

# Programming Against Interfaces

---



**Mosh Hamedani**

[programmingwithmosh.com](http://programmingwithmosh.com) @moshhamedani

# Dependency Inversion Principle

---

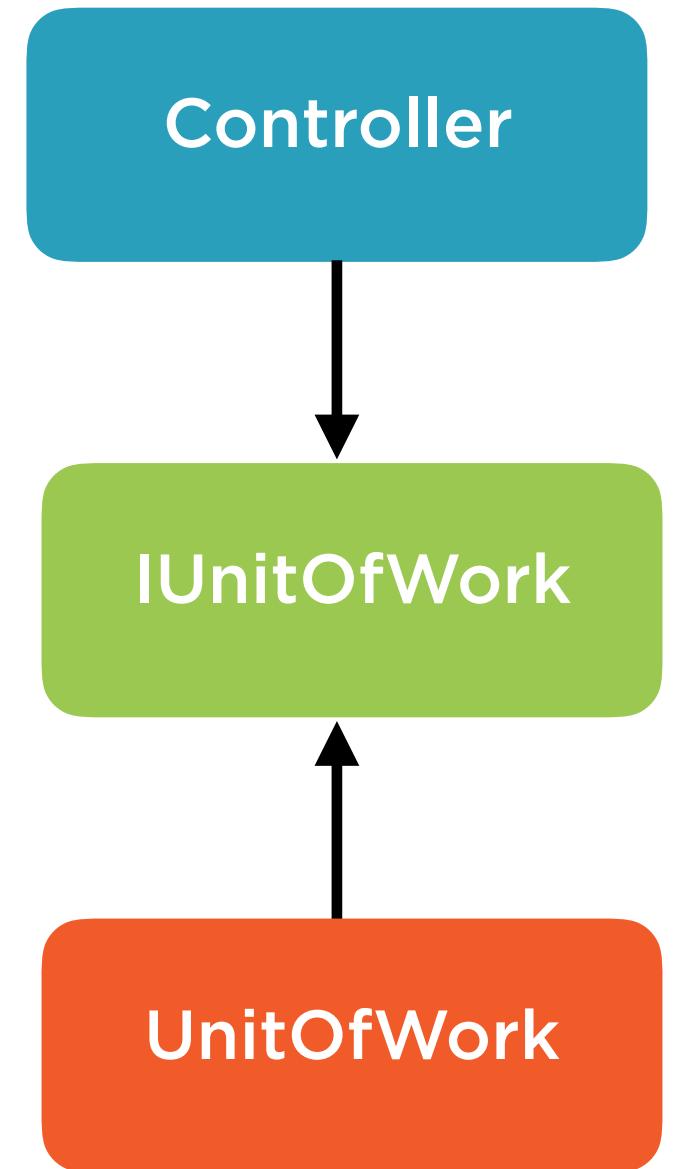
# Dependency Inversion Principle

A: High-level modules should not depend on low-level modules. Both should depend on abstractions.

**Controller**



**UnitOfWork**



# IUnitOfWork

IUnitOfWork

Complete()  
Gigs: GigRepository

# Dependency Inversion Principle

B: Abstractions should not depend on details. Details should depend on abstractions.

# DI Frameworks

**Unity**

**StructureMap**

**Autofac**

**Ninject**

...

# IUnitOfWork

IUnitOfWork

Complete()  
Gigs: **IGigRepository**

# Convention Over Configuration

**UnitOfWork**

**IUnitOfWork**

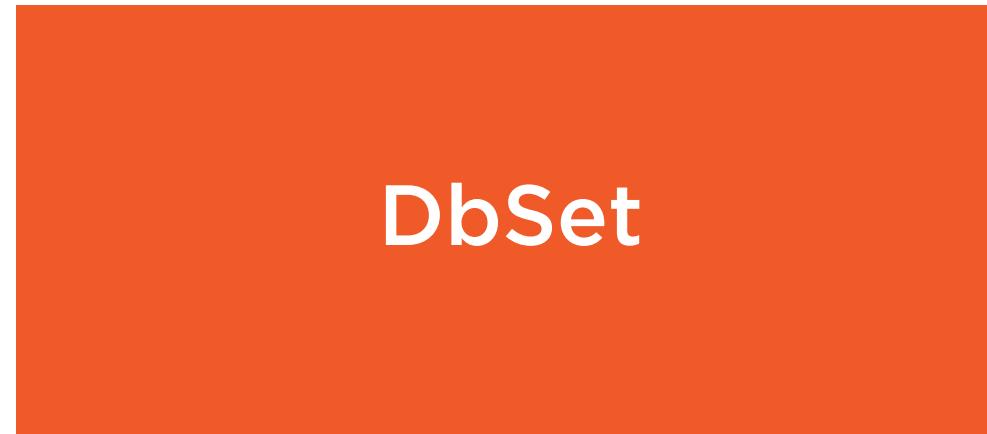
**Repository pattern is already  
implemented in Entity Framework.**

**No need to re-implement it!**



# Entity Framework

<< repository >>



**DbSet**

Add()  
Remove()  
Where()

<< unit of work >>



**DbContext**

SaveChanges()

# Repository

- Minimizes duplicate query logic**
- Provides better separation of concerns**
- Decouples from persistence frameworks**

# IQueryable

```
_context.Attendances
    .Where(a => a.AttendeeId == userId)
    .Select(a => a.Gig)
    .Include(g => g.Artist)
    .Include(g => g.Genre)
    .ToList();
```

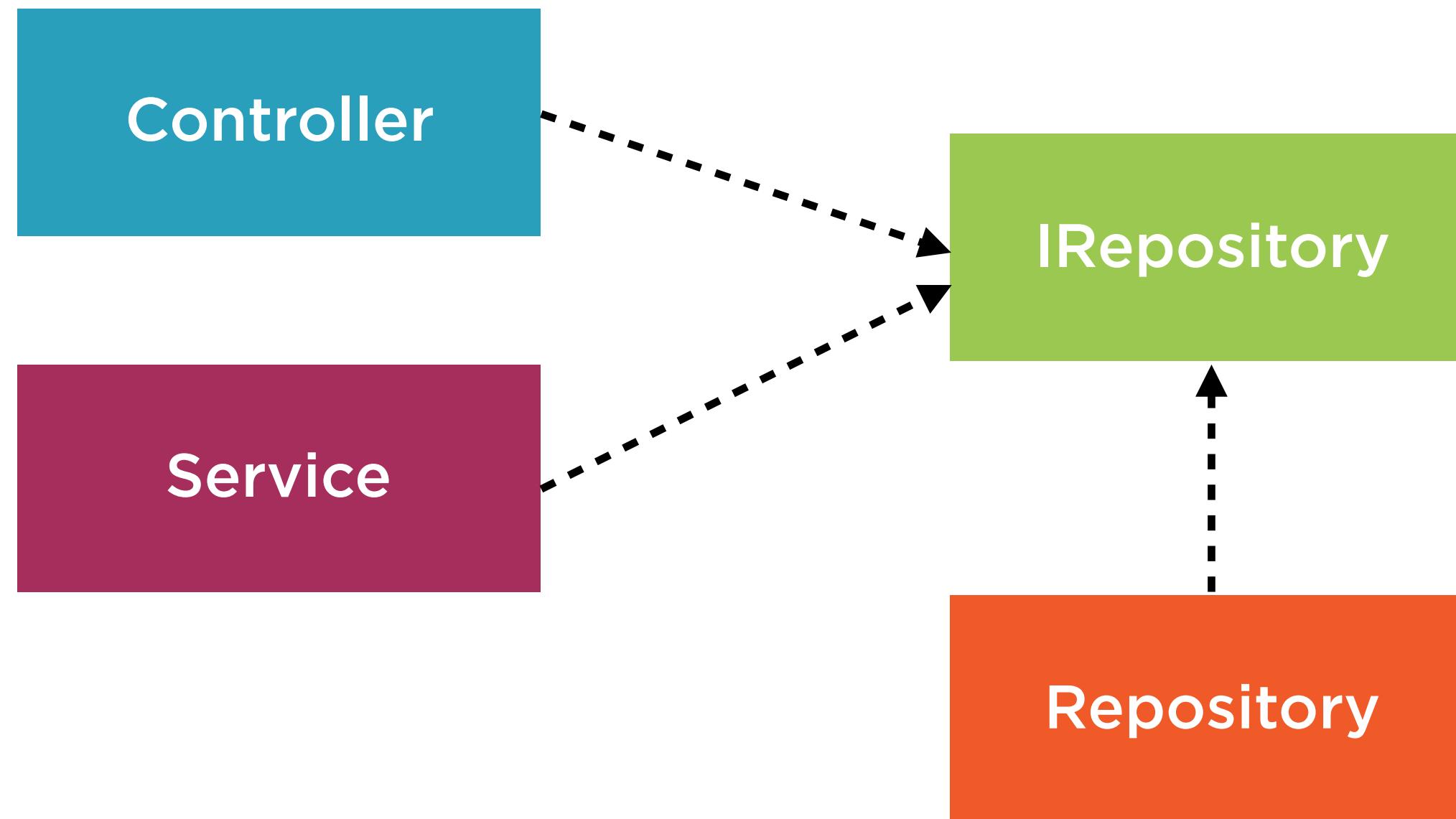
# Extension Methods

```
public static IEnumerable<Gig>
    GetGigsUserAttending(this DbSet<Gig>, string userId)
```

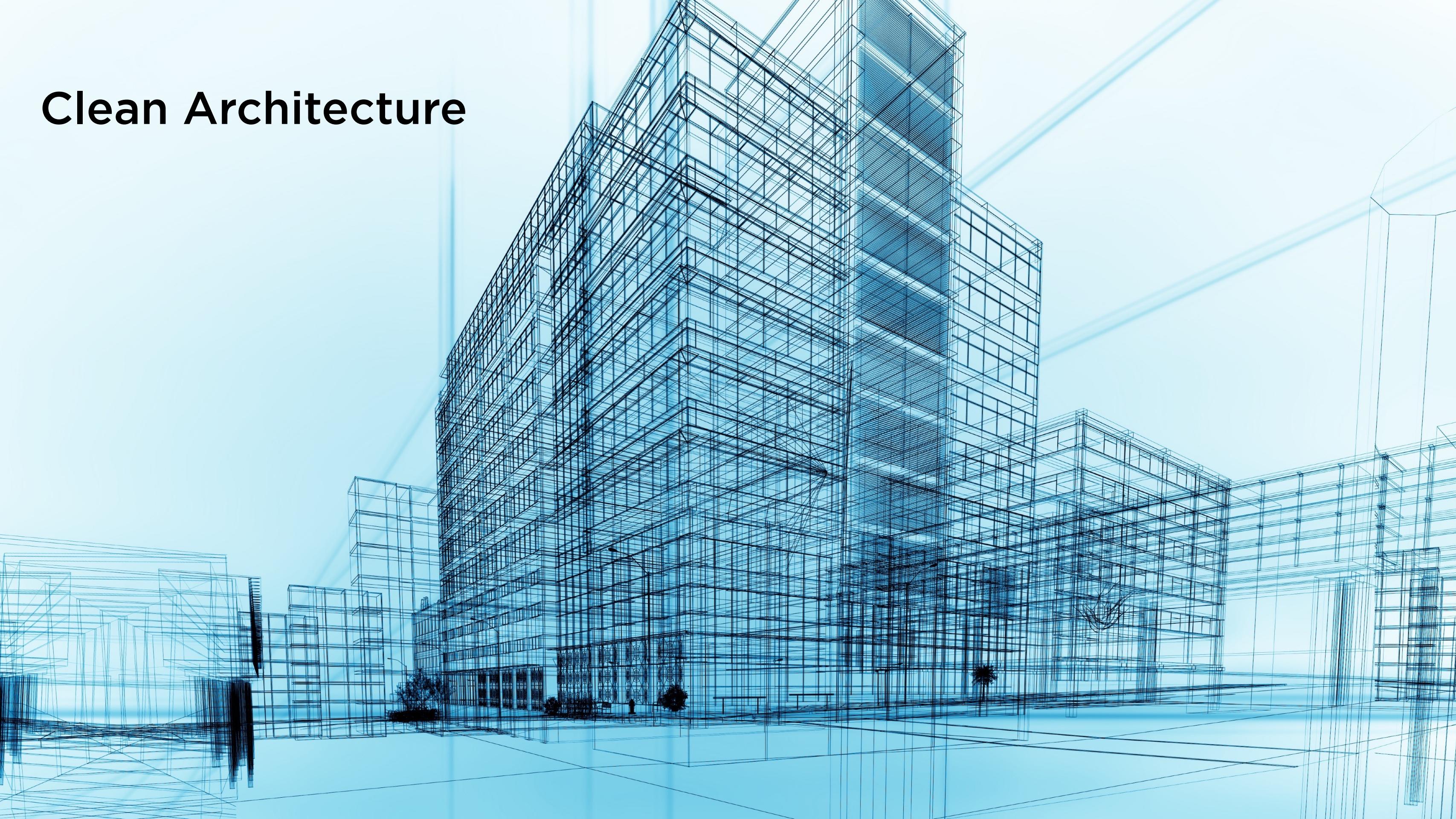
# IQueryable

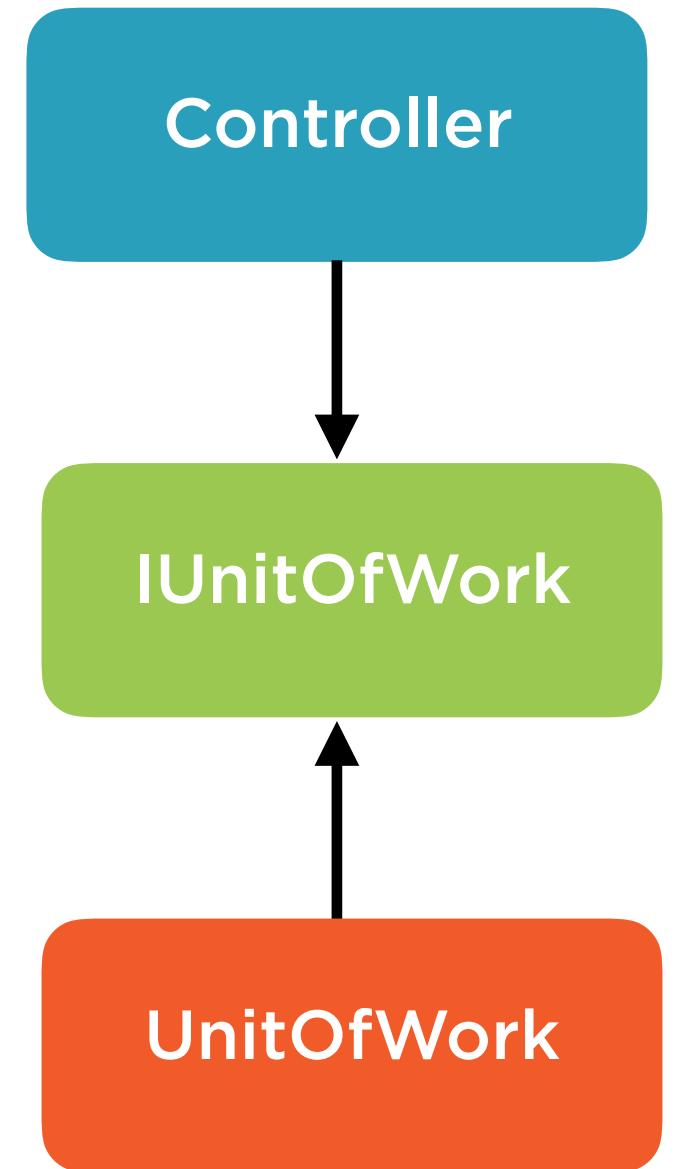
```
_context.Attendances
    .Where(a => a.AttendeeId == userId)
    .Select(a => a.Gig)
    .Include(g => g.Artist)
    .Include(g => g.Genre)
    .ToList();
```

# Decoupled Architecture



# Clean Architecture



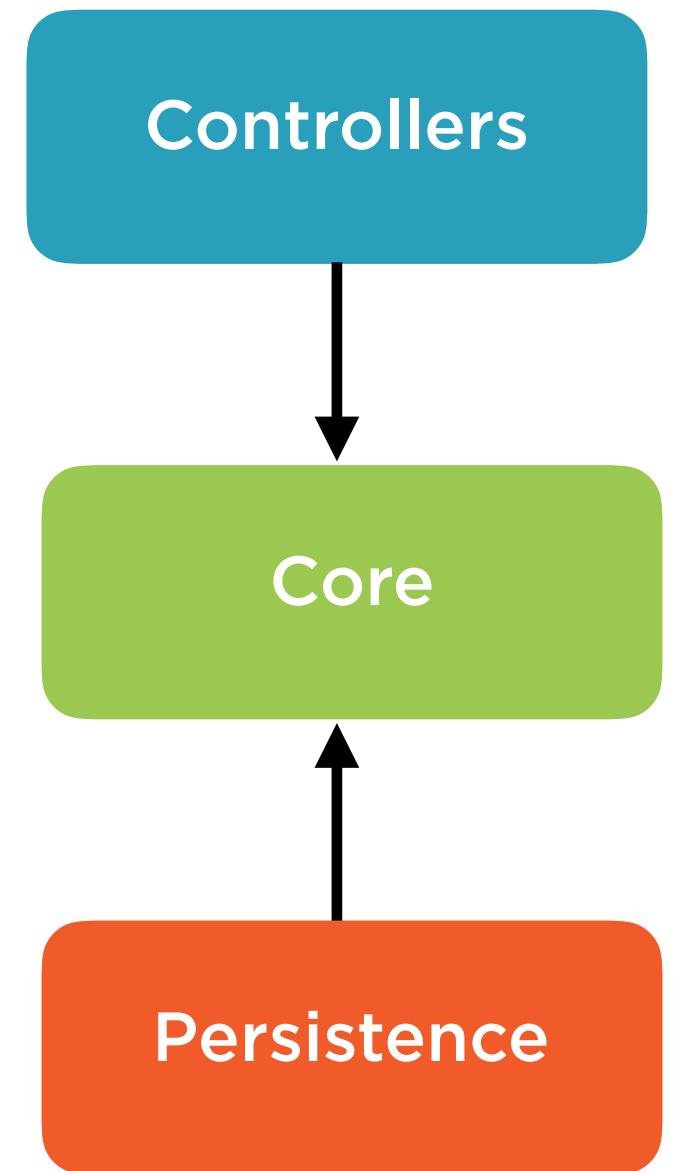


Controllers

Persistence



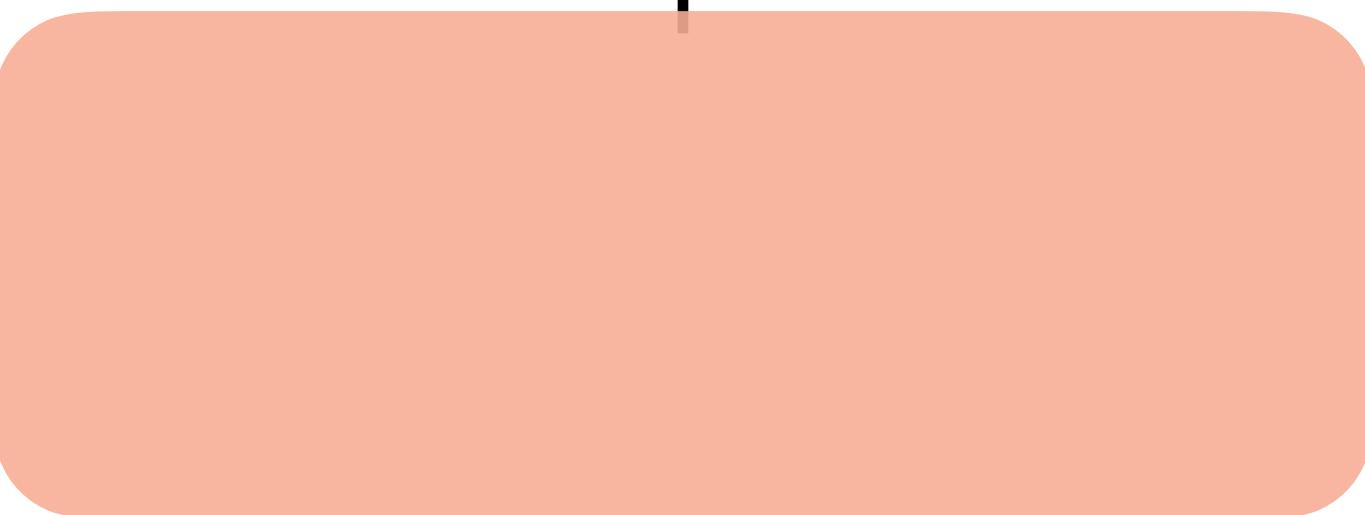




**Core**



**Persistence**



**Core**

IUnitOfWork

IRepository

**Persistence**



## Core

IUnitOfWork

IRepository

## Persistence

UnitOfWork

Repository



# Dependency Inversion Principle

A: High-level modules should not depend on low-level modules. Both should depend on abstractions.

# Abstractions

# Abstractions

Entities

# Abstractions

Entities

Dtos

# Abstractions

Entities

Dtos

ViewModels