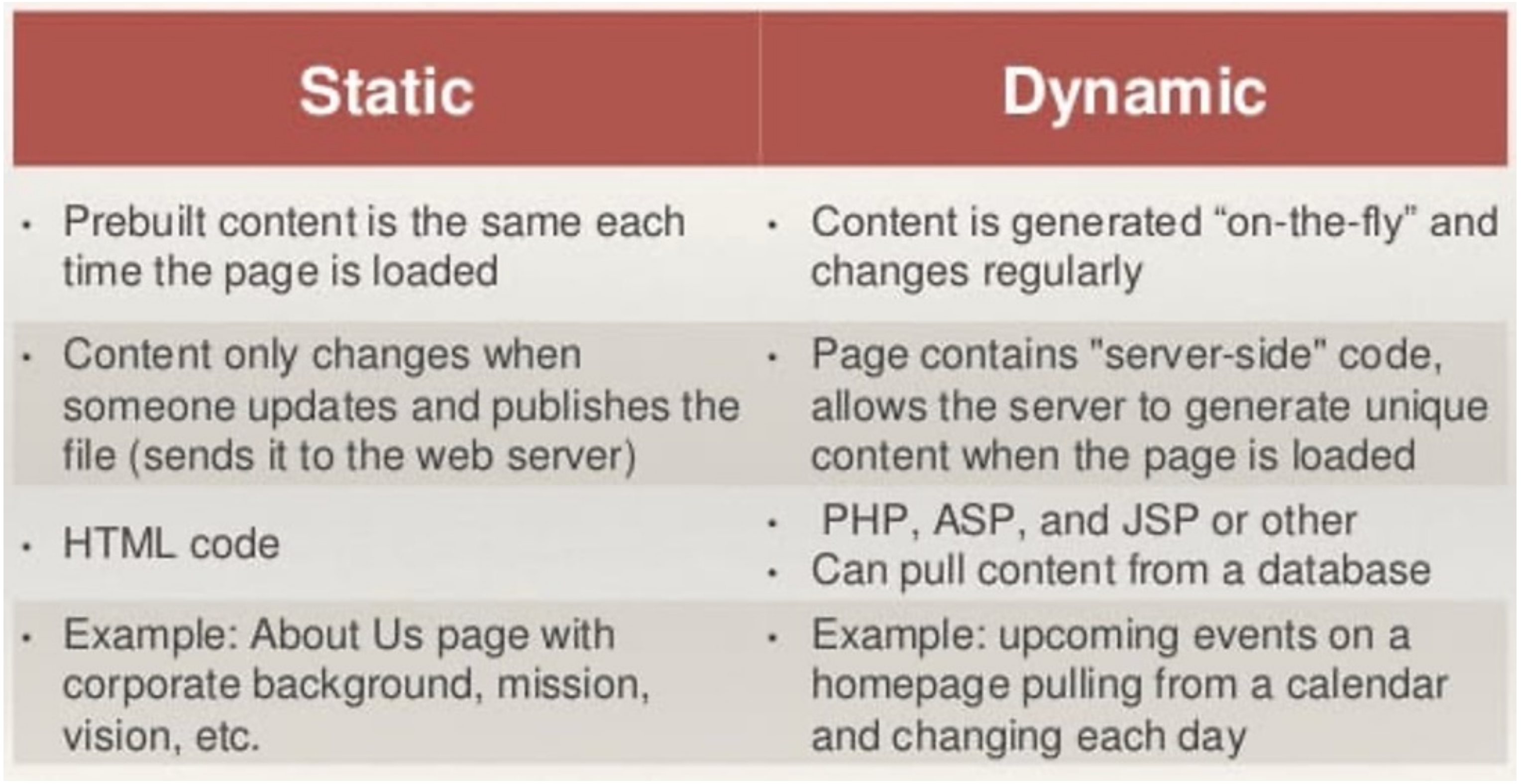
Dipartimento di Scienze della Comunicazione

reramo@unite.it

Static vs Dynamic web applications

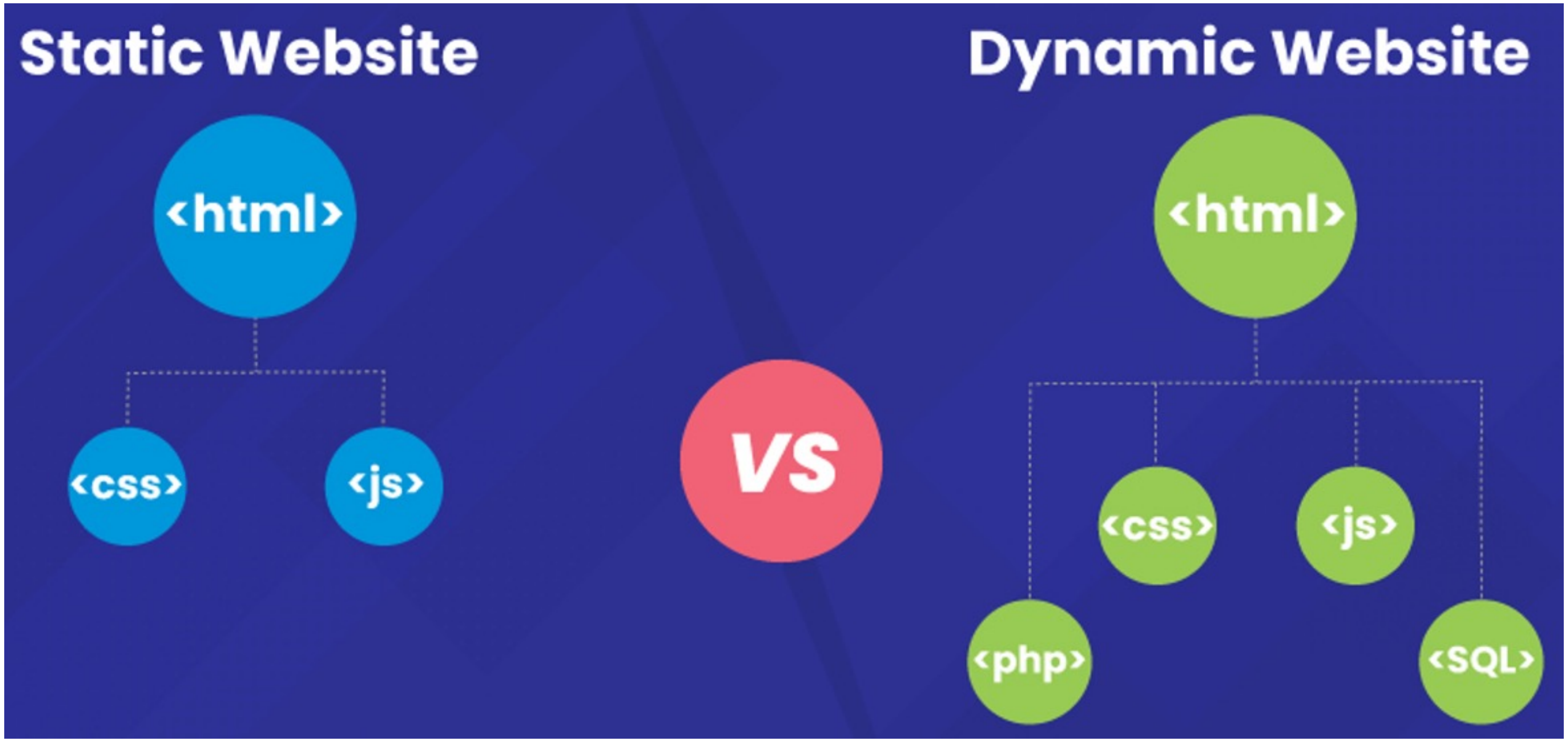
- When computing power was at a premium, web pages were often served as unchanging text files. This type of static content is very efficient, but it can quickly become stagnant if not updated.
- With the rise of low-cost computing and faster Internet speeds, developers created server-side platforms and architectures that could generate user-specific content on the fly. This form of content is known as **dynamic content.**
  - Dynamic content is a webpage (component) that changes based on user signals that include in-session behavior, user data, and user-characteristics

Static vs Dynamic web applications

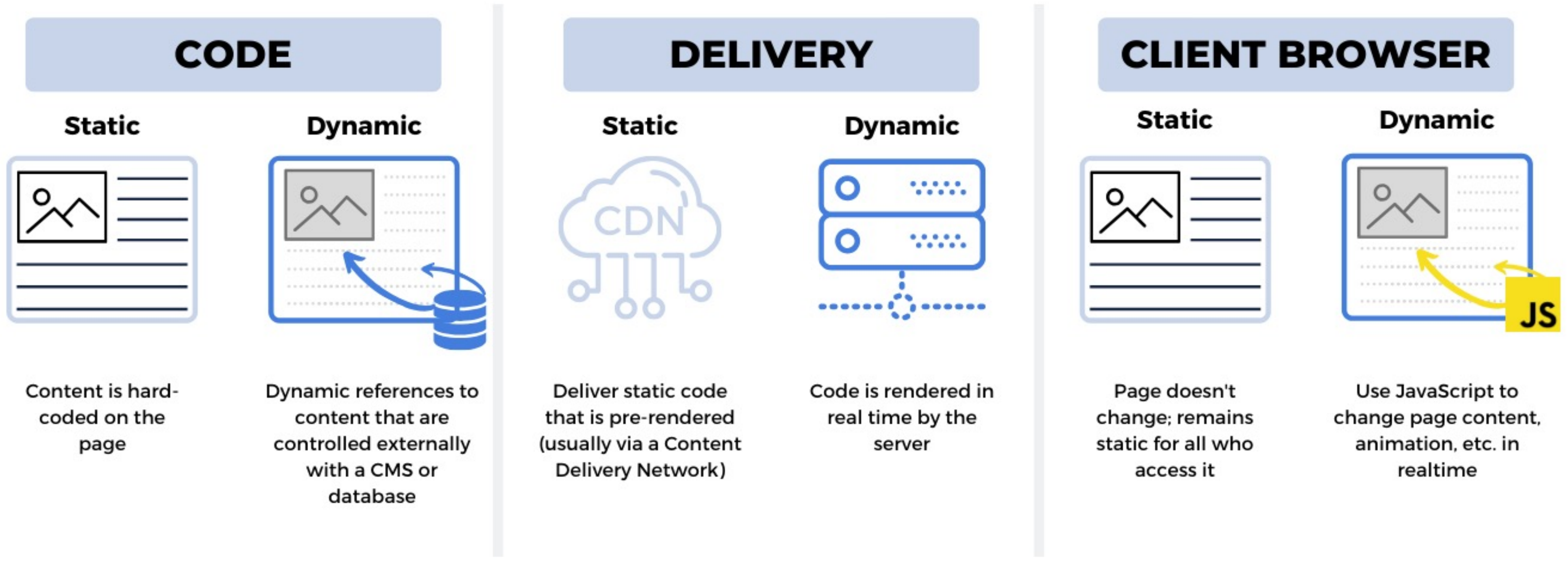


Static	Dynamic
• Prebuilt content is the same each time the page is loaded	• Content is generated "on-the-fly" and changes regularly
• Content only changes when someone updates and publishes the file (sends it to the web server)	• Page contains "server-side" code, allows the server to generate unique content when the page is loaded
• HTML code	• PHP, ASP, and JSP or other • Can pull content from a database
• Example: About Us page with corporate background, mission, vision, etc.	• Example: upcoming events on a homepage pulling from a calendar and changing each day

Static vs Dynamic web applications



Static vs Dynamic web applications



CODE

Static

Content is hard-coded on the page

Dynamic

Dynamic references to content that are controlled externally with a CMS or database

DELIVERY

Static

Deliver static code that is pre-rendered (usually via a Content Delivery Network)

Dynamic

Code is rendered in real time by the server

CLIENT BROWSER

Static

Page doesn't change; remains static for all who access it

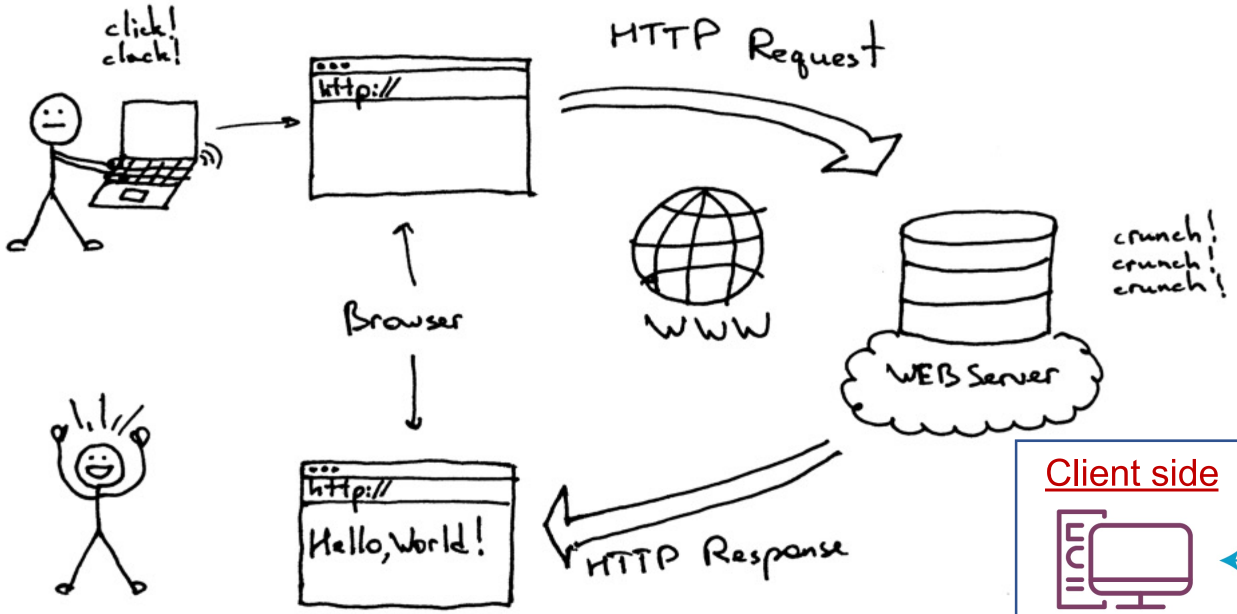
Dynamic

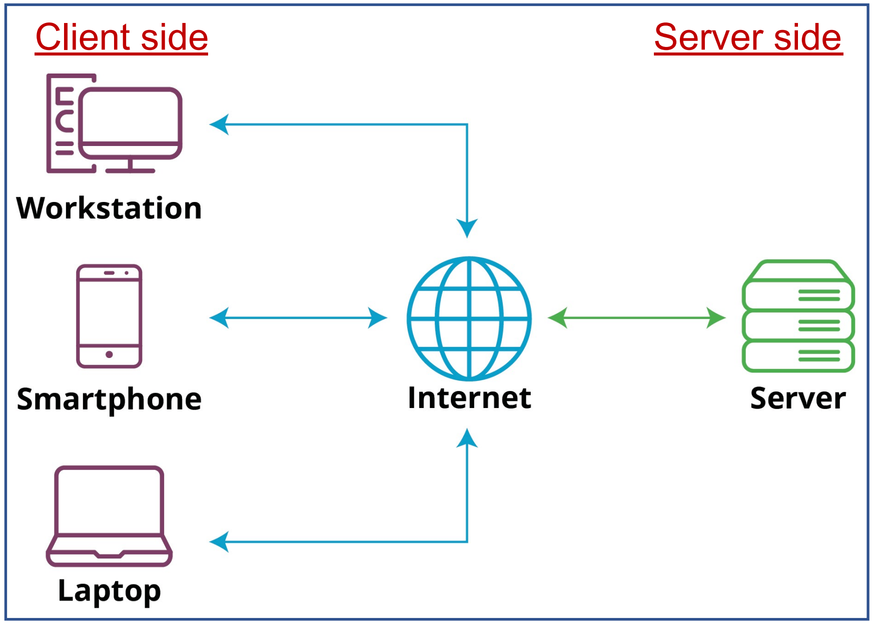
Use JavaScript to change page content, animation, etc. in realtime

Architectures for dynamic content generation

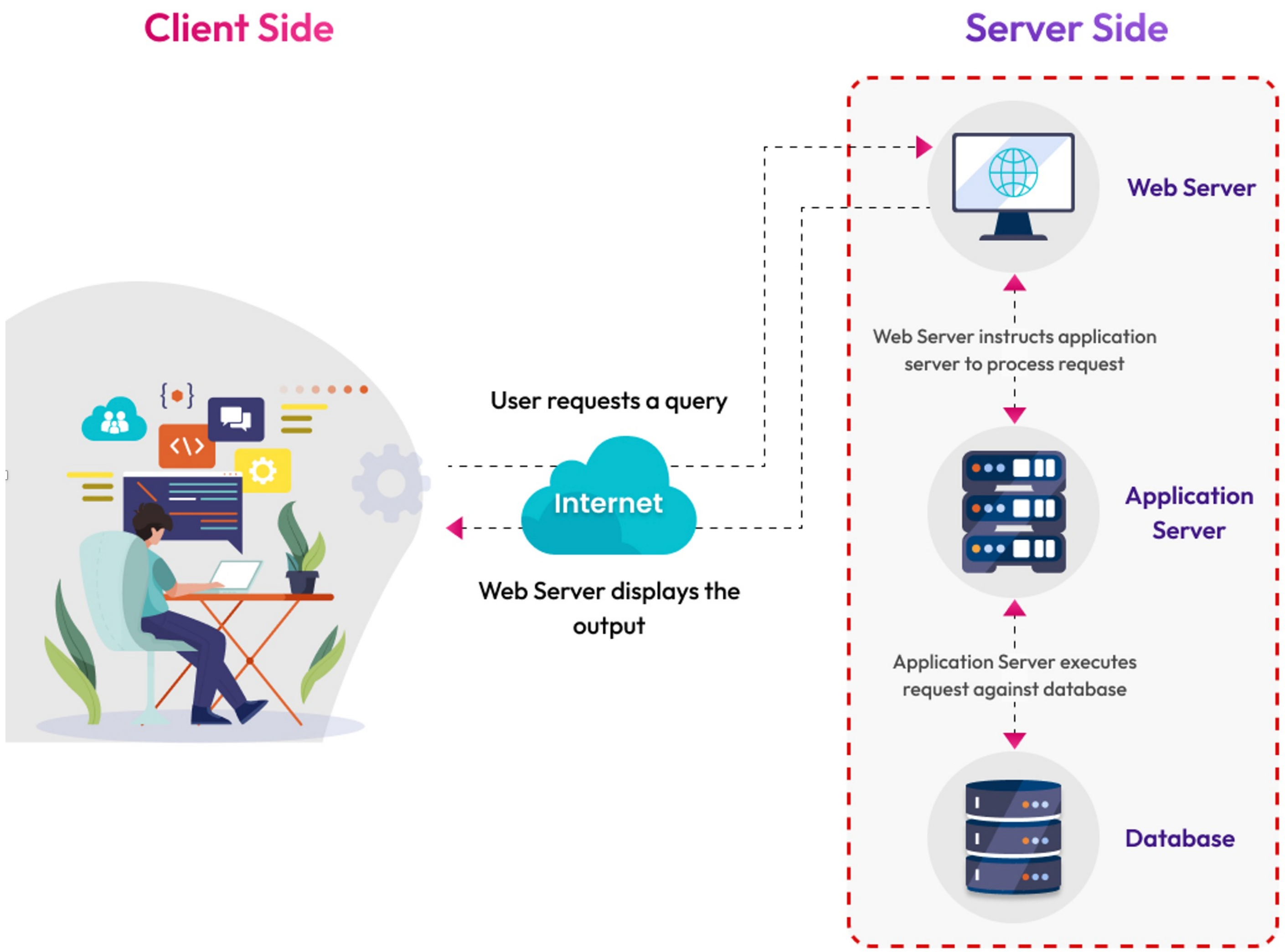
- » HTTP limits for dynamic and interactive application development
- » Dynamic content creation: passing parameters and typing data in HTTP, query string and request body, difference between GET and POST method
- » Web application customization: identification, authentication, authorization, role-based control of access, session management
- » Connection between databases and the web: data modeling for the web, database virtualization and connection
- » Programming of the presentation by means of templates: code modularization, server-side template concept

Client-Server architecture *(recap)*





Client-Server architecture *(recap)*


Client Side
Server Side
User requests a query
Internet
Web Server displays the output
Web Server
Web Server instructs application server to process request
Application Server
Application Server executes request against database
Database

Significance of “tiers”

» N-tier architectures have the same components
  - Presentation
  - Business/Logic
  - Data

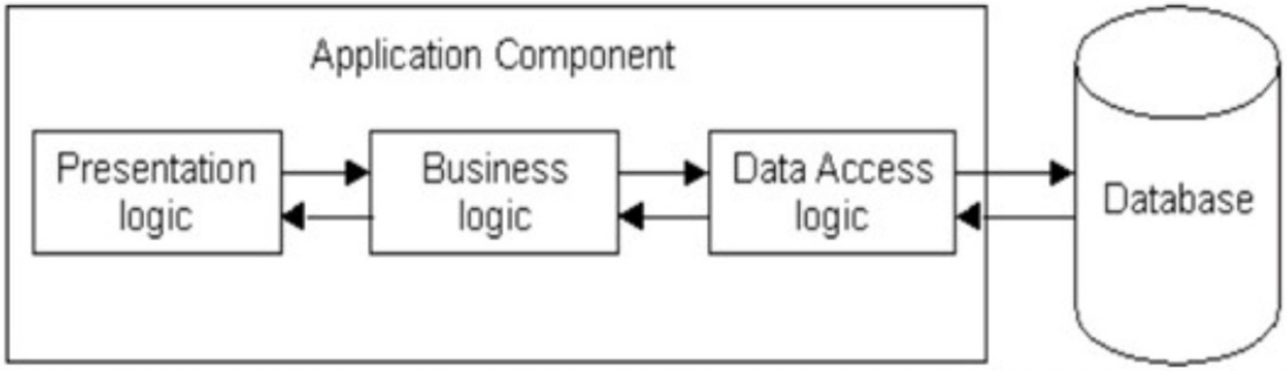
» N-tier architectures try to separate the components into different tiers/layers
  - Tier: physical separation
  - Layer: logical separation

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

Significance of “tiers”

» Database runs on Server
  - Separate from client
  - Easy to switch to a different database

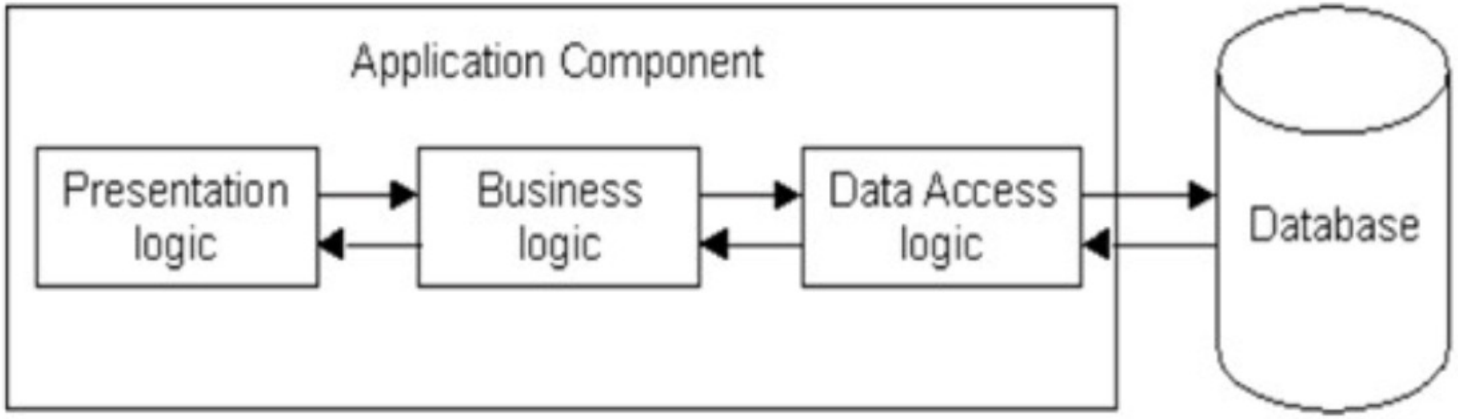
» Presentation and logic layers still tightly connected
  - Heavy load on server
  - Potential congestion on network
  - Presentation still tied to business logic



1-Tier Architecture

» All 3 layers are on the same machine
  - All code and processing kept on a single machine

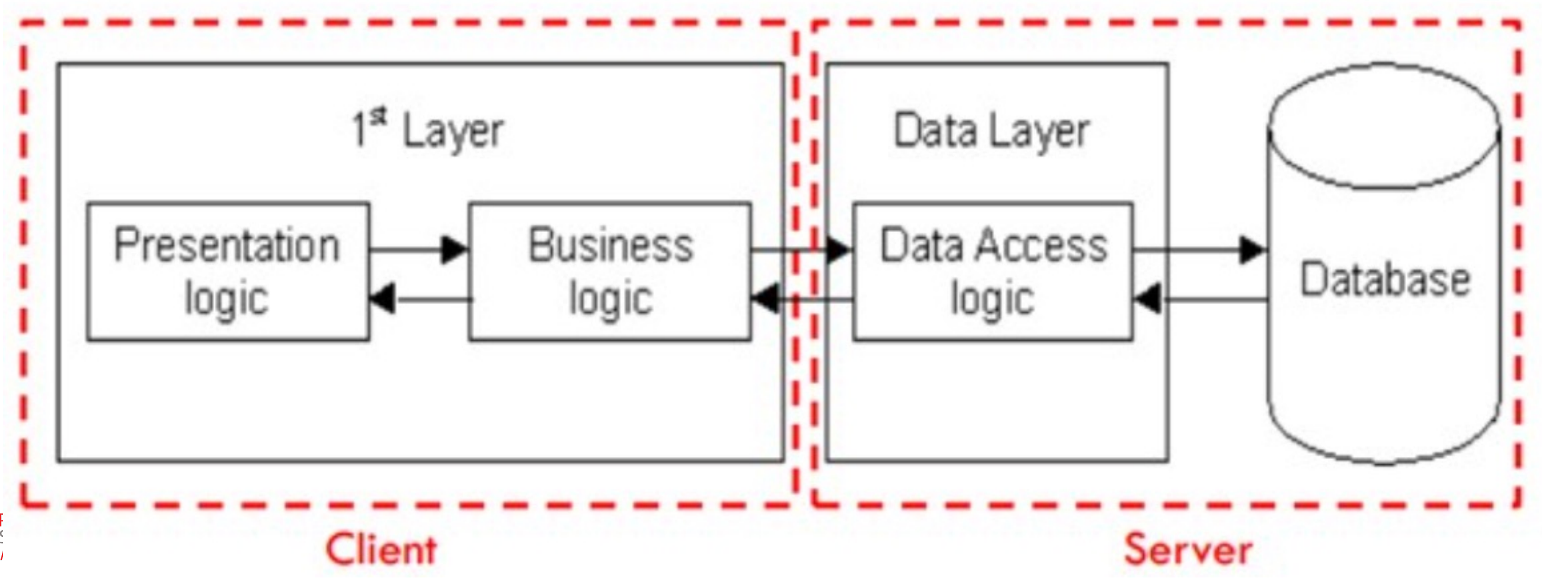
» Presentation, Logic, Data layers are tightly connected
  - Scalability, Portability, Maintenance



2-Tier Architecture

» Database runs on Server
  - Separated from client
  - Easy to swotch to a different database

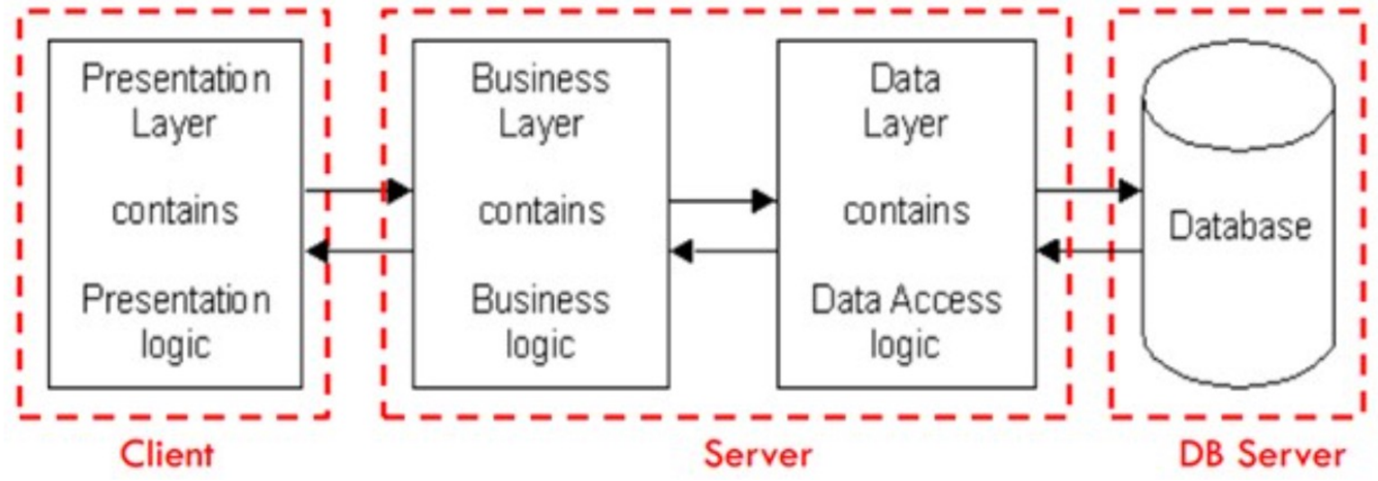
» Presentation and logic layers still tightly connected (coupled)



3-Tier Architecture

» Each layer can ptentially run on a different machine

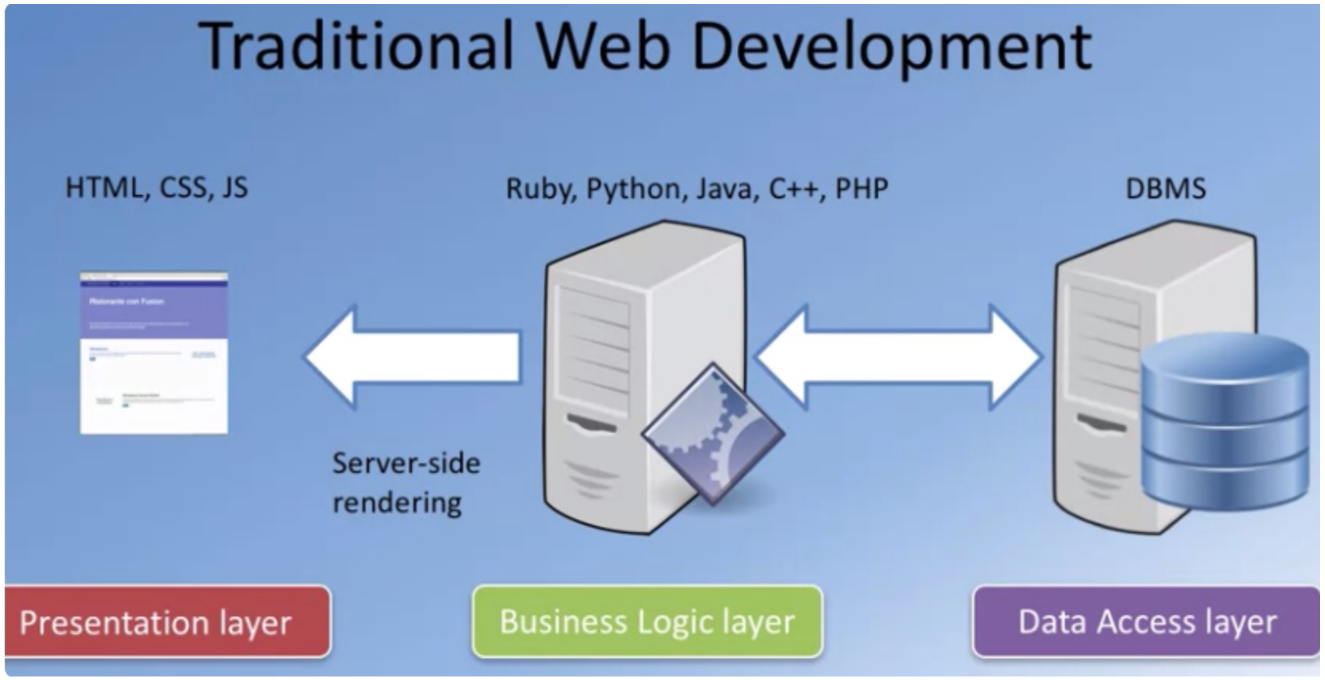
» Presentation, logic and data layers disconnected



3-Tier Architecture

» Each layer can ptentially run on a different machine

» Presentation, logic and data layers disconnected



Presentation tier

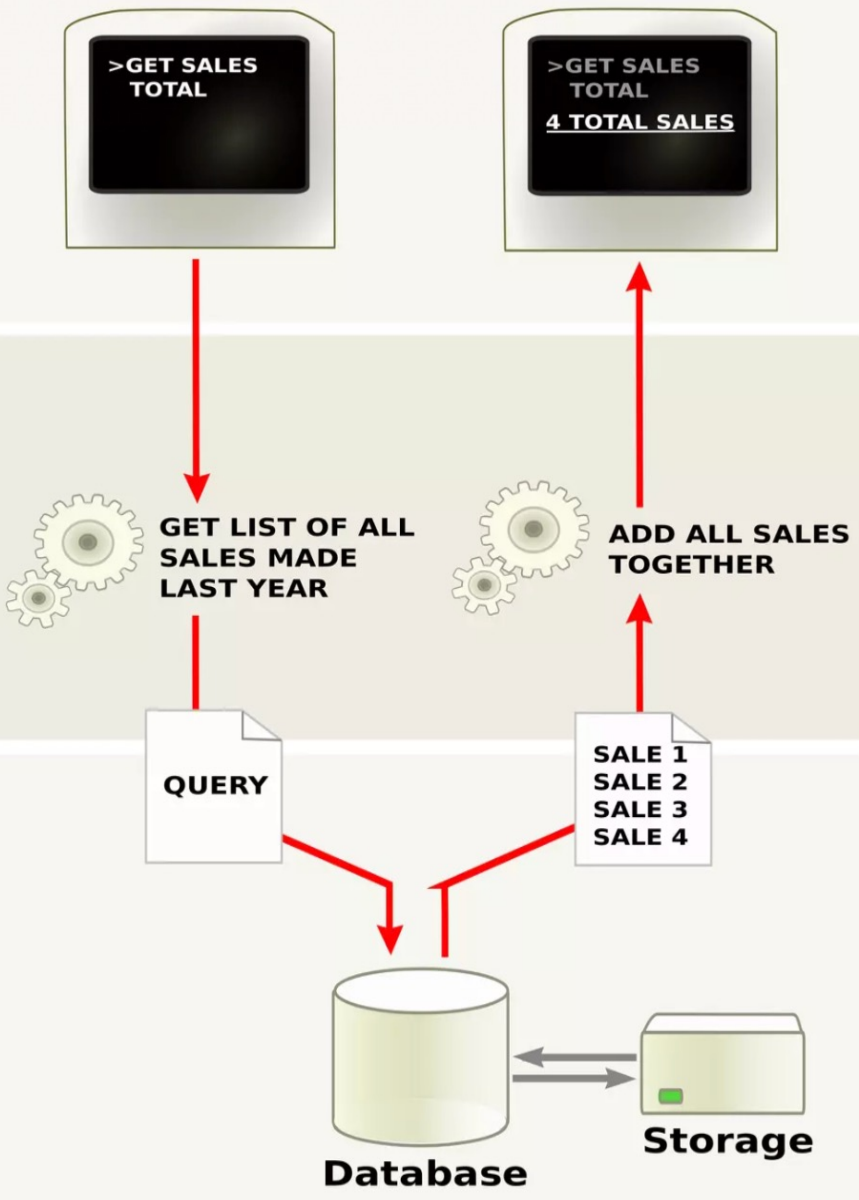
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

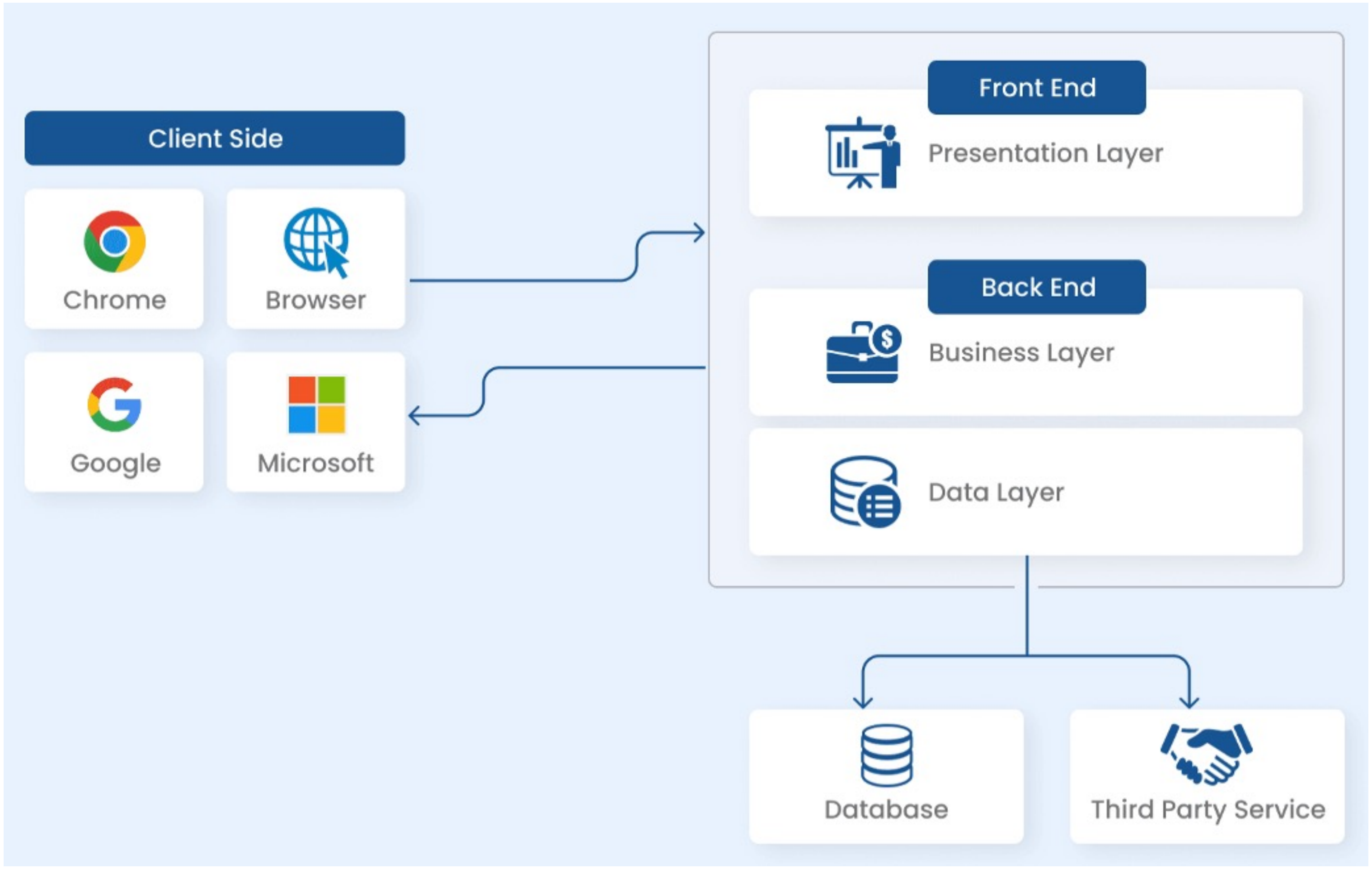
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

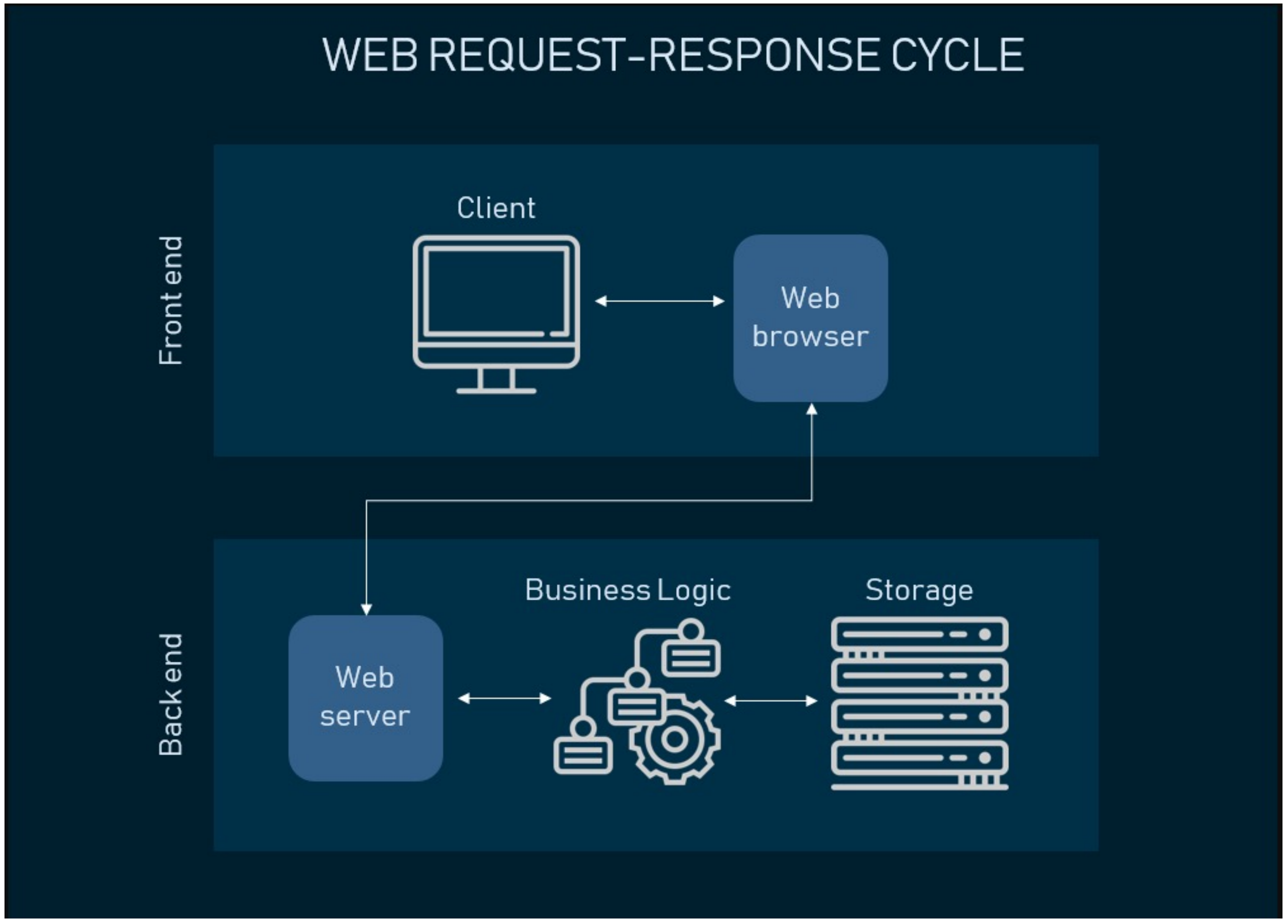
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



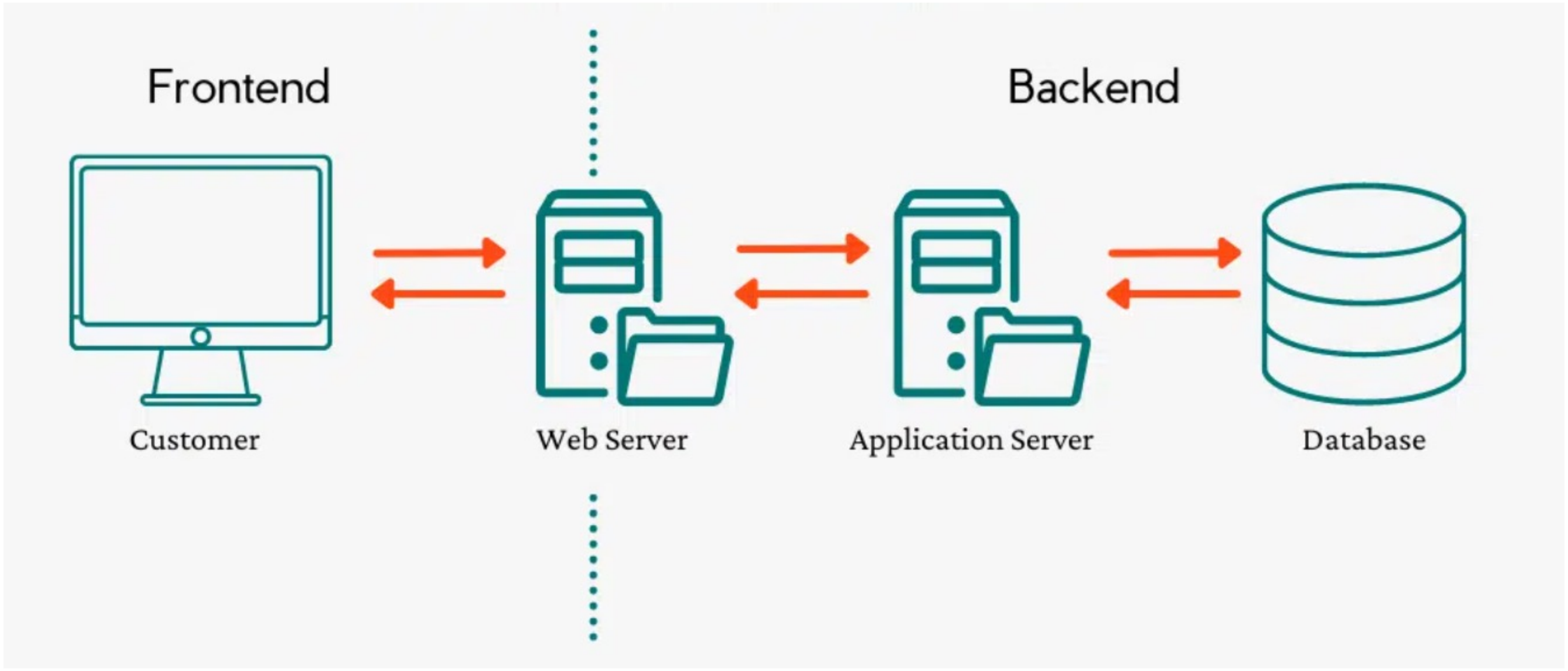
Standard Web application archi



Web request-response cycle



Frontend vs Backend



https://elevatex.de/blog/it-insights/frontend-vs-backend-vs-fullstack-differences/

Frontend vs Backend

» **Frontend:** It is the presentation layer, it deals with interactivity and user experience; it includes design, menus, posts, pages, media, comments and navigation; it concerns Web design.

- The frontend simplifies and abstracts data and processes of the backend by representing and visualizing them as user-friendly accesses.
- It can, for example, be a graphical user interface (GUI).
- This applies equally to software applications, websites on the Internet, and computer games.

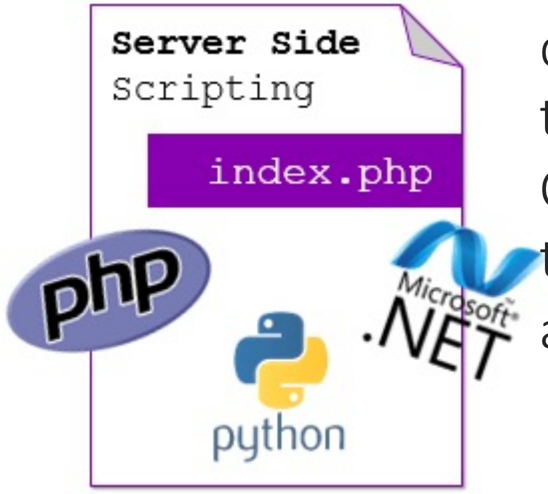
https://elevatex.de/blog/it-insights/frontend-vs-backend-vs-fullstack-differences/

Frontend vs Backend

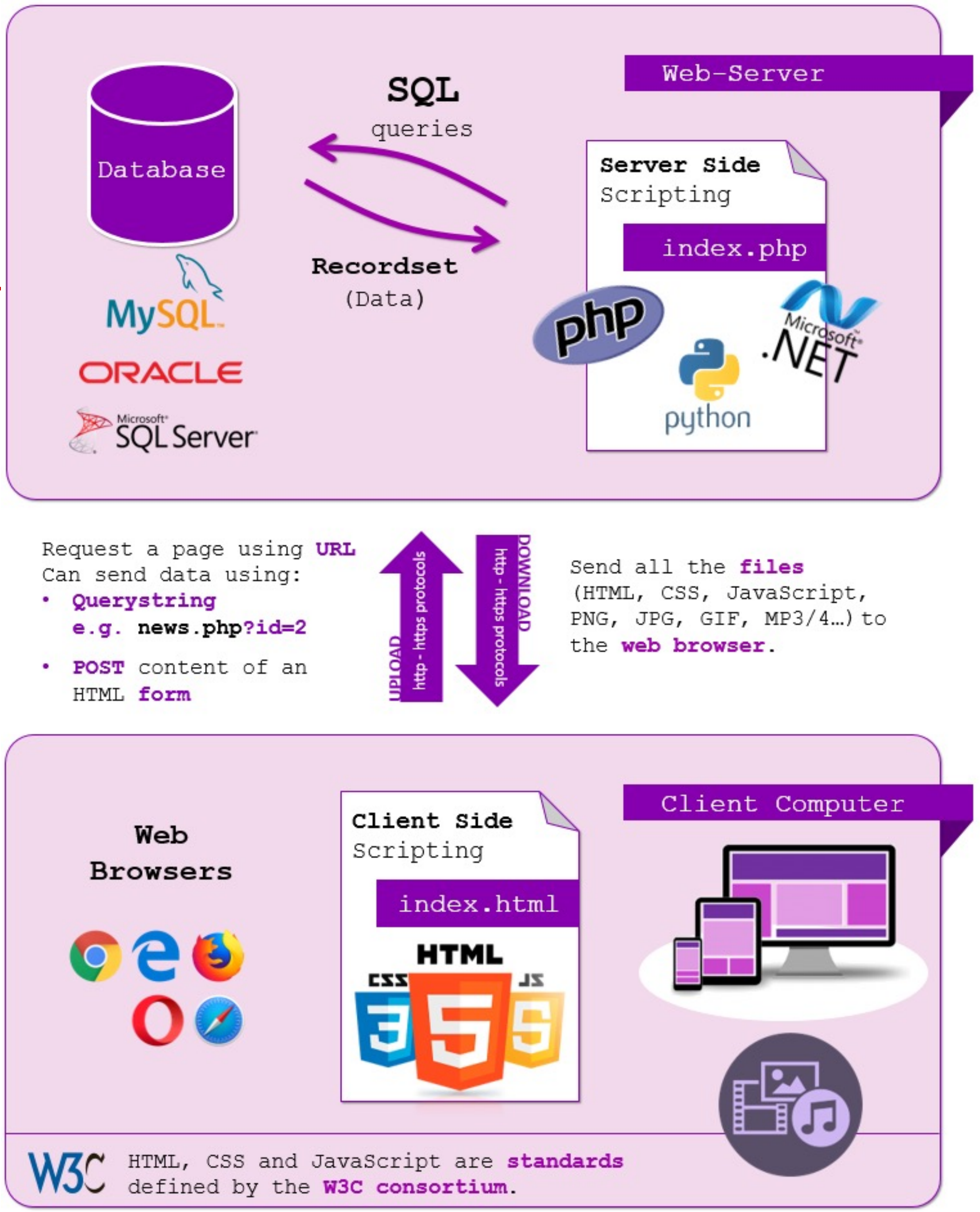
- Backend: It is responsible for all processes that run in the background of a website or software. This applies, for example, to servers, databases, or saved data
  - In the web, the Internet plays a crucial link between the backend and the frontend
  - *Example: Private photo albums on a cloud*
    - The frontend is the part the user interact with
    - The backend ensures that the photo albums are saved on a server. The user can access the photo albums over the Internet at any time

https://elevatex.de/blog/it-insights/frontend-vs-backend-vs-fullstack-differences/

Client-Server Technologies in a Web-Based Application



- Client-side technologies:
  - HTML, CSS, JavaScript
- Server-side technologies:
  - PHP, ASP, .Net, C#, Python, etc.
- Database to store and retrieve information:
  - MySQL, Oracle, SQL Server
- Server-side scripts interact with the database using SQL queries to select, update, insert or delete data. They then combine this data with HTML tags to create HTML, CSS and JavaScript code that is then sent to the client computer to be viewed in the web-browser as a static web page would be viewed



Request a page using **URL**
Can send data using:
- **Querystring**
  e.g. news.php**?id=2**
- **POST** content of an
  HTML **form**

Send all the **files** (HTML, CSS, JavaScript, PNG, JPG, GIF, MP3/4…) to the **web browser**.

HTML, CSS and JavaScript are **standards** defined by the **W3C consortium**.

https://www.101computing.net/client-server-technologies-in-a-web-based-application
/