Fundamentals: The Dependency Inversion Principle Part 2

Steve Smith

http://pluralsight.com/

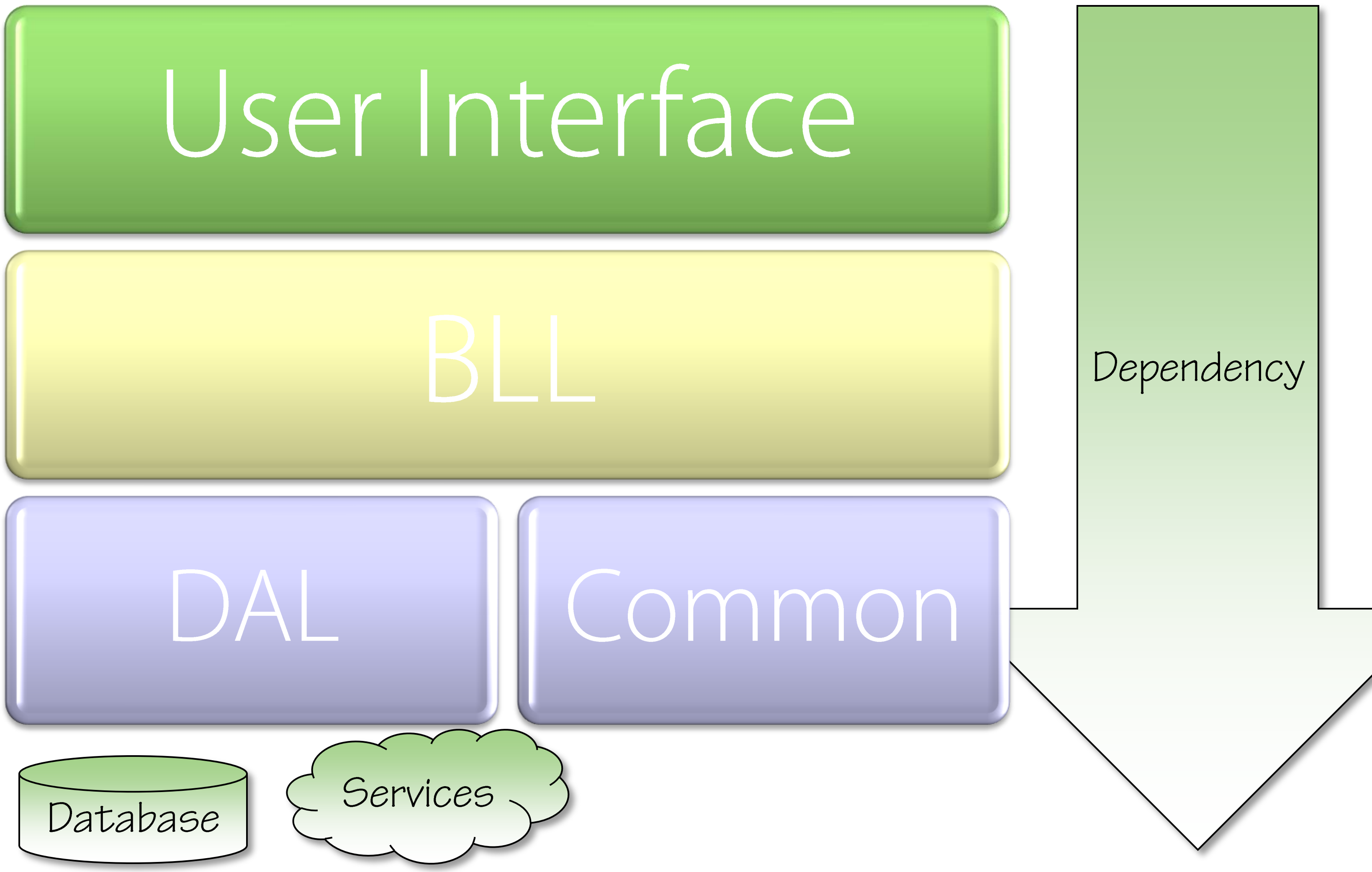
Outline

- Project Dependencies
- The Problem
- An Example
- Refactoring to Apply DIP
- Related Fundamentals

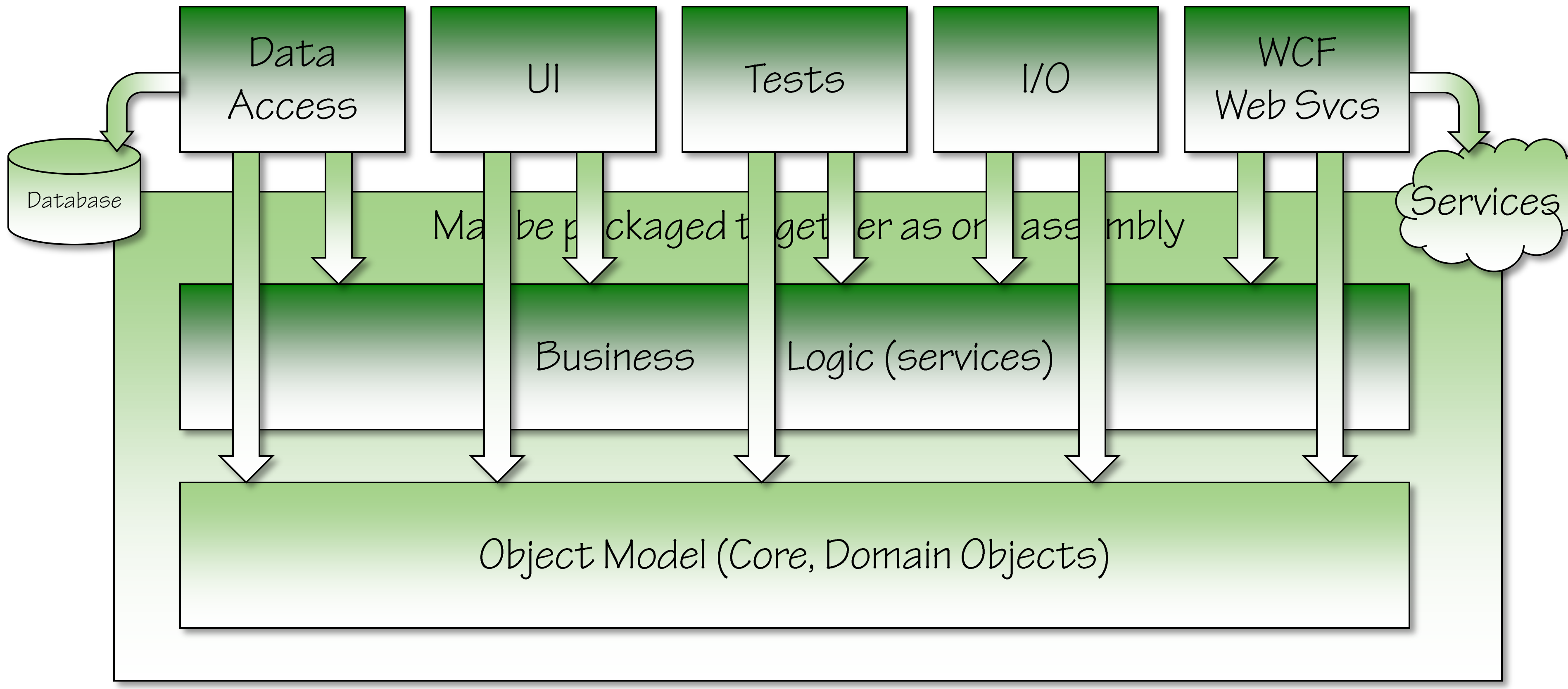
Layered / Tiered Application Design

- **Separate Logical (and sometimes physical) Layers**
  - For instance
  - User Interface (UI)
  - Business Logic Layer (BLL)
  - Data Acces Layer (DAL)
- **Supports Encapsulation and Abstraction**
  - Work at the abstraction level appropriate
  - Each level only knows about one level deep (ideally)
- **Provides Units of Reuse**
  - Lowest levels generally are most reusable

Traditional (Naïve) Layered Architecture



Inverted Architecture


Data Access
UI
Tests
I/O
WCF Web Svcs
Database
Services
May be packaged together as one assembly
Business Logic (services)
Object Model (Core, Domain Objects)

Demo

Violating DIP with Projects and Assemblies

The Problem

- **Dependencies Flow Toward Infrastructure**
- **Core / Business / Domain Classes Depend on Implementation Details**

- **Result**
  - Tight coupling
  - No way to change implementation details without recompile (OCP violation)
  - Difficult to test

Dependency Injection

- Dependency is ***transitive***
  - If UI depends on BLL depends on DAL depends on Database
    Then *everything* depends on the Database

- **Depend on *abstractions* (DIP)**

- **Package interfaces (abstractions) with the client (ISP)**

- **Structure Solutions and Projects so Core / BLL is at center, with fewest dependencies**

Demo

Refactoring to a Better Design

Summary

- **Don't Depend on Infrastructure Assemblies from Core**
- **Apply DIP to reverse dependencies**
- **Related Fundamentals:**
  - Open Closed Principle
  - Interface Segregation Principle
  - Strategy Pattern
- **Recommended Reading:**
  - Agile Principles, Patterns, and Practices by Robert C. Martin and Micah Martin [http://amzn.to/agilepppcsharp]
  - http://www.martinfowler.com/articles/injection.html

For more in-depth online developer training visit

pluralsight
see what you can learn

on-demand content from authors you trust