TECHORAMA
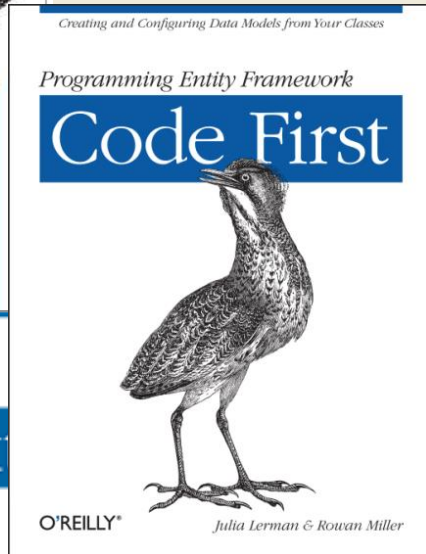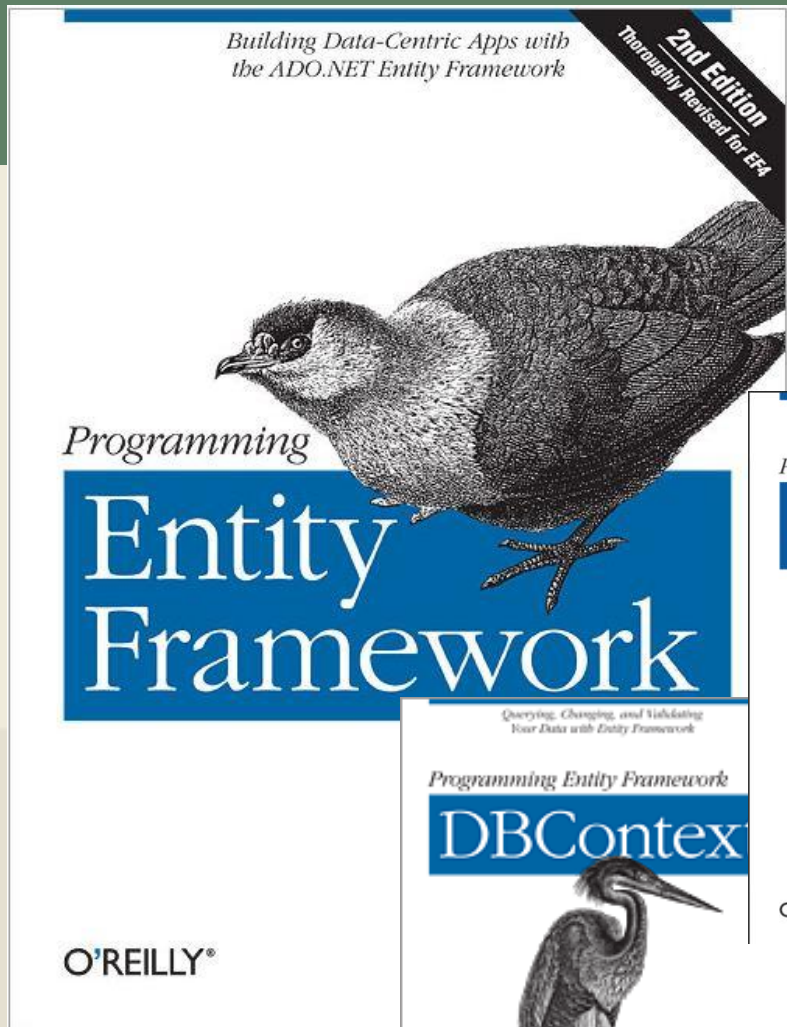
MAY 12-13 2015

Entity Framework 7
Who Are You & What Have You Done to my ORM?

Julie Lerman @julielerman thedatafarm.com

2nd Edition
Thoroughly Revised for EF4

Building Data-Centric Apps with the ADO.NET Entity Framework

Programming
Entity Framework

O'REILLY®

Programming Entity Framework
DBContext

Querying, Changing, and Validating Your Data with Entity Framework

O'REILLY®
Julia Lerman & Rowan Miller

Creating and Configuring Data Models from Your Classes

Programming Entity Framework
Code First

O'REILLY®
Julia Lerman & Rowan Miller

My Courses on
pluralsight

- **Domain-Driven Design Fundamentals**
- **Looking Ahead to Entity Framework 7**
- **EF6 Ninja Edition: What's New in EF6**
- **Automated Testing for Fraidy Cats Like Me**
- **Getting Started with Entity Framework 5**
- **Entity Framework in the Enterprise**
- **Entity Framework Code First Migrations**
- **Data Layer Validation with Entity Framework 4.1+**
- **Entity Framework 4.1 - DbContext Data Access**
- **Entity Framework 4.1 - Code First**
- **Querying the Entity Framework**
- **Designer Supported EDM Customization**
- **Entity Framework and Data Models**
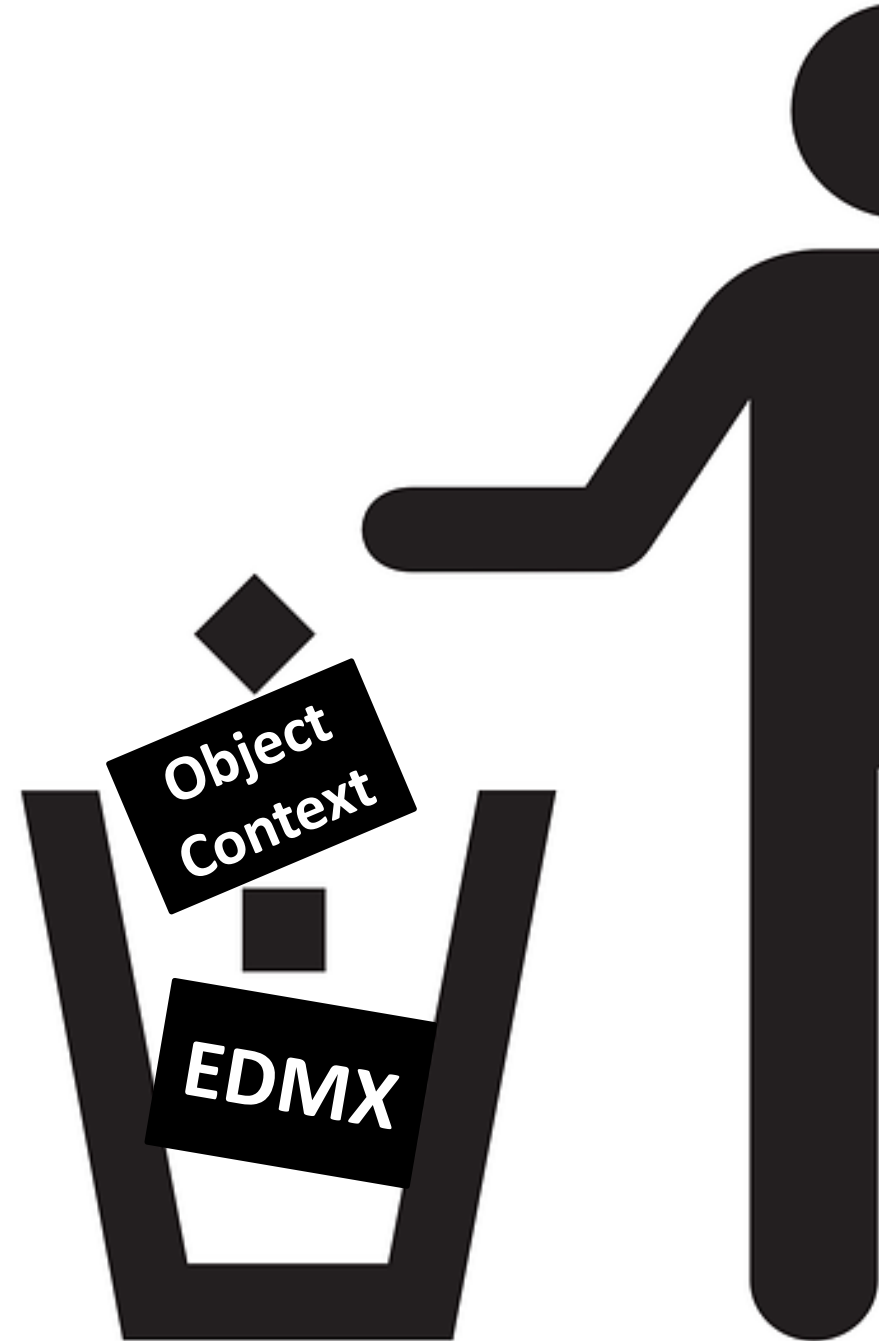- **Entity Framework 4.0 By Example**

TECHORAMA

@julielerman

so much technical debt!

@julielerman

A complete rewrite

Familiar APIs

Better APIs

Not backwards compatible

@julielerman

"New Platforms
New Data Stores"
Features We've Longed For

Object Context

EDMX

Expected Progression of EF7

@julielerman

EF7 Core

SQL Server provider for EF7

Support for Relational providers

Redis provider for EF7

System. Interactive.Async

In-Memory provider for EF7

Some other RDBMS Provider

Azure Table Storage provider for EF7

System.Data. Common

Support for Migrations

Some NoSQL Provider

Some Other NoSQL Provider
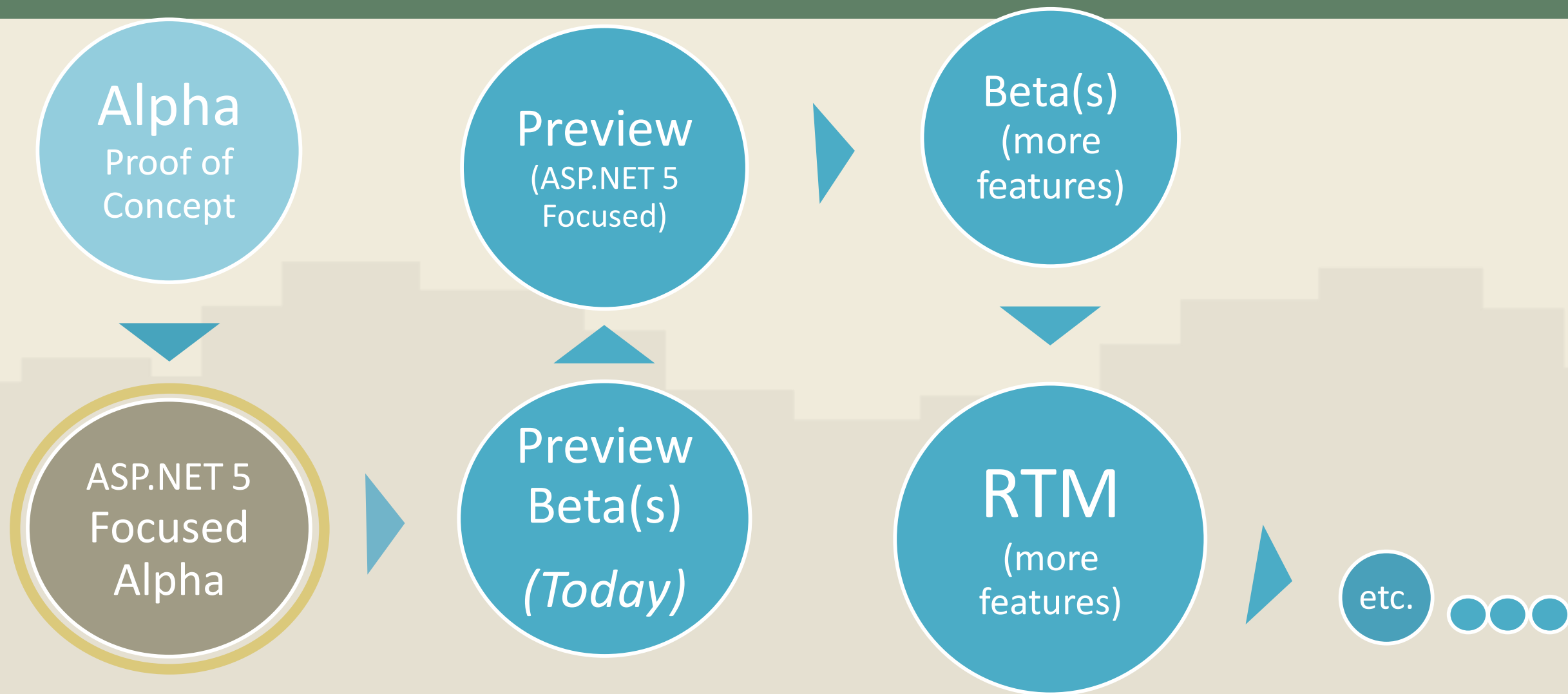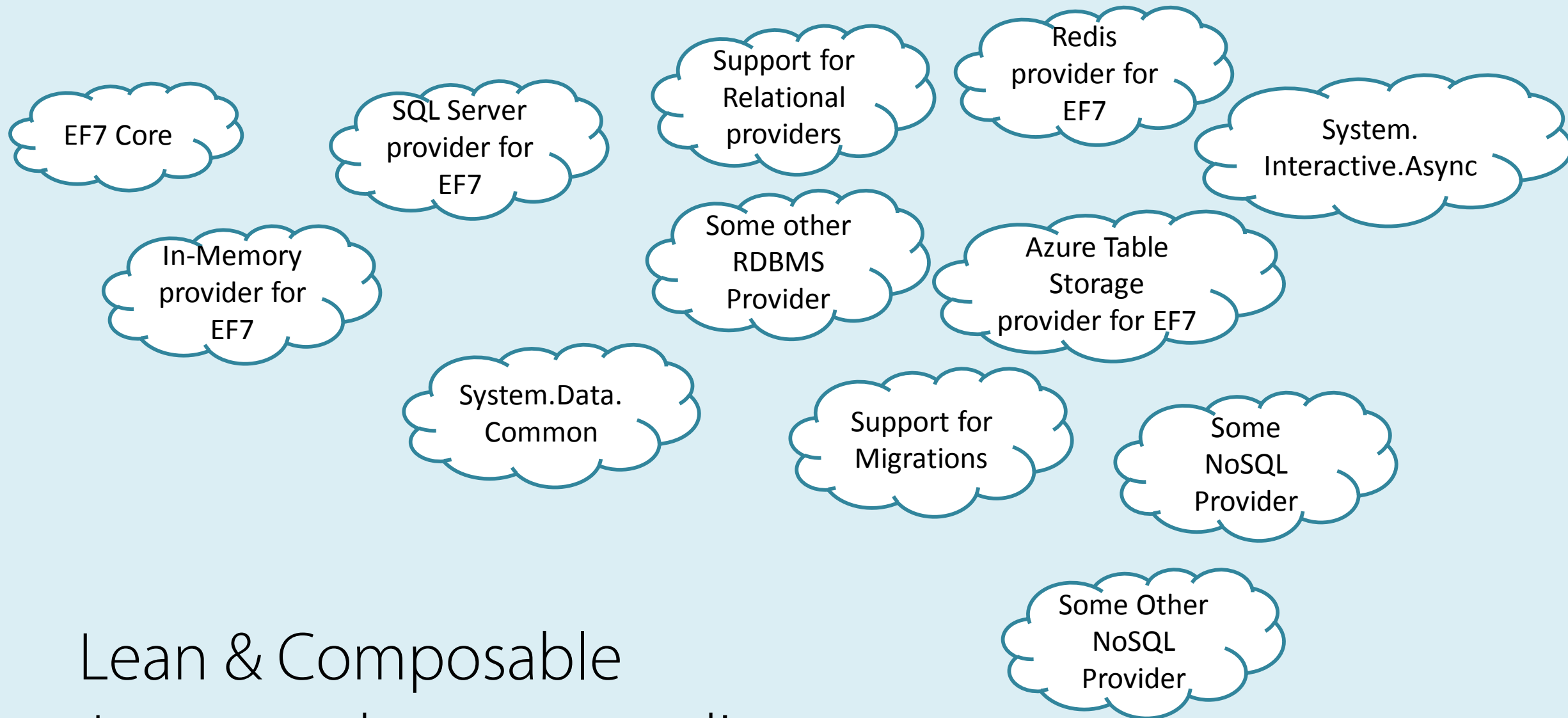
Lean & Composable
Just get what you need!

How We Got Here

Client Side
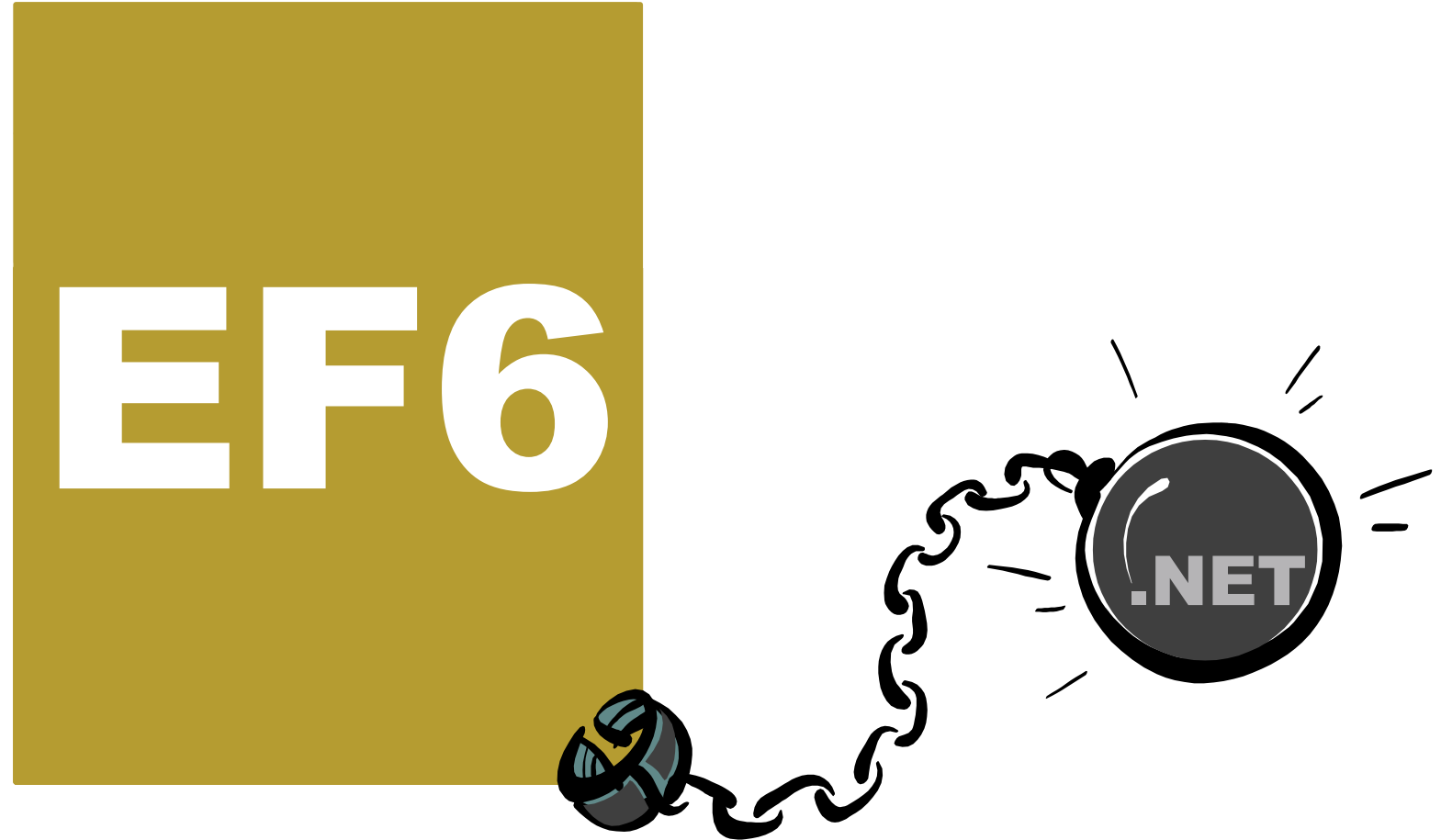
Console

WPF

Windows Form

Server Side

ASP.NET Web Forms

ASP.NET MVC

ASP.NET WebAPI

WCF Data Services

EF6
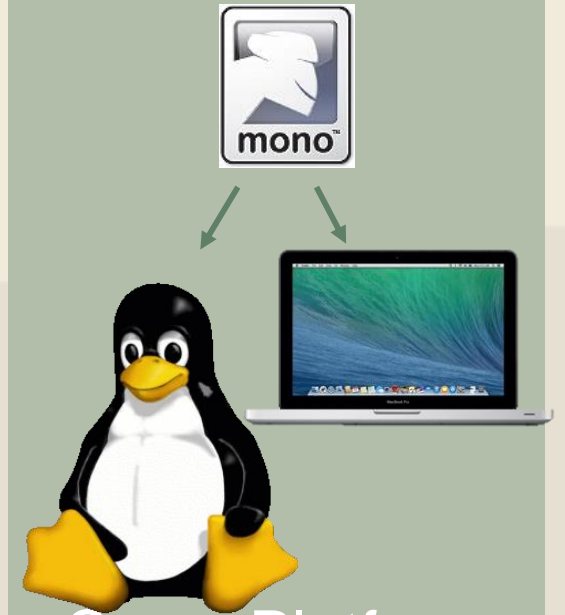
.NET

EF7 Won't Require Full .NET Framework

Windows Phone

Windows Store

ASP.NET 5
W10 UAP

Core CLR

mono

Cross Platform
CLR

& .NET Framework

TECHORAMA

Data Stores

Relational Databases

Adding SQLite for Device Support
Adding PostgreSQL for Mac & Linux Support

Non-Relational Providers

NoSQL (proof of concept: Azure Table Storage)

In-Memory (for integration testing w EF, not DB)

Redis Cache (proof of concept)

More . . .

TECHORAMA

Familiar APIs

Entity Model

DbContext API

Migrations

LINQ to Entities

Change Tracking

SaveChanges

@julielerman

Familiar APIs & Behavior

Demo

Enable Long-Requested Features

Batch CUD	Unique Constraints*	More Provider Control of Queries	Shadow State for Entities
In-Memory Provider for Fakes & Test Doubles	IOC Friendly	Type Conversions	Default Database Values

More @ data.uservoice.com

TECHORAMA

@julielerman

EF7 Coding Enhancements

Demo

@julielerman

Disconnected Graphs in EF7

Affect Root Only

DbContext.Add/Range, AddAsync, Attach, Update, Remove
DbSet.Add/Range, AddAsync, Attach, Remove, Update
DbContext.Entry(entity).SetState, SetStateAsync

Affect Full Graph

DbContext.ChangeTracker.AttachGraph(entity) /Async
DbContext.ChangeTracker.AttachGraph(entity, Action) /Async

TECHORAMA

Biggest Planned Feature Cuts

EDMX Support

ObjectContext API

Entity SQL

Metadata Workspace API

Overly Complex Mappings

MEST (Multiple Entities for Single Type) Mapping

Automatic Migrations

@julielerman

Lighter Weight API

Code Based Modeling Only
Reverse engineer database still supported

DbContext API Only
Lower level functions from ObjectContext will be in DbContext

Simpler Metadata API
Losing some edge case mapping capabilities

@julielerman

What About EF6?

- EF6 will be around for a long time
- Supported & maintained by EF Team
- Open Source so Community Contribs invited
- I'm investing in EF6 w a new Pluralsight course
- EF6 & EF7 will continue in parallel although most innovation will happen in EF7

TECHORAMA

What About My Existing EF5/EF6 Projects?

- You are not required to upgrade

- Moving existing code may take some effort

- Only move if you have a compelling reason

- Consider moving only a subset if possible
  – Move that into separate project(s)
  – Might mean breaking a model apart

TECHORAMA

What if We Are Starting a New Project Soon?

- (Soon == while EF7 is still an alpha*)
- Use EF6
- Use DbContext
- Use Code First
- Keep logic as close to EF7 patterns as possible
- Build it with transition to EF7 in mind

*This guidance was written in May 2015

TECHORAMA

Github.com/aspnet/entityframework

My Pluralsight course on EF7:
bit.ly/PS-EF7Alpha

TECHORAMA