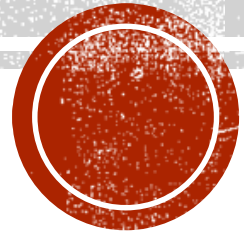
РАЗВОЈ СОФТВЕРА 2

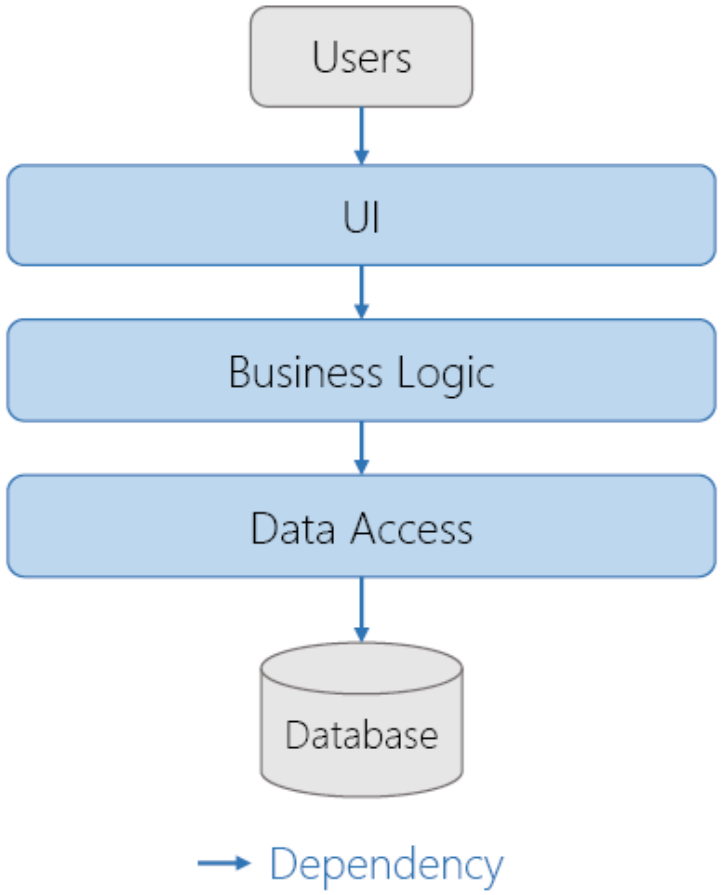
Слојевита архитектура. Апликативни слој.



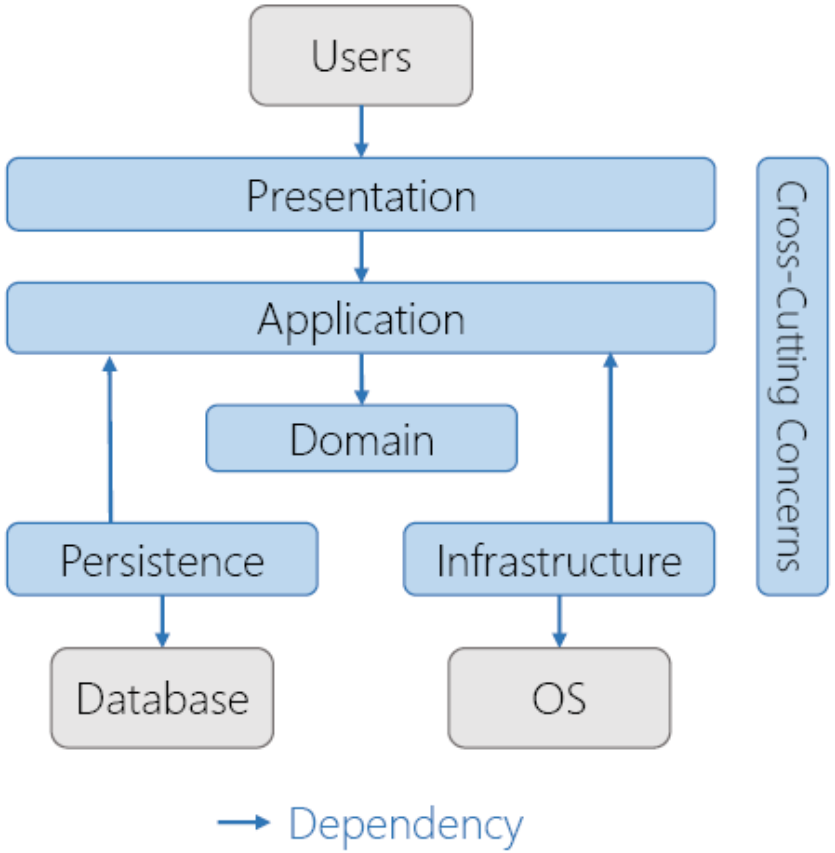
What Are Layers?

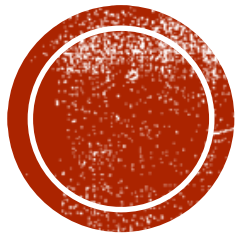
Levels of abstraction
Single-Responsibility Principle
Developer roles / skills
Multiple implementations
Varying rates of change

Classic 3-Layer Architecture



Modern 4-Layer Architecture

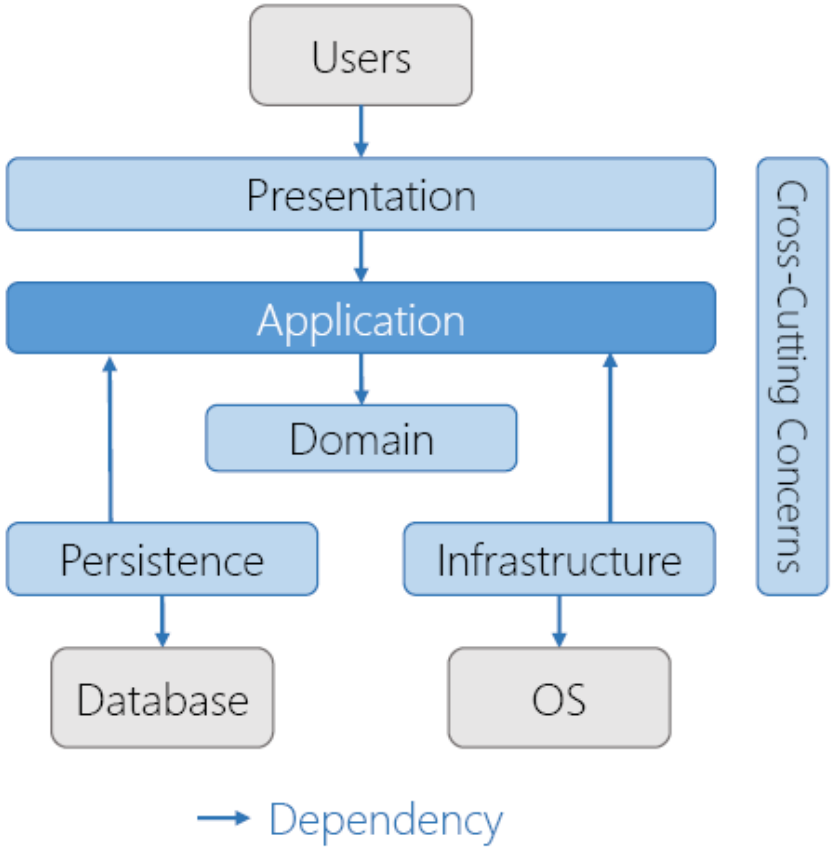




АПЛИКАТИВНИ СЛОЈ

Application Layer

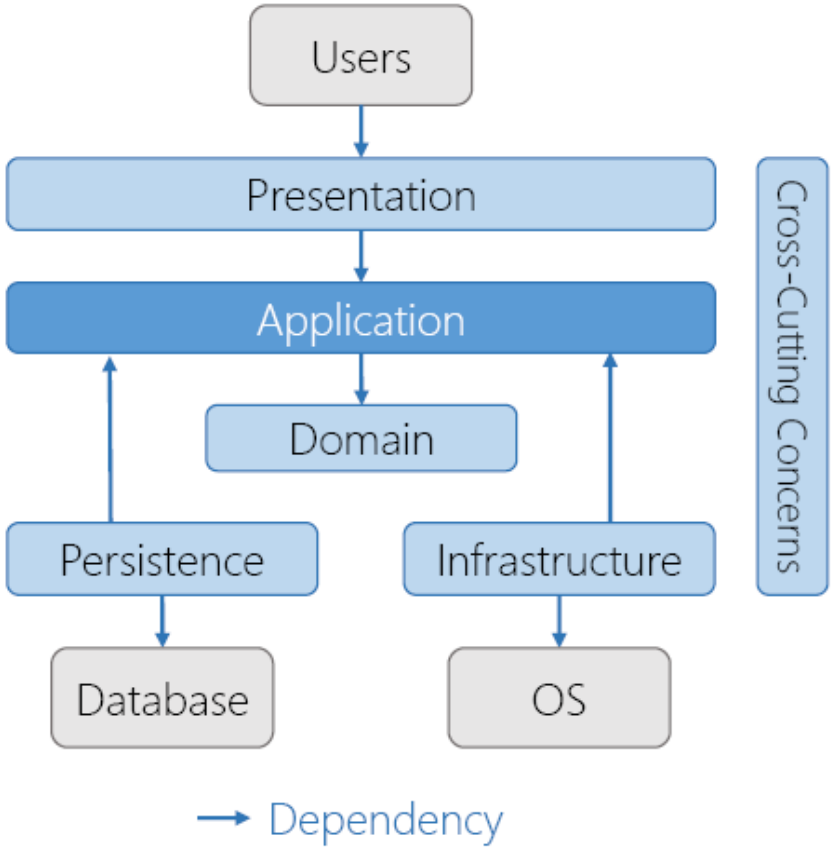
Implements use cases



Application Layer

Implements use cases

High-level application logic

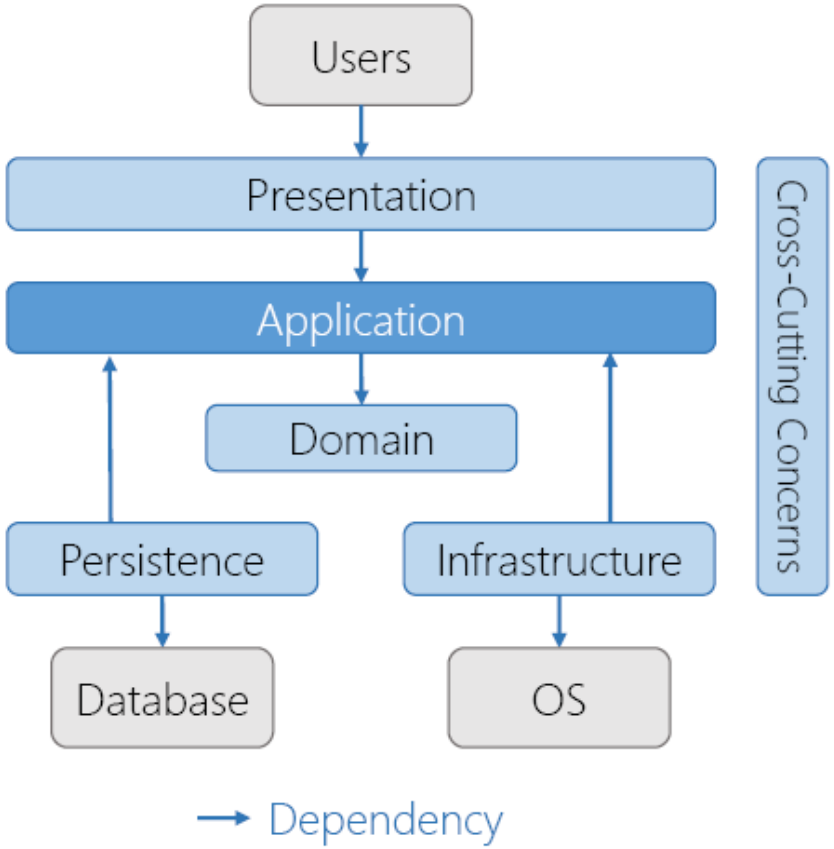


Application Layer

Implements use cases

High-level application logic

Knows about domain



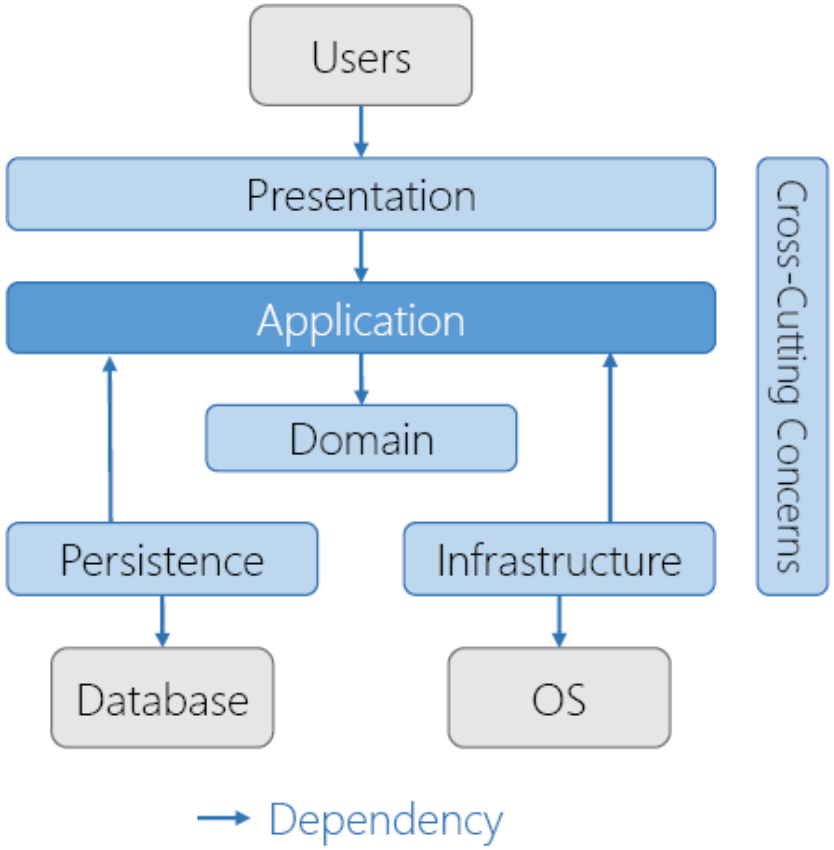
Application Layer

Implements use cases

High-level application logic

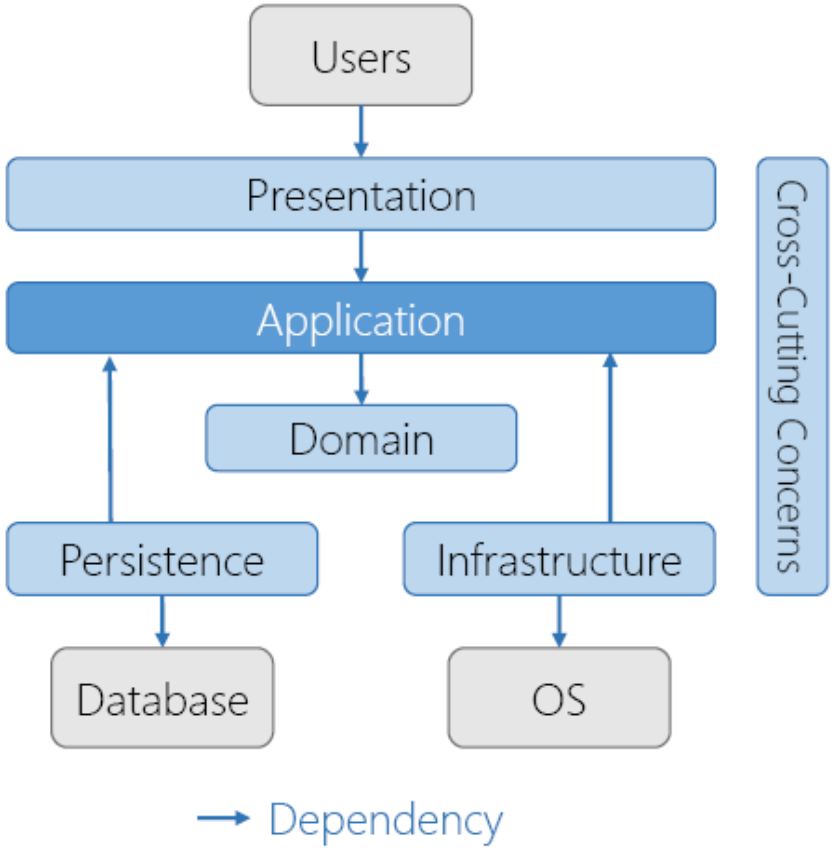
Knows about domain

No knowledge of other layers



Application Layer

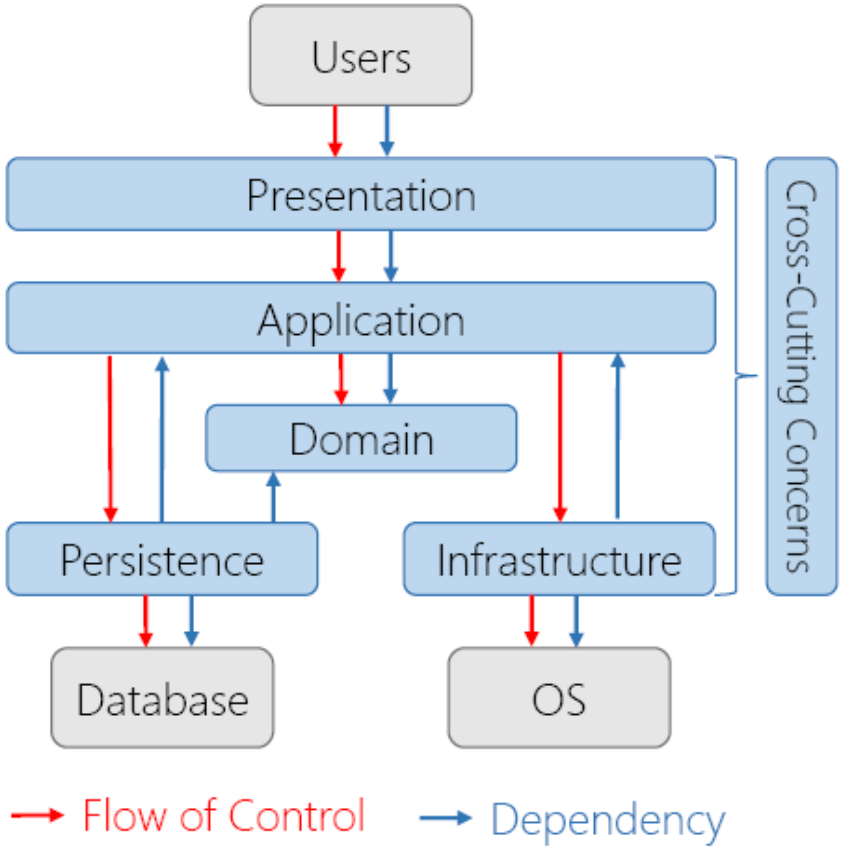
Implements use cases
High-level application logic
Knows about domain
No knowledge of other layers
Contains interfaces for details



ЗАВИСНОСТИ ИЗМЕЂУ СЛОЈЕВА

Layer Dependencies

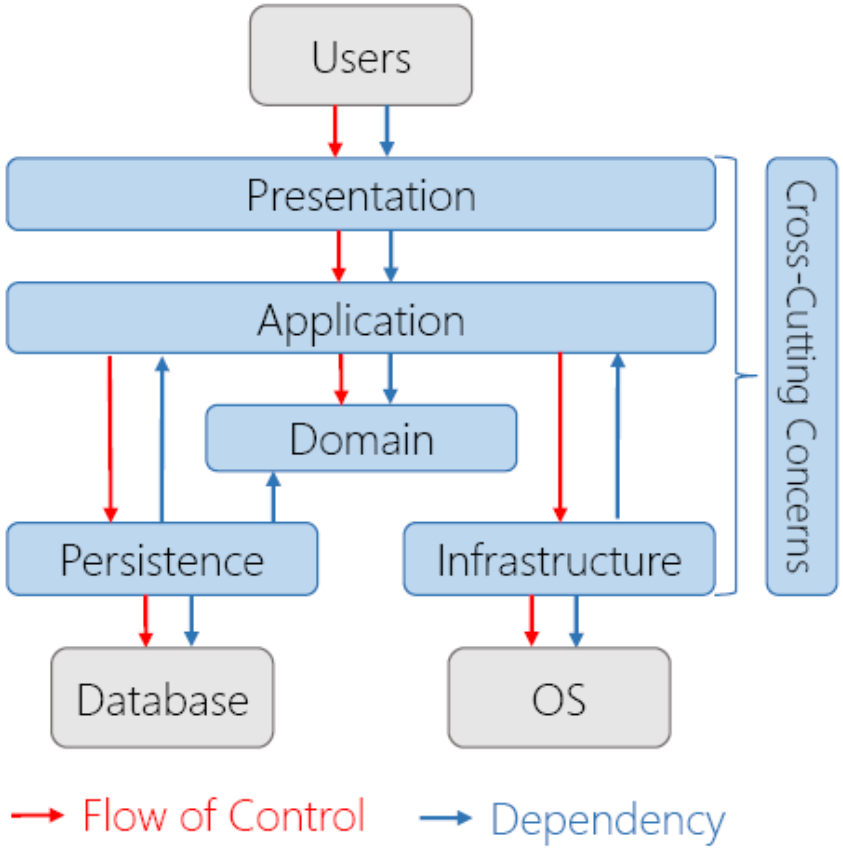
Dependency inversion



Layer Dependencies

Dependency inversion

Inversion of control

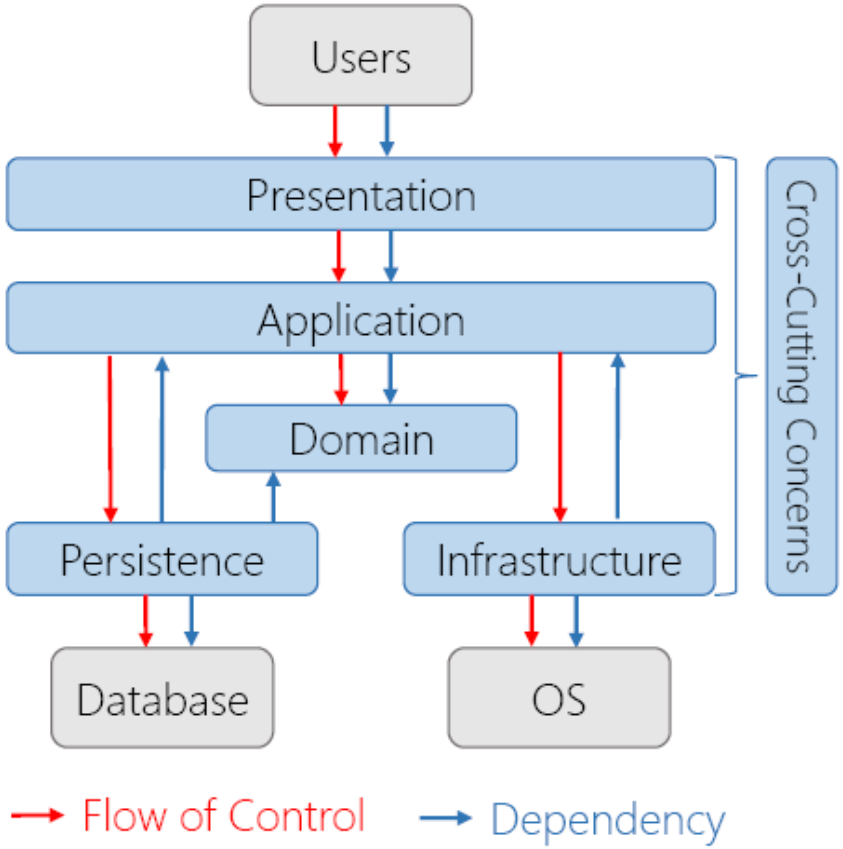


Layer Dependencies

Dependency inversion

Inversion of control

Independent deployability



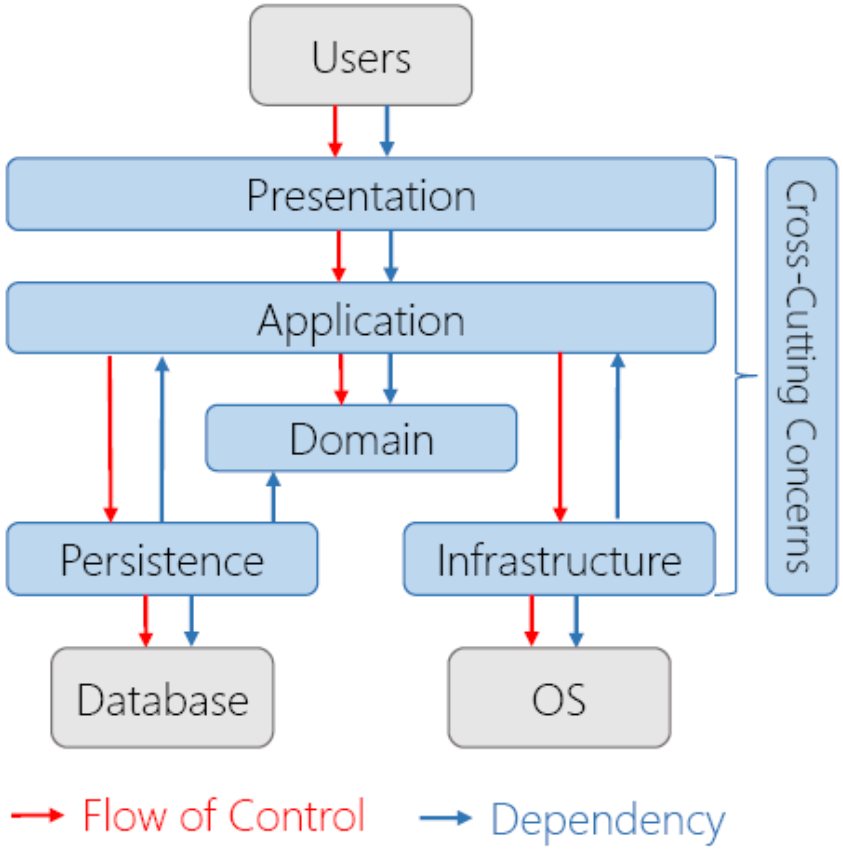
Layer Dependencies

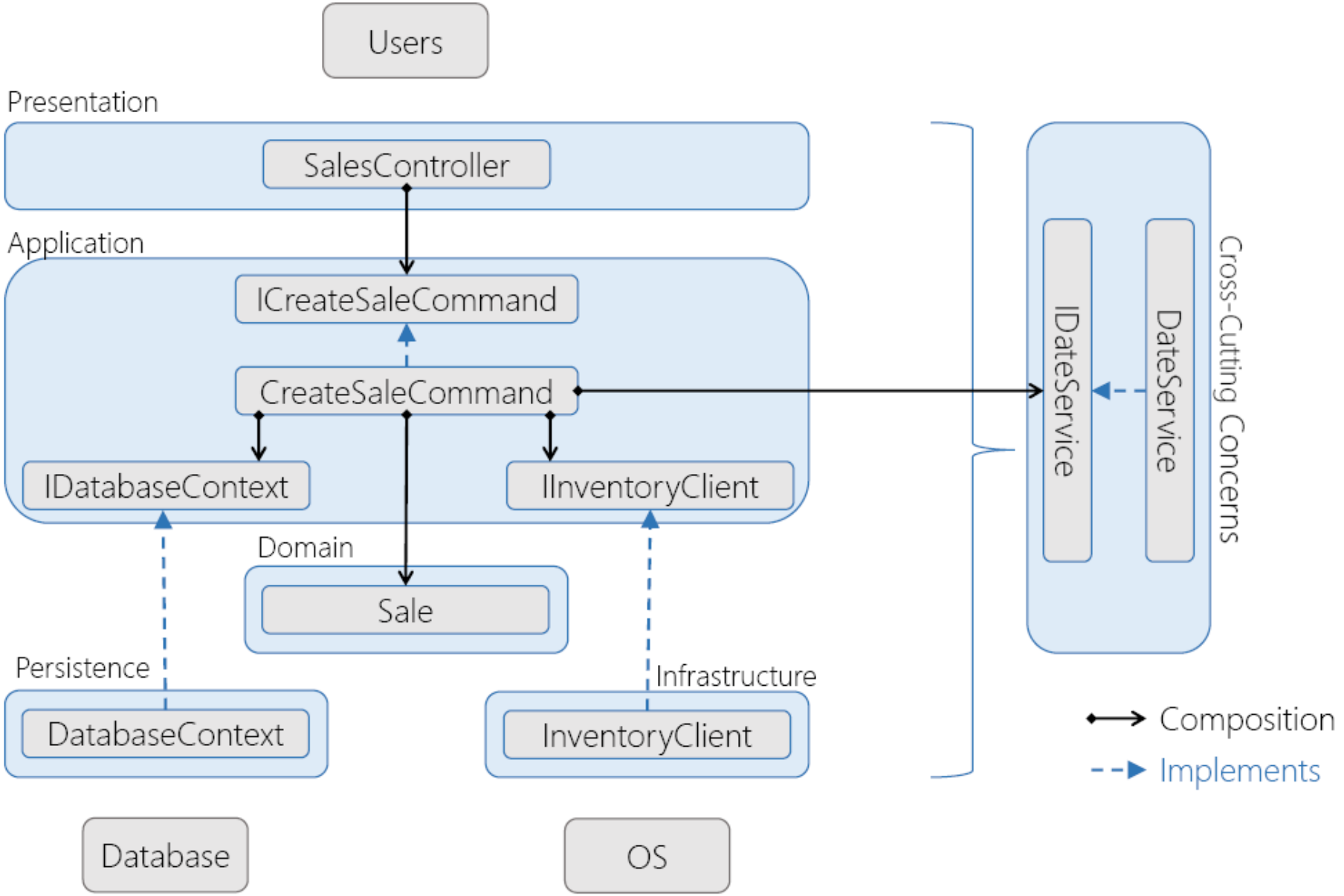
Dependency inversion

Inversion of control

Independent deployability

Flexibility and maintainability




Users
Presentation
SalesController
Application
ICreateSaleCommand
CreateSaleCommand
IDatabaseContext
IInventoryClient
Domain
Sale
Persistence
DatabaseContext
Infrastructure
InventoryClient
Database
OS
IDateService
DateService
Cross-Cutting Concerns
Composition
Implements

ПРЕДНОСТИ И НЕДОСТАЦИ КОРИШЋЕЊА АПЛИКАТИВНОГ СЛОЈА

Why Use an Application Layer?

Pros

Focus is on use cases

Easy to understand

Follows DIP

Why Use an Application Layer?

Pros

Focus is on use cases

Easy to understand

Follows DIP

Cons

Additional cost

Requires extra thought

IoC is counter-intuitive