

Application Layer



Matthew Renze

SOFTWARE CONSULTANT

@matthewrenze www.matthewrenze.com



Overview



Application Layer

Dependency Inversion Principle

Pros and Cons

Demo

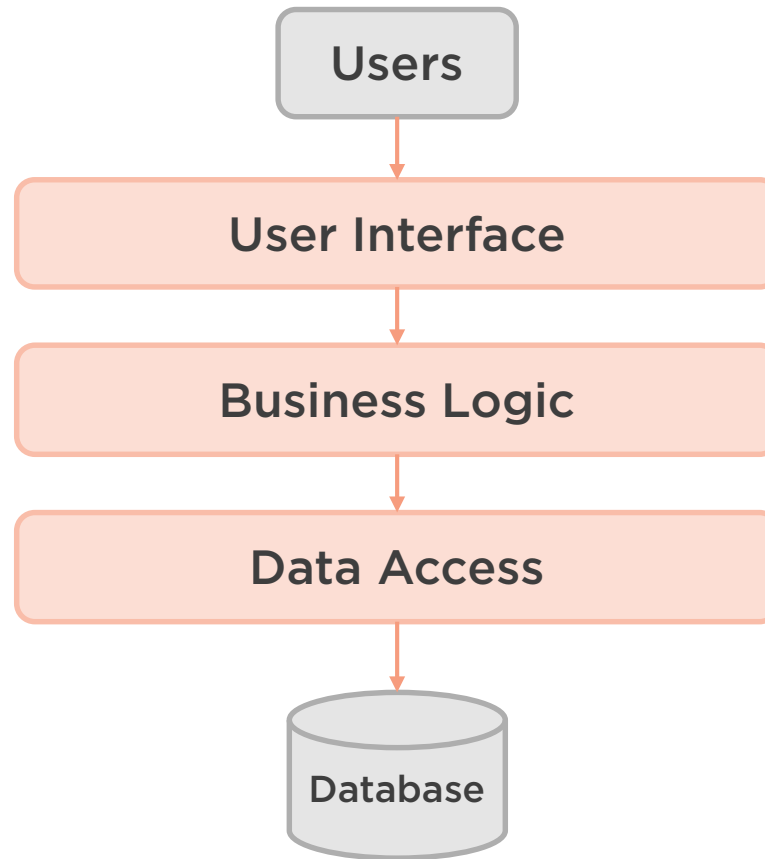


What Are Layers?

Levels of abstraction
Single-responsibility
Isolate roles and skills
Multiple implementations
Varying rates of change



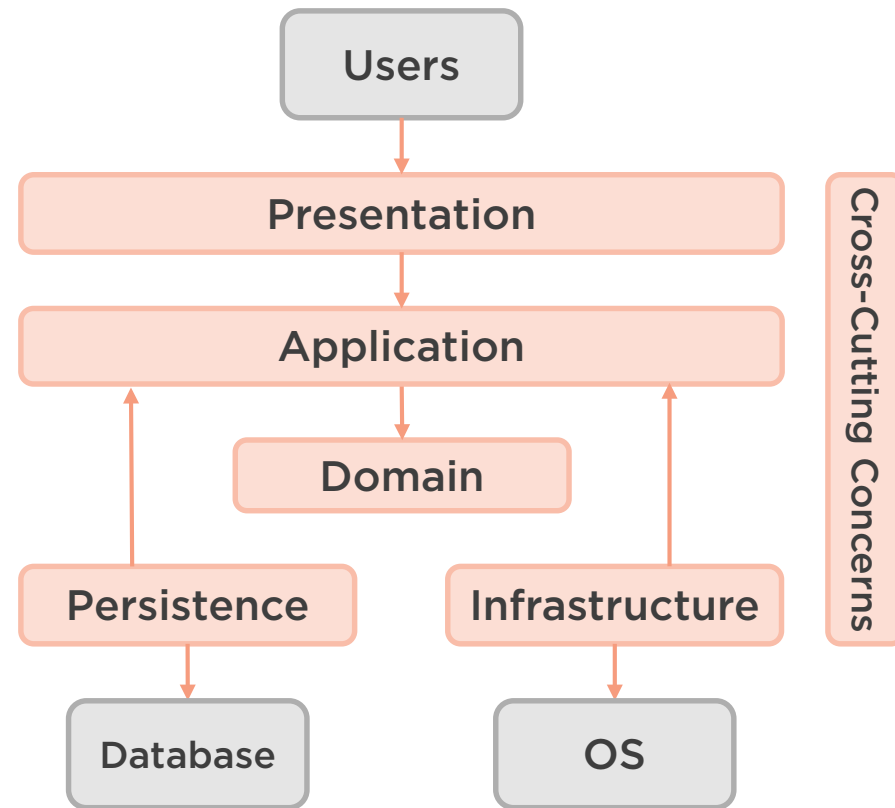
Classic Three-layer Architecture



→ Dependency



Modern Four-layer Architecture

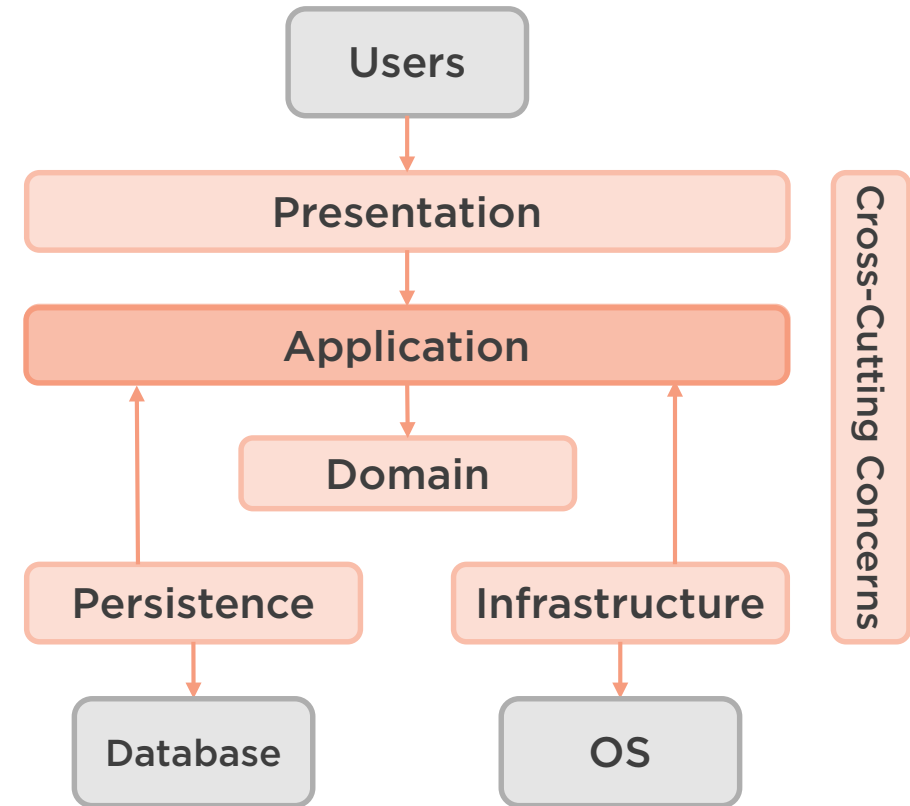


→ Dependency



Application Layer

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

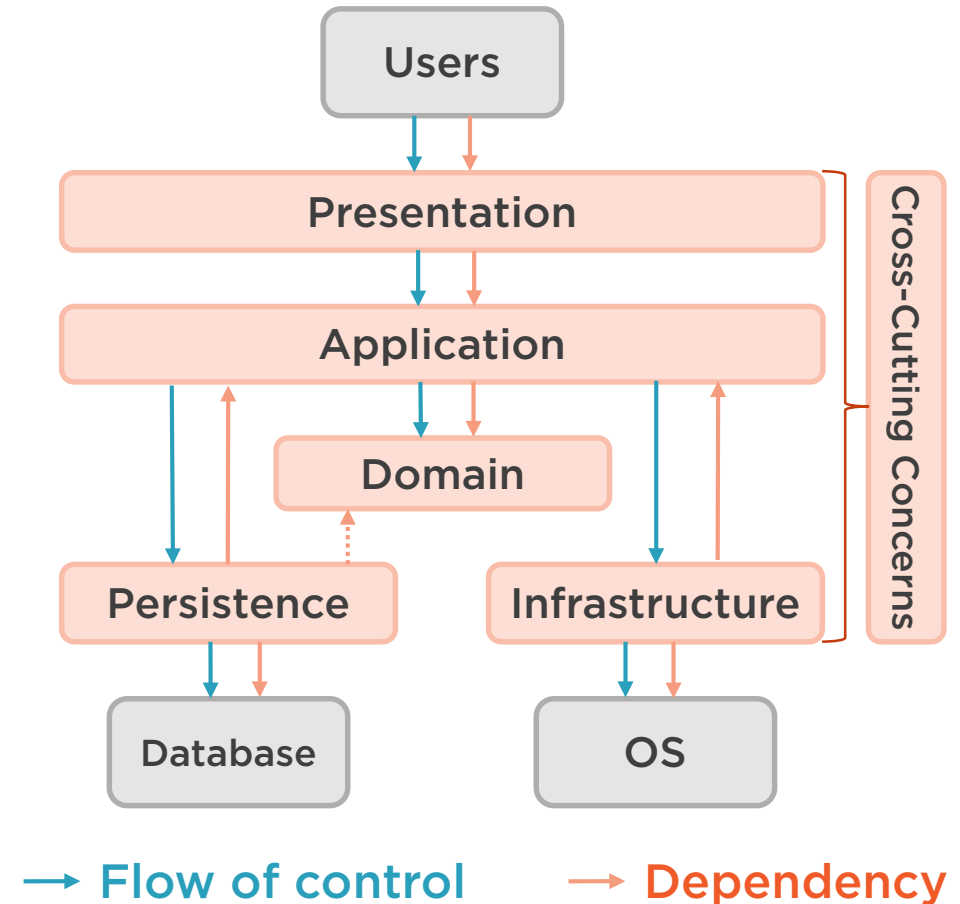


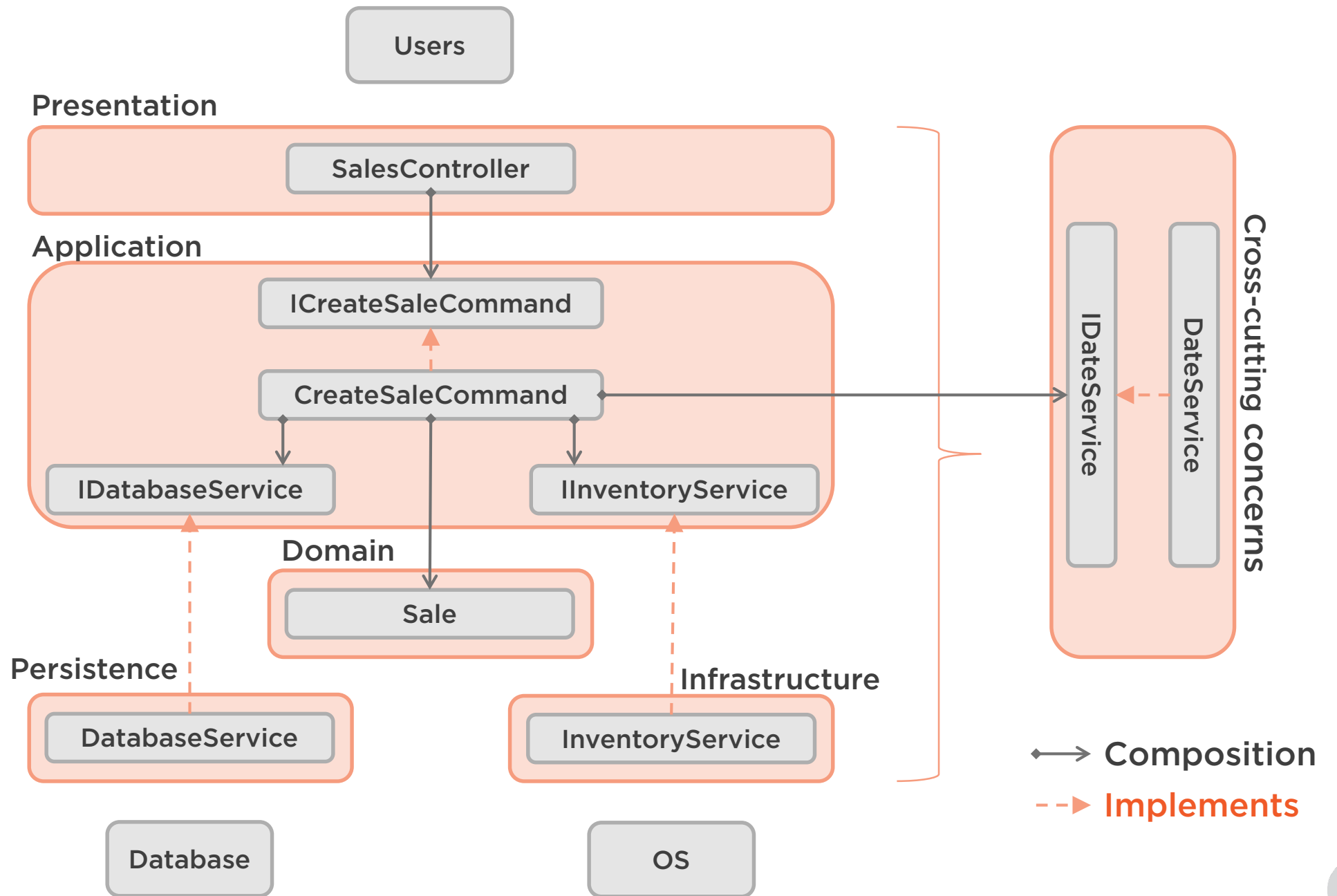
→ Dependency



Layer Dependencies

Dependency inversion
Inversion of control
Independent deployability
Flexible and maintainable





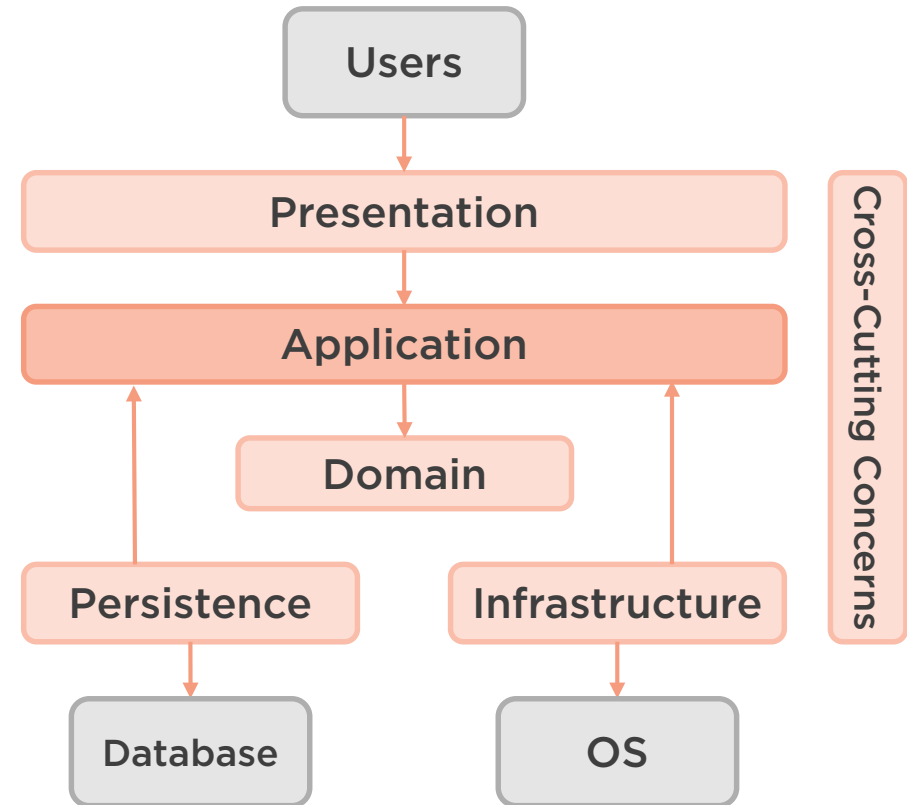
Why Use an Application Layer

Pros

Focus on use cases

Easy to understand

Follows DIP



Why Use an Application Layer

Pros

Focus on use cases

Easy to understand

Follows DIP

Cons

Additional layer cost

Requires extra thought

IoC is counter-intuitive



Set Up



Show Application Project



Show SalesController Class



Show ICreateSaleCommand Interface



Show Create Sale Model



Show CreateSaleCommand Class (Part 1)



Show CreateSaleCommand Class (Part 2)



Show IDatabaseService Interface



Show DatabaseService Class



Other Dependencies



Wrap Up



Summary



Application Layer

Dependency Inversion Principle

Pros and Cons

Demo

