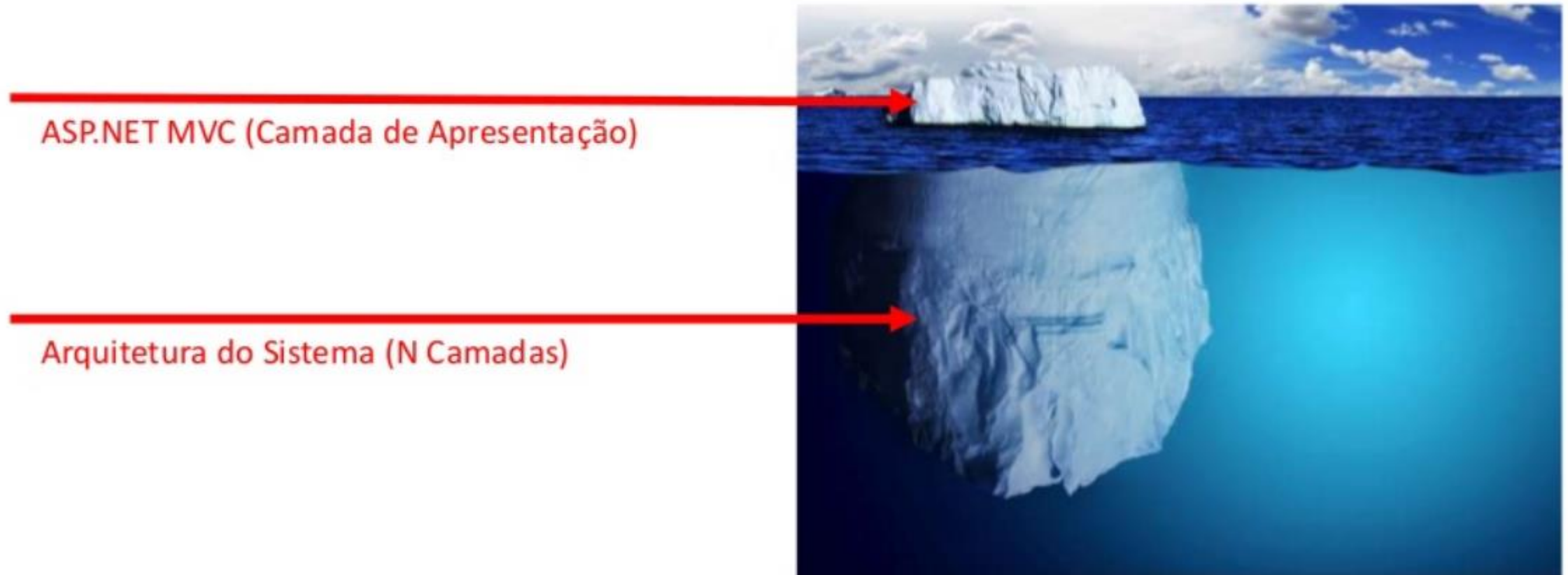


# Arquitetura de Aplicações

---

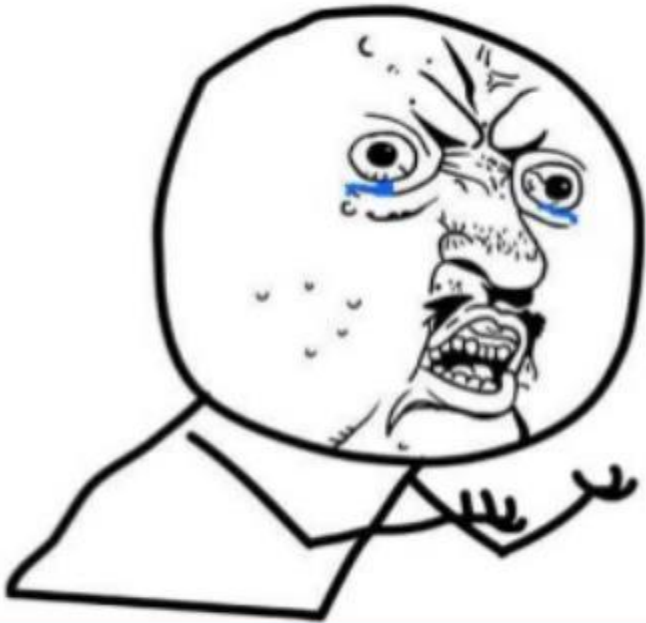
# Modelo de Arquitetura MVC

- ASP.NET MVC é apenas a camada de apresentação.
- O padrão MVC visa a separação de responsabilidades e nada mais.



# Mas e o Template do Visual Studio?

- Mas no projeto do template não tem camadas...
- Mas nos exemplos da internet...
- Mas nos exemplos dos livros...
- Mas... Mas...



*São assim apenas para ser o mais simples e didático possível ao ensinar como o ASP.NET MVC funciona! Não serve para ser o modelo de sua arquitetura.*

# Na minha empresa funciona assim...

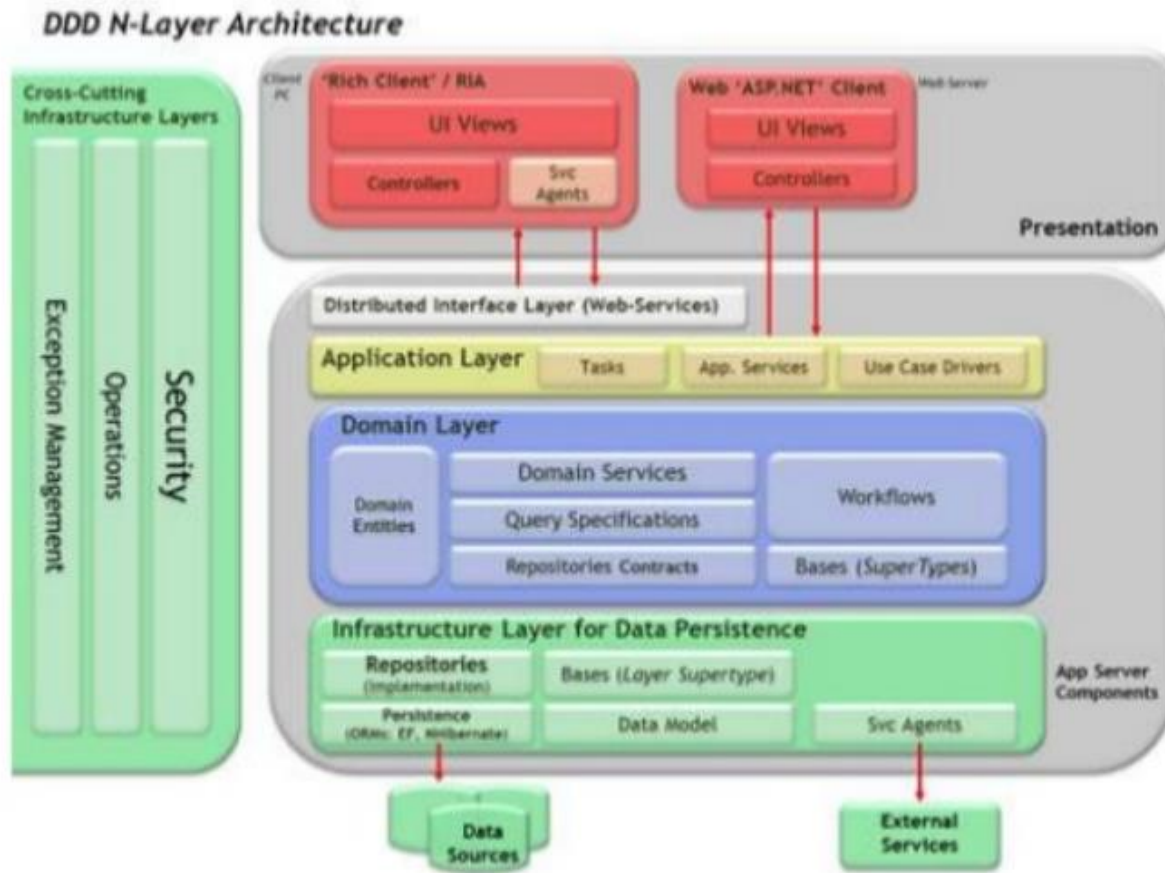
Business Object, Layer Object, Value Object

- ClienteVO
- ClienteBO
- ClienteDAO



*OUCH!... Mais um caso de  
"arquitetura" BOLOVO  
detectada, mas ainda há  
salvação.*

# DDD – Domain Driven Design



- Uma abordagem de design
- Complexa (no início)
- Desenvolvimento guiado pelo Domínio
- Isolamento de Responsabilidades
- Abstração de Camadas
- Indicado para aplicações corporativas



# DDD – Domain Driven Design



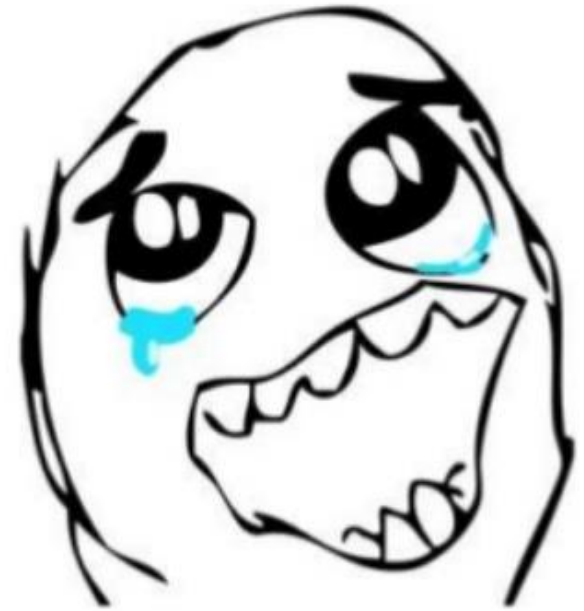
- Entidades
- Objetos de Valor
- Contratos (Interfaces)
- Services
- Factories

# Então quer dizer que...

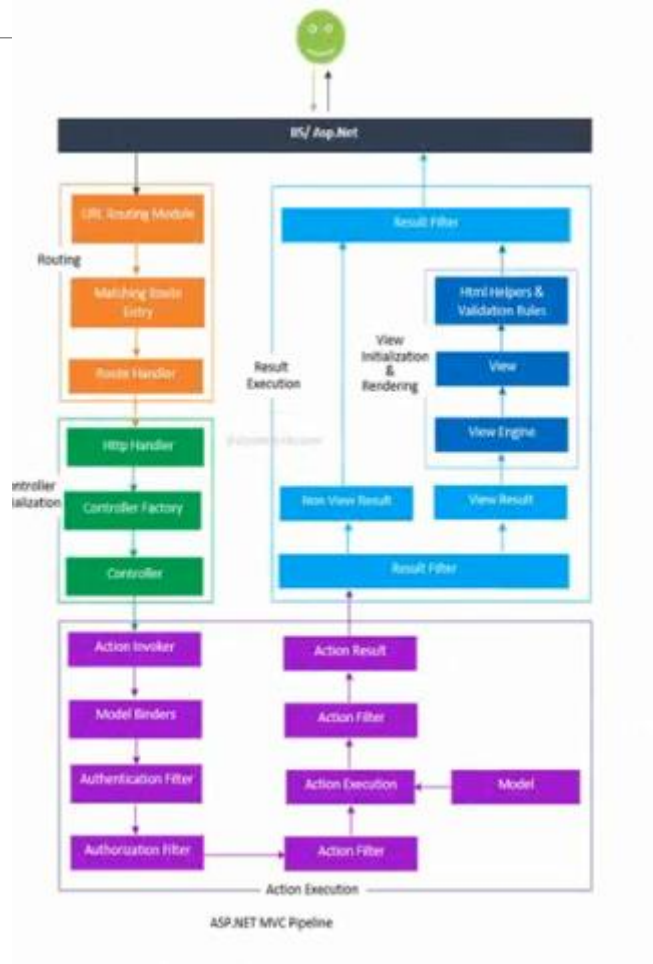
Se eu montar minha aplicação utilizando ASP.NET MVC e aplicar DDD para guiar meu desenvolvimento, utilizar ORM's, Repositórios, IoC, mapear minhas Entidades de Domínio com as ViewModels e entre outros patterns eu terei uma boa arquitetura?

Sim!

Ela será responsável, escalável, testável, reutilizável e sua manutenção será muito mais fácil e tranquila.



# ASP.NET Pipeline



Essencial para conhecer o ciclo de vida da aplicação

Totalmente manipulável



# ASP.NET Identity

---



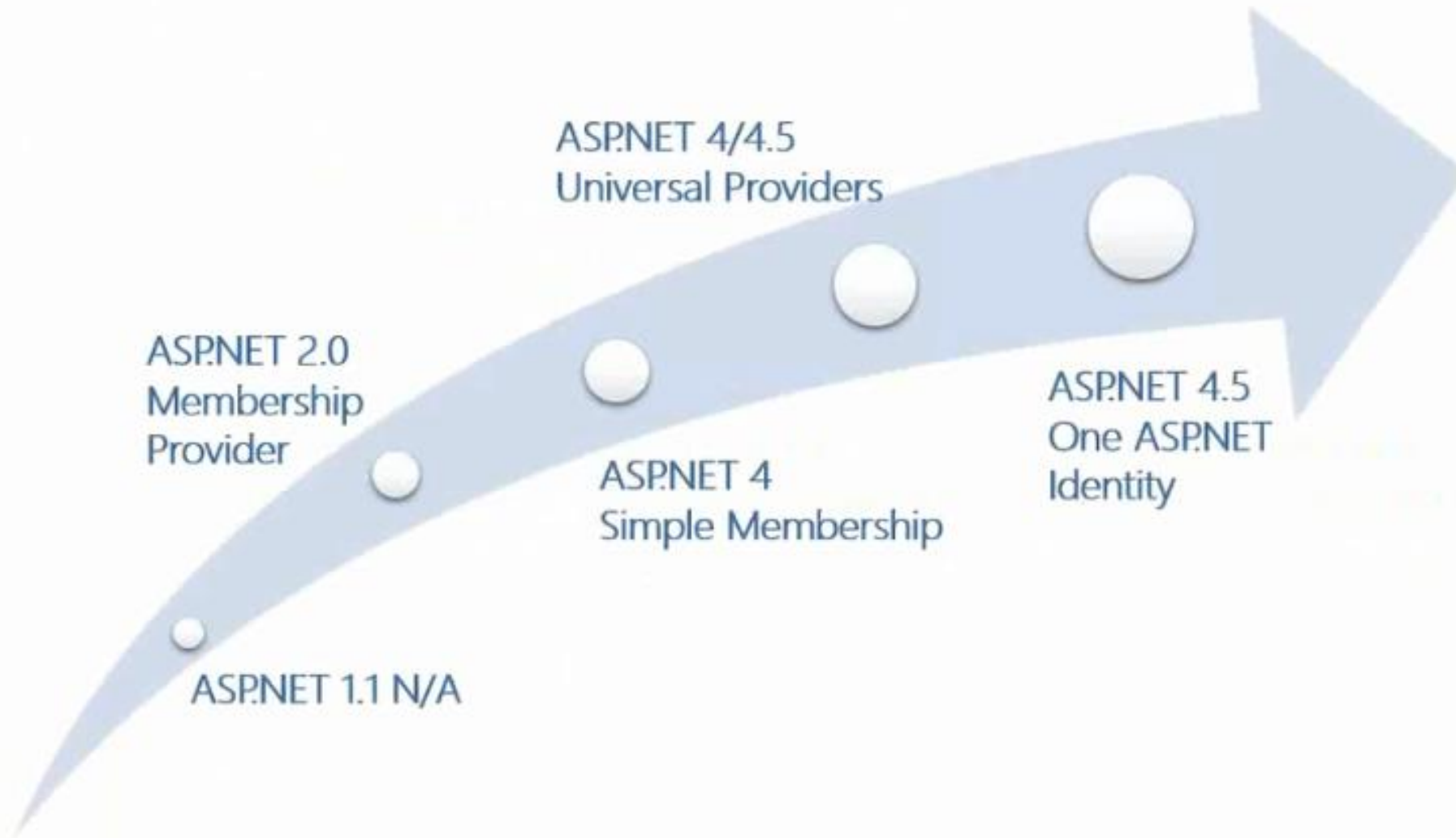
Autenticação

Autorização

OAuth e OpenID

Features

# ASP.NET Identity



# ASP.NET Identity

Lançado com o ASP.NET MVC 5

Parte do ONE ASP.NET

Customização do perfil do usuário simplificado (escrito em Code First)

Controle de Persistência de dados (EF ou outros)

Totalmente testável (Unity Tests)

Role Provider (separação de acessos por perfil)

Claims Based (isso aqui é TOPZERA)

Autenticação em redes sociais (FB, Twitter, Google+ e Microsoft Accounts)

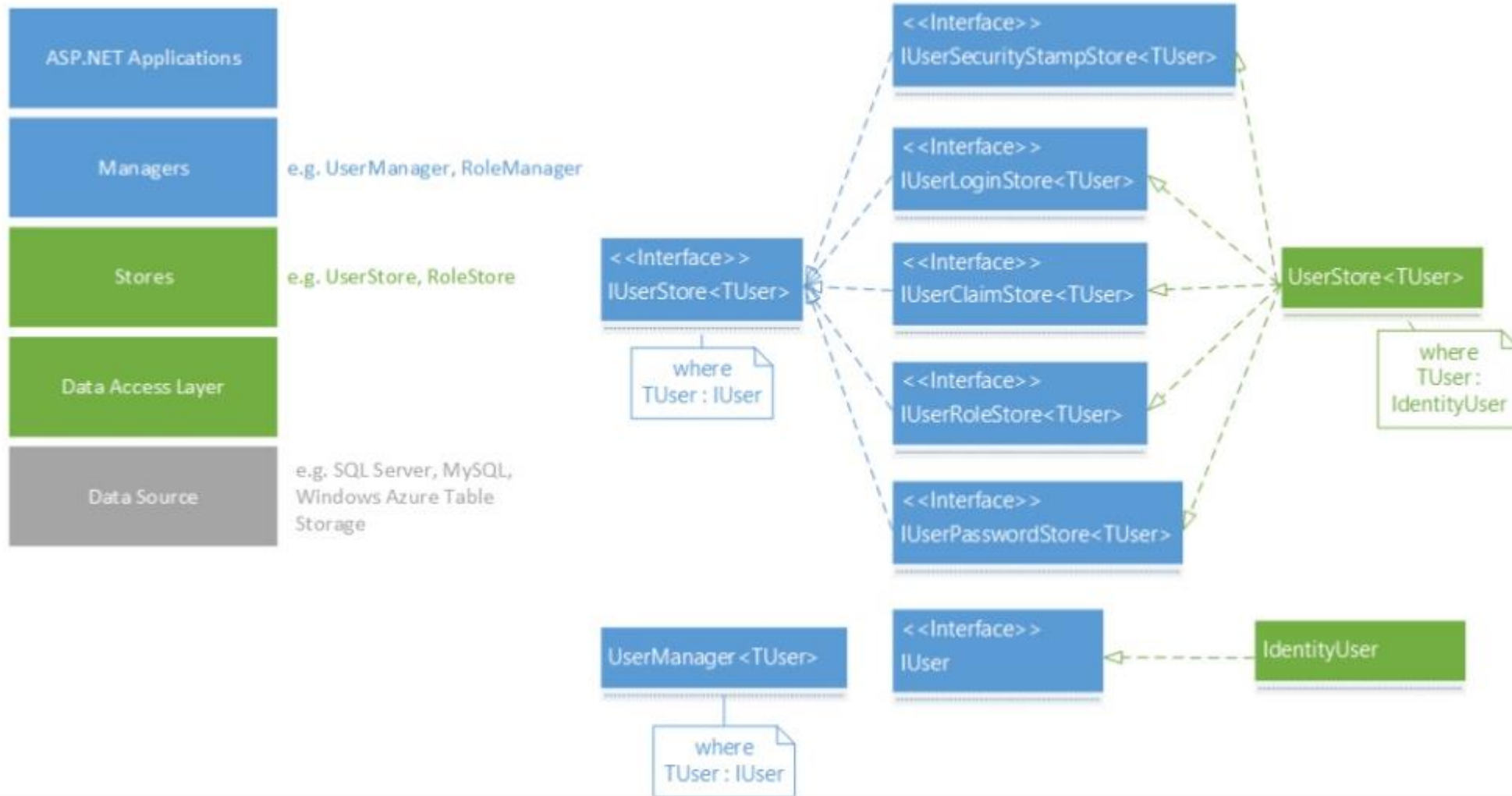
Integração com Active Directory (on-premises e Azure)

Integração com OWIN (OWIN Middleware based)

Entregue via NuGet (Nuget Everywhere)

TOTALMENTE Extensível

# Como funciona o ASP.NET Identity



# ASP.NET Identity

Two Factor Authentication

Account Lockout

Account Confirmation

Password Reset

Security Stamp (Sign Out Everywhere)

Primary Key Extensible for Users and Roles

Support IQueryable on Users and Roles

Delete User Account

Identity Factory Middleware – CreatePerOwinContext (UserManager, DbContext)

Enhanced Password Validator

SignInManager (V2.1 – mais fácil para aplicar Features



# REFERENCIAS

---

- EVANS, Eric – Domain Driven Design – Atacando as complexidades no Coração do Software
- Clean Architecture for .NET Applications – <https://paulovich.net/2018/05/15/clean-architecture-for-net-applications/>
- Eduardo Pires MVP (curso ASP.NET MVC) – [www.eduardopires.net.br](http://www.eduardopires.net.br)
- ASP.NET Identity <https://www.asp.net/identity>