ONE PAGE SUMMARY

IT & Software › Software Architecture › Layered Architecture

Introduction to Layered Architecture

Layered Architecture: Organizing Software for Clarity and Modularity

DEFINITION

> Layered architecture is a software design pattern that separates concerns into distinct layers. Each layer has specific responsibilities, promoting better organization and maintainability.

KEY CONCEPTS

Presentation Layer

This layer handles user interface and user experience, presenting data to users.

Business Logic Layer

This layer contains the core functionality and rules of the application, processing data.

Data Access Layer

This layer manages data storage and retrieval, interacting with databases or external services.

Separation of Concerns

Each layer focuses on a specific aspect of the application, reducing complexity.

Inter-layer Communication

Layers communicate with each other through well-defined interfaces, ensuring loose coupling.

EXAMPLES

→ Web applications with distinct UI, business logic, and database layers.

→ Mobile apps separating presentation, logic, and data management.

→ Enterprise systems using layered architecture for scalability and maintenance.

MEMORY TIPS

★ Think of layers like a cake: each layer has its own flavor and purpose.

★ Remember 'PBD' for Presentation, Business, Data layers.

★ Visualize a building: each floor serves a different function but is part of the whole.

COMMON MISTAKES

✗ Mixing responsibilities across layers, leading to tightly coupled code.

✗ Neglecting to define clear interfaces for layer communication.

✗ Overcomplicating the architecture with too many layers.

QUICK RECAP

Layered architecture organizes software into distinct layers, each with specific roles. This structure enhances maintainability and clarity, making it easier to manage complex applications.