Breaking Tight Coupling with Dependency Injection

Loose Coupling For the Win!

What is Dependency Injection?

- Dependency Injection is a software design pattern that allows a choice of component to be made at run-time rather than compile time.
  - Wikipedia 2012

What is Dependency Injection?

- Dependency injection is a software design pattern that allows the removal of hard-coded dependencies and makes it possible to change them, whether at run-time or compile-time.
  - Wikipedia 2013

What is Dependency Injection?

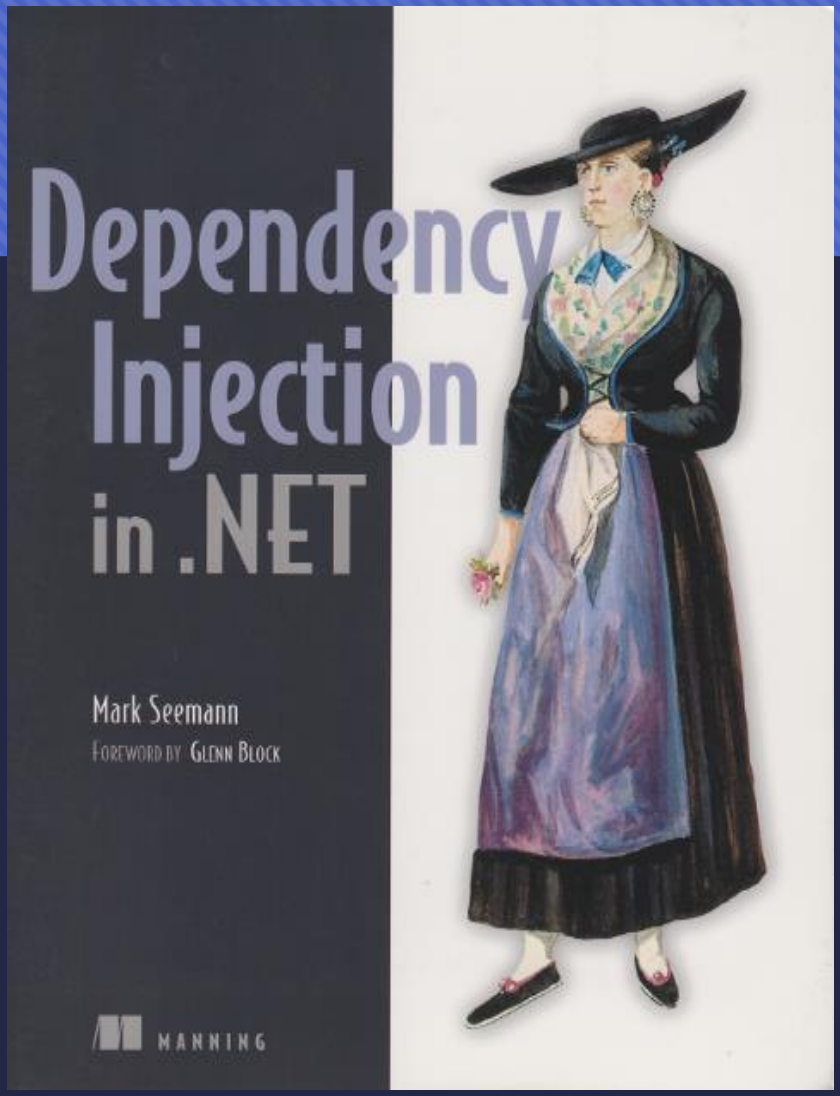
- Dependency injection is a software design pattern that implements inversion of control and allows a program design to follow the dependency inversion principle. The term was coined by Martin Fowler.
  - Wikipedia 2014

What is Dependency Injection?

- Dependency Injection is a set of software design principles and patterns that enable us to develop loosely coupled code.
  - Mark Seeman

Who is Mark Seeman?

- Dependency Injection in .NET
  - Mark Seeman



Primary Benefits

- Extensibility
- Late Binding
- Parallel Development
- Maintainability
- Testability
- Adherence to S.O.L.I.D. Design Principles

DI Concepts

- DI Design Patterns
  - Constructor Injection
  - Property Injection
  - Method Injection
  - Ambient Context
  - Service Locator
- Object Composition
- Composition Root

DI Containers

- Unity
- MEF
- Castle Windsor
- Ninject
- Autofac
- StructureMap
- Spring.NET
- and many others

Application Layers

View

- MainWindow

View Model

- MainWindowViewModel

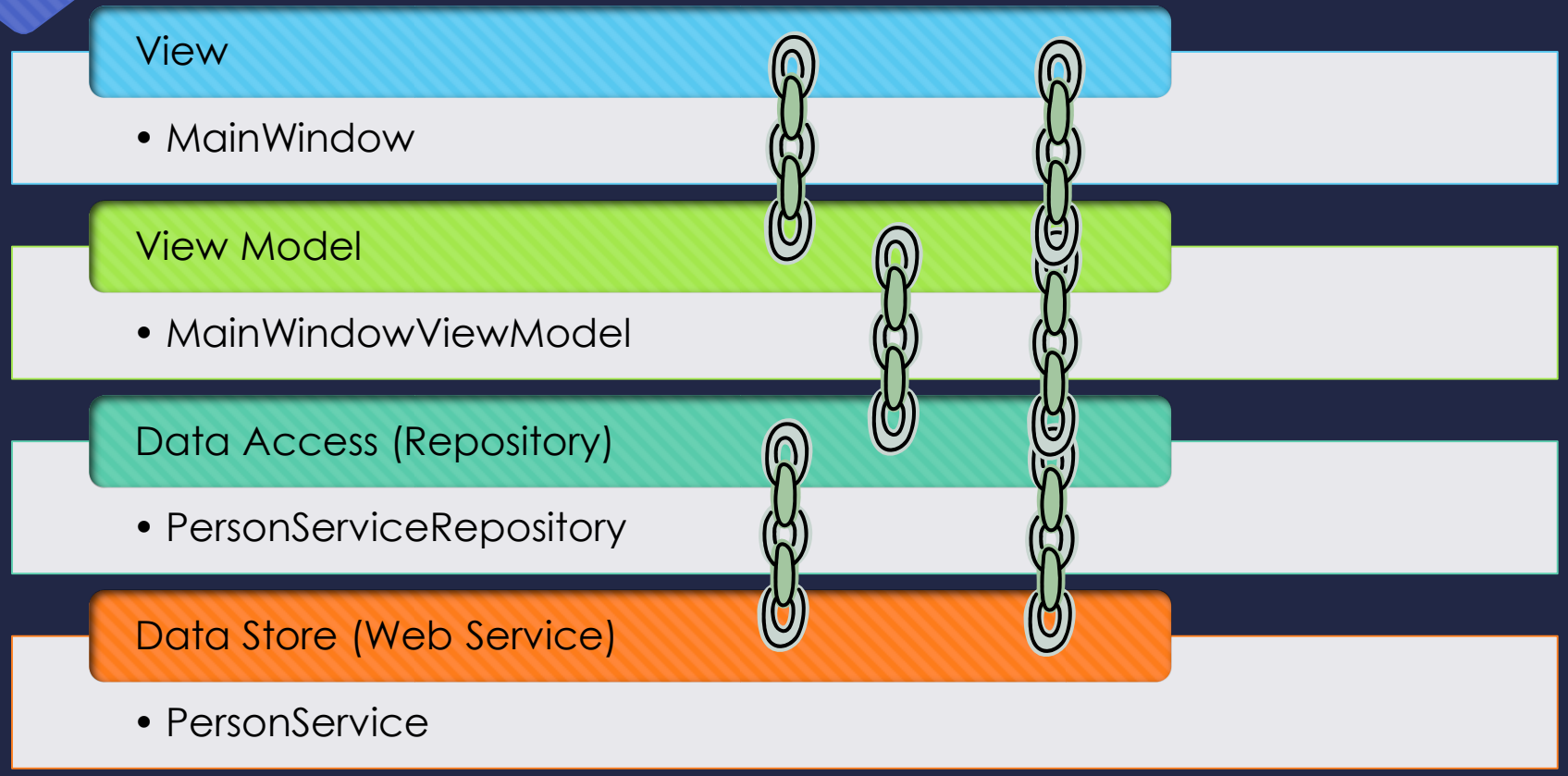
Data Access (Repository)

- PersonServiceRepository

Data Store (Web Service)

- PersonService

Tight Coupling



Loose(r) Coupling

View

- MainWindow

View Model

- MainWindowViewModel

Data Access (Repository)

- PersonServiceRepository

Data Store (Web Service)

- PersonService

Primary Benefits

- Extensibility
- Late Binding
- Parallel Development
- Maintainability
- Testability
- Adherence to S.O.L.I.D. Design Principles

a href="…"

- http://www.amazon.com/Dependency-Injection-NET-Mark-Seemann/dp/1935182501/
  - Dependency Injection in .NET by Mark Seeman

Breaking Tight Coupling with Dependency Injection

Next Up: Lab