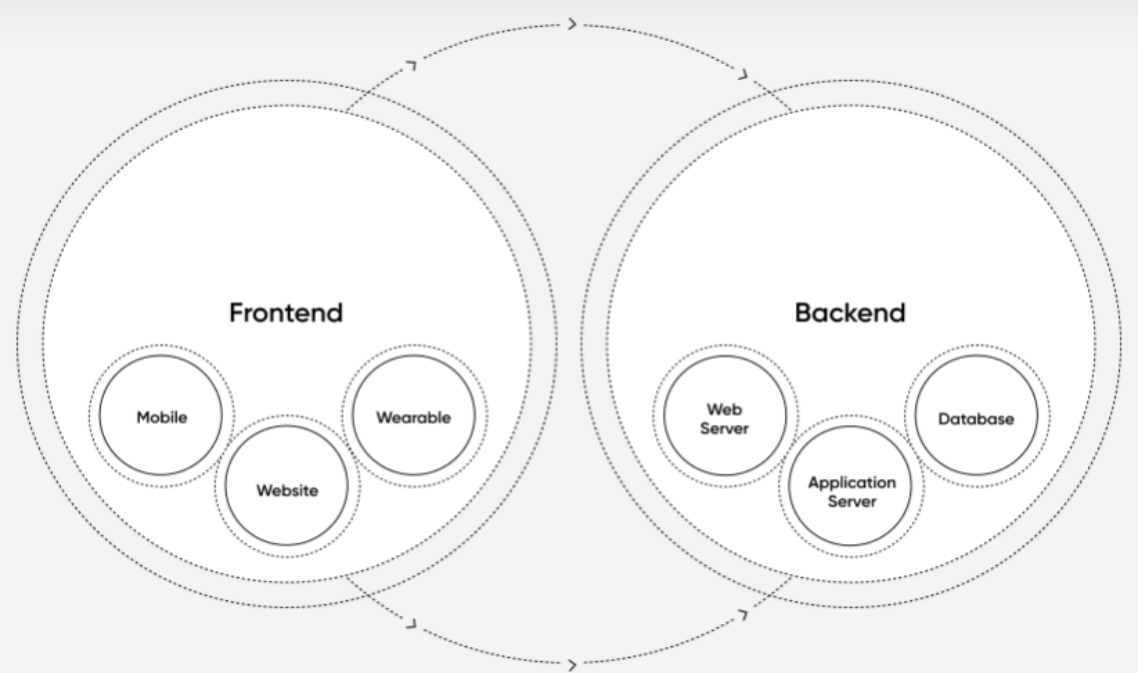
BACK-END SOFTWARE DEVELOPMENT

ECC 811 – SOFTWARE ENGINEERING

Wednesday, July 9, 2025

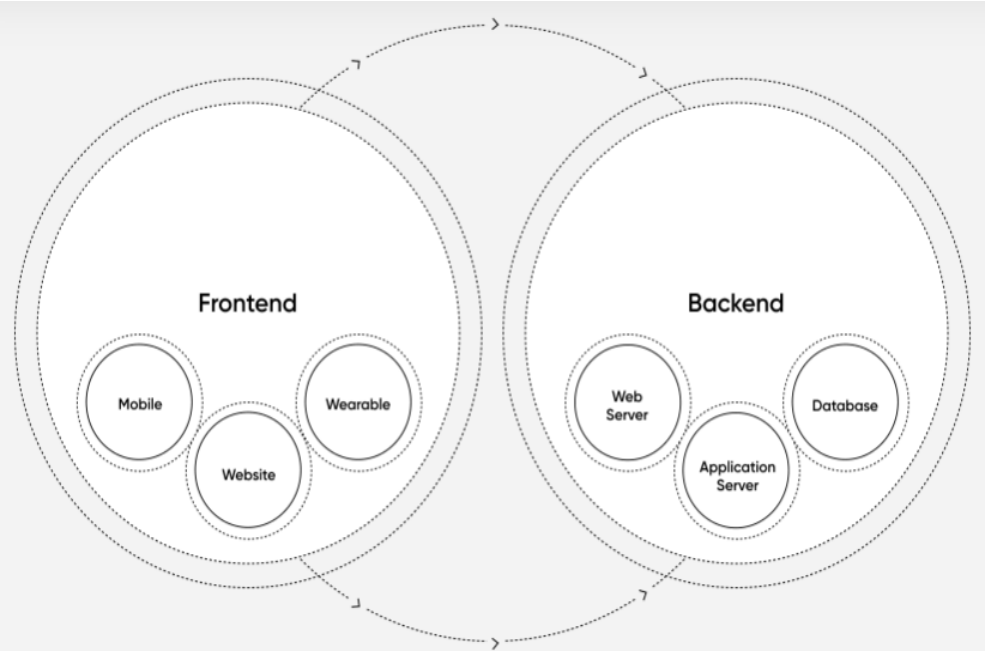
WHAT IS BACK-END SOFTWARE?

1. **Back-end Software** refers to parts of a computer application or a program's code that allow it to operate and that cannot be accessed by a user.
2. Most data and code are stored and accessed in the back-end of a computer system. Typically, the code is is developed in one or more programming languages, e.g Python, PHP, C++, Java, Node.js.



WHY BACK-ENDS ARE NECESSARY?

1. **Security:** SOME part of our code needs to be “trusted”
   a) Validation, security, etc. that we don’t want to allow users to bypass
2. **Performance:**
   a) Avoid duplicating computation (do it once and use cache)
   b) Do heavy computation on more powerful machines
   c) Do data-intensive computation “nearer” to the data
3. **Compatibility:** Have software that runs on many user platforms.



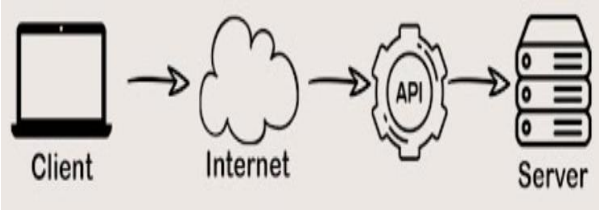
BACK END Vs FRONT END

FRONT END	BACK END
Advantages • Very responsive (low latency) **Disadvantages** • Security • Performance • Unable to share between front-ends	**Advantages** • Easy to refactor between multiple clients • Logic is hidden from users • Good for security, compatibility, and intensive computation **Disadvantages** • High latency (Interactions require a round-trip to server and back)

ESSENTIAL SKILLS FOR BACK-END DEVELOPERS

Essential Skills For Back-end Developers are

1. **Backend Programming Languages** such as Object-Oriented Programing Languages (Python, Java and .NET)
2. **Functional Programming Languages** such as Matlab, SQL
3. **Databases** stores the data and site content in a way that makes it simple to recover, arrange, alter, and store information.
4. **Server** are computer system that receives requests for web-based files which includes HTML, CSS, Javascript, etc., and sends those replies to the client. Some examples of servers are Apache, Nginx, IIS servers.
5. **APIs (Application Programming Interface)** are a set of rules and definitions that allow different clients, software, or services to communicate with each other over the internet. The server is the one that provides the API, and the client is the one who consumes it.



RESPONSIBILITIES OF BACK-END DEVELOPERS /01

1. Coordinate with frontend developers and develop server-side algorithms to transfer data efficiently to the client-side web applications.
2. Cross-collaborate with PM(Project Manager), and QA (Quality Assurance) engineers to optimize and develop a quality User Experience (UX).
3. Make sure the application is fast and performs the same whenever the user traffic changes.
4. Collaborate with the stakeholders to understand their particular needs, then translate those into technical requirements and come up with the most effective and efficient technical solution.
5. Optimize the applications for increasing response time and efficiency.

RESPONSIBILITIES OF BACK-END DEVELOPERS /02

6. Analyze the requirements and goals, handle bugs and errors, and come up with efficient and fast solutions.
7. Interact with the database for storing data.
8. Manage and develop APIs (they help two software communicate with each other over the internet) that are executed across the devices.
9. Building the architecture of the system while keeping in mind the scalability, speed, and stability of the applications.
10. Implementing security structures and their best practices.
11. Writing reusable code and libraries for future implementation.