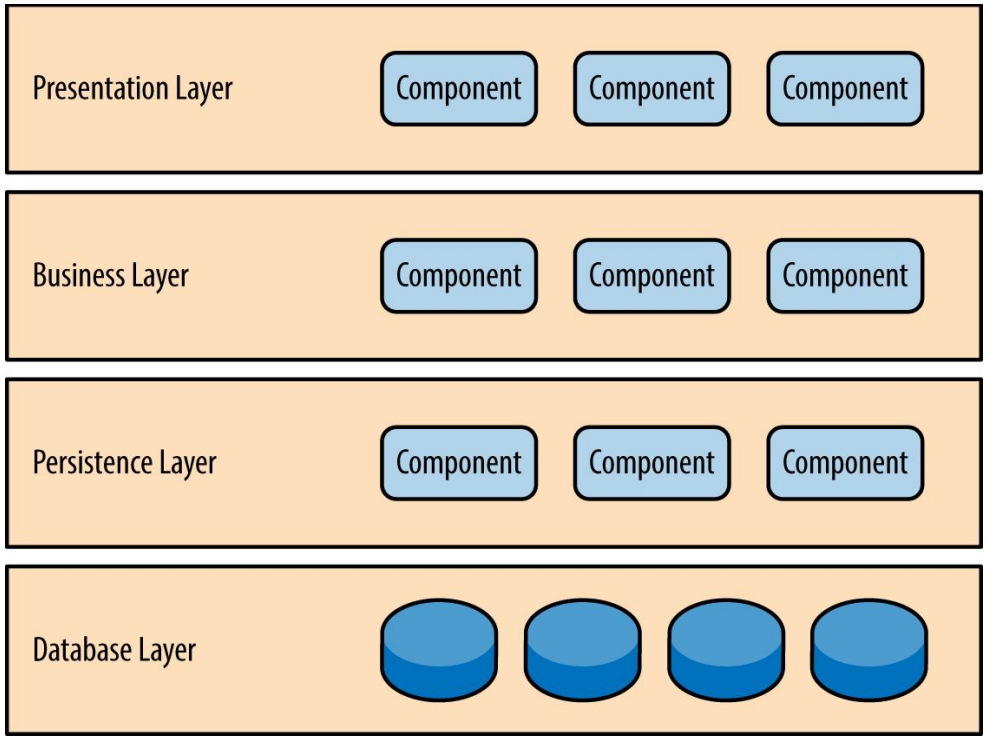
Layered Architecture

Definition

- The layered architecture style is one of the most common architectural styles. The idea behind Layered Architecture is that modules or components with similar functionalities are organized into horizontal layers. As a result, each layer performs a specific role within the application.
- The layered architecture style does not have a restriction on the number of layers that the application can have, as the purpose is to have layers that promote the concept of separation of concerns. The layered architecture style abstracts the view of the system as a whole while providing enough detail to understand the roles and responsibilities of individual layers and the relationship between them.
- **Vocabulary for components and connectors:**
    - The connector between each layer can be a function call, a query request, a data object or any connector that conveys request or information.
    - The naming of layers is quite flexible, but usually a presentation layer, a business layer and a physical layer are always present.
    - The presentation layer contains all of the classes responsible for presenting the UI/visualization to the end-user and handling bowser communication logic. Ideally, this is the only layer that customers interact with.
    - The business/logic layer contains all the logic that is required by the application to meet its functional requirements. This layer usually deals with data aggregation, computation and query requests. This is where the main logic of the application is represented.
    - The data/physical + persistence layer is where retrievable information is stored. This layer consists of both logical and physical aspects. While the logical schema specifies conceptual model of data, the physical schema implements the logical model into physical database platform.

Topological constraints

- The architecture itself is a topological constraint as it is a specific way of organizing the software systems.
- Connecters only interact with two layers, and usually, no skipping of layers is allowed.
- Communication is usually between one user and one interface/system.
- Data flow goes two ways (user to system, and back).



[1]

Applicable problems

- Software that requires separate layers of processing and security.
- Data driven software, CRUD applications.

Resilience to change

- Since the separation of concerns is the main property of the architecture, each layer of software has a specific function. This makes updating individual layers easy, and also allows teams to separate workloads.

Negative behaviours

- Layered software usually results in tightly coupled software components, and monolithic applications.
- Security can become an issue if bypassing layers is allowed. The Business Layer usually acts as an integrity check for passing data.
- If not properly designed and managed, communication between layers can become complicated.

[1] Source: https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html

Supported NFPs

- Easy to implement/test: Layered architecture is one of the easiest structures to implement. Since every layer has a specific function, testing is easy since layers can be mocked
- Flexibility: In some sense, any software can be abstracted into layers. Layered architecture is very open and broad in its implementation, thus leading to many practical applications.
- Security: Security can be implemented at every layer. If layers are skipped, extra precautions must be made.

Inhibited NFPs

- Low scalability: Layered architecture results in a rigid structure of software implementation and highly coupled software groups, resulting in a system that is hard to scale and hard to update. A change to a single layer must be verified that it does not crash the entire system.
- Low Performance: Data must travel through every layer and processed, slowing down the performance.

Comparison with other architectures

- Client-server, layered, and pipe and filter architectures are similar in their objective.
    - Client-server can be thought of as a variation of layered architecture with two layers.
    - Pipe and filter only allows unidirectional flow of information, whereas client-server and layered architectures allow bidirectional flow.

Credit:

Students:

Derma team (Yihan Duan, Lishi Wang Stephen Lau, Omar Qadri)

Instructor and TAs:

Mei Nagappan, Achyudh Ram Keshav Ram, Aswin Vayiravan, Wenhan Zhu