



TECHORAMA

DEEP KNOWLEDGE IT CONFERENCE

May 22-24 | 2018

Antwerp, Belgium



MAPPING DDD DOMAIN TYPES WITH EF CORE (2.1)

JULIE LERMAN
THEDATAFARM.COM @JULIELERMAN

TECHORAMA

DEEP KNOWLEDGE IT CONFERENCE

May 22-24 | 2018

Antwerp, Belgium



Julie Lerman 
Pluralsight Author

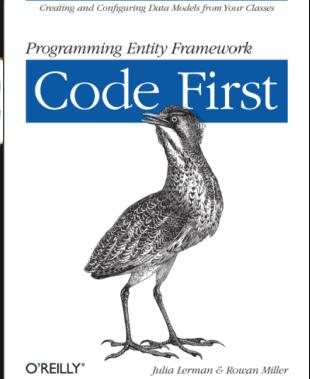
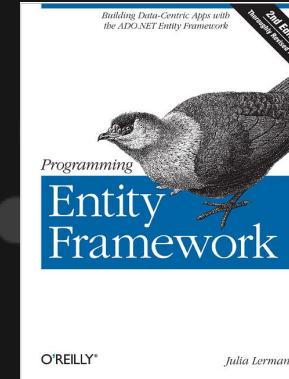
Julie Lerman has been programming and mentoring developers for over 25 years and brings a wealth of experience and knowledge to share with Pluralsight subscribers. She is the leading independent authority on the Entity Framework and has been using and teaching EF since its inception.

COURSES AUTHORED

20

All time

TOPICS AUTHORED



TOTAL RATINGS

Creating and Configuring Data Models from Your Classes

Programming Entity Framework

Code First



Courses by Julie

Entity Framework Core 2: Mappings

by Julie Lerman

Intermediate

1h 59m



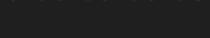
Entity Framework Core 2: Getting Started

by Julie Lerman

Beginner

2h 42m

6 February 2018



Cross-platform SQL Server Management for Developers Using VS Code

by Julie Lerman

Intermediate

2h 21m

11 September 2017



EF Core 1.0
June 2016

EF Core 2.0
August 2017

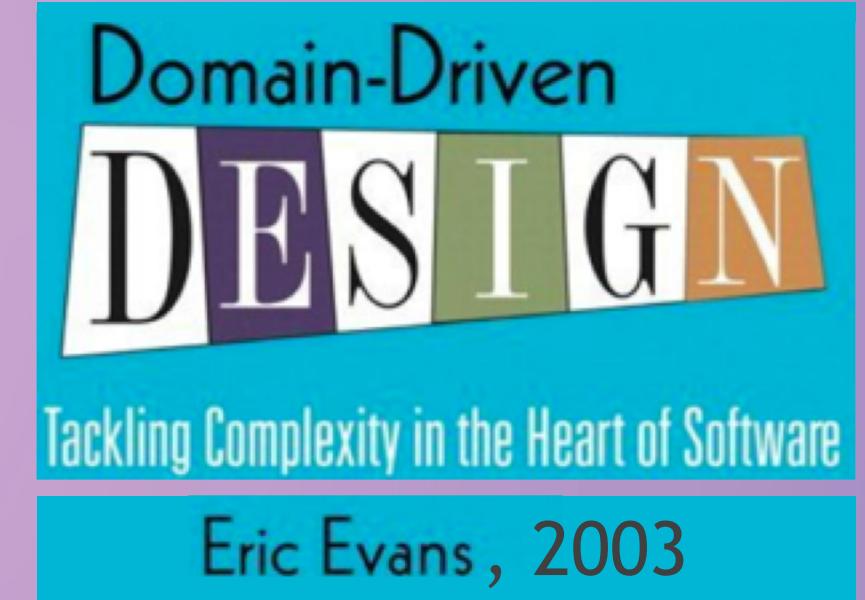
EF Core 2.1

RC1

May 9

RTM

“1st half 2018”

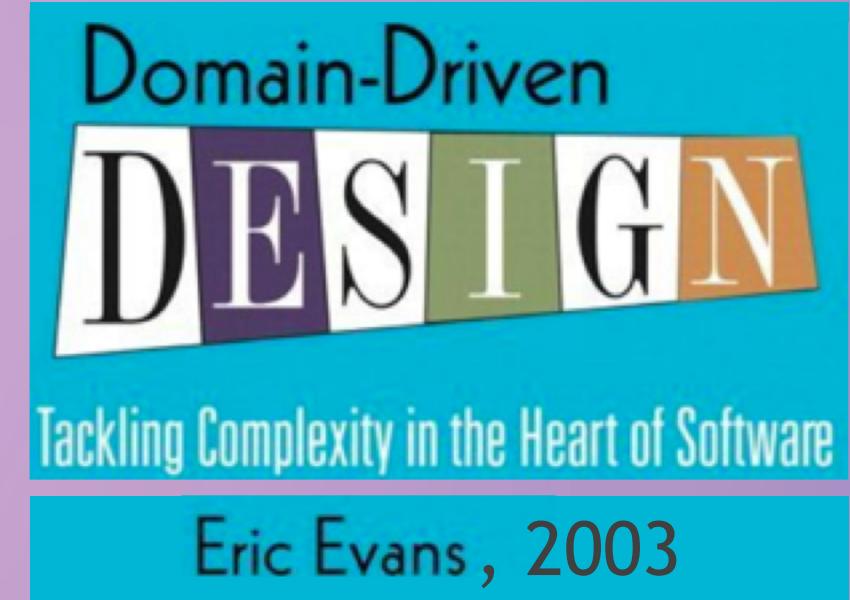


Strategic Design

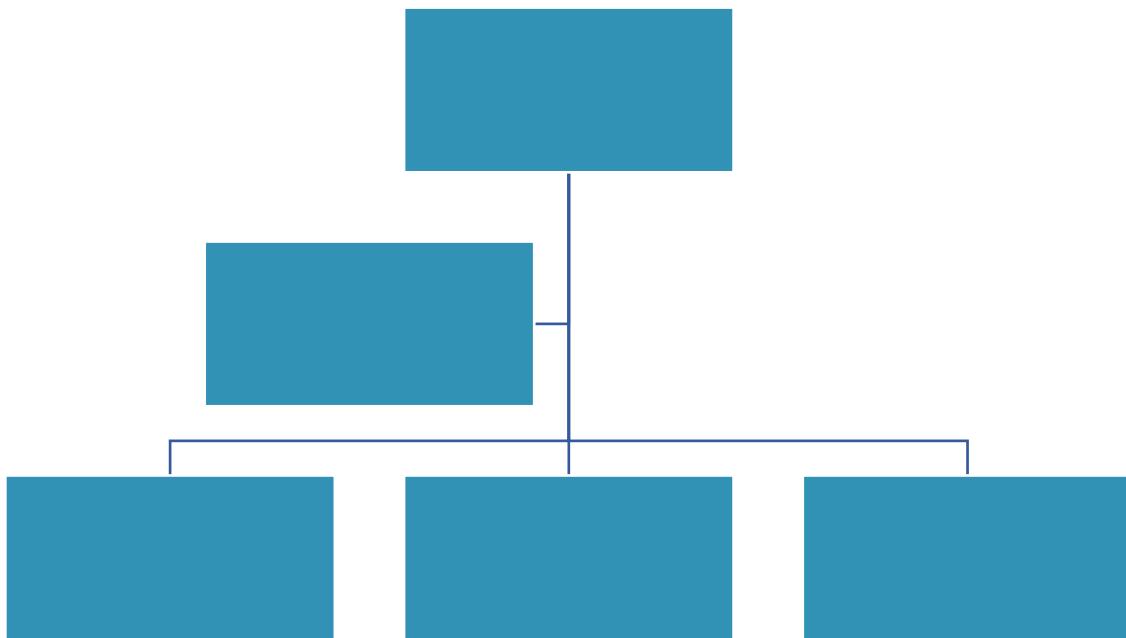


Tactical Design

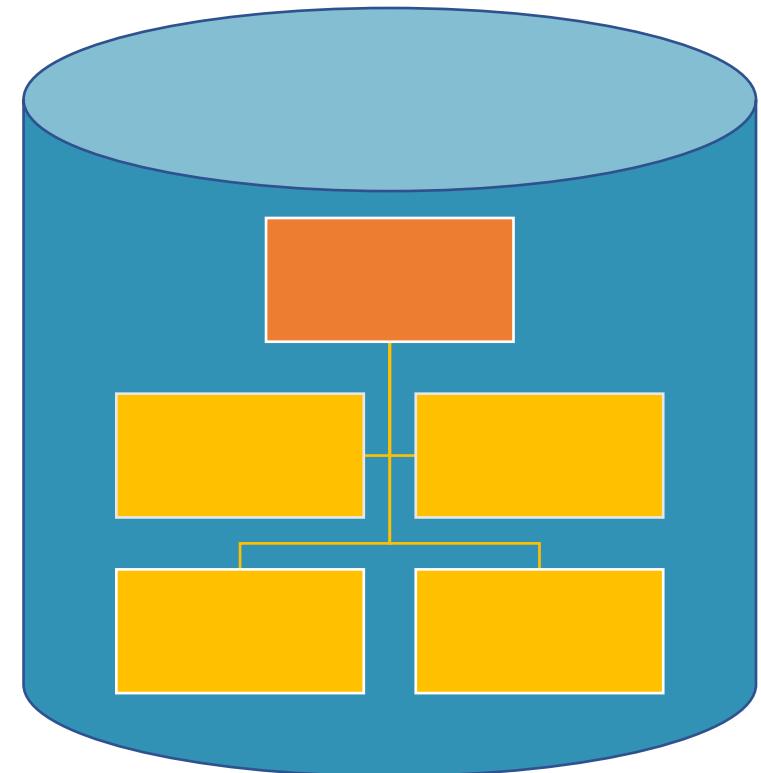
**D is for DOMAIN
NOT DATA PERSISTENCE**



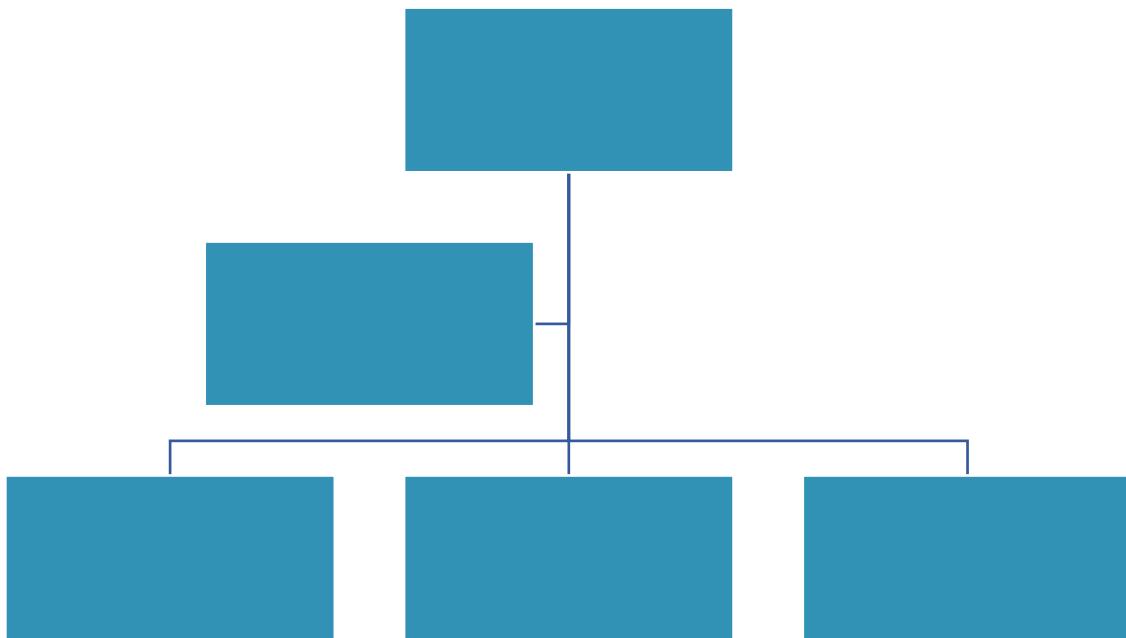
Domain Model



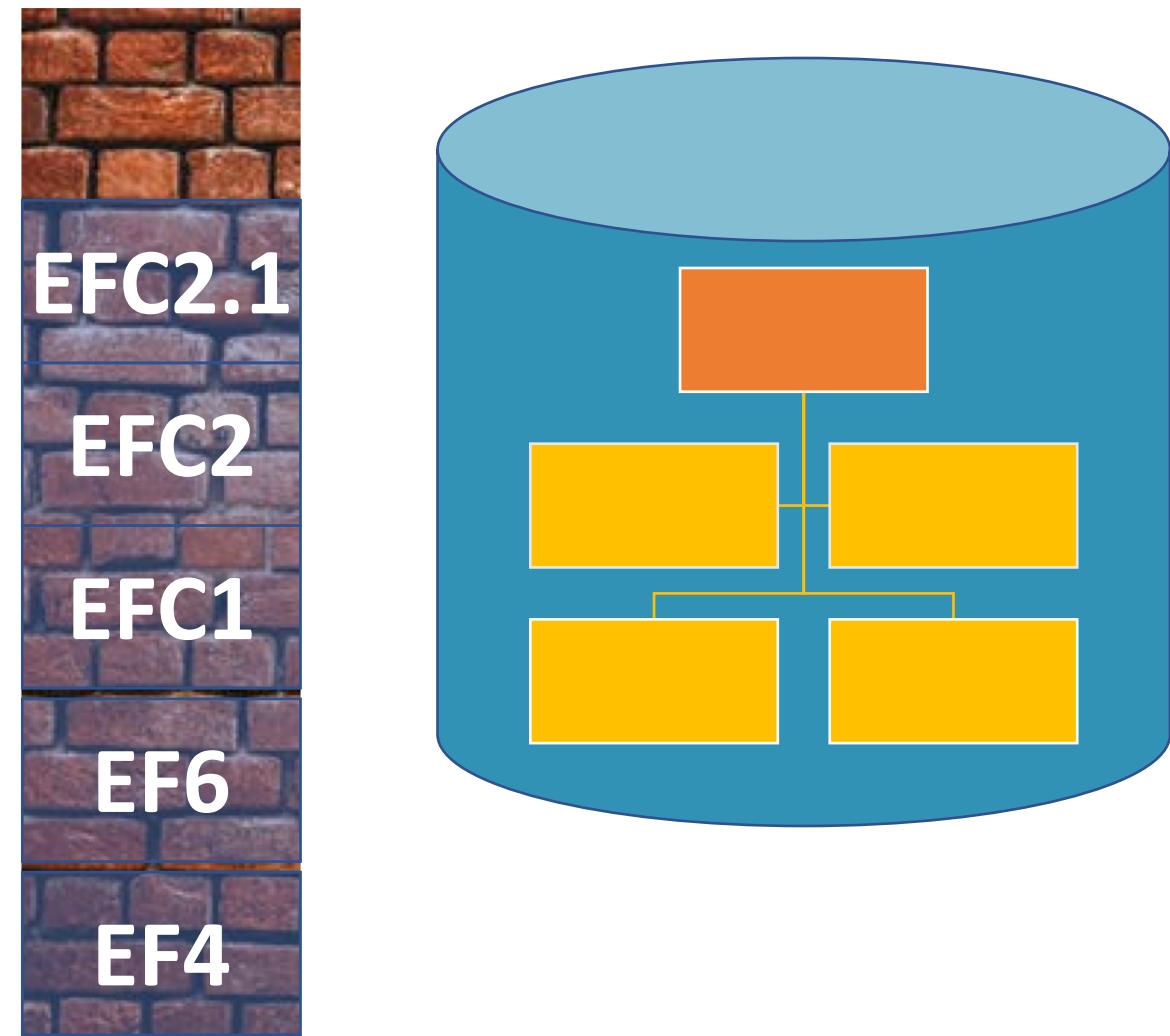
Data Model



Domain Model

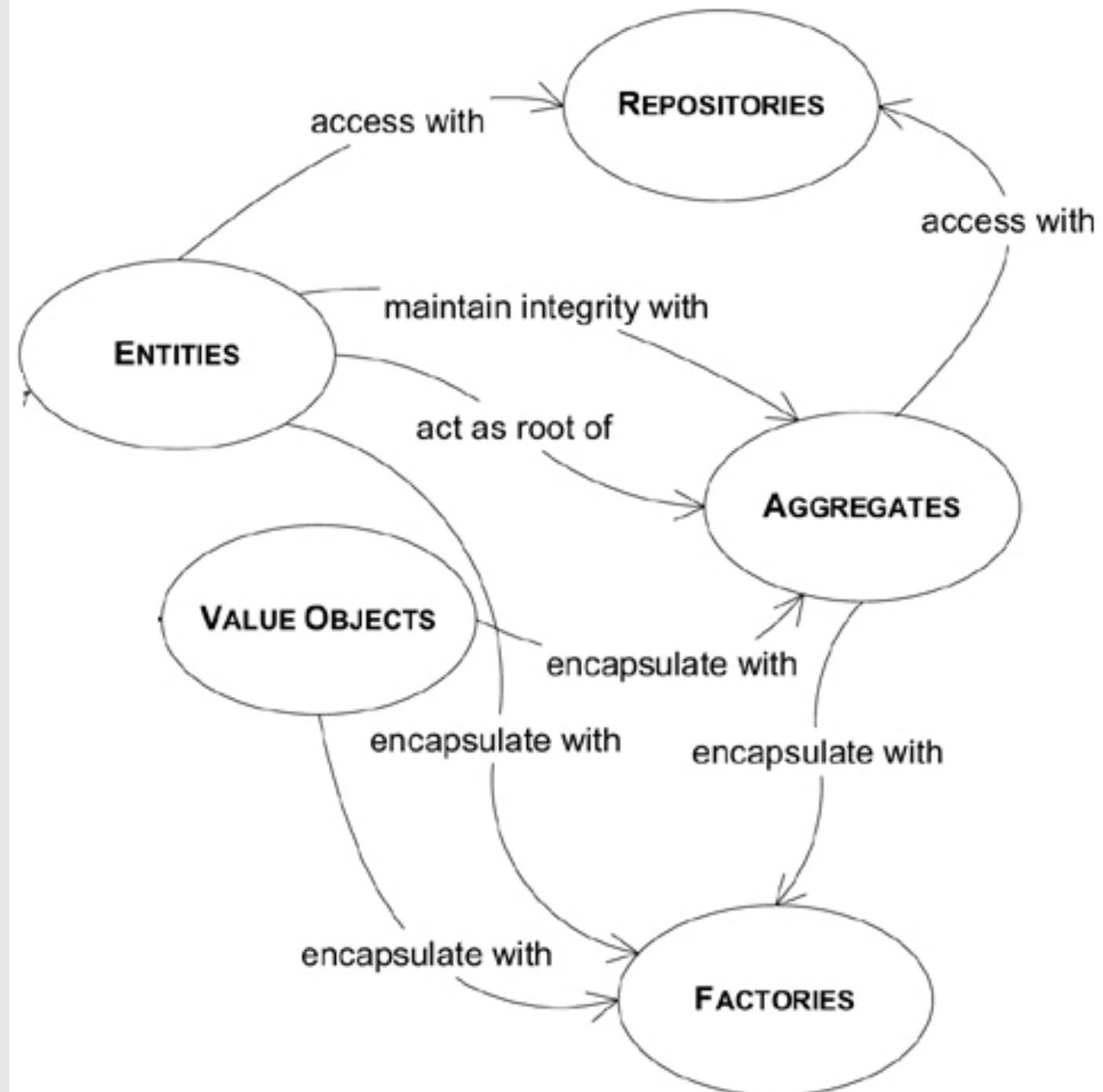


Data Model



EF Core 2.1 makes it easier to
keep your mappings
out of the way of
good domain design

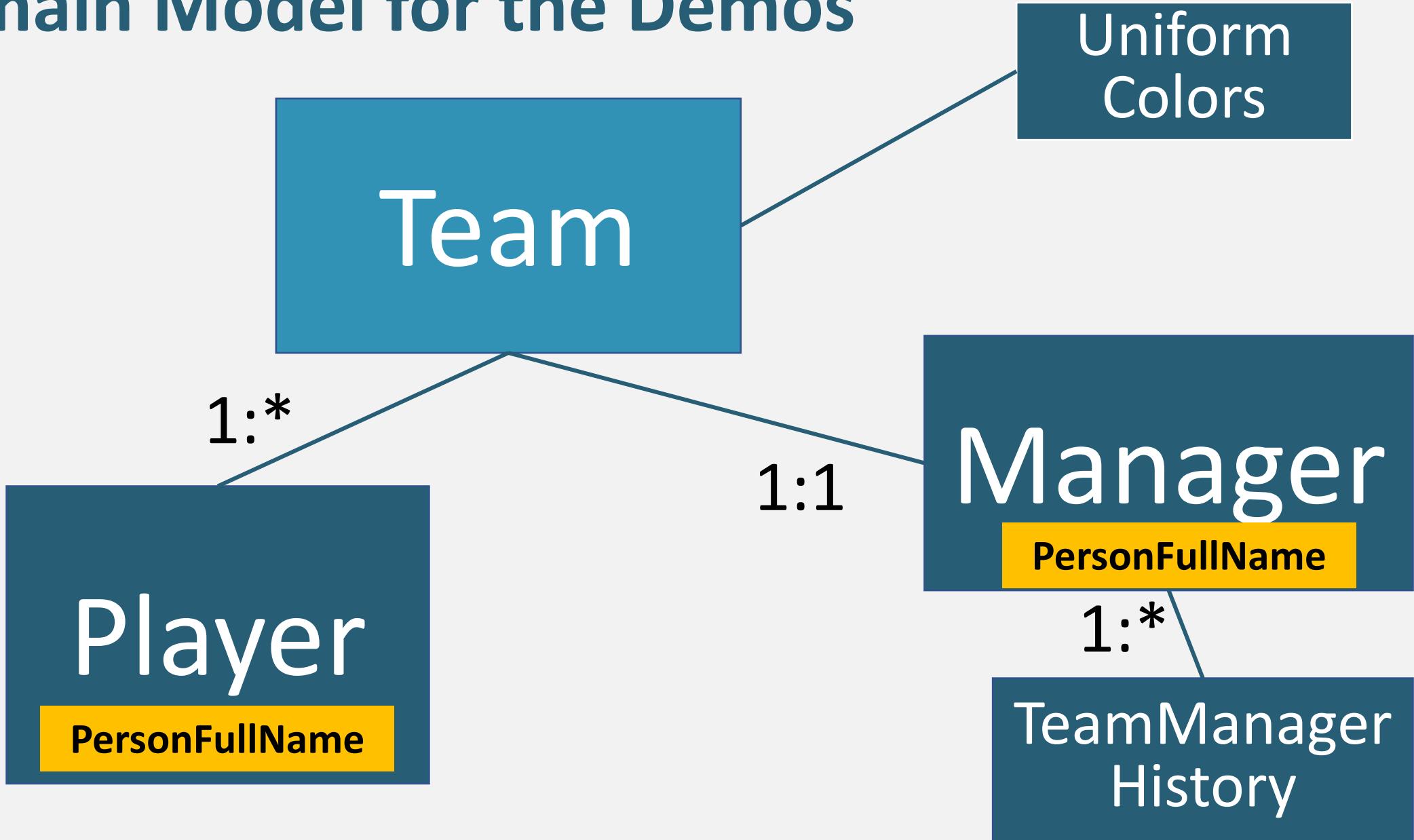
Key elements of Tactical Design



Source:

Domain-Driven Design, Eric Evans, Addison-Wesley, 2003

Domain Model for the Demos



Patterns You'll See

One-to-one by convention

Backing fields

IEnumerable collections allows encapsulation

Complex type support for mapping value objects

Value conversions

Shadow properties

More tricks:

- Encapsulate a one-to-one relationship

- Define a many-to-many

- Materialize [most] entities without a parameterless ctor

Build

`dotnet build`

Validate DbContext

`dotnet ef dbcontext info --startup-project/console`

Run Tests

`dotnet test`

Straight-forward 1-to-1

Now by convention

Navigation properties no longer
required on both end

Relationship via FK, not shared PK

Encapsulate Scalar Properties with backing fields

Encapsulate Collections

Map value Objects & complex types as EF Core “Owned Types”

Owned Type

No identity key

Used as a property of an entity

Value Object

Value Object Attributes

No identity key

Used as a property of an entity

Immutable

Equals compares all properties

GetHashCode for all properties

Owned Type Attributes

No identity key

Used as a property of an entity

Immutable

Equals compares all properties

GetHashCode for all properties

Encapsulating the One-to-One

Mapping Many-to-Many

Value Conversions

SHADOW PROPERTIES

Resources

Code & Slides github.com/julielman/2018DDDMappings (soon!)

My Pluralsight Author Page bit.ly/2t6m83w

- EF Core 2: Getting Start
- EF Core 2: Mappings,
- Domain-Driven Design Fundamentals
- Entity Framework in the Enterprise

MSDN Mag Data Points:

- DDD-Friendlier EF Core 2.0 (Sept & Oct 2017)
- EF Core 2 Owned Entities Temporary Work-Arounds (April 2018)
- EF Core 2.1 Query Types (July 2018)

ASP.Net Core: dot.net

EF Core: docs.microsoft.com/efcore