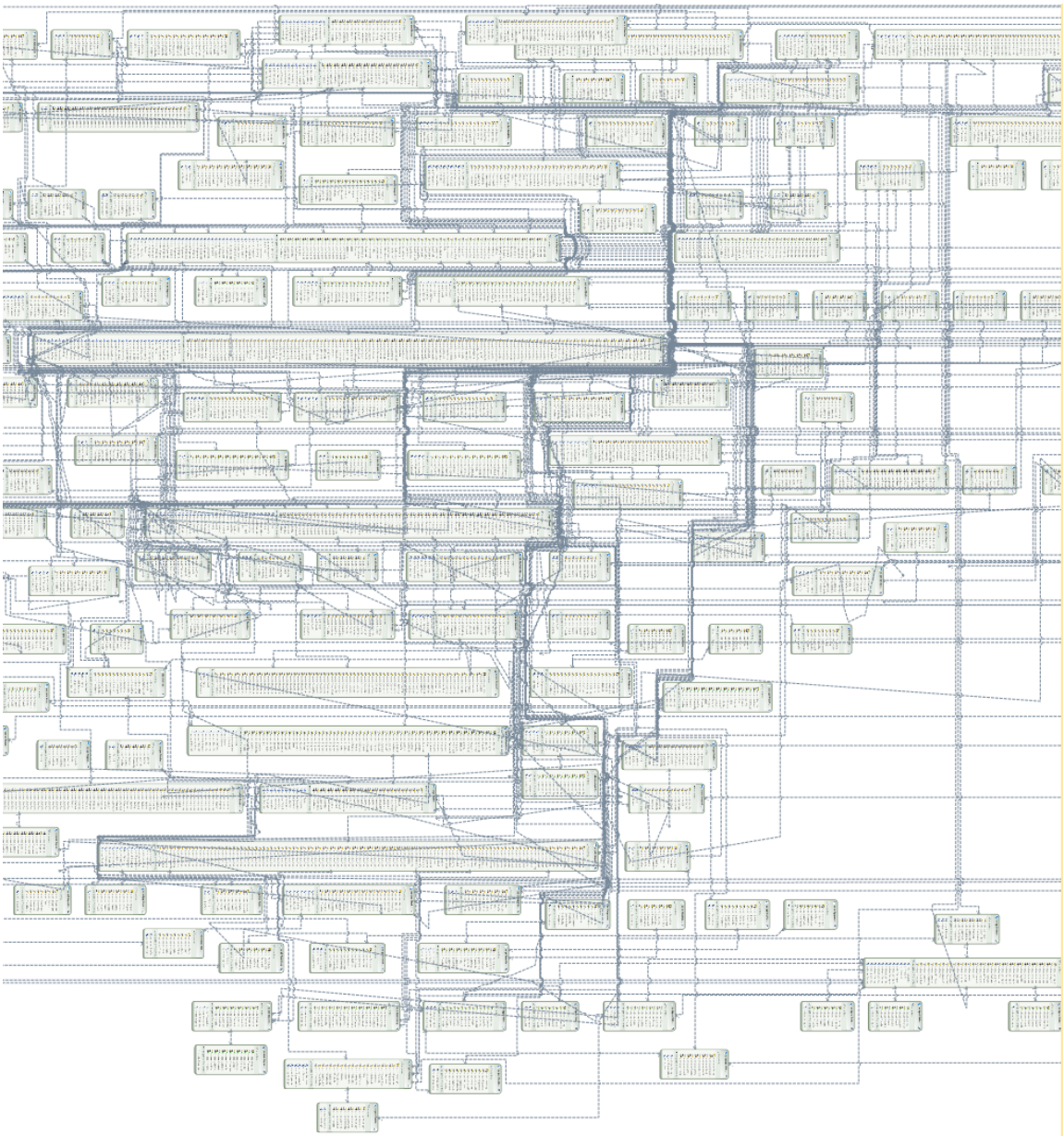
DDD and EF Core: Preserving Encapsulation

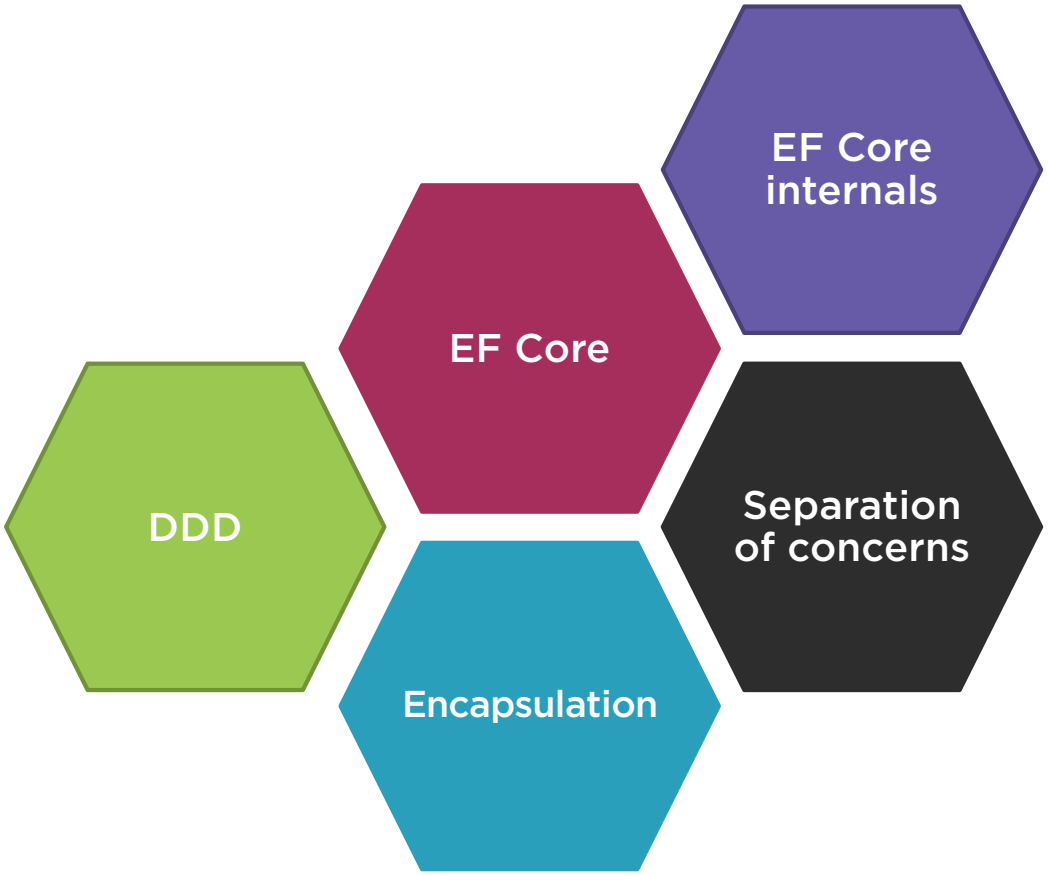
INTRODUCTION

Vladimir Khorikov

@vkhorikov www.enterprisecraftsmanship.com



CQRS


EF Core internals
EF Core
DDD
Separation of concerns
Encapsulation

Overview

Introduction

Working with Many-to-one Relationships

Working with Lazy Loading

Mapping Backing Fields

Mapping Value Objects

Implementing a Domain Event Dispatcher

Prerequisites

DDD

“Domain-Driven Design in Practice” by Vladimir Khorikov

“Refactoring from Anemic Domain Model Towards a Rich One” by Vladimir Khorikov

EF Core

“Entity Framework Core 2: Mappings” by Julie Lerman

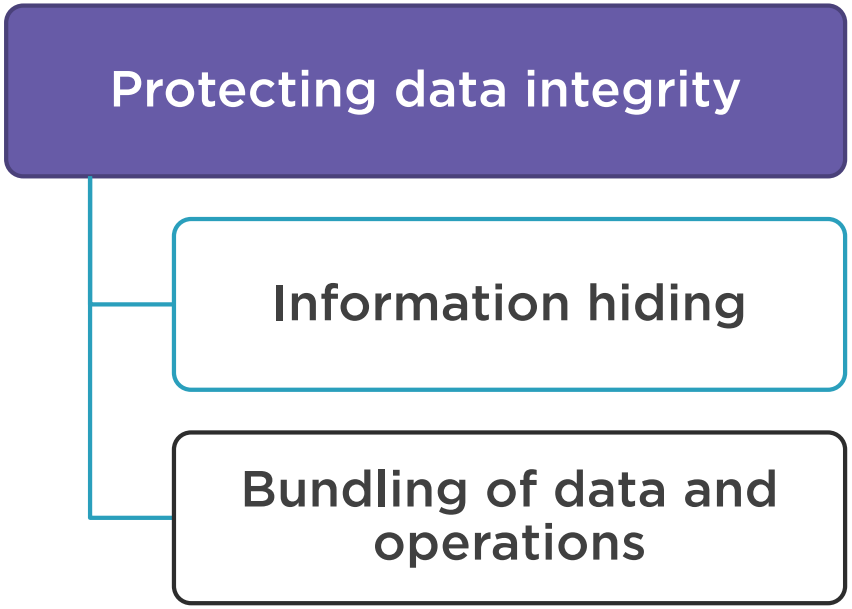
Encapsulation and Separation of Concerns

Encapsulation

Separation of concerns

Encapsulation is an act of protecting data integrity.

Encapsulation



Encapsulation

Refactoring from Anemic Domain Model Towards a Rich One

by Vladimir Khorikov

Building bullet-proof business line applications is a complex task. This course will teach you an in-depth guideline into refactoring from Anemic Domain Model into a rich, highly encapsulated one.

Resume Course | Bookmark | Add to Channel | Download Course

Table of contents | Description | Transcript | Exercise files | Discussion | Learning Check | Recommended

This course is part of: Domain-Driven Design Path

Expand All

Course author

Vladimir Khorikov

Vladimir Khorikov is a Microsoft MVP and has been professionally involved in software development for more than 10 years. Nowadays he specializes in rescuing legacy code bases and helping teams...

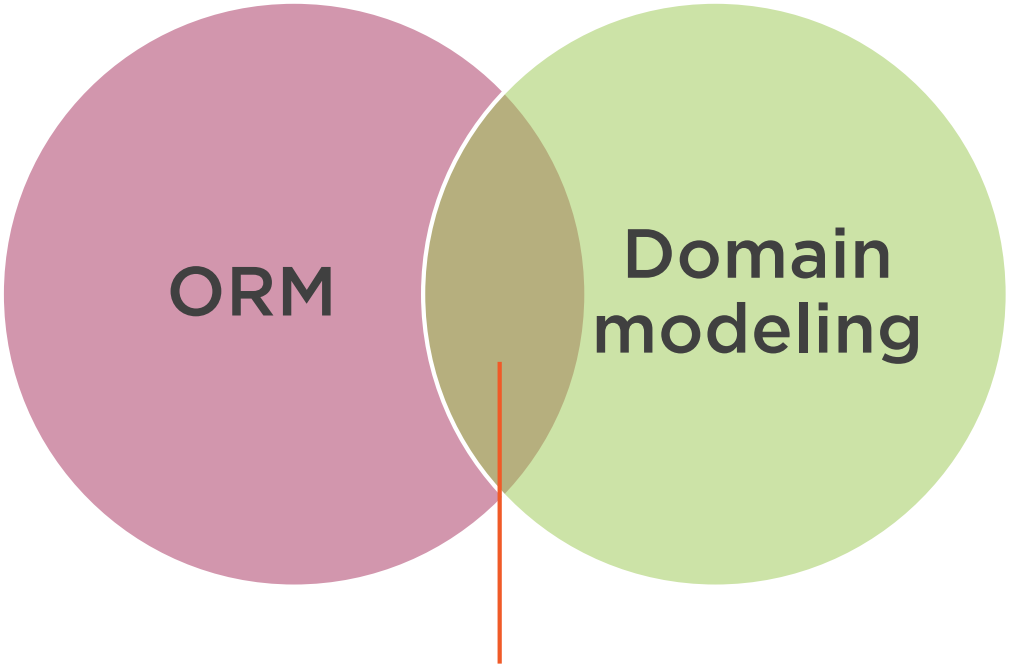
Course info

Level	Intermediate
Rating	★★★★★ (194)
My rating	★★★★★
Duration	3h 36m
Released	13 Nov 2017

Share course

Each application concern should have its own place in the code base and not overlap with other concerns.

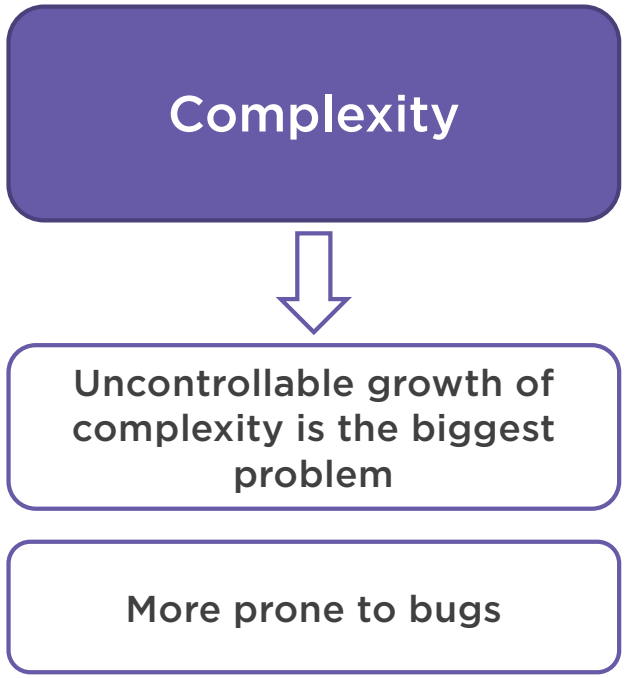
Separation of Concerns



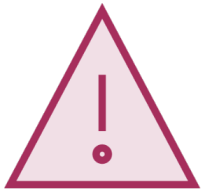
There should be no code in the intersection

Single responsibility principle applied on a larger scale

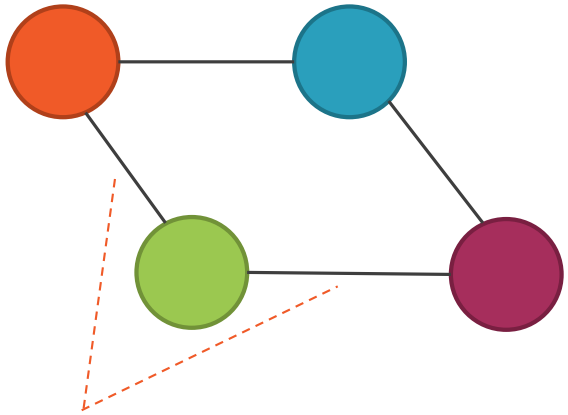
Encapsulation and Separation of Concerns



Complexity

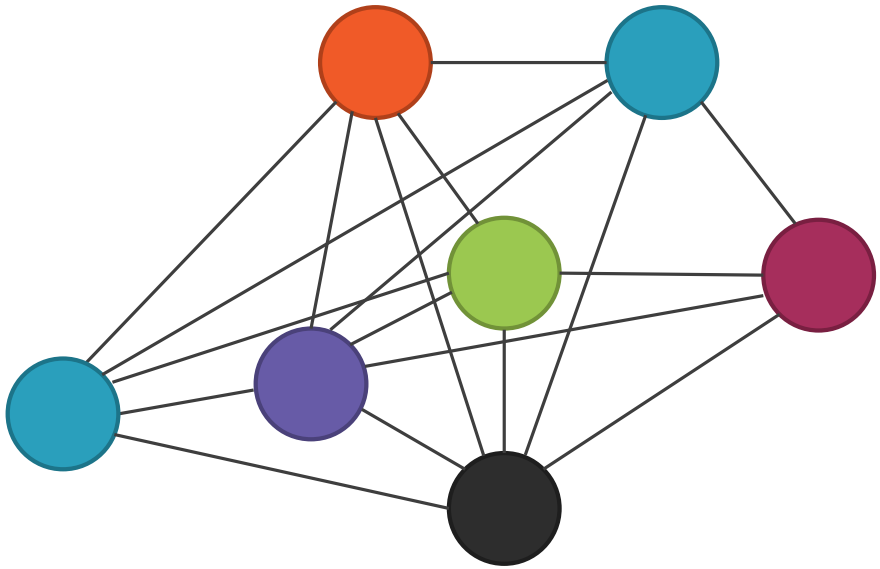


Complexity emerges from coupling



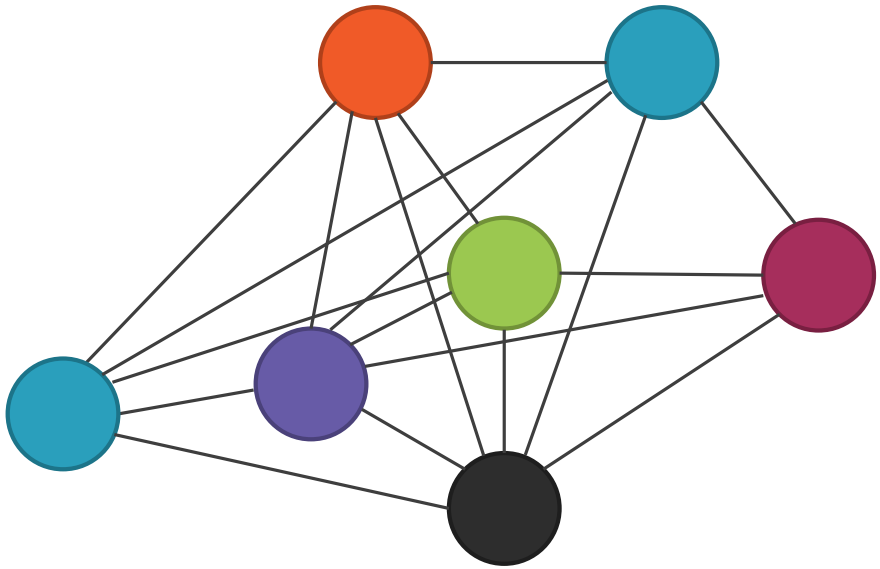
Coupling is connections between code elements

Complexity

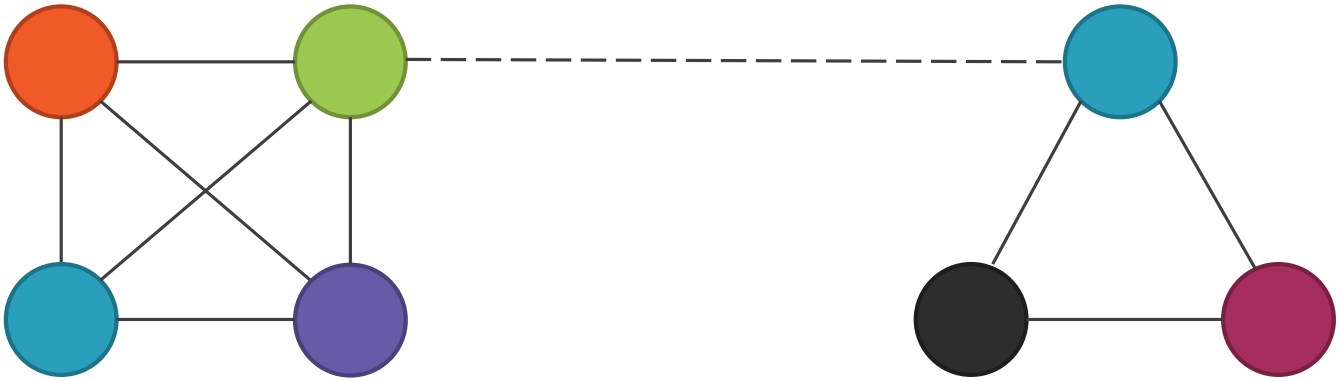


- ✗ Exponential growth of complexity
- ✗ Dissolves into a big ball of mud

Complexity



Complexity



Complexity

High cohesion, low coupling

Cohesion = internal coupling

Coupling = external coupling

Complexity

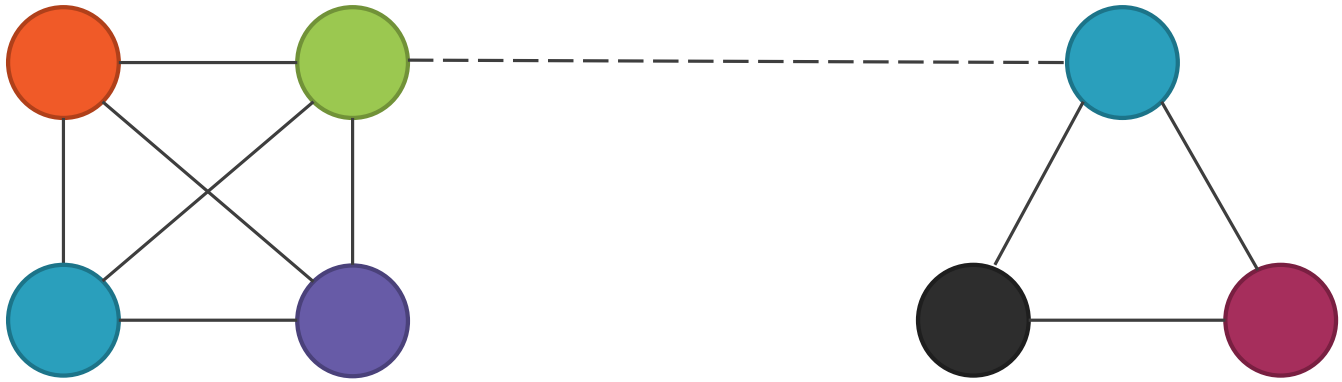
Separation of concerns reduces the rate of complexity growth

Domain modeling

ORM mappings

Database

Complexity



Encapsulation reduces internal coupling

Separation of Concerns: Example

```
public class Course
{
    [Column("Is_active", TypeName = "char(1)")]
    public bool IsActive { get; set; }

    [Column("Students_enrolled", TypeName = "int")]
    public int NumberOfStudents { get; set; }
}
```

- ✗ Violates the principle of separation of concerns
- ✗ Domain modeling and ORM mapping in the same class

Separation of Concerns: Example

```
// DbContext
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Course>(x =>
    {
        x.Property(p => p.IsActive)
            .HasColumnName("Is_active");
        x.Property(p => p.NumberOfStudents)
            .HasColumnName("Students_enrolled");
    });
}
```

Use fluent mapping instead

Separation of Concerns: Example

```
public class CourseDto
{
    [RegularExpression("[a-zA-Z]{1,50}")]
    public string Name { get; set; }
}
```

Data container

Input validation

Encapsulation: Example

```
public class Course
{
    public bool IsActive { get; set; }
    public int NumberOfStudents { get; set; }
}
```

Inactive courses cannot have students

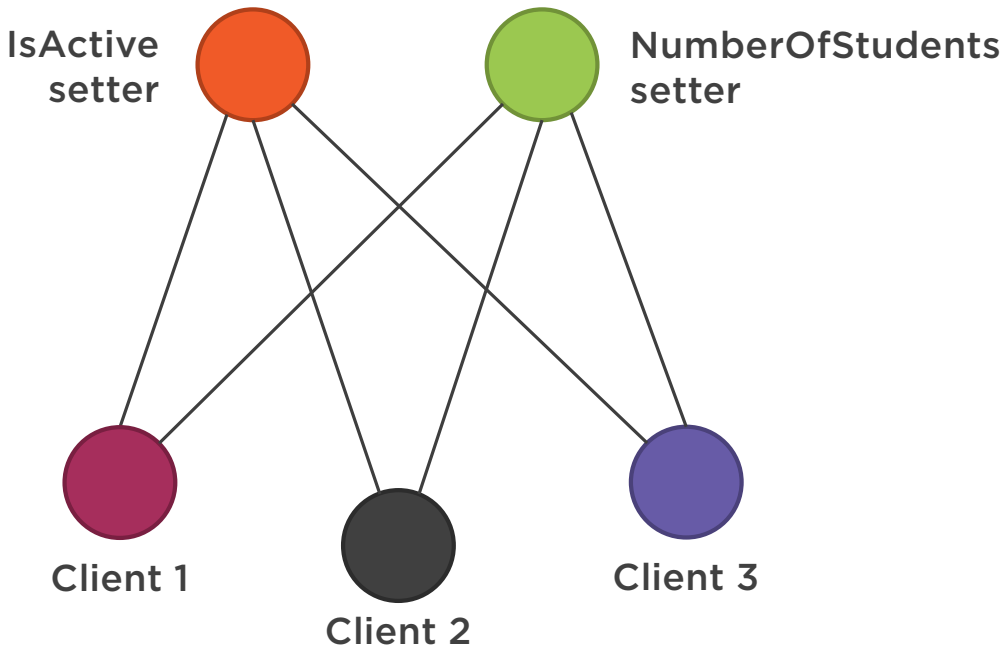
No way to maintain invariants in the current implementation

Encapsulation: Example

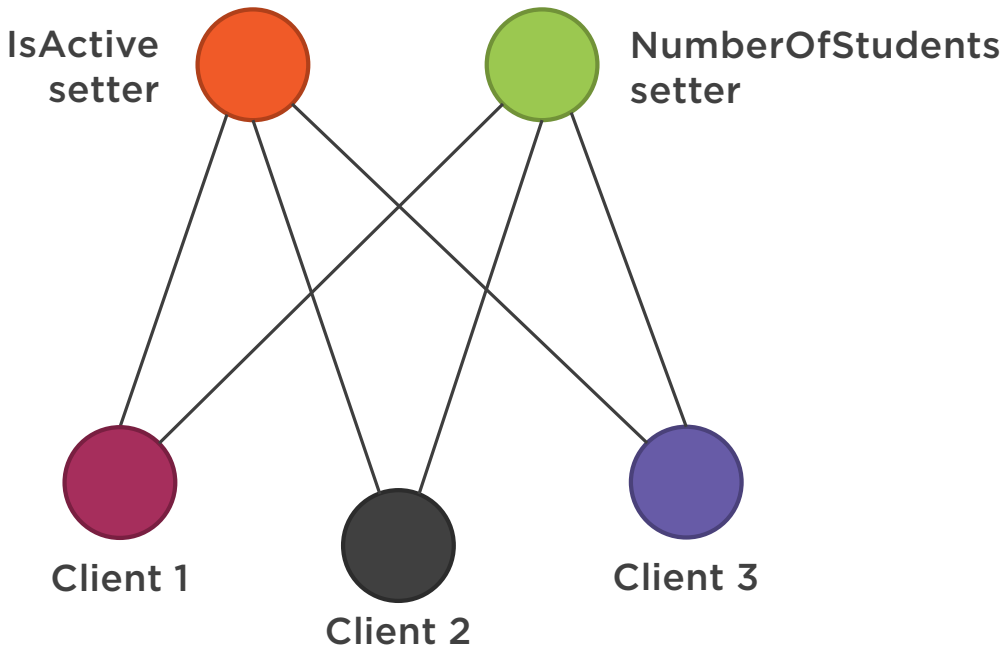
```
public class Course
{
    public bool IsActive { get; private set; }
    public int NumberOfStudents { get; private set; }

    public void Disable()
    {
        IsActive = false;
        NumberOfStudents = 0;
    }
}
```

Maintains encapsulation

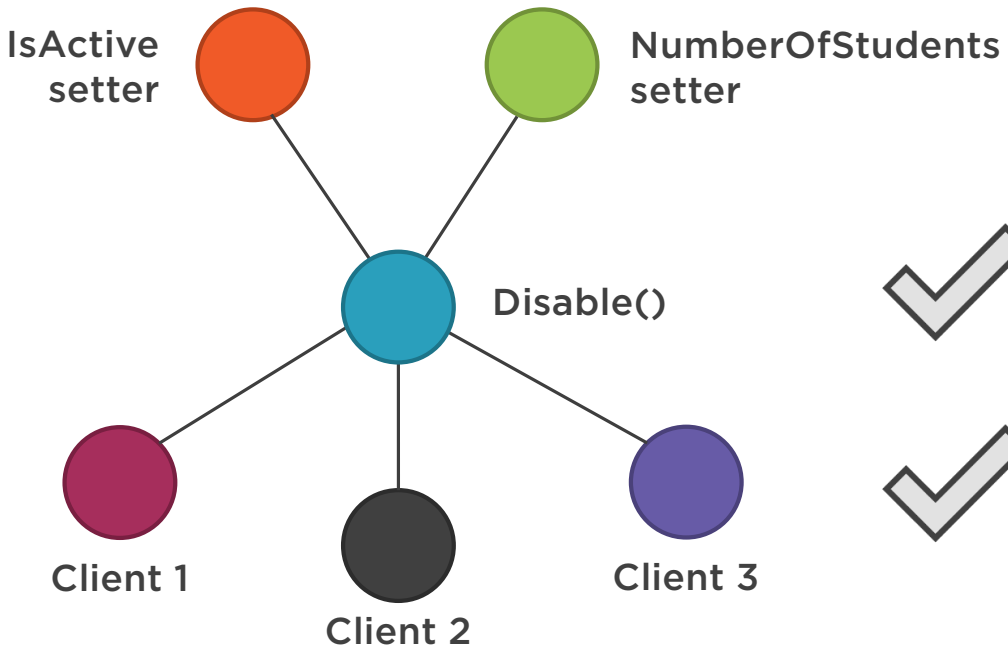
Encapsulation: Example



Encapsulation: Example



Encapsulation: Example



- Reducing the number of connections
- Upkeep of invariants

Encapsulation: Example

How this relates to EF Core?

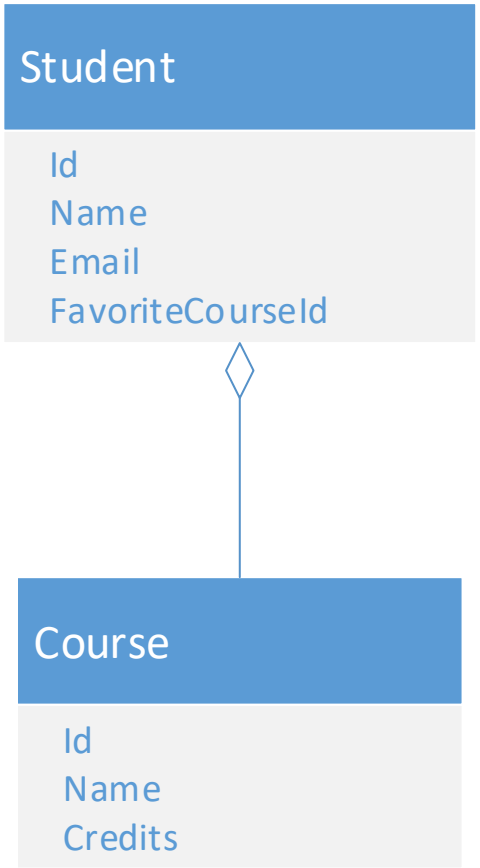
EF Core often requires you to compromise on encapsulation

Learn how to preserve encapsulation

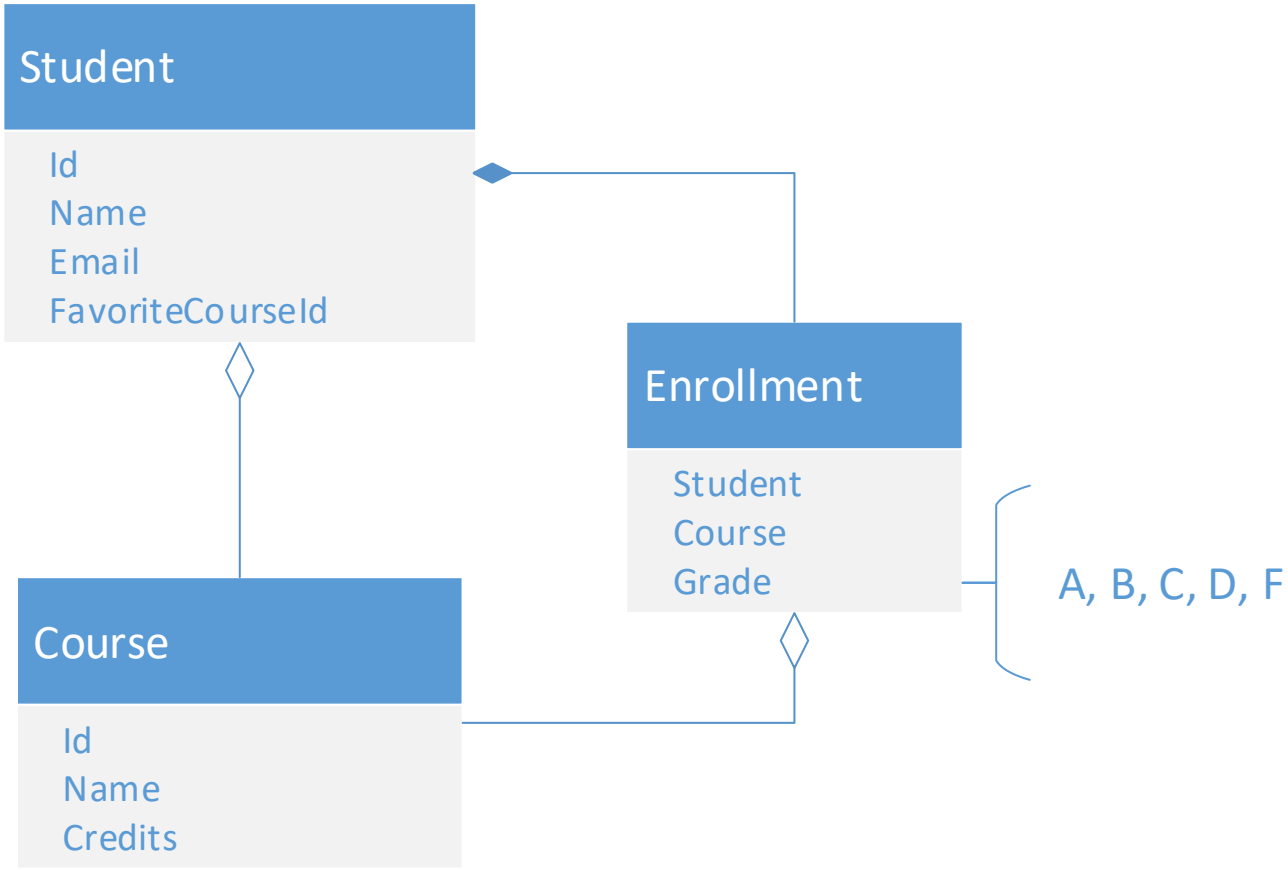
Encapsulation: Example

ORM and the database are important too

Sample Application Introduction



Sample Application Introduction



Sample Application Introduction

vkhorikov / DddAndEFCore

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Source code for the DDD and EF Core Pluralsight course (link is coming soon)

Edit

Manage topics

1 commit | 1 branch | 0 packages | 1 release | 1 contributor | MIT

Tag: module-1 | New pull request | Create new file | Upload files | Find file | Clone or download

vkhorikov Initial — Latest commit aabb560 1 hour ago

src/App	Initial	1 hour ago
.gitignore	Initial	1 hour ago
Database-initial.sql	Initial	1 hour ago
DddAndEFCore.sln	Initial	1 hour ago
DddAndEFCore.sln.DotSettings	Initial	1 hour ago
LICENSE	Initial	1 hour ago
README.md	Initial	1 hour ago

http://bit.ly/ddd-ef

Sample Application Introduction

All Branches | Show Remote Branches | Date Order | Jump to:

Graph	Description	Date	Author	Commit
master, module-1, origin/master	Initial	19 Nov 2019 19:24	Vladimir <vladimir	aabb560

Pending files, sorted by file status | Search

Commit: aabb560ac820cc16cfed822f661aee332d0f40c4 [aabb560]
Author: Vladimir <vladimir.khorikov@gmail.com>
Date: Tuesday, November 19, 2019 7:24:19 PM
Committer: Vladimir

Initial

.gitignore

.gitignore

File Contents | Reverse hunk

```
1  + out
2  + .svn
3  + obj
4  + bin
5  + .idea
6  + _ReSharper*
7  + *.sln.GhostDoc.xml
8  + *.dotCover
9  + *.suo
```

Sample Application Introduction

vkhorikov / DddAndEFCore

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Source code for the DDD and EF Core Pluralsight course (link is coming soon)

Edit

Manage topics

1 commit | 1 branch | 0 packages | 1 release | 1 contributor | MIT

Tag: module-1 | New pull request | Create new file | Upload files | Find file | Clone or download

vkhorikov Initial — Latest commit aabb560 1 hour ago

src/App	Initial	1 hour ago
.gitignore	Initial	1 hour ago
Database-initial.sql	Initial	1 hour ago
DddAndEFCore.sln	Initial	1 hour ago
DddAndEFCore.sln.DotSettings	Initial	1 hour ago
LICENSE	Initial	1 hour ago
README.md	Initial	1 hour ago

http://bit.ly/ddd-ef

Sample Application Introduction

vkhorikov / DddAndEFCore

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Releases | Tags

Draft a new release

4 minutes ago

module-1 ...

aabb560 zip tar.gz

Summary

Encapsulation and Separation of Concerns help tackle code complexity

Complexity stems from coupling between code elements

- Separation of Concerns reduces complexity by minimizing coupling between buckets
- Encapsulation reduces complexity by minimizing coupling inside the buckets

The use of EF Core often damages encapsulation and separation of concerns

This course will help you preserve domain model encapsulation while using EF Core

Summary

Encapsulation and Separation of Concerns help tackle code complexity

Complexity stems from coupling between code elements

- Separation of Concerns reduces complexity by segregating those code elements into buckets and minimizing coupling between those buckets
- Encapsulation reduces complexity by minimizing coupling inside the buckets

The use of EF Core often damages encapsulation and separation of concerns

This course will help you preserve domain model encapsulation while using EF Core

In the Next Module

Working with Many-to-one Relationships