

Introduction to Architecture Basics

Master the fundamentals of software architecture for effective system design.

DEFINITION

Software architecture is the high-level structure of a software system, defining its components and their interactions. It serves as a blueprint for both the system and the project developing it.

KEY CONCEPTS

Components

These are the individual parts of a software system, such as modules or services.

Connectors

Connectors define how components communicate and interact with each other.

Architectural Styles

These are design patterns like microservices or monolithic architecture that guide system structure.

Quality Attributes

These are non-functional requirements like performance, scalability, and security that affect design decisions.

Documentation

Proper documentation is essential for understanding and maintaining the architecture over time.

EXAMPLES

- A web application using a microservices architecture for better scalability.
- An e-commerce platform structured as a monolithic application for simplicity.
- A mobile app that communicates with a backend via RESTful APIs.

MEMORY TIPS

- ★ Remember 'C-C-A-Q-D' for Components, Connectors, Architectural styles, Quality attributes, and Documentation.
- ★ Visualize architecture as a city layout: buildings (components) connected by roads (connectors).
- ★ Think of architectural styles as different recipes for the same dish, each with unique ingredients.

COMMON MISTAKES

- ✗ Neglecting non-functional requirements during the design phase.
- ✗ Overcomplicating the architecture with unnecessary components.
- ✗ Failing to document architectural decisions and rationale.

QUICK RECAP

Software architecture is crucial for structuring systems effectively. Focus on components, connectors, and quality attributes to ensure a robust design. Always document your architecture for future reference.