Software Architecture

By Mirolim Majidov

What is software architecture?

Software Architecture

Software Architecture is the high-level design of a software system, similar to how building architecture defines the structure of a building. It involves **planning, structuring, and defining** the system's components, how they interact.

Key points:

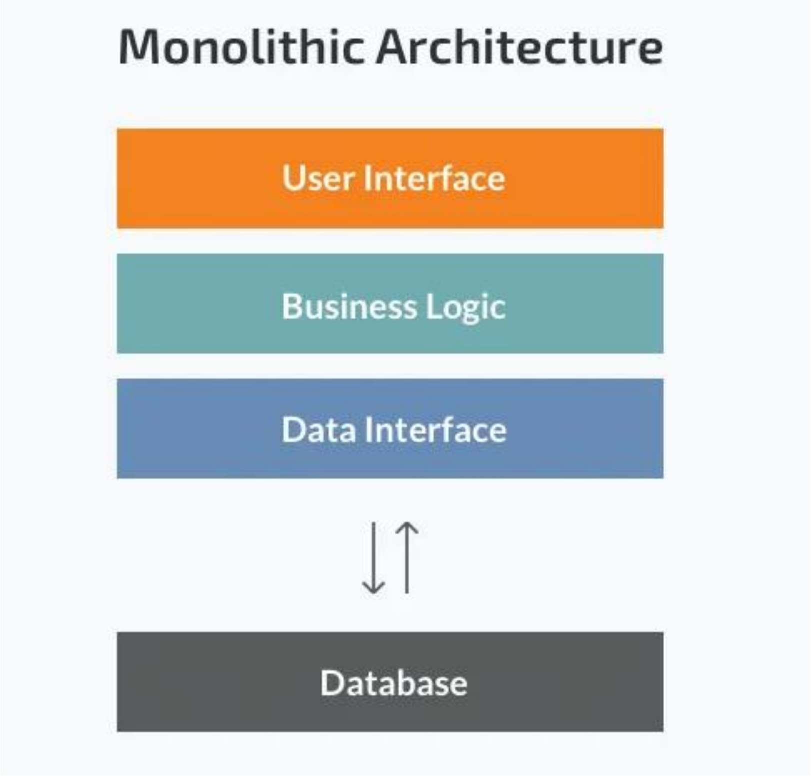
- **High-level structure** (not low-level coding details).
- **Components and their interactions** (modules, services, APIs, databases, etc.).
- **Guiding principles/patterns** (scalability, modularity, maintainability, security, etc.).
- **Constraints** (performance vs. flexibility, cost vs. maintainability).

A few types of Software Architectures

1. Monolithic Architecture
2. Clean Architecture
3. Domain-Driven Design (DDD)
4. Microservices Architecture
5. Modular Monolithic Architecture

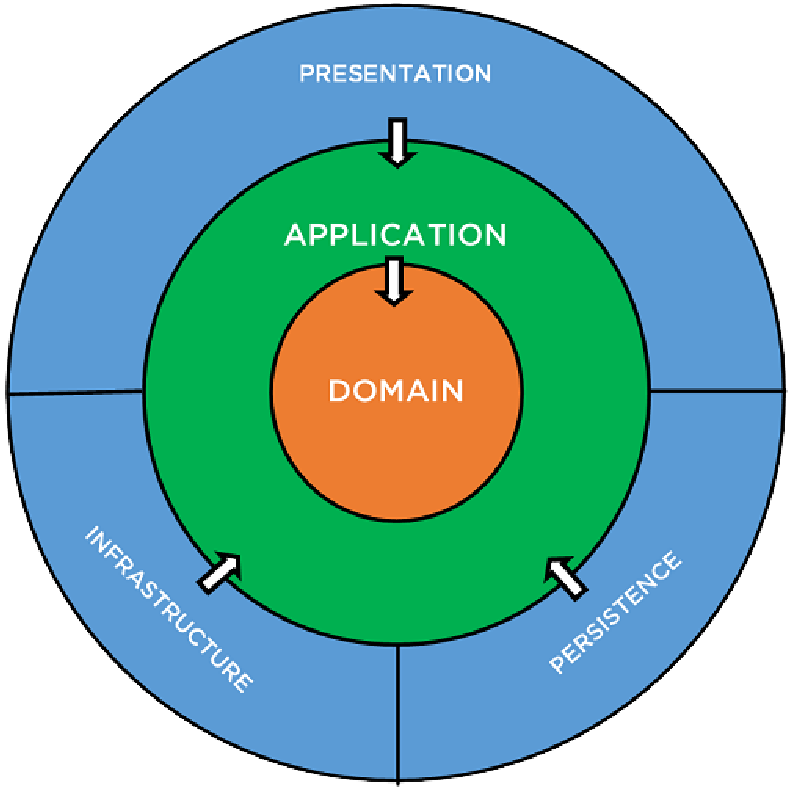
1. Monolithic Architecture

- **Definition**: A single, unified application where all components (UI, business logic, data access) are combined into one deployable unit.
- **Focus**: Simple to develop and deploy.



2. Clean Architecture

- **Definition**: A design pattern for organizing code into layers: entities, use cases, interface adapters, and frameworks/infrastructure.
- **Focus**: Internal structure of a system or service.



3. Domain-Driven Design

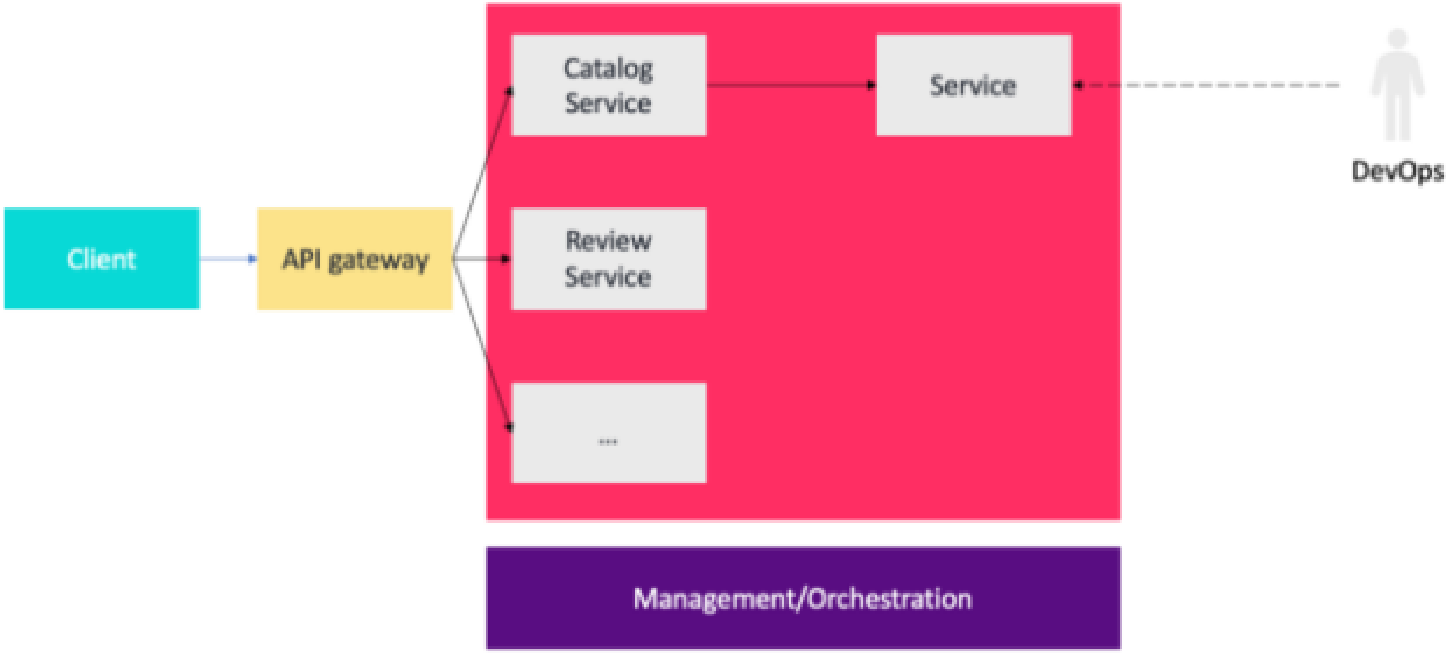
Domain-Driven Design (DDD) is an architectural and design approach primarily focused on solving complex business problems by organizing software applications around the core concepts and processes of the business domain.

Key principles and concepts of DDD:

1. Ubiquitous Language
2. Bounded Contexts
3. Entities
4. Value Objects
5. Aggregates
6. Repositories
7. Services
8. Domain Events

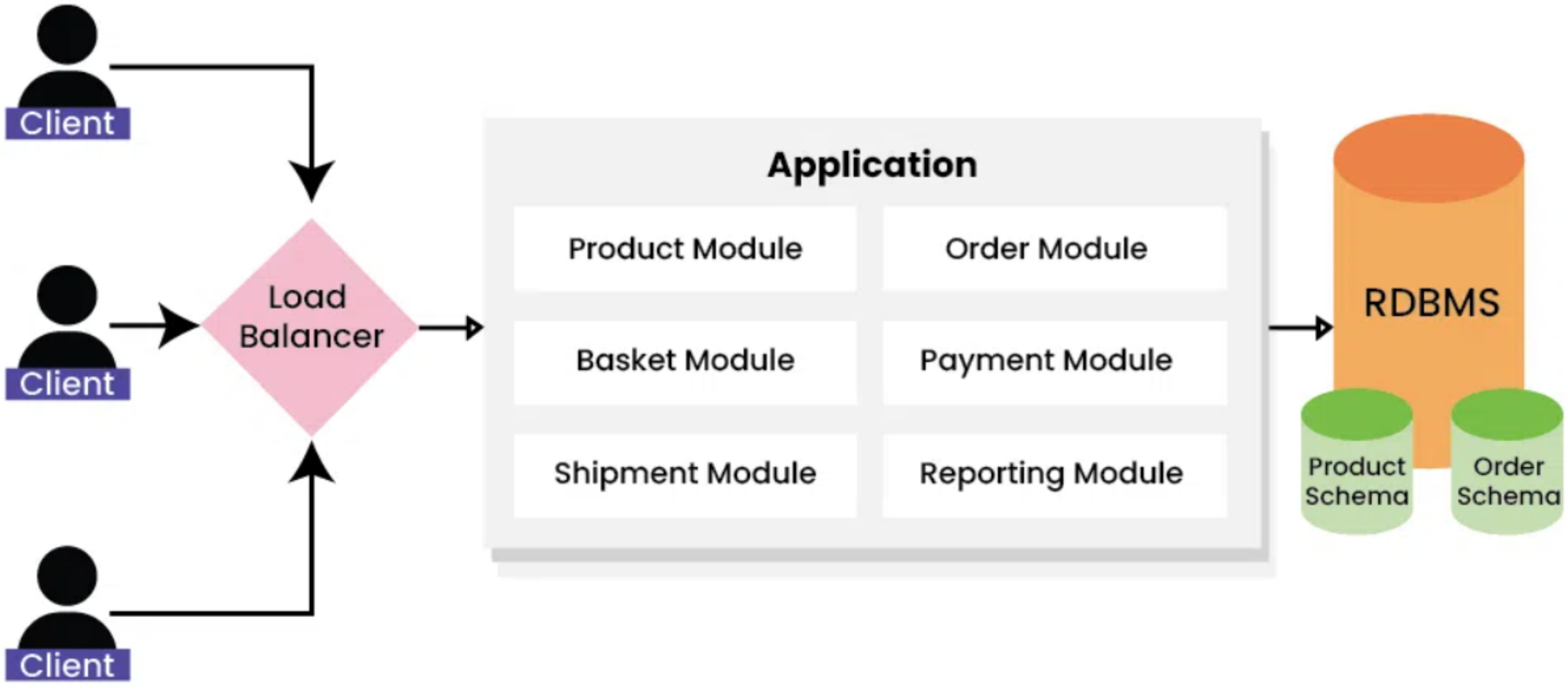
4. Microservices Architecture

- **Definition:** The application is split into independent, deployable services, each responsible for a specific business capability.
- **Focus**: System-level architecture (independent deployment, scaling, autonomy).



5. Modular Monolithic Architecture

- **Definition:** A single deployable application (like a monolith) but structured internally into well-defined modules with strict boundaries.
- **Focus**: Code modularity inside a single deployable unit.



Comparison Table

Architecture	Level	Deployment	Focus	Database	Complexity	Best For
Monolithic	System	Single	Unified app	Shared	Low-medium	Small/medium apps
Modular Monolith	System	Single	Code modularity	Shared	Medium	Medium apps, maintainability
Microservices	System	Multiple	Service autonomy	Per-service	High	Large/complex apps, scaling
Clean Architecture	Design	N/A	Internal structure	Flexible	Medium	Testable, maintainable apps
DDD	Design/Methodology	N/A	Business domain modeling	Flexible	High	Complex business domains