

# Entity Framework Core: Getting Started

---

INTRODUCING A NEW, LIGHTWEIGHT VERSION OF EF



**Julie Lerman**

MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK

@julielerman    thedatafarm.com



# Who Is This Course For?



New to EF Core



New to EF



Manager/CTO



# Course Overview

What is EF Core and what about EF6?

Creating a data model and database with EF Core

Using EF Core to interact with your data

Using EF Core in your applications

Using EF Core with cross-platform tools in multiple environments

# Module Overview



What is Entity Framework?

What is Entity Framework Core?

Why a fresh start and a new name?

EF Core, .NET Framework and .NET Core

Highlights of familiar features

Highlights of new features

What about EF6 in old and new projects?



# EF Core 1.1

## Visual Studio 2015 Update 3

*This course will get updated for future EF Core*

*There are minor updates throughout  
after VS2017 & .NET Core tooling release in March 2017*

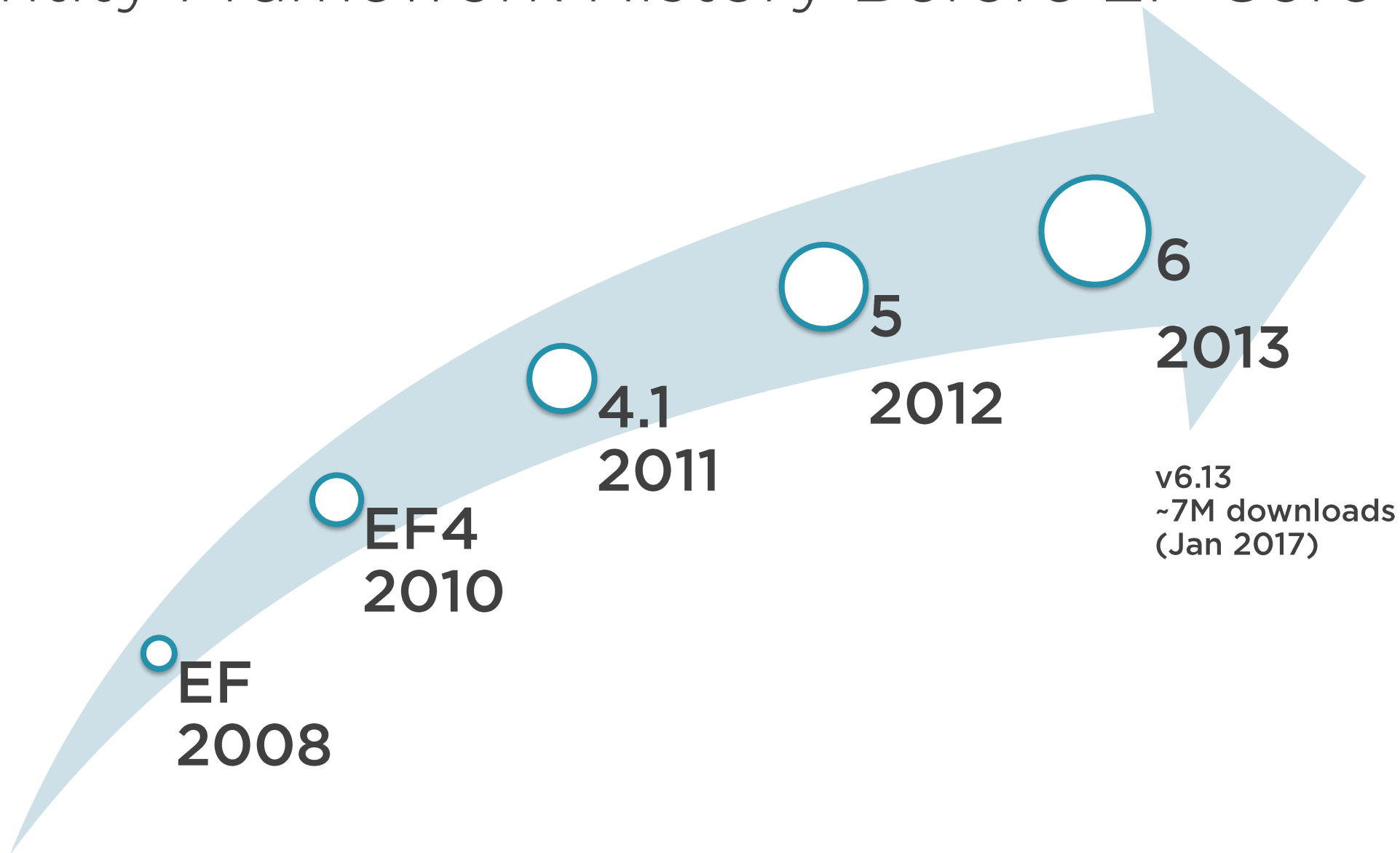


# Entity Framework

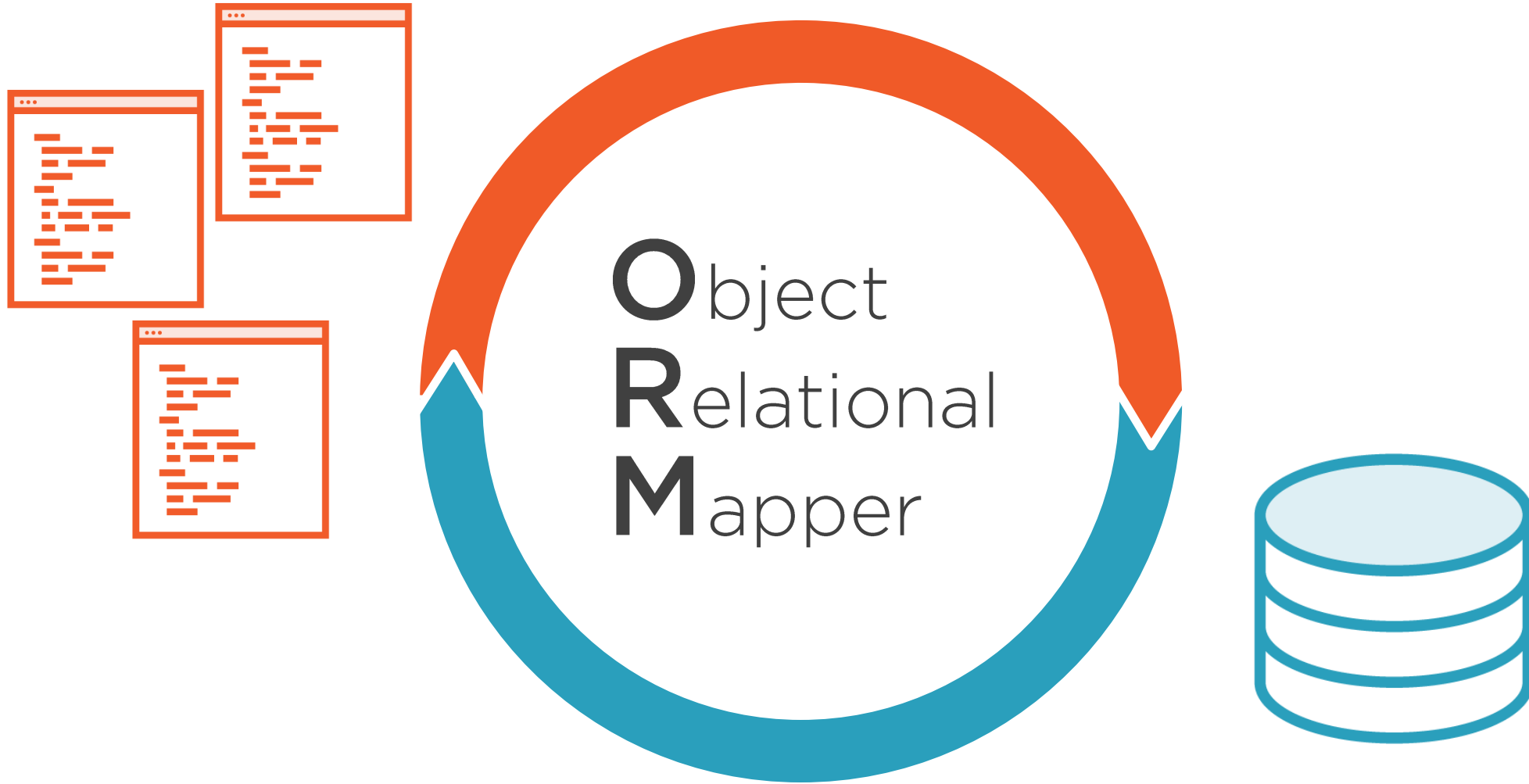
Microsoft's official data access technology  
for .NET development



# Entity Framework History Before EF Core



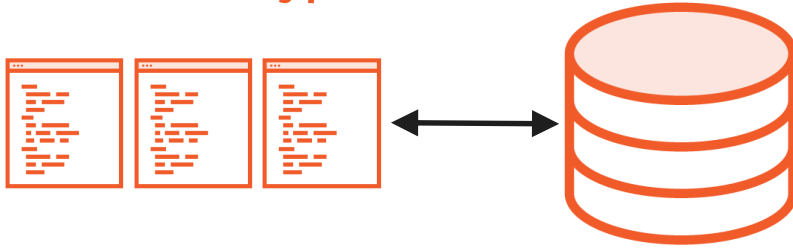
# EF Is an ORM



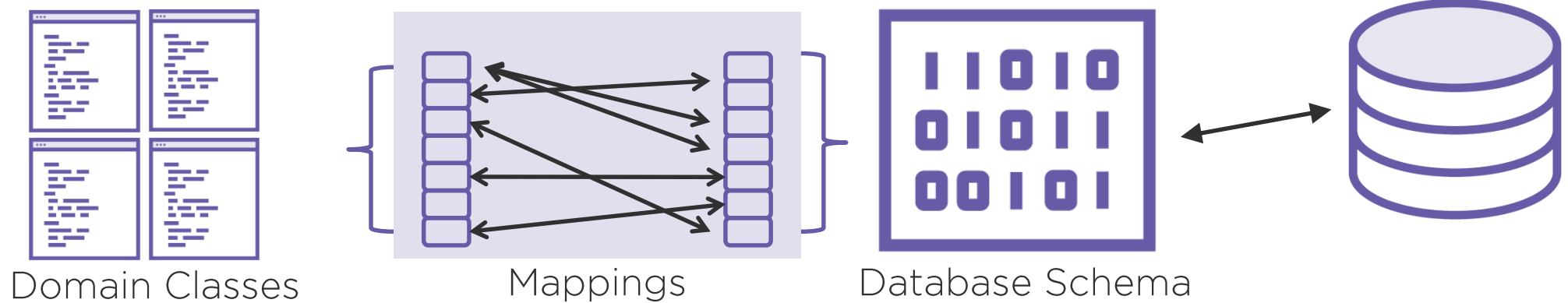


# EF Maps Differently Than Most ORMs

## Typical ORM



## Entity Framework



# Why Entity Framework?

Developer productivity

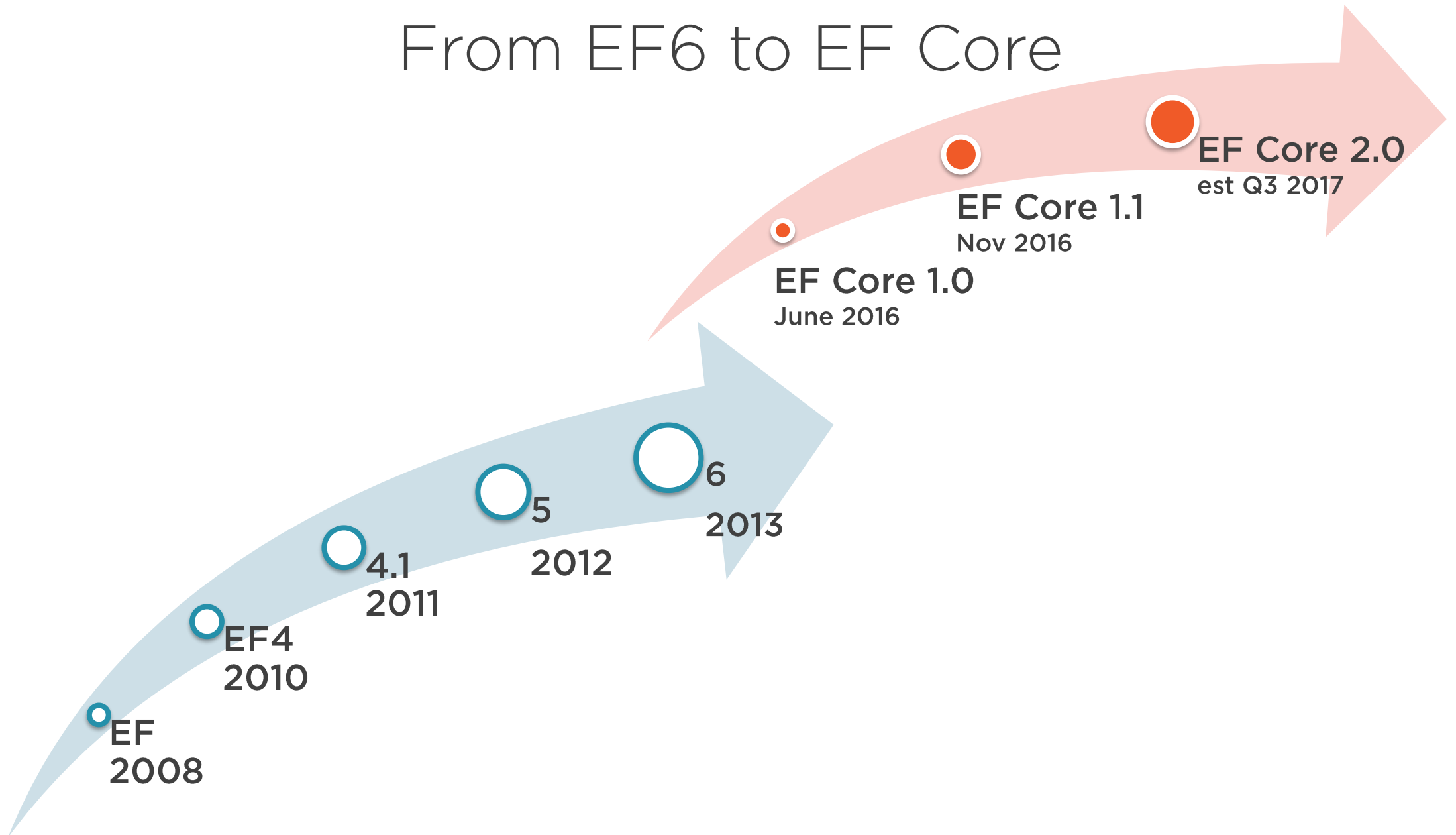
First class member  
of  
Microsoft .NET stack

Consistent query syntax  
with  
LINQ to Entities

Focus on domain,  
not on DB, connections,  
commands, etc.



# From EF6 to EF Core



**“Entity Framework (EF) Core** is a lightweight and extensible version of the popular Entity Framework data access technology.”

**EF Core Documentation**



Re-written from scratch.



No more legacy code  
holding back innovation.



No more staples, tape or  
“adhesive bandages”



# .NET Core Themes



Composable APIs



Cross-platform

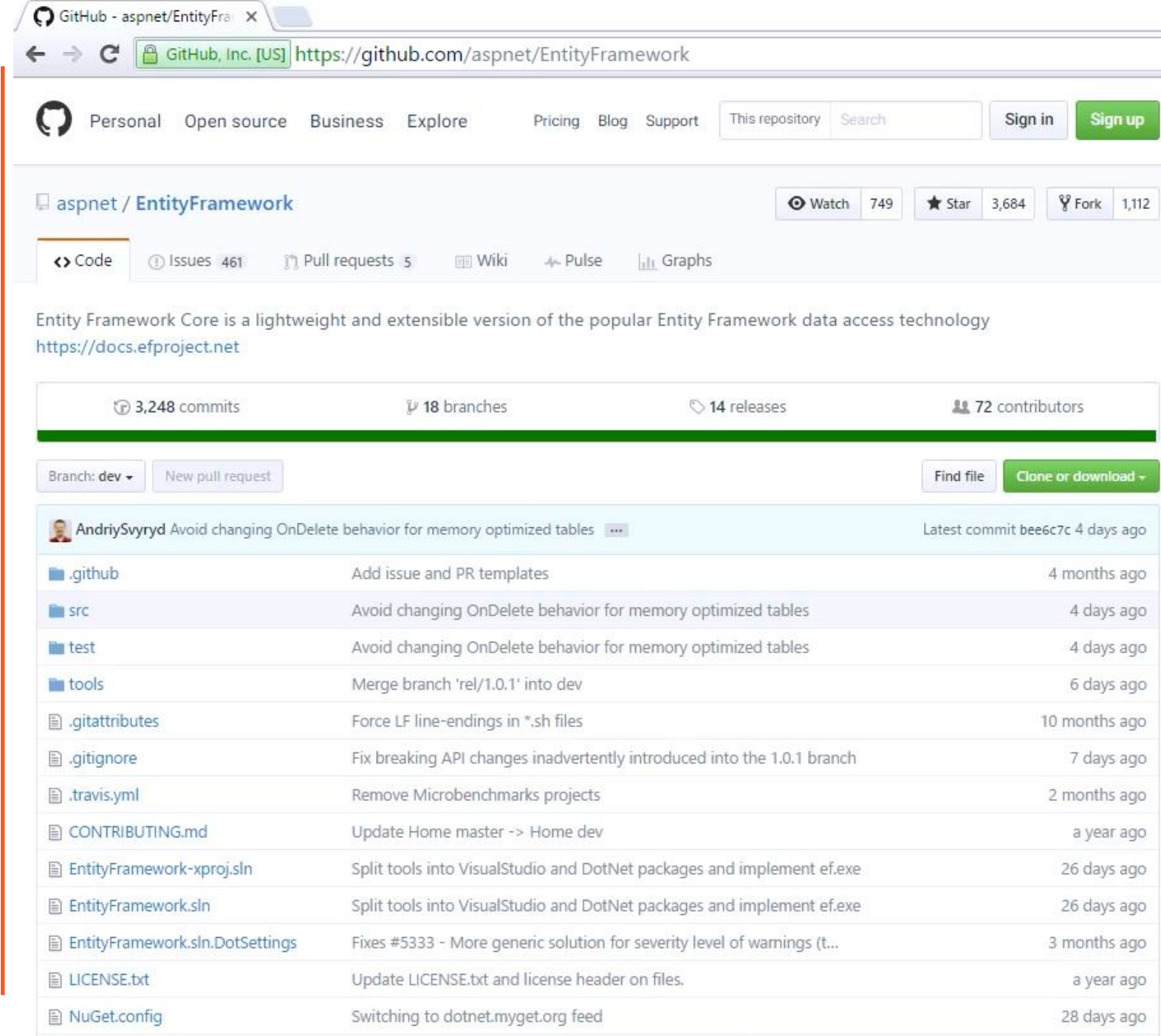


Modern  
Software Practices





# Open Source



GitHub - aspnet/EntityFramework

Personal Open source Business Explore Pricing Blog Support This repository Search Sign in Sign up

aspnet / EntityFramework

Watch 749 Star 3,684 Fork 1,112

Code Issues 461 Pull requests 5 Wiki Pulse Graphs

Entity Framework Core is a lightweight and extensible version of the popular Entity Framework data access technology  
<https://docs.efproject.net>

3,248 commits 18 branches 14 releases 72 contributors

Branch: dev New pull request Find file Clone or download

AndriySvyryd Avoid changing OnDelete behavior for memory optimized tables Latest commit b6e6c7c 4 days ago

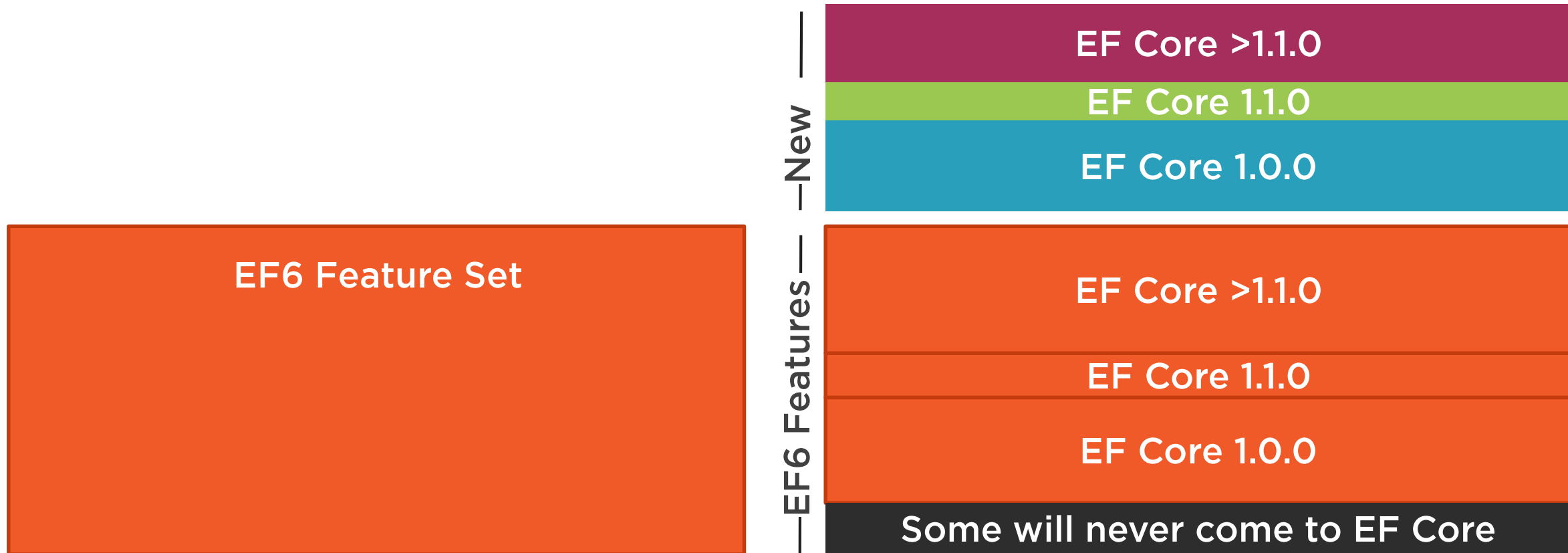
.github	Add issue and PR templates	4 months ago
src	Avoid changing OnDelete behavior for memory optimized tables	4 days ago
test	Avoid changing OnDelete behavior for memory optimized tables	4 days ago
tools	Merge branch 'rel/1.0.1' into dev	6 days ago
.gitattributes	Force LF line-endings in *.sh files	10 months ago
.gitignore	Fix breaking API changes inadvertently introduced into the 1.0.1 branch	7 days ago
.travis.yml	Remove Microbenchmarks projects	2 months ago
CONTRIBUTING.md	Update Home master -> Home dev	a year ago
EntityFramework-xproj.sln	Split tools into VisualStudio and DotNet packages and implement ef.exe	26 days ago
EntityFramework.sln	Split tools into VisualStudio and DotNet packages and implement ef.exe	26 days ago
EntityFramework.sln.DotSettings	Fixes #5333 - More generic solution for severity level of warnings (t...	3 months ago
LICENSE.txt	Update LICENSE.txt and license header on files.	a year ago
NuGet.config	Switching to dotnet.myget.org feed	28 days ago

# Where *Can* You Use EF Core?

		.NET 4.5.1+	.NET Core	UWP
OS		Windows	Windows, macOS, Linux	Windows 10
App Types		<i>Any .NET 4.5.1+</i> Console WinForms WPF ASP.NET 4 etc..	ASP.NET Core Web Apps & APIs Libraries Services Console Apps	Mobile PC HoloLens Xbox Surface Hub IoT Devices



# EF6 Features vs EF Core Features




For .NET Apps,  
use EF Core only if it has all  
of the features you need.

Otherwise, use EF6



# ASP.NET Core & EF6

 | Developer Network ▾


magazine ▾ Issues and downloads ▾ Subscribe Submit article

Issues and downloads / 2016 / February 2016 / Data Points - Refactoring an ASP.NET 5/EF6 Project and Dependency Injection

FEBRUARY 2016

Data Points - Refactoring an ASP.NET 5/EF6 Project and Dependency Injection

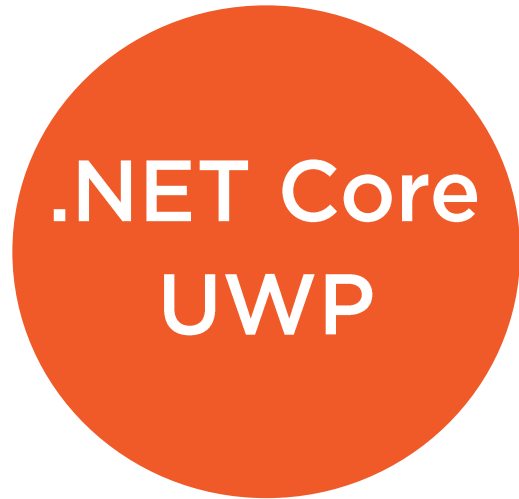
By [Julie Lerman](#) | February 2016 | [Get the Code](#)



Dependency injection (DI) is all about loose coupling ([bit.ly/1TZWWtW](http://bit.ly/1TZWWtW)). Rather than hardcoding classes you depend on into other classes, you request them from somewhere else, ideally your class's constructor. This follows the Explicit Dependencies Principle, more clearly informing users of your class about the collaborators it requires. It also allows you to build more flexibility into your software for scenarios such as alternate configurations of a class's object instance, and it's really beneficial for writing automated tests for such classes. In my world, which is fraught with Entity Framework code, a typical example of coding without loose coupling is creating a repository or controller that instantiates a DbContext directly. I've done this thousands of times. In fact, my goal with this article is to apply what I've learned about DI to the code I wrote in my column, "The EF6, EF7 and ASP.NET 5 Soup" ([msdn.com/magazine/dn973011](http://msdn.com/magazine/dn973011)). For example, here's a method where I instantiated a DbContext directly:

```
public List<Ninja> GetAllNinjas() {
```





EF Core



EF Core  
EF6




# EF6 Features Not Coming to EF Core

---




# No EDMX/Designer in EF Core



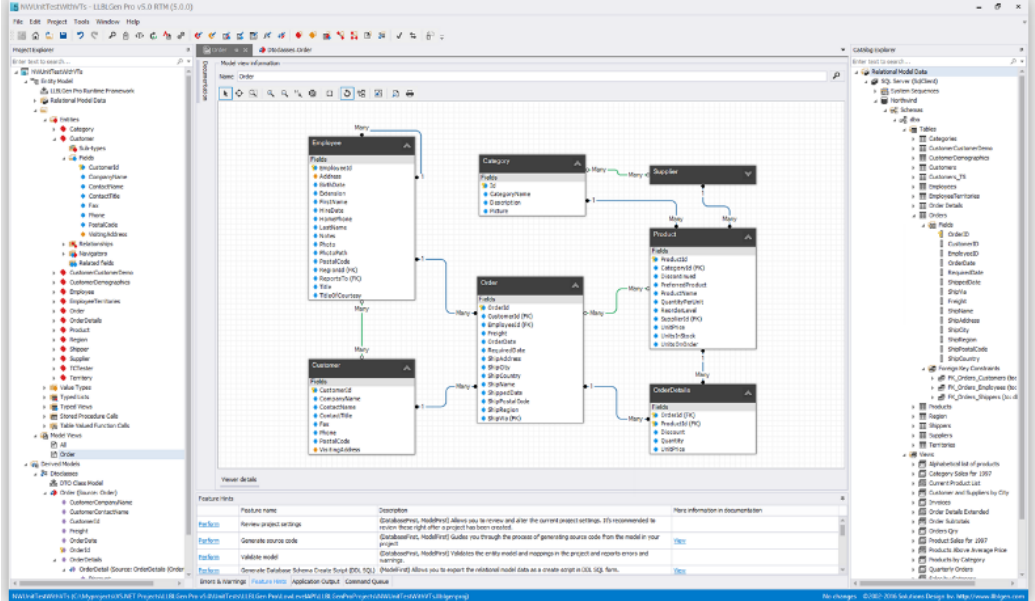
**Entity Developer**

First-class ORM model designer for NHibernate, Entity Framework, LINQ to SQL and Telerik Data Access



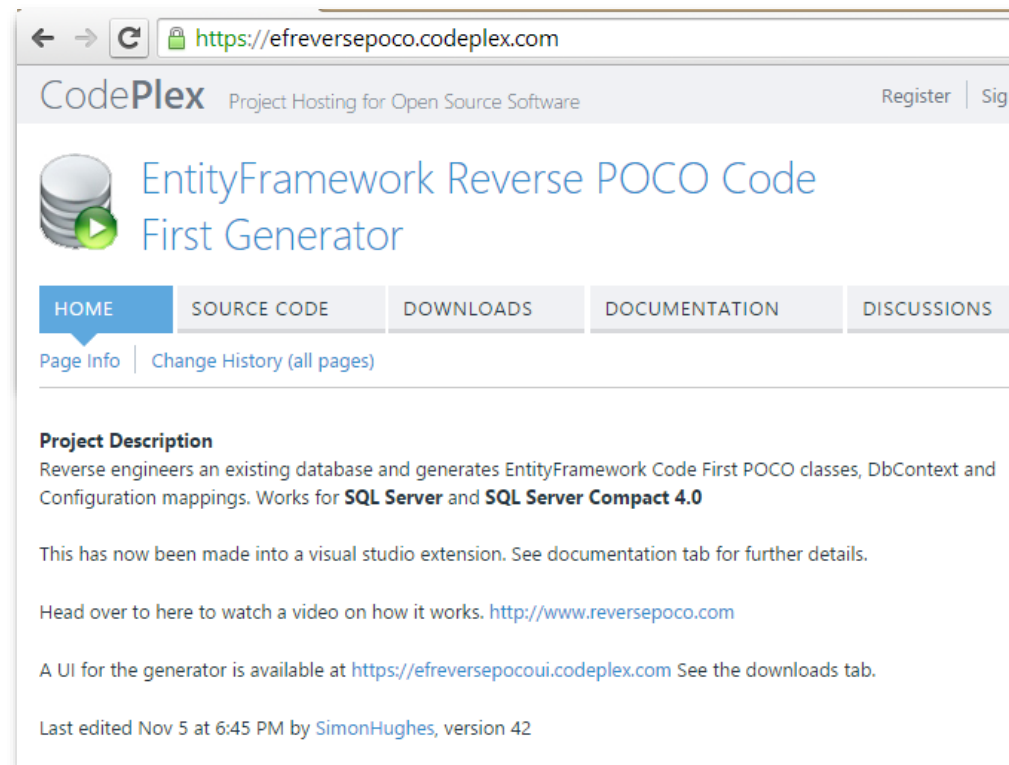
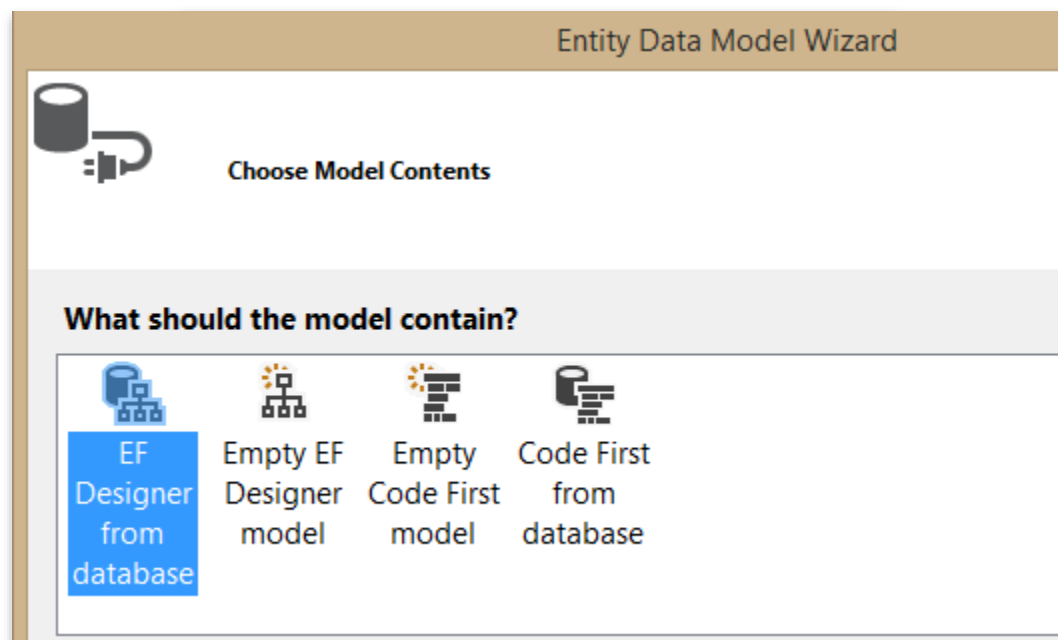
Buy Try Product Information Resources Support

Use any .NET ORM framework

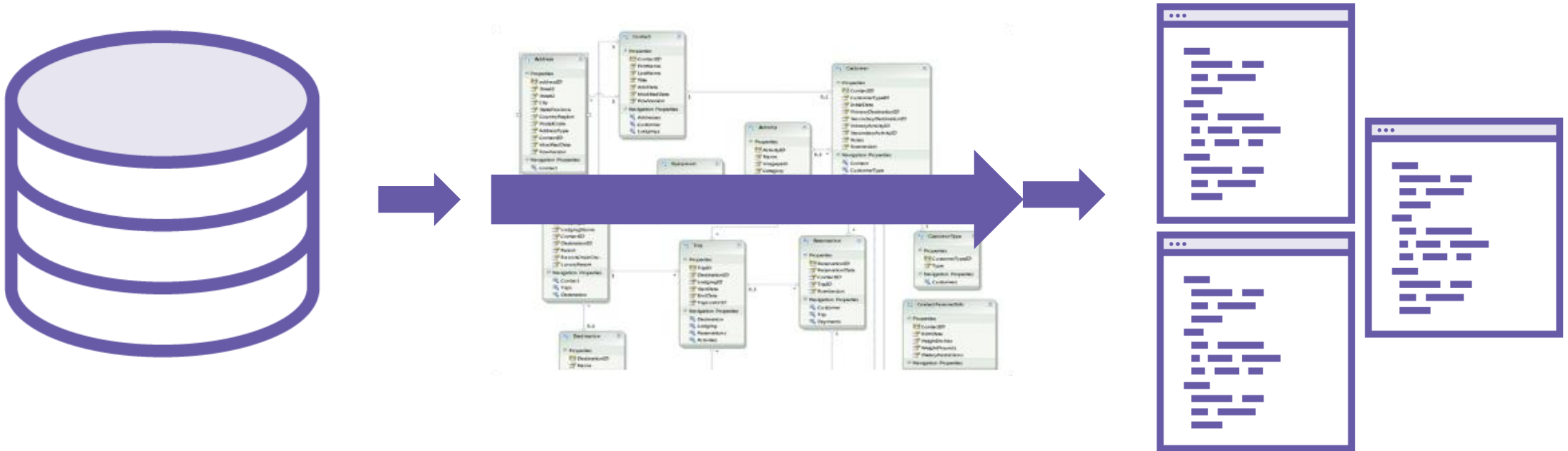




# No EDMX/Designer in EF Core

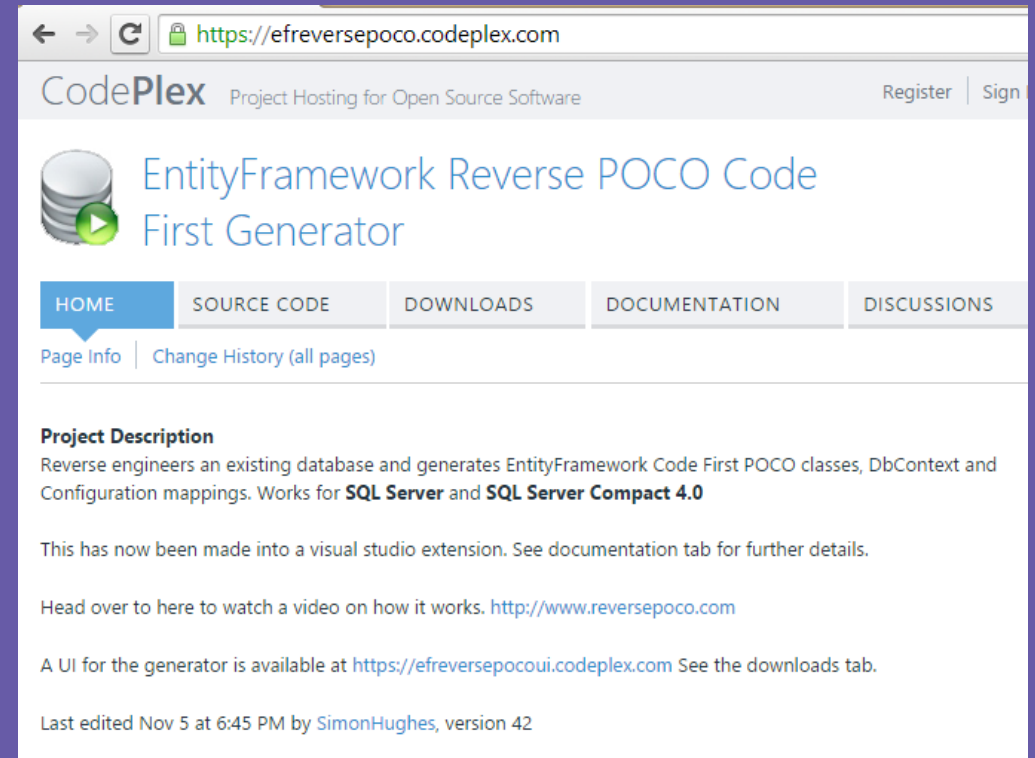
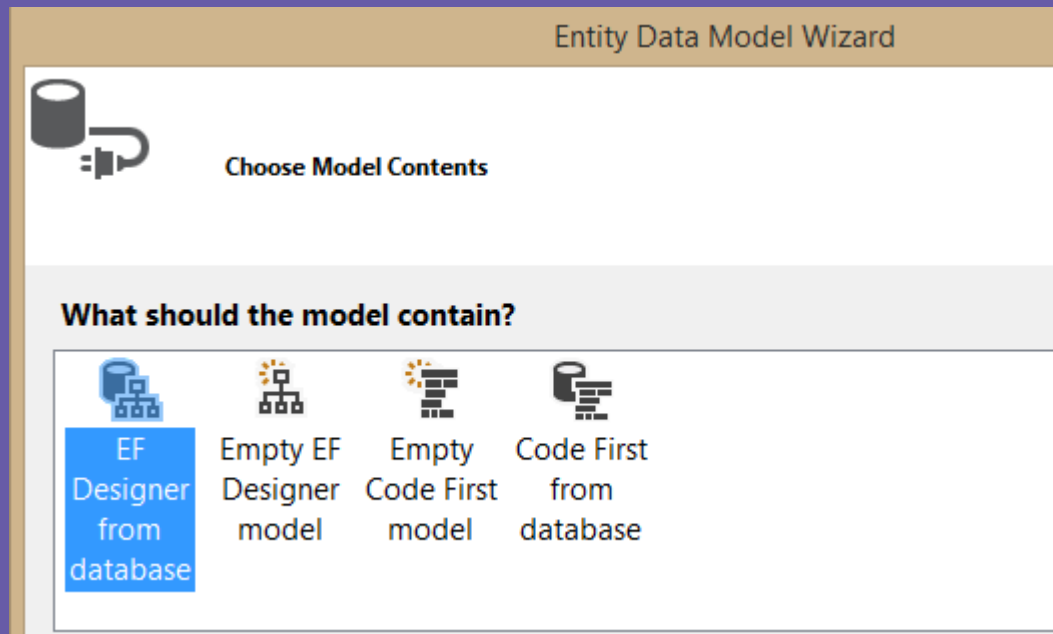


# No Designer? What About Database First?



# No Designer? What About Database First?

## Reverse Engineer to “Code First” in EF6



Reverse Engineer in EF Core

# EF Core Migrations

**scaffold** command



# Biggest Planned Feature Cuts

**EDMX Support**

**ObjectContext API**

**Entity SQL**

**Metadata Workspace API**

**Overly complex mappings**

**MEST\* mapping**

***Automatic Migrations***

\*Multiple Entities for Single Type

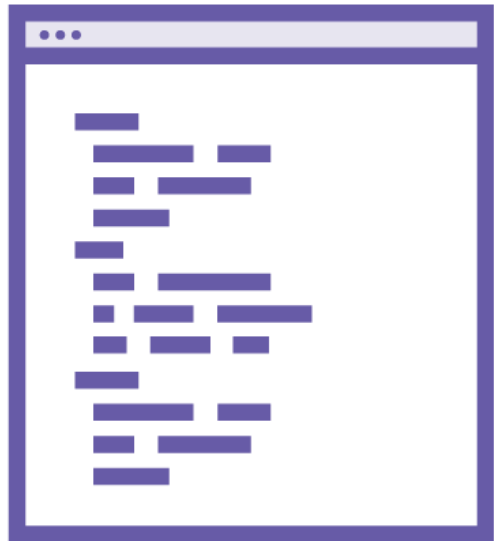


# How EF Works

---



# EF Puzzle Pieces



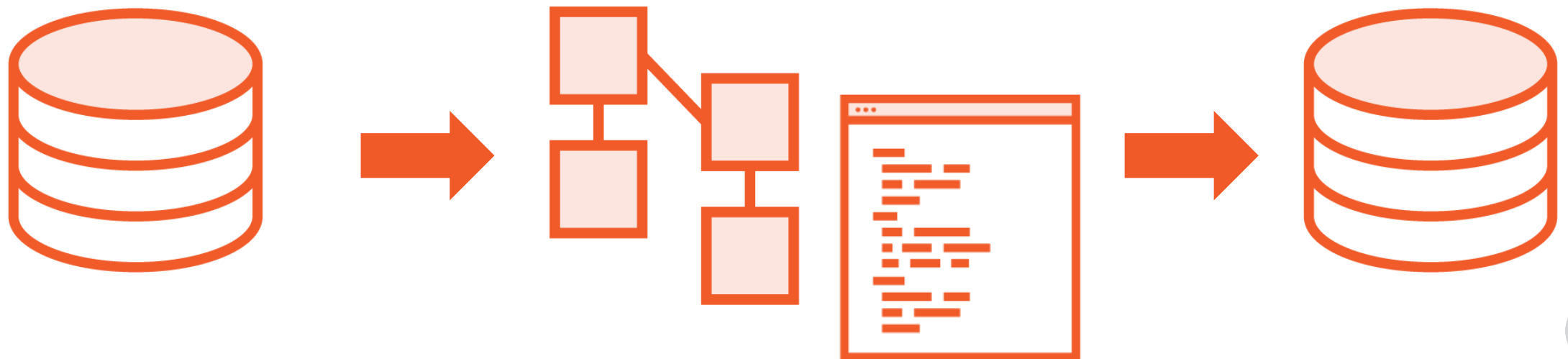
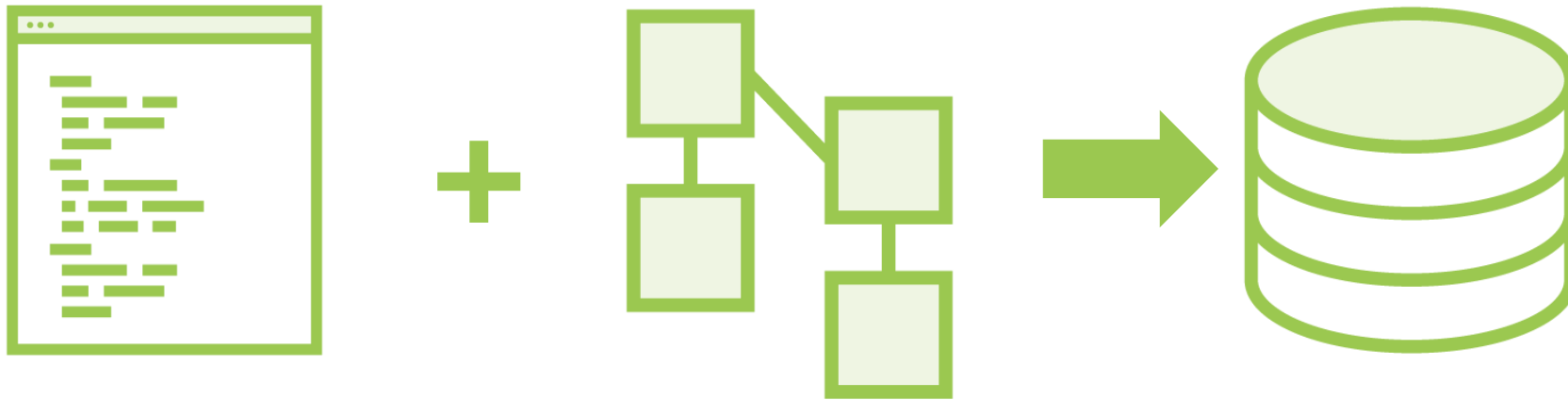
Data model  
LINQ queries  
Save changes



Track changes  
Build SQL  
Execute SQL  
Create objects



# Model Creation Options





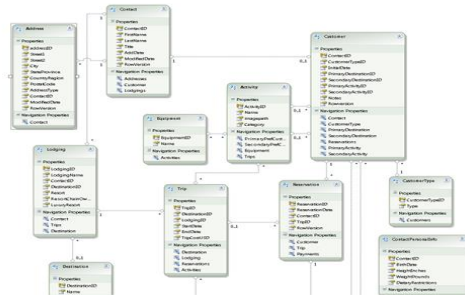
# Mapping Classes & Properties to Database



<pre>public class Customer {     public int Id {get;set;}     public string FirstName {get;set;}     public DateTime DateOfBirth {get;set;} }</pre>	<p>← Relies on Data Model/Custom →</p> <p>← Convention →</p> <p>← Custom Mapping →</p> <p>← Convention →</p>	<p>Table Name: Customers</p> <p>Id (PK, int, not null)</p> <p>First_Name(nvarchar(30),not null)</p> <p>DateOfBirth(date, not null)</p>
---	--	--

# Basic Workflow

Define model



Express & execute query

(from p in people select p)  
.ToList()

EF determines  
& executes SQL

**SELECT \* from people**

EF transforms results  
into your types

3	Ms.	Donnie	F.	Carreras
4	Ms.	Janet	M.	Gates
5	Mr.	Lucy	NULL	Harrington
6	Mr.	Joop	X.	Carroll
7	Mr.	Dominic	P.	Gash
10	Ms.	Kathleen	M.	Garza
11	Ms.	Kathleen	NULL	Harding
12	Mr.	Johnny	A.	Caprio
16	Mr.	Christopher	R.	Beck
18	Mr.	David	J.	Liu
19	Mr.	John	A.	Beaver

User modifies data



Code triggers save

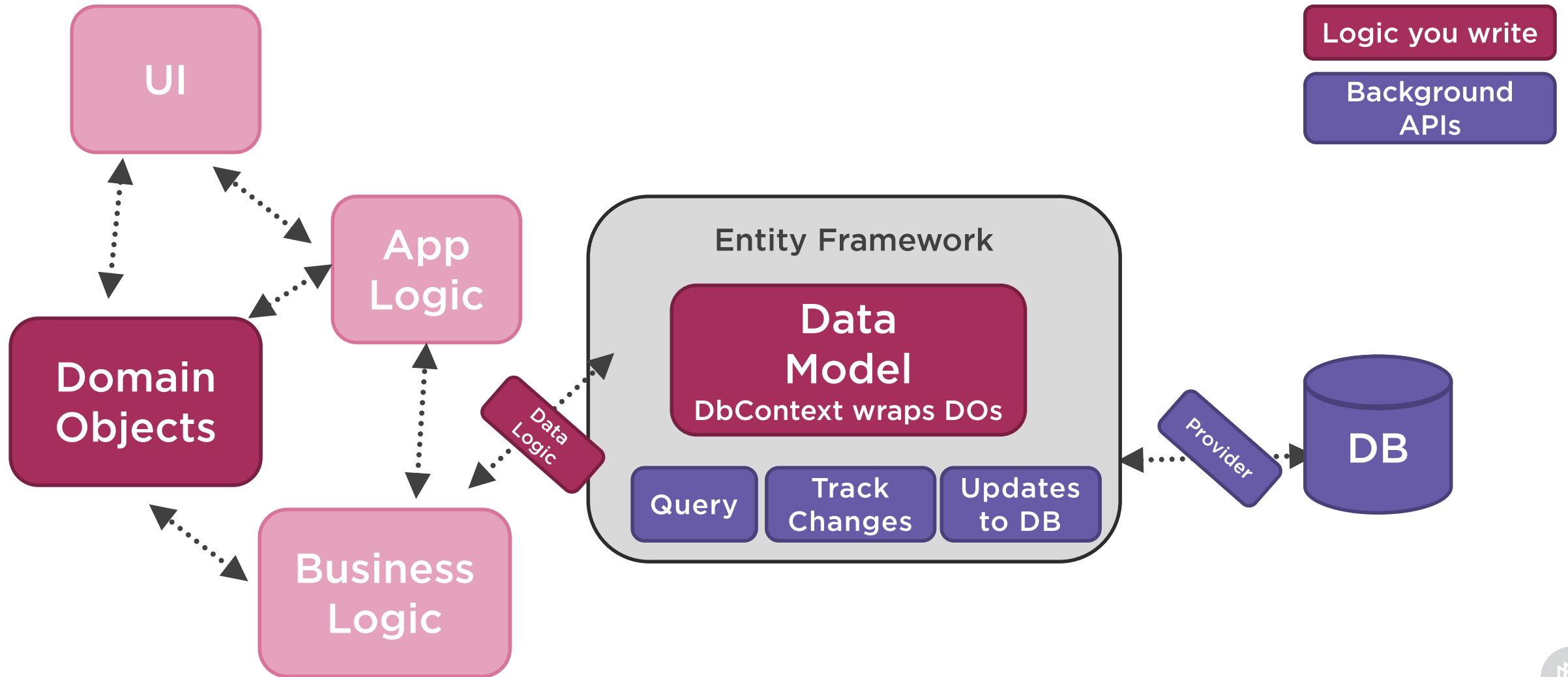
**DbContext.SaveChanges**

EF determines & executes SQL

**UPDATE people  
SET Firstname='Julie'  
WHERE id=3**



# EF in Your Software Architecture



# EF Core Is Still Entity Framework

Data Model

DbContext API

Migrations

LINQ to Entities

Change Tracking

SaveChanges



# EF Core Enables Long-requested Features

Batch Insert,  
Update,  
Delete

Unique  
constraints

Smarter  
queries

Smarter  
disconnected  
patterns

In-Memory  
provider for  
testing

Extensible &  
IoC friendly

Mapping to  
backing fields  
& IEnumerableables

Smarter &  
simpler  
mapping

More at [data.uservice.com](http://data.uservice.com)



[bit.ly/EFCoreRoadmap](https://bit.ly/EFCoreRoadmap)



# What About EF6?

---
















EF6 or EF Core?

**Starting a  
new app**

**Want to  
update an  
old app**





	EF6	EF Core 1.1
Production Ready		
Actively Evolving		
Visual Designer		*expect 3 <sup>rd</sup> party support
Backwards Compatible		
Full .NET Support		 *4.5.1 +
.NET Core Support		
Lightweight API(s)		
Better APIs, New Features		
Non-Relational Data		*future support
Open-Source (on GitHub)		



For .NET apps,  
use EF Core only if it has all  
of the features you need.

Otherwise, use EF6



# Updating EF6 to EF Core





Updating EF6 to EF Core

# Existing Code

EF6

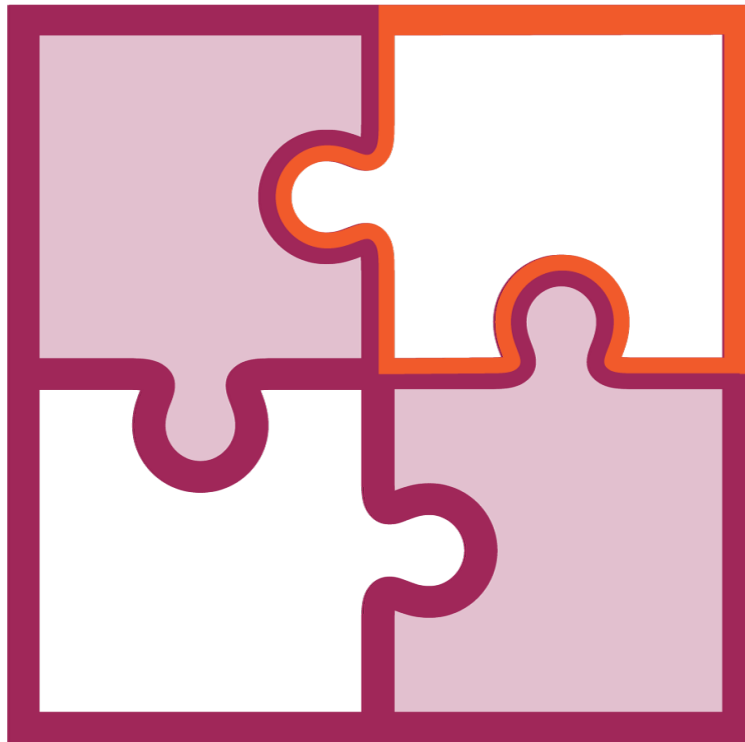
EF Core



Build with EF6,  
keeping EF Core in mind



# Isolate and Update Parts of Your App



```
public Thing GetThatThing()  
{  
    return Data.GetThatThing();  
}
```

IData



EF6.Data : IData



EFCore.Data : IData





# Review

Entity Framework Core overview

Why a fresh start and a new name?

EF Core vs EF6

Highlights of new features

Highlights of familiar features

EF Core, .NET Framework and .NET Core

# Resources

Pluralsight: Domain-Driven Design Fundamentals [bit.ly/PS-DDD](https://bit.ly/PS-DDD)  
Entity Framework in the Enterprise [bit.ly/PS-EFEnt](https://bit.ly/PS-EFEnt)

Entity Framework Core on GitHub [github.com/aspnet/entityframework](https://github.com/aspnet/entityframework)  
Entity Framework 6 on GitHub [github.com/aspnet/entityframework6](https://github.com/aspnet/entityframework6)

EF Core Roadmap [bit.ly/efcoreroadmap](https://bit.ly/efcoreroadmap)

Data Points - EF Core Change-Tracking Behavior: Unchanged, Modified and Added, Aug 2016  
<http://msdn.microsoft.com/magazine/mt767693>

Data Points - Refactoring an ASP.NET 5/EF6 Project and Dependency Injection,  
Feb 2016 <http://msdn.microsoft.com/magazine/mt632269>

ASP.NET 5 is dead - Introducing ASP.NET Core 1.0 and .NET Core 1.0 by Scott Hanselman  
[hanselman.com/blog/ASPNET5IsDeadIntroducingASPNETCore10AndNETCore10.aspx](http://hanselman.com/blog/ASPNET5IsDeadIntroducingASPNETCore10AndNETCore10.aspx)

Some Insights into Features Being Dropped in the Move From EF6 to EF Core  
[thedatafarm.com/data-access/droppedfromefcore](http://thedatafarm.com/data-access/droppedfromefcore)



# Entity Framework Core: Getting Started

---

INTRODUCING A NEW, LIGHTWEIGHT VERSION OF EF



**Julie Lerman**

MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK

@julielerman    thedatafarm.com

