

宁波DOTNET社区(NBDNC)线下交流分享会

MAY 2018

1. DotNet开发框架ABP介绍
2. Azure云介绍及演示

DOTNET 开发框架 ABP 介绍

BY 王伯承

目录

1. 自我介绍
2. ABP框架介绍
3. ABP实践经验分享
4. ABP的成功案例

I. 自我介绍

- 王伯承(personball)，2010年起从事dotnet开发，至今八年多；
- 前五年在宁波工作，经历过八六人才网、土拨鼠、浙江大道集团；
- 2015年8月到上海，跟随国内最早ABP开发框架的引入者——罗阳铭(<http://www.cnblogs.com/mienreal/>)，先后就职“要买车”和“健康724”。积累丰富的ABP框架实践经验，并根据实践经验开源ABP扩展库github.com/personball/abplus；
- 个人博客：<http://personball.com>

2.ABP框架介绍



企业级开发框架需要什么？

基础框架组件的功能：

- 1、基础框架组件独立、通用，可用于多个不同项目。类似于daxnet的Apworks框架。
- 2、对项目实现模块化开发提供了支持，每个模块有独立的EF DbContext，可单独指定数据库。
- 3、对DDD的技术实现进行了封装，让项目以极精简的代码，专注于业务领域。
- 4、多租户支持，每个租户的数据自动隔离，业务模块开发者不需要手动操作TenantId。
- 5、集成ASP.NET Identity，实现登录认证、功能权限授权&验证、角色和用户管理。
- 6、集成Log4Net，实现日志记录。
- 7、集成AutoMapper，实现Dto类与实体类的双向自动转换。
- 8、实现UnitOfWork模式，为应用层和仓储层的(会写数据库的)方法自动实现数据库事务。
- 9、可通过ApplicationService的方法自动建立相应的WebApi方法，ajax可直接调用，不需要写ApiController和Action。
- 10、调用ApplicationService的方法时，自动验证权限和参数有效性(用相应的Attribute标注)。
- 11、继承自FullAuditedEntity基类的领域实体，会自动实现软删除（在数据库中用IsDeleted字段进行标注）。
- 12、实现一系列扩展方法，简化编码。

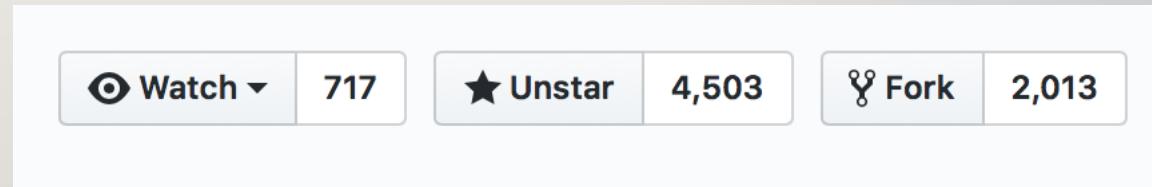
2. 什么是ABP？

- ABP是“ASP.NET Boilerplate Project (ASP.NET样板项目)”的简称。
- 开源时间：2014年5月4日发布在 Github
- 整个2017年作者全职做ABP框架后，版本升得飞快

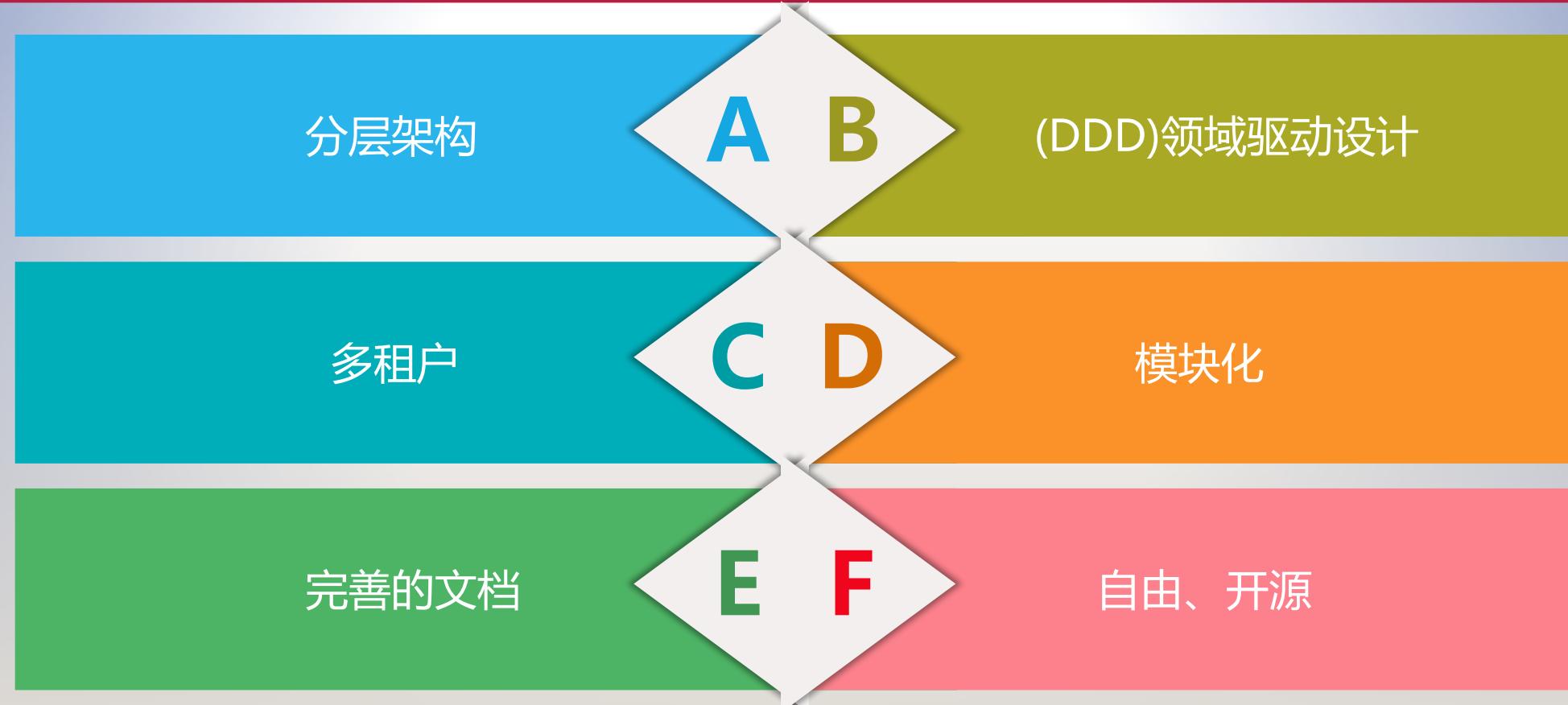
时间	Commits	Releases	版本号
2017年3月7日	-	-	V1.5.0
2017年4月7日	3023	124	V2.0.0
2018年4月16日	4612	163	V3.5.0

2. IABP是什么？

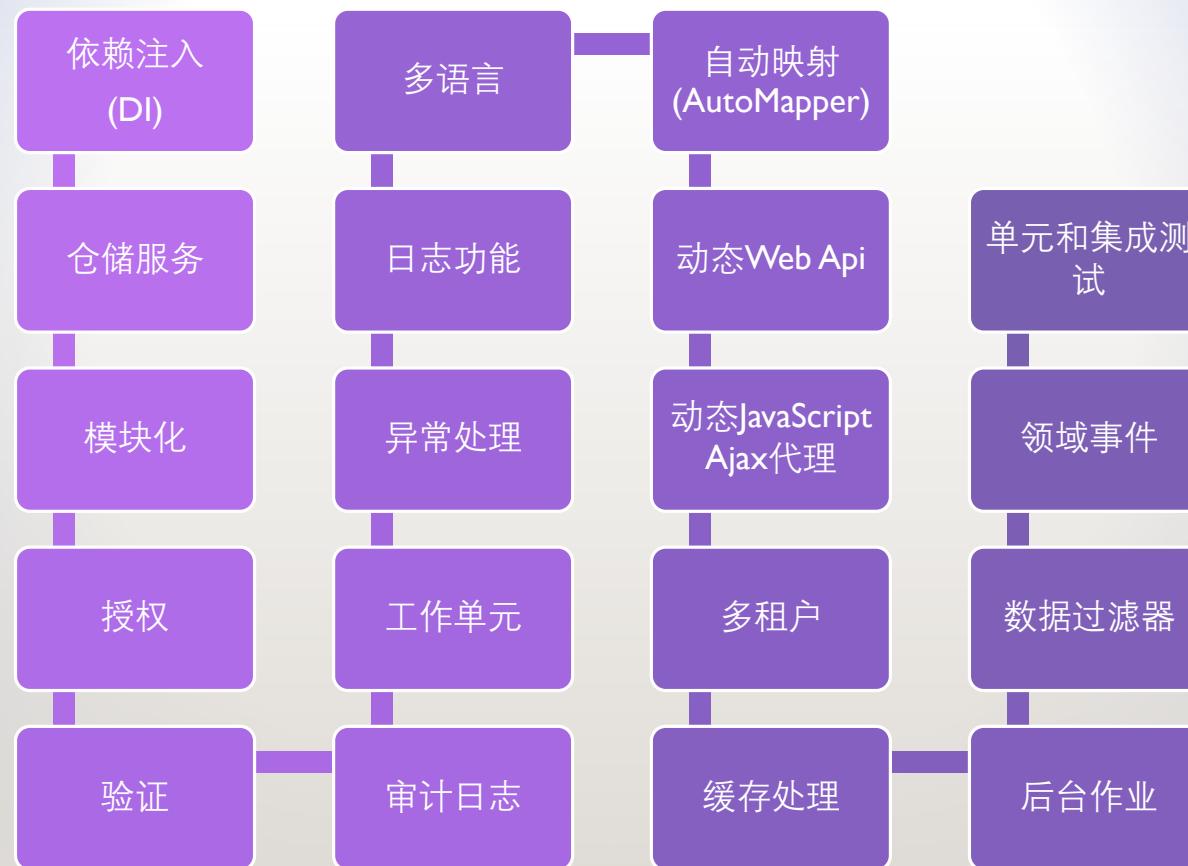
- ASP.NET Boilerplate 是一个用最佳实践和流行技术开发现代WEB应用程序的新起点.
 - 它旨在成为一个通用的WEB应用程序框架和项目模板。
 - ASP.NET Boilerplate 基于DDD的经典分层架构思想，实现了众多DDD的概念（但没有实现所有DDD的概念）。
-
- 官网：<https://www.aspnetboilerplate.com>
 - github.com/aspnetboilerplate



2.2 ABP框架的理念

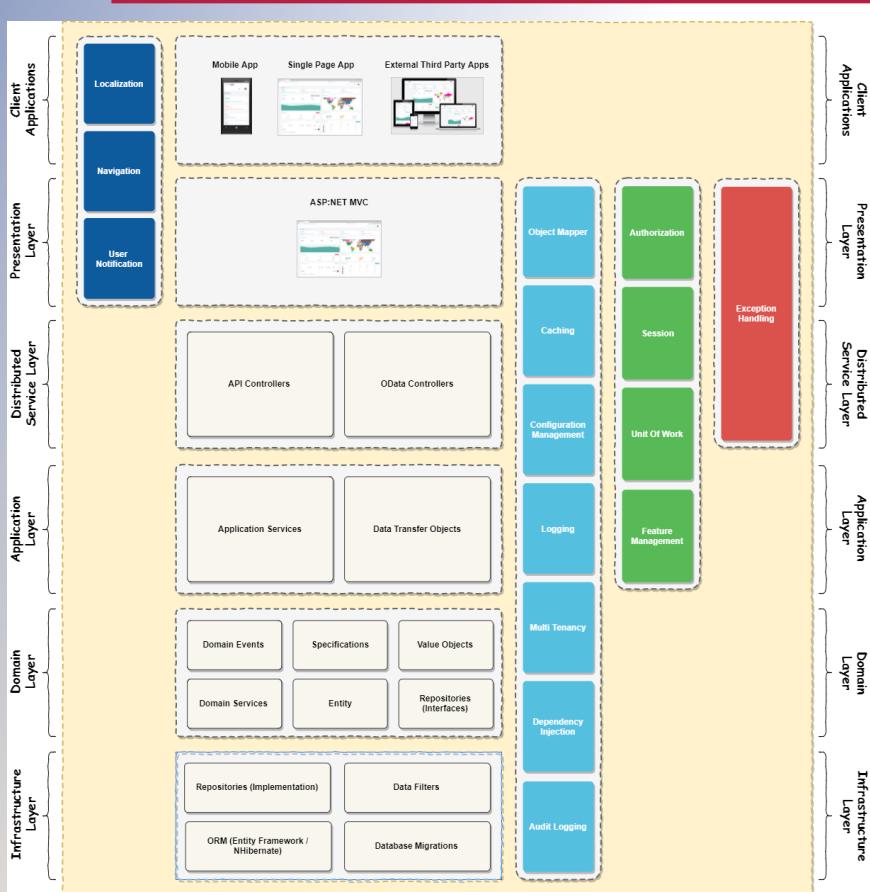


2.3 ABP提供的现成机制



DRY :
Don't Repeat Yourself !

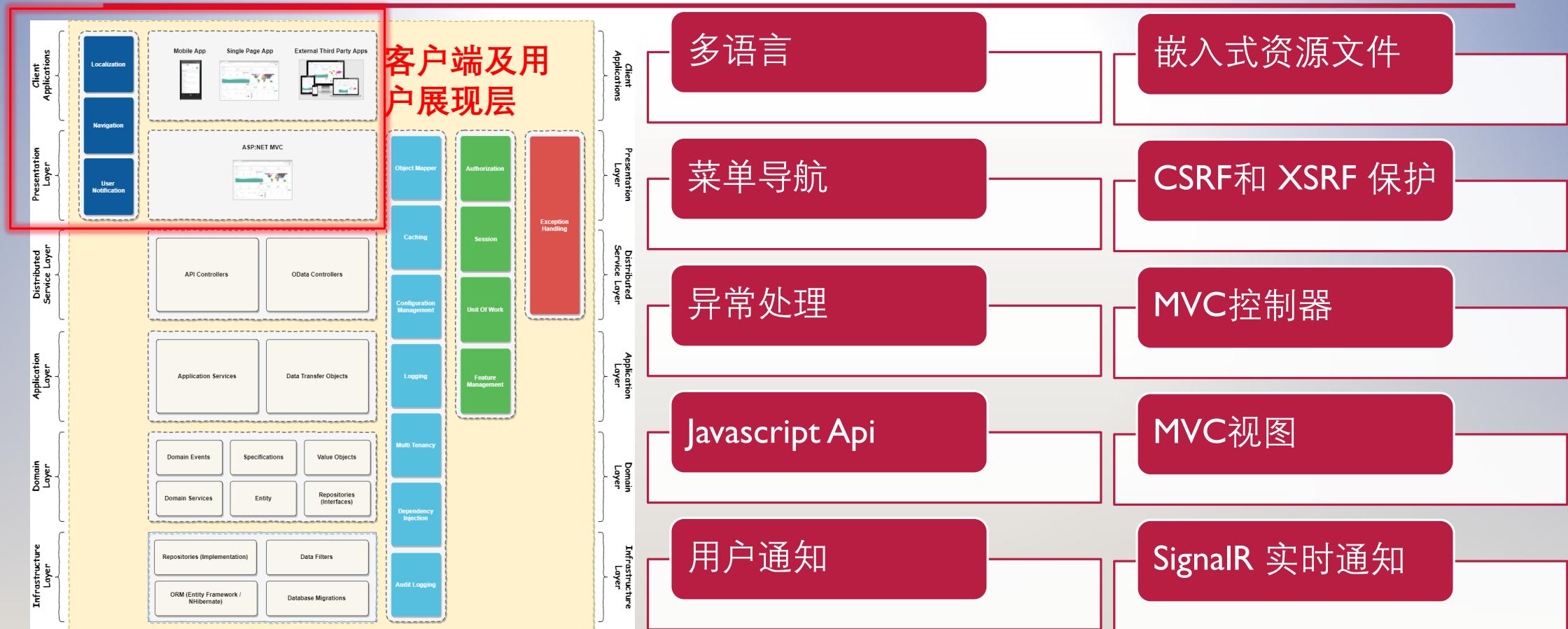
2.4 ABP的分层架构



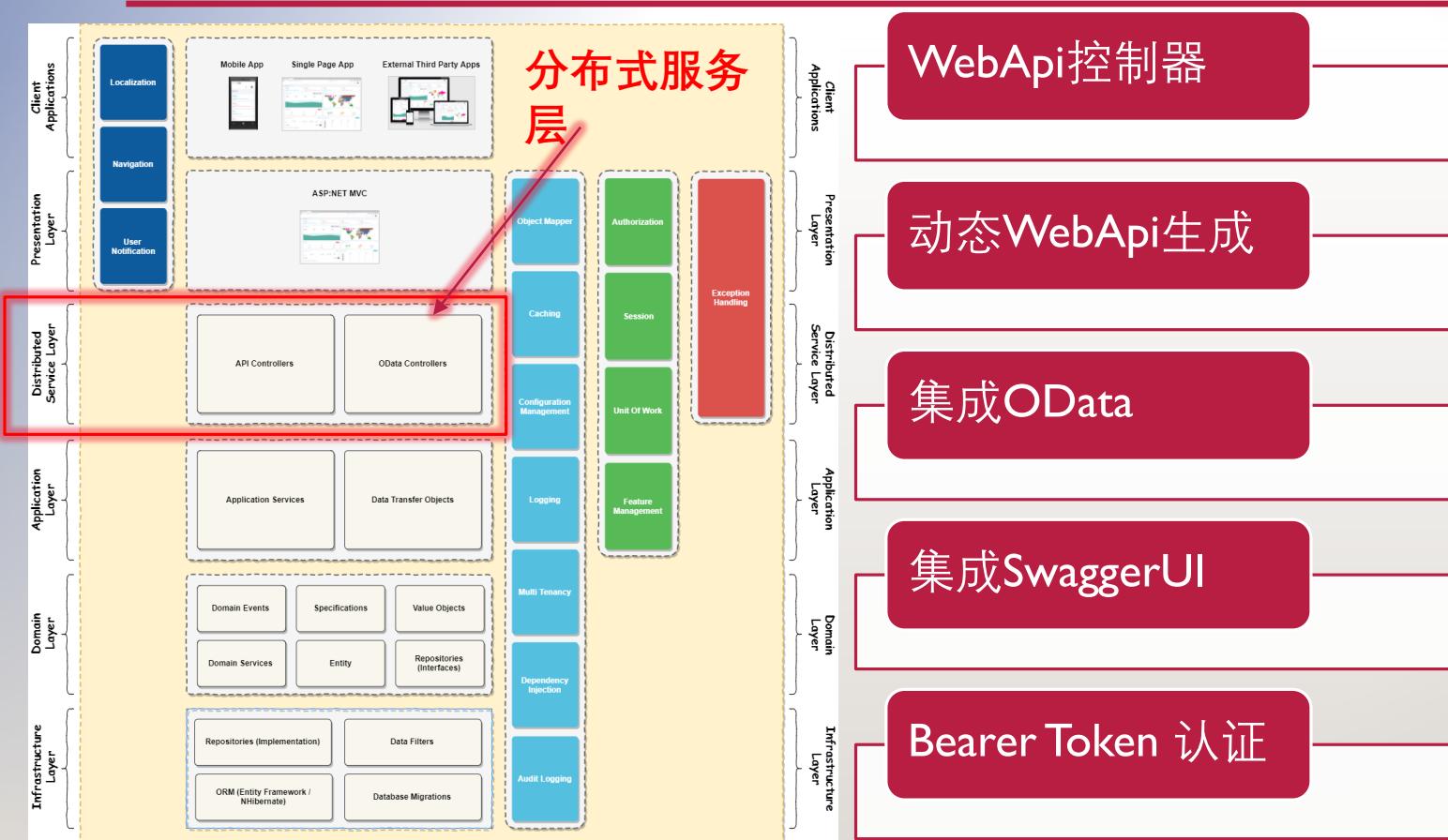
Layers

Presentation	View Models (Javascript)	Views (HTML/CSS)	Localization
Web	Web API Controllers	MVC Controllers	
Application	Application Services	DTOs	DTO Mappers
Domain	Entities	Value Objects	Repositories
Infrastructure	Business Classes	Domain Services	Unit of Work
	Domain Events	ORM	DB Migrations
Others			
Server Side	Dependency Injection	Logging	

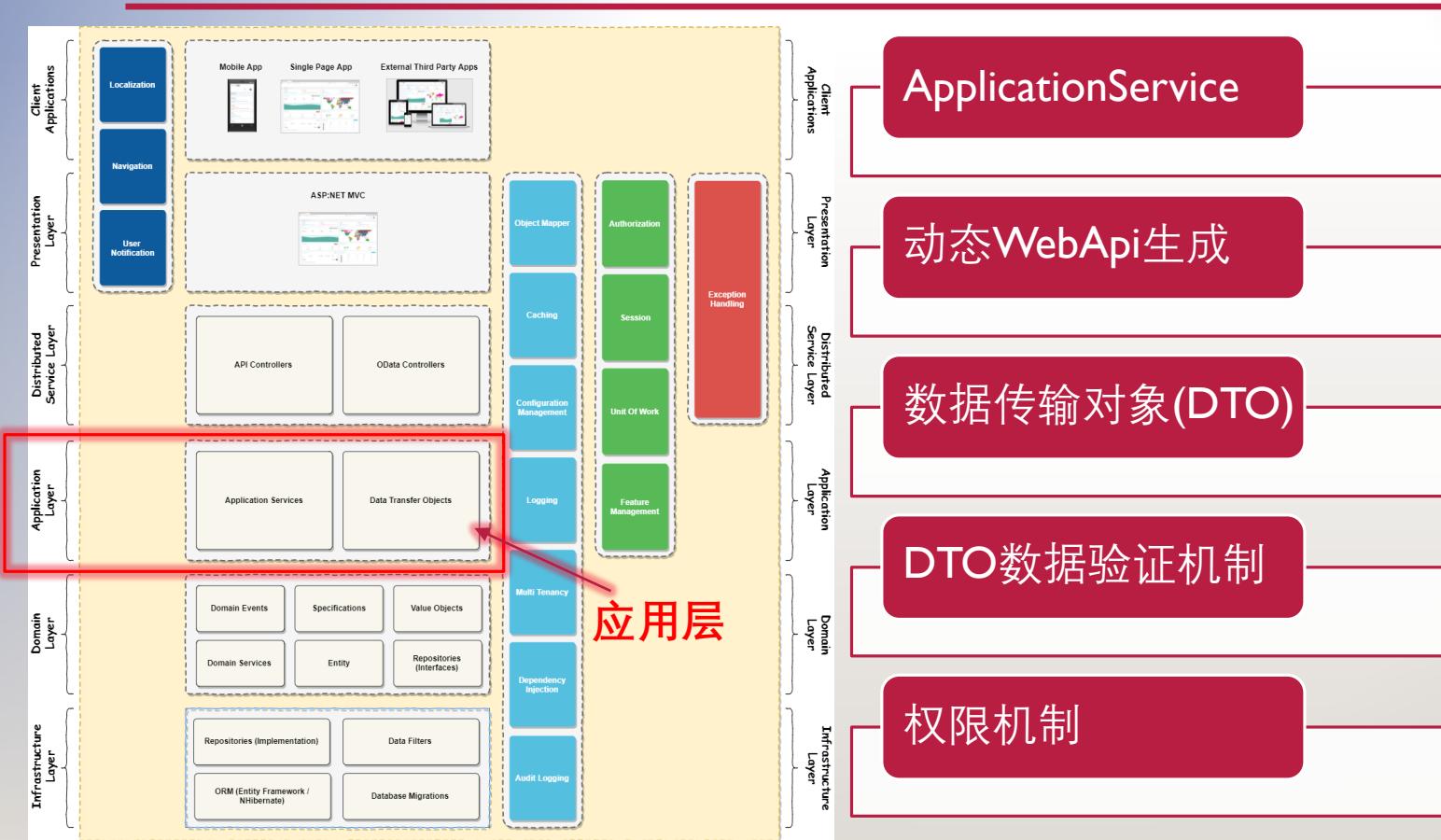
2.4 ABP的分层架构



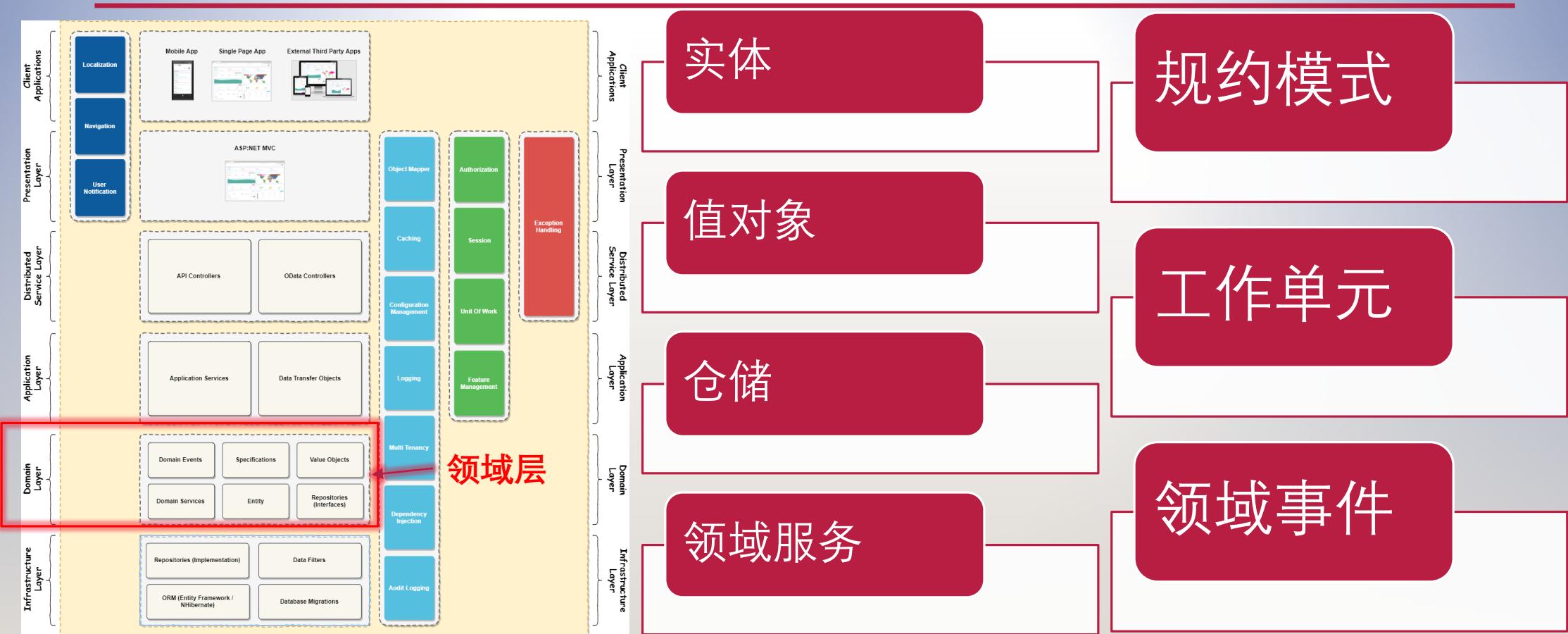
2.4 ABP的分层架构



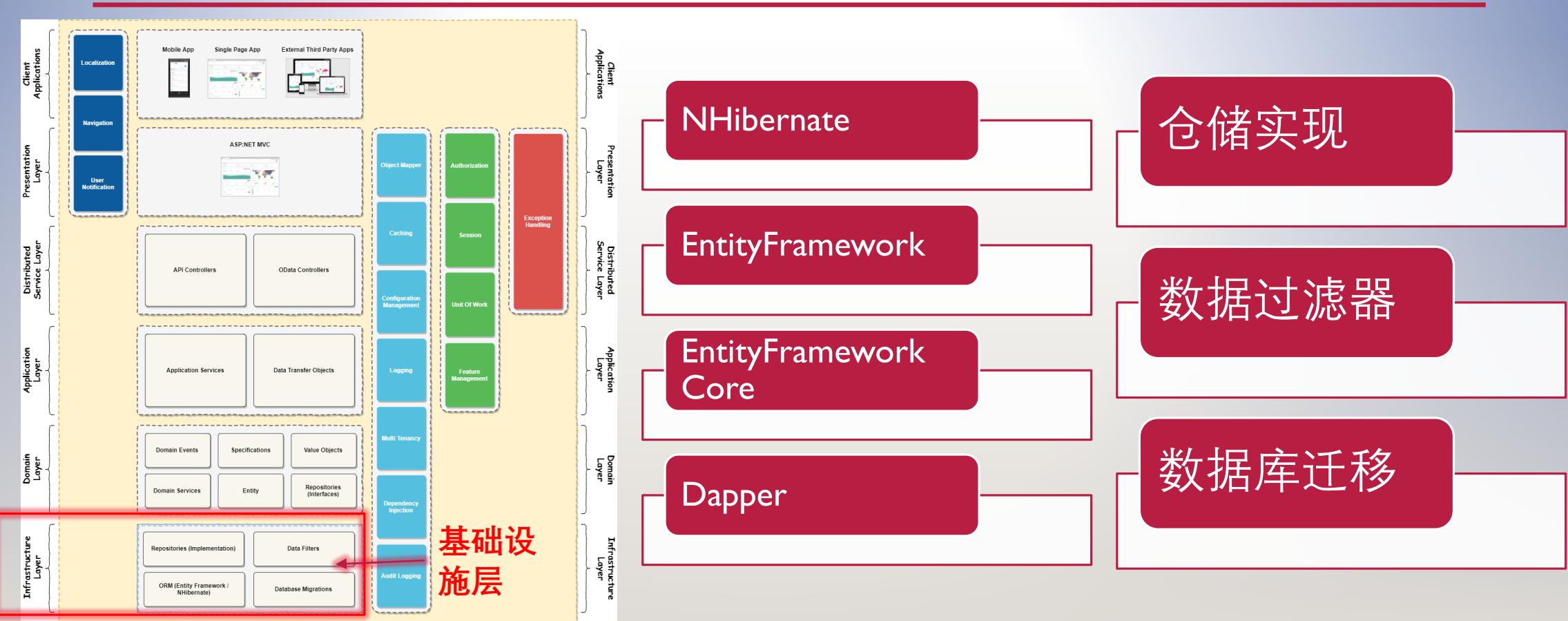
2.4 ABP的分层架构



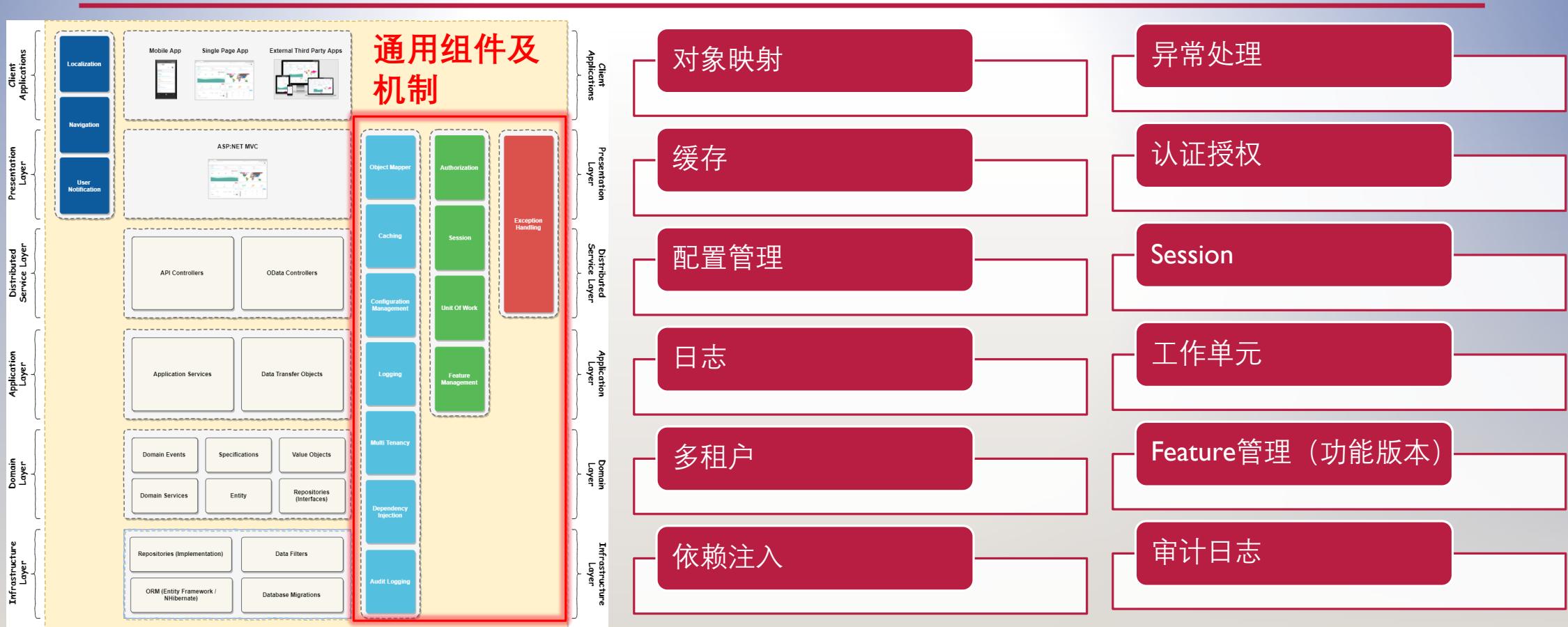
2.4 ABP的分层架构



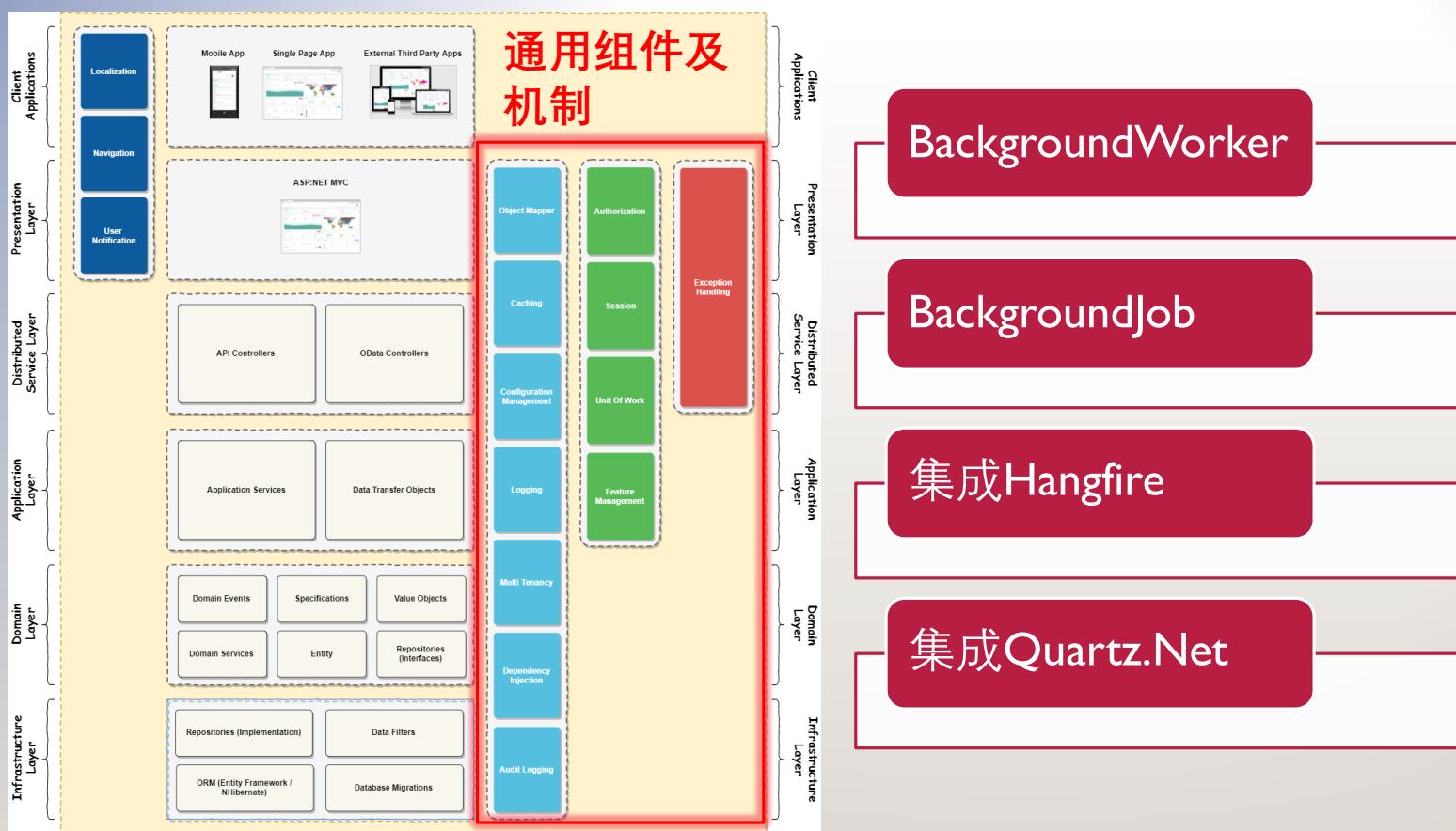
2.4 ABP的分层架构



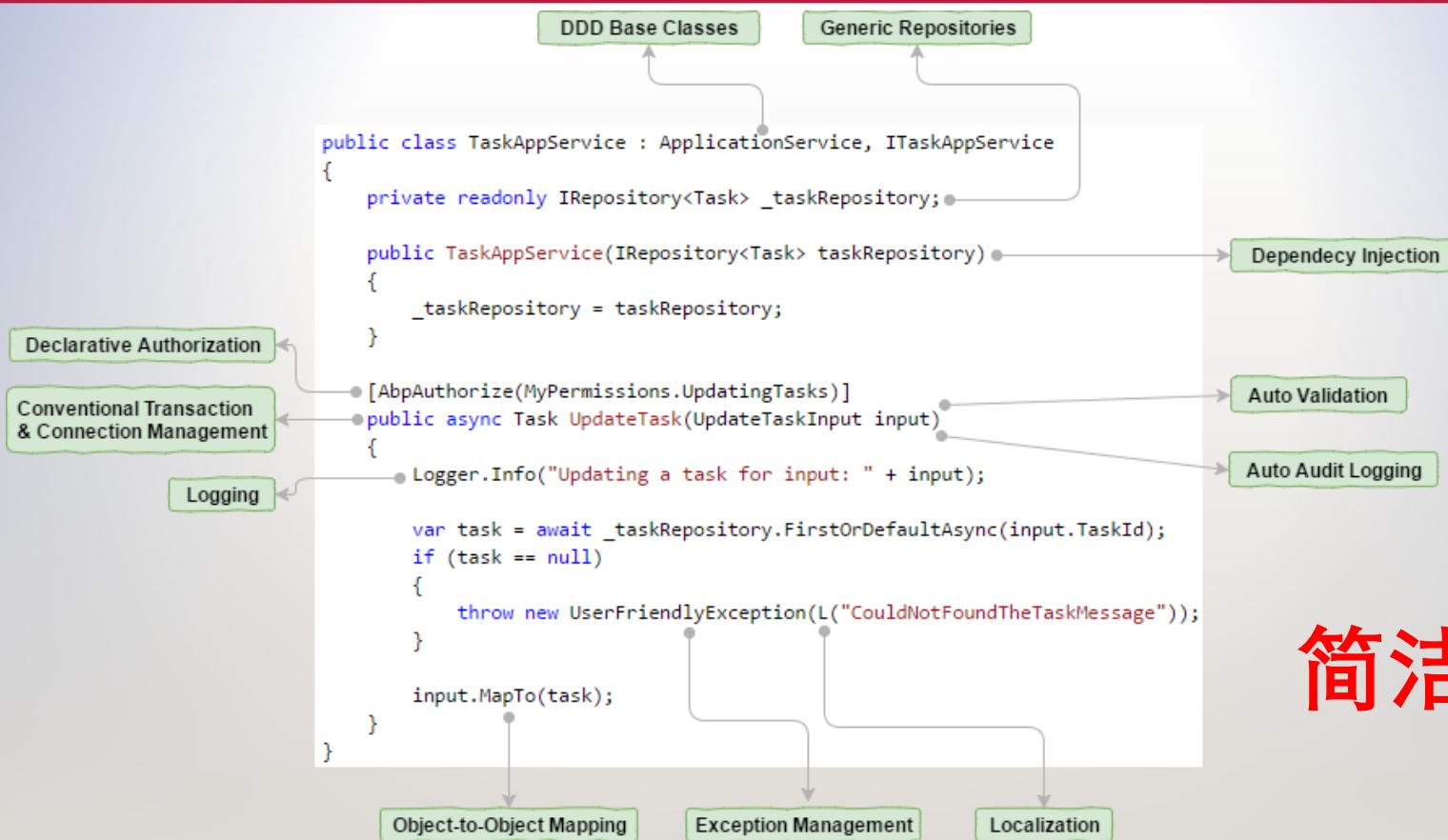
2.4 ABP的分层架构



2.4 ABP的分层架构



2.4 ABP的分层架构

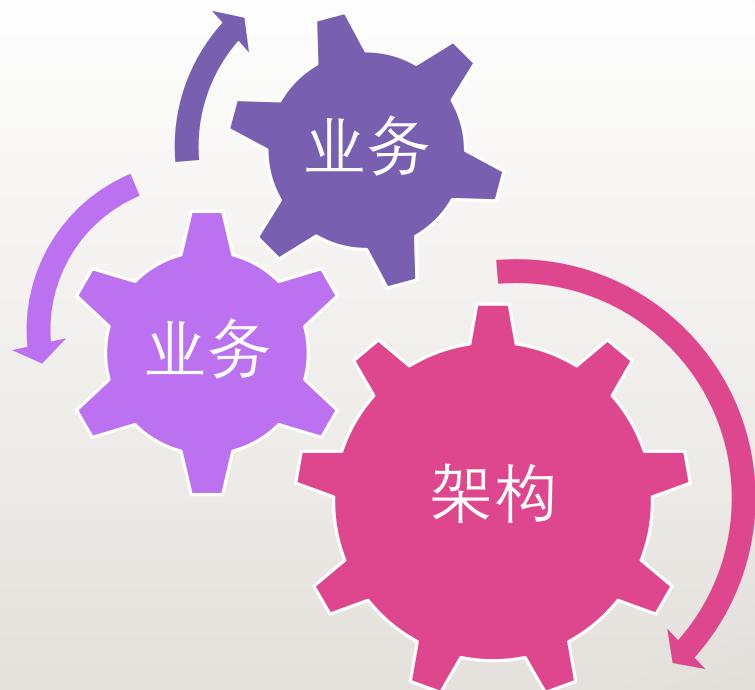


简洁！

2.5 ABP几个常见名词区分

- ABP (基础框架, 通用, 自由开源)
- Module Zero (提供了ABP所有基础概念的默认实现, 用户、角色、权限、配置、语言、审计日志、BackgroundJob, 特别是各种机制的默认存储实现)
- AspNet Zero (作者用以支持ABP框架开发的, 重点投入前端界面开发的收费版企业级启动项目模板)

3.ABP实践经验分享



3.1 架构方面——针对ABP进行了消息队列机制的扩展



重点在于：

1. 在不同宿主中，我们可以统一跑一个框架，用同一份业务代码。
2. 通过不同形式的部署（Web或Services）分别承担适合各自场景的任务，以灵活设计同步或异步的逻辑处理方案。

3.2 原724项目介绍



3.2 原724项目介绍

Libs

Boss.Ecom.Application

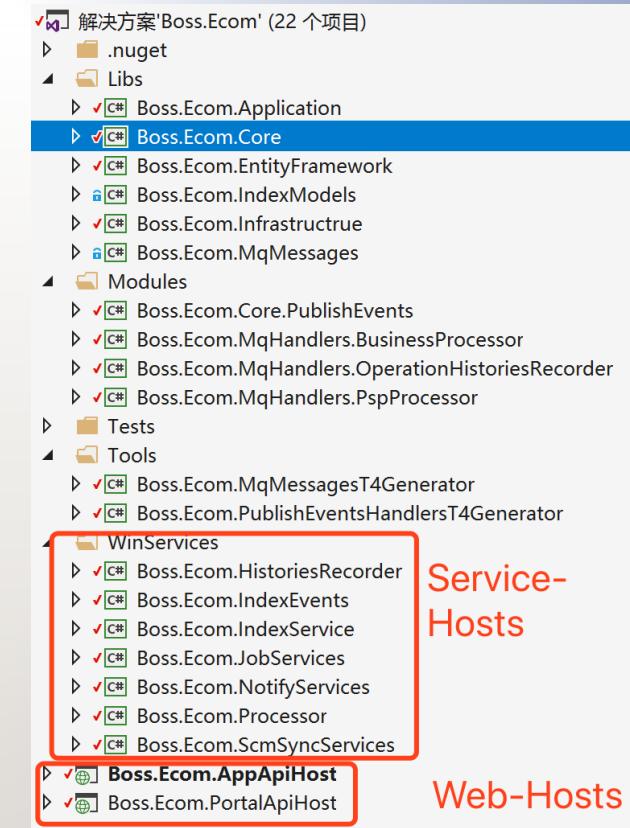
Boss.Ecom.Core

Boss.Ecom.EntityFramework

Boss.Ecom.IndexModels

Boss.Ecom.Infrastructure

Boss.Ecom.MqMessages



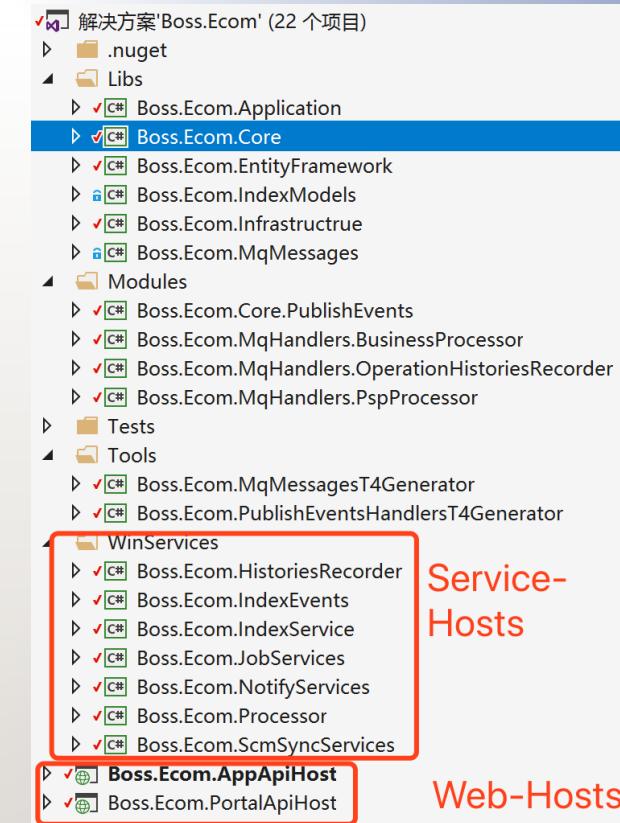
3.2 原724项目介绍

Modules Boss.Ecom.Core.PublishEvents

Boss.Ecom.MqHandlers.BusinessProcessor

Boss.Ecom.MqHandlers.OperationHistoriesRecorder

Boss.Ecom.MqHandlers.PspProcessor

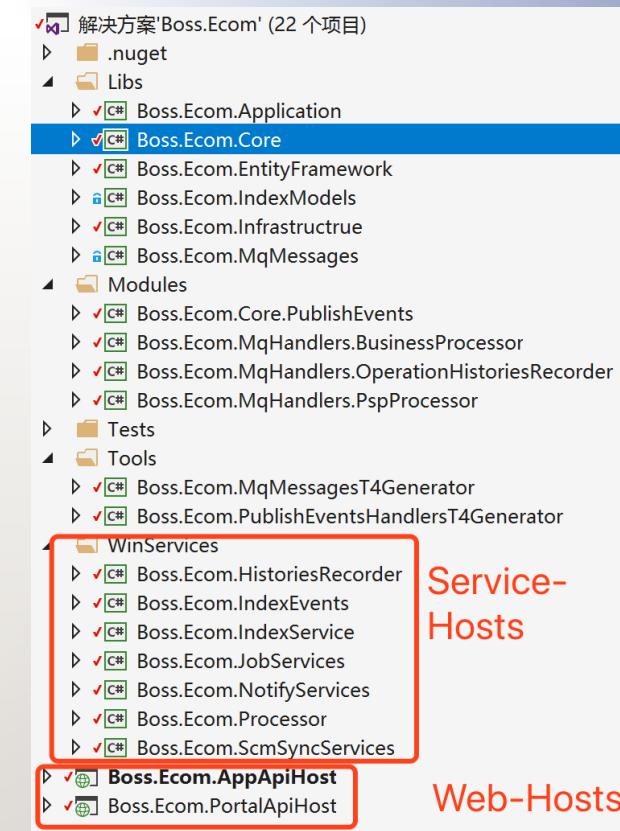


3.2 原724项目介绍

Tools

Boss.Ecom.MqMessagesT4Generator

Boss.Ecom.PublishEventsHandlersT4Generator



3.2 原724项目介绍

WinServices

Boss.Ecom.HistoriesRecorder

Boss.Ecom.IndexEvents

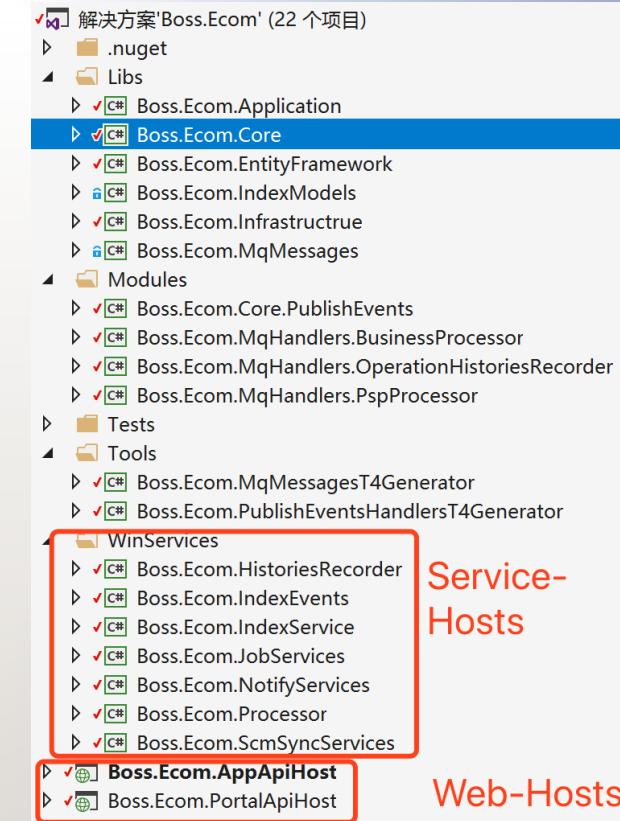
Boss.Ecom.IndexService

Boss.Ecom.JobServices

Boss.Ecom.NotifyServices

Boss.Ecom.Processor

Boss.Ecom.ScmSyncServices

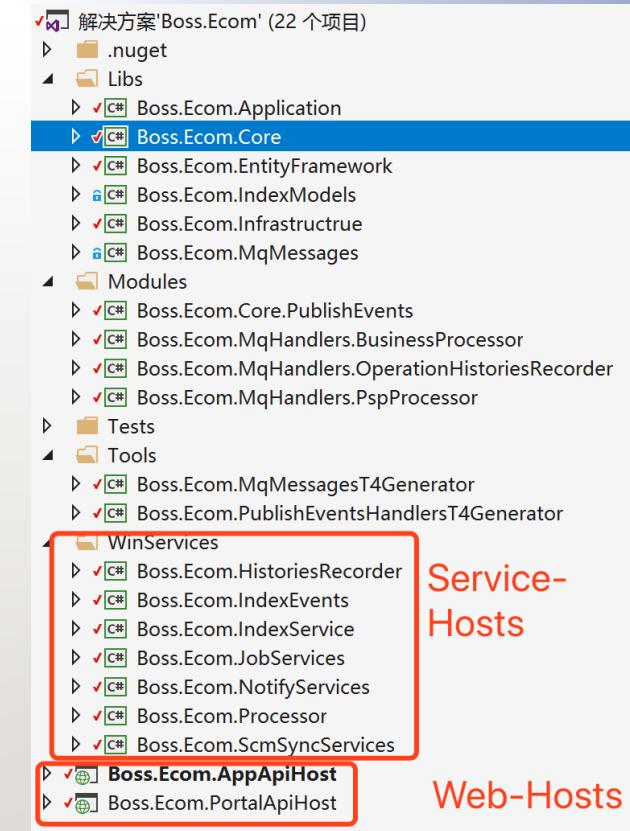


3.2 原724项目介绍

Web(ApiHost)

Boss.Ecom.AppApiHost

Boss.Ecom.PortalApiHost



3.2 原724项目介绍

ELK

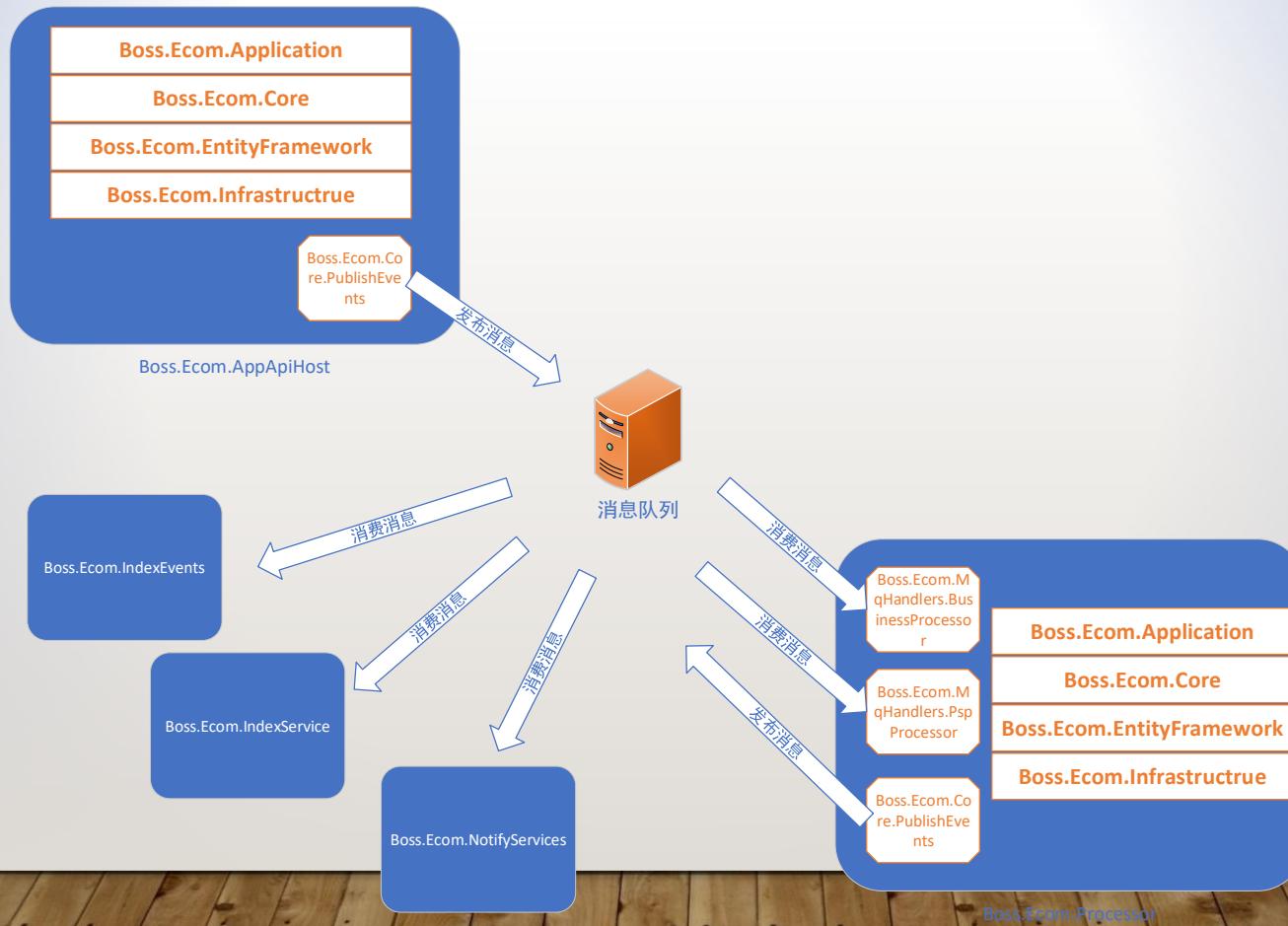
Abp 审计日志收集，通过消息队列对接

应用日志收集， Nlog, filebeat

nginx 访问日志收集， filebeat

H5 异常日志收集，由js汇报， webapi 中转到 redis 队列对接

3.2 原724项目介绍



3.3 业务方面

业务逻辑代码放到
领域层

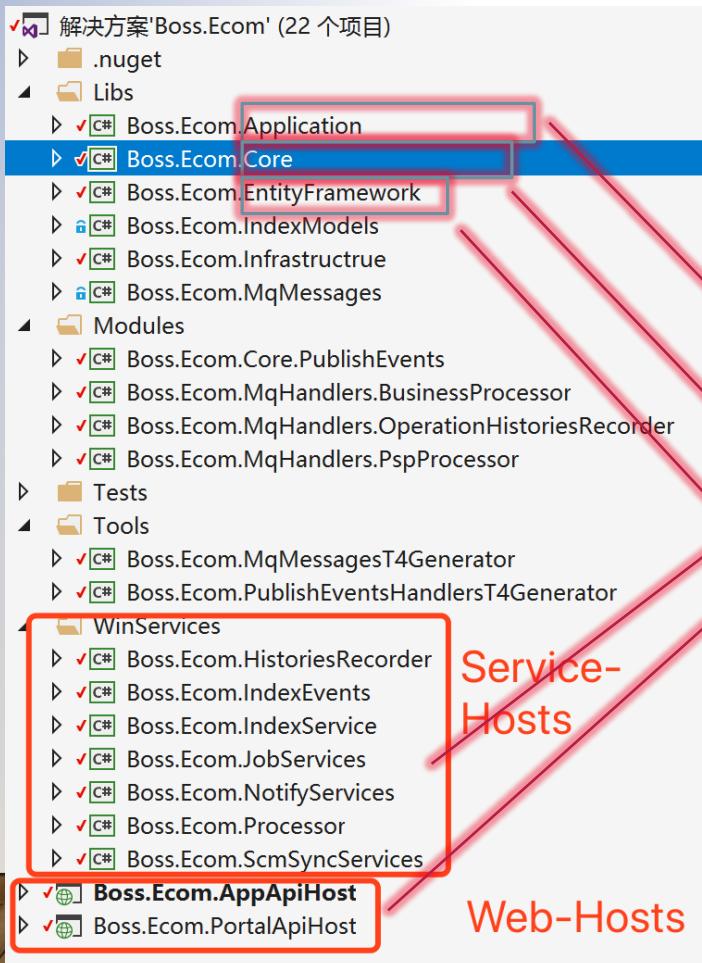


尽量给领域层一个
纯内存的执行环境

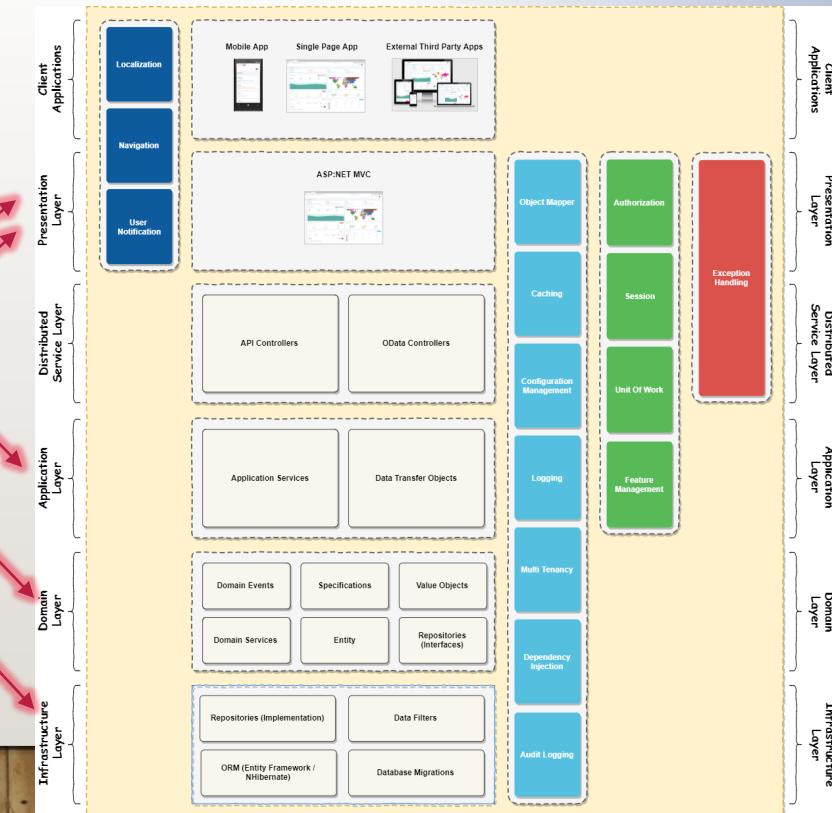


运用设计模式更好
的写出可维护可拓
展的业务代码（业
务规则建模）

3.3.1 理解命名空间的作用



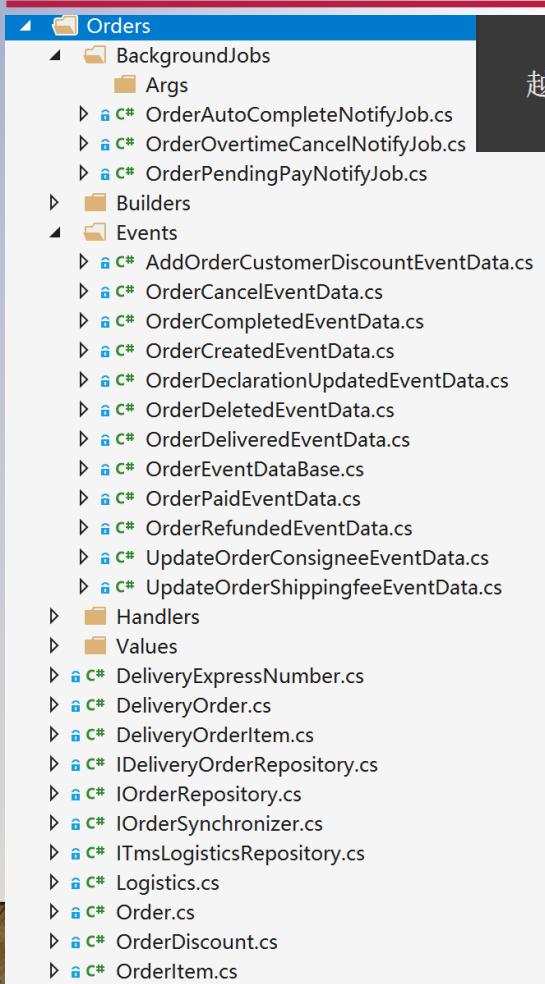
程序集命名体现了水平分层的层次



3.3.1 理解命名空间的作用



3.3.1 理解命名空间的作用



越是在上级空间中的对象，越具有可共享倾向（暂不考虑其他保护级别），越是往下级目录靠，其实就是越特殊

如果说，

程序集是水平划分代码（划分层次，程序集名称的最后一段代表其所在层次）

那么，

命名空间是垂直划分代码（划分概念，命名空间Personball.Demo.abc中的abc代表这部分代码所专注的领域和概念或者说提供哪方面的功能）

这是正交的关系。

为了提供功能，必然需要跨越多个层次，在每个层次中都保证同样的目录结构（vs新建类或接口的所处命名空间），那么

同一个领域或概念在不同层次中是天然可见的，不需要特地用using语句引入命名空间

3.3.2 业务异常及错误码设计

```
using System;
using Abp;
using Abp.Logging;

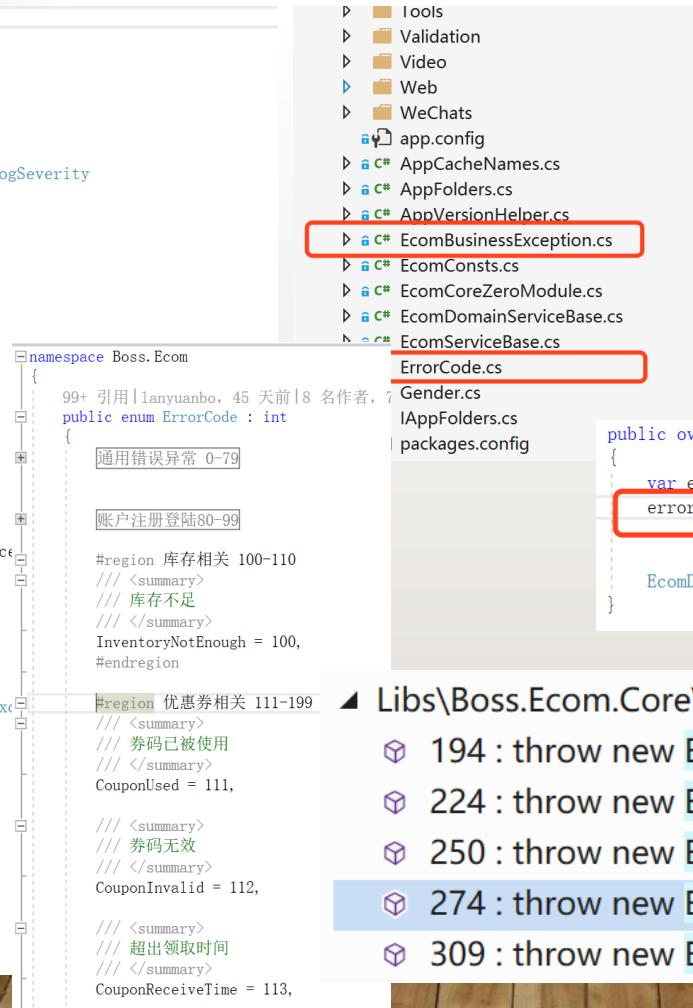
namespace Boss.Ecom
{
    99+ 引用 | personball, 255 天前 | 1 名作者, 2 项更改
    public class EcomBusinessException : AbpException, IHasErrorCode, IHasLogSeverity
    {
        99+ 引用 | personball, 364 天前 | 1 名作者, 1 项更改
        public EcomBusinessException(ErrorCode errorCode)
            : base(errorCode.ToString())
        {
            Code = (int)errorCode;
        }

        99+ 引用 | personball, 364 天前 | 1 名作者, 1 项更改
        public EcomBusinessException(ErrorCode errorCode, string message)
            : base(message)
        {
            Code = (int)errorCode;
        }

        0 个引用 | personball, 364 天前 | 1 名作者, 1 项更改
        public EcomBusinessException(ErrorCode errorCode, Exception innerException)
            : base(errorCode.ToString(), innerException)
        {
            Code = (int)errorCode;
        }

        0 个引用 | personball, 364 天前 | 1 名作者, 1 项更改
        public EcomBusinessException(ErrorCode errorCode, string message, Exception innerException)
            : base(message, innerException)
        {
            Code = (int)errorCode;
        }

        8 个引用 | personball, 364 天前 | 1 名作者, 1 项更改
        public int Code { get; set; }
        0 个引用 | personball, 255 天前 | 1 名作者, 1 项更改
        public LogSeverity Severity { get; set; } = LogSeverity.Warn;
    }
}
```



```
<!--ErrorCodes Start-->
<text name="UnknownException" value="未知异常"/>
<text name="ExistDuplicateItem" value="指定的对象存在重复"/>
<text name="ObjectIsNotActive" value="指定对象未激活"/>
<text name="CanNotBeDeletedSinceDependency" value="其他项依赖导致无法删除"/>
<text name="RequestParametersError" value="请求参数错误"/>
<text name="RequestHasOvertime" value="请求超时"/>
<text name="RequestSignError" value="请求签名错误"/>
<text name="ItemNotExists" value="对象不存在"/>
<text name="CanNotBeChangedSinceDependency" value="其他项依赖导致无法更改"/>
<text name="InventoryNotEnough" value="库存不足"/>
<text name="CouponUsed" value="券码已被使用"/>
<text name="CouponInvalid" value="券码无效"/>
<text name="CouponReceiveTime" value="超出领取时间"/>
<text name="CouponIsScrap" value="优惠券不能被领取"/>
<text name="CouponRepeat" value="不能重复领取"/>
<text name="NotCoupon" value="您还没有券"/>
<text name="CouponOverdue" value="优惠券过期了"/>

public override void PostInitialize()
{
    var errorInfoBuilder = LocManager.Resolve<IErrorInfoBuilder>();
    errorInfoBuilder.AddExceptionConverter(LocManager.Resolve<CustomExceptionErrorInfoConverter>());
}

EcomDbContextHelper.EntityFrameworkWarmUp();
}
```

Libs\Boss.Ecom.Core\Orders\Order.cs (5)

- ⑤ 194 : throw new EcomBusinessException(ErrorCode.OrderCannotDelete);
- ⑤ 224 : throw new EcomBusinessException(ErrorCode.OrderCannotPaid);
- ⑤ 250 : throw new EcomBusinessException(ErrorCode.OrderCannotComplete);
- ⑤ 274 : throw new EcomBusinessException(ErrorCode.OrderCannotCancel);
- ⑤ 309 : throw new EcomBusinessException(ErrorCode.OrderCannotRefund);

3.3.3 领域层设计

```
/// <summary>
/// 分润账户余额（可提现余额）
/// </summary>
11 个引用 | personball, 281 天前 | 1 名作者, 3 项更改
public int BalanceInCent { get; private set; }

/// <summary>
/// 冻结金额（提现中）
/// </summary>
8 个引用 | personball, 281 天前 | 1 名作者, 3 项更改
public int FrozenInCent { get; private set; }
```

关闭setter

```
/// <summary> 余额增加
2 个引用 | personball, 243 天前 | 2 名作者, 4 项更改
public PspAccountHistory Increase(int amountInCent, string remark) ...

/// <summary> 余额冻结
1 个引用 | personball, 320 天前 | 1 名作者, 2 项更改
public void Freeze(int amountInCent) ...

/// <summary> 冻结解冻
1 个引用 | personball, 320 天前 | 1 名作者, 2 项更改
public void Unfreeze(int amountInCent) ...
/// <summary> 冻结扣减
1 个引用 | personball, 320 天前 | 1 名作者, 2 项更改
public PspAccountHistory Decrease(int amountInCent, string remark) ...
```

添加methods

```
public PspAccountHistory Increase(int amountInCent, string remark)
{
    if (amountInCent < 0)
    {
        throw new ArgumentOutOfRangeException(nameof(amountInCent), $"nameof(amountInCent) 不能为负数");
    }

    if (amountInCent == 0)
    {
        return null;
    }

    BalanceInCent += amountInCent;

    DomainEvents.Add(new Events.PspAccountIncreasedEventData
    {
        AccountId = this.Id,
        AccountLevelId = this.AccountLevelId,
        IncreasedAmountInCent = amountInCent,
        OrganizationId = this.OrganizationId,
        UserId = this.UserId,
        IncreasedType = increasedType
    });
}

return history;
```

考虑事件

3.3.3领域层设计

The screenshot shows the Visual Studio IDE interface. On the left is the code editor with `OrderManager.cs` open. The code implements the `IOrderBuilder` interface, specifically the `CreateOrder` method. A red box highlights the construction of the `OrderBuilderContext` object, which is built from various decorators. On the right is the Solution Explorer window, which displays the project structure for the `Orders` solution. A second red box highlights the `Builders` folder, which contains all the concrete implementations of the `IDecorator` interface.

```
public async Task<OrderBuilderContext> CreateOrder(OrderBasicInfo orderBasicInfo, bool isPreviewMode)
{
    //构建builder
    var orderBuilder = IocManager.Instance.Resolve<DefaultOrderBuilder>(new
    {
        ...;

        //构建上下文
        var context =
            new OrderBuilderContext(
                new TaxCouponDecorator( //税优惠券
                    new SelectDefaultTaxCouponDecorator( //税优惠券取默认
                        new TaxActivitiesDecorator( //税促销活动
                            new ShippingFeeCouponDecorator( //运费优惠券
                                new SelectDefaultShippingFeeCouponDecorator( //运费优惠券取默认
                                    new GoodsCouponsDecorator( //商品优惠券
                                        new SelectDefaultGoodsCouponsDecorator( //商品优惠券取默认
                                            new ShippingFeeActivityDecorator( //运费促销活动
                                                new GoodsActivitiesDecorator( //商品促销活动
                                                    orderBuilder //订单基本信息构建
                                                ), _freightCalculator
                                            ), _couponRepository, _couponHistoryRepository
                                        )
                                    )
                                )
                            ), _taxCalculator
                        )
                    ), _promotionManager
                );
            ...
            context.LoadUnexpiredActivities(canUseActivities);

            //设置预览模式
            if (isPreviewMode)
            {
                context.SetPreviewMode(orderBasicInfo, needDefaultCoupons);
            }
            else
            {
                context.SetCreateNewMode(orderBasicInfo);
            }

            //构建订单
            await context.Build();
    });
}
```

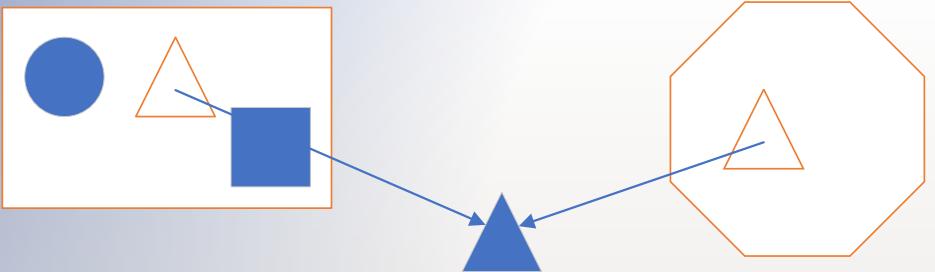
解决方案资源管理器显示了以下文件结构：

- Orders
- BackgroundJobs
- Builders
 - Decorators
 - GoodsActivitiesDecorator.cs
 - GoodsCouponsDecorator.cs
 - SelectDefaultGoodsCouponsDecorator.cs
 - SelectDefaultShippingFeeCouponDecorator.cs
 - SelectDefaultTaxCouponDecorator.cs
 - ShippingFeeActivityDecorator.cs
 - ShippingFeeCouponDecorator.cs
 - TaxActivitiesDecorator.cs
 - TaxCouponDecorator.cs
 - CouponWithHistoryId.cs
 - DefaultOrderBuilder.cs
 - IOrderBuilder.cs
 - OrderBasicInfo.cs
 - OrderBuilderContext.cs
 - Events
 - Handlers
 - Values
 - DeliveryExpressNumber.cs
 - DeliveryOrder.cs
 - DeliveryOrderItem.cs
 - IDeliveryOrderRepository.cs
 - IOrderRepository.cs
 - IOrderSynchronizer.cs
 - ITmsLogisticsRepository.cs
 - Logistics.cs
 - Order.cs
 - OrderDiscount.cs
 - OrderItem.cs
 - OrderItemGoodsIdentifier.cs
 - OrderItemPackageItem.cs
- DeliveryOrder.cs
- DeliveryOrderItem.cs
- IDeliveryOrderRepository.cs
- IOrderRepository.cs
- IOrderSynchronizer.cs
- ITmsLogisticsRepository.cs
- Logistics.cs
- Order.cs
- OrderDiscount.cs
- OrderItem.cs
- OrderItemGoodsIdentifier.cs
- OrderItemPackageItem.cs

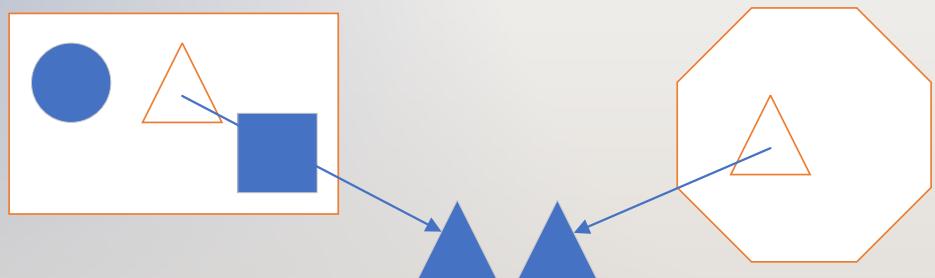
3.3.4事件与消息



3.3.5 值对象的不可变性



引用类型的值对象必须具备不可变性，
以防止多个实体持有同一个值对象引用时，
其中一个实体修改了该对象但并未意识到影响了另一个实体
就会造成意料之外的影响



必须关闭属性的set访问器！

```
12 个引用 | personball, 304 天前 | 1 名作者, 3 项更改
public class PspBuyerInfo : ValueObject<PspBuyerInfo>
{
    0 个引用 | personball, 311 天前 | 1 名作者, 1 项更改
    protected PspBuyerInfo() ...

    1 个引用 | personball, 304 天前 | 1 名作者, 1 项更改
    public PspBuyerInfo(long buyerAccountId, long buyerParentAccountId, int buyerOrganizationId) ...

    /// <summary> 买家账号
    9 个引用 | personball, 305 天前 | 1 名作者, 2 项更改
    public long BuyerAccountId { get; private set; }

    /// <summary> 买家所属组织
    3 个引用 | personball, 304 天前 | 1 名作者, 2 项更改
    public int BuyerOrganizationId { get; private set; }

    /// <summary> 推荐人
    2 个引用 | personball, 305 天前 | 1 名作者, 2 项更改
    public long BuyerParentAccountId { get; private set; }
}
```

4.ABP的成功案例



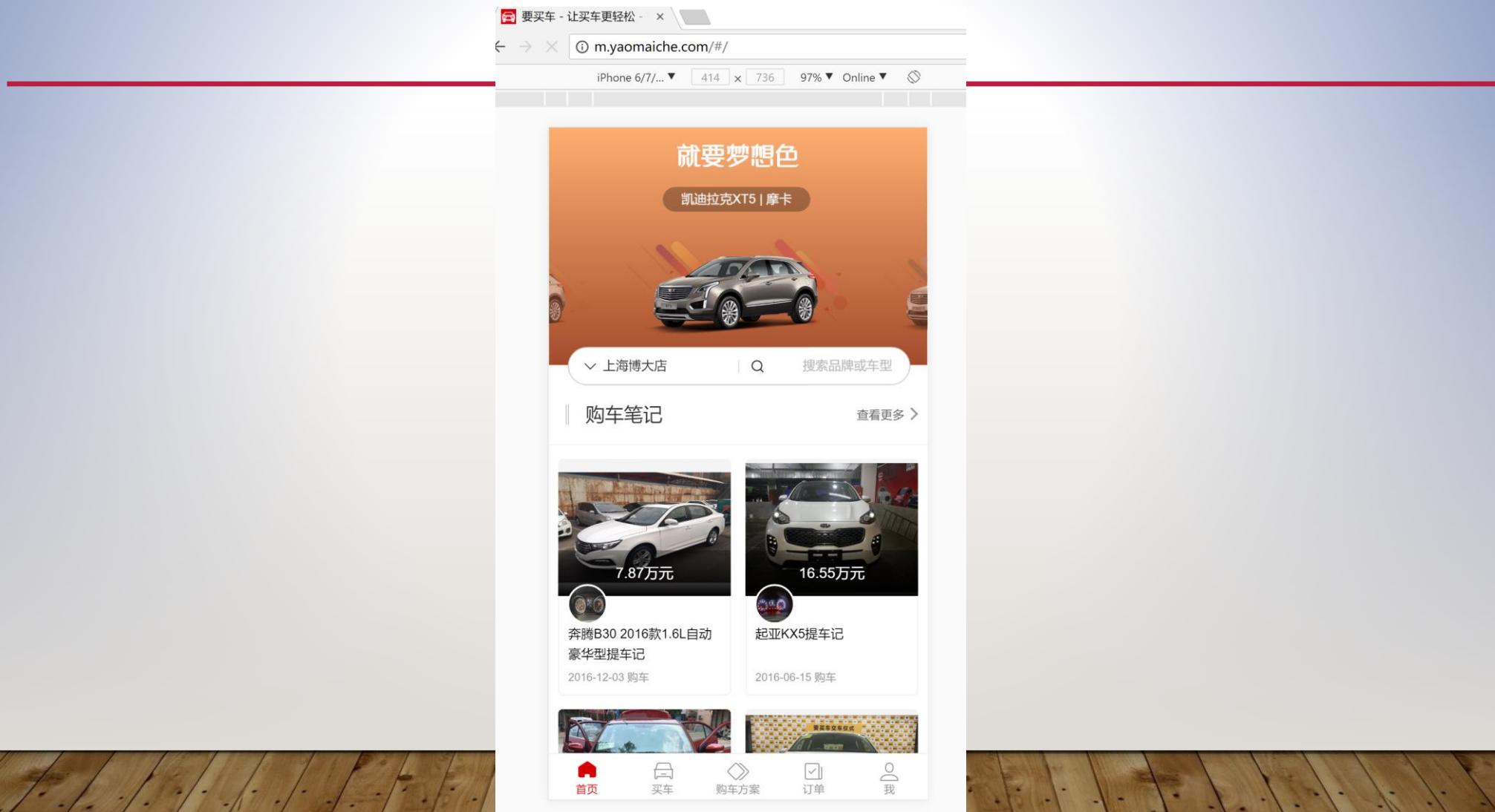
4. IABP的成功案例——积微物联(成都)

The screenshot displays the homepage of Jiweweixuan (积微物联), a platform for waste recycling and resource management. The top navigation bar includes links for '交易' (Trade), '物流' (Logistics), '区块链' (Blockchain), '大数据' (Big Data), and a red '登录' (Login) button. Below the navigation is a search bar and a menu with links to '客服中心', '积微账户', '积微家族', and '手机循环'.

The main content area features several service modules:

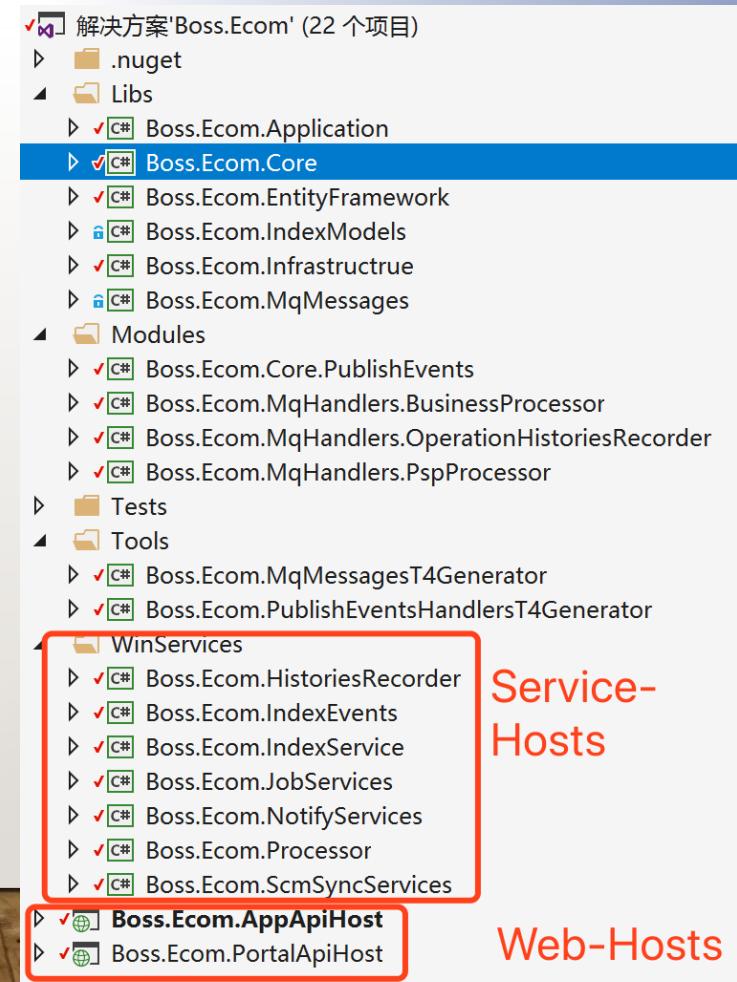
- 我们的服务 (OUR SERVICES):** Includes '积微钢铁' (Jiweweixuan Steel), '积微运网' (Jiweweixuan Logistics Network), '积微循环' (Jiweweixuan Recycling), and '积微钒网' (Jiweweixuan Vanadium Network).
- 积微循环 (Jiweweixuan Recycling):** Shows a green banner with the slogan '积微循环 —— 保护环境，爱护自然' (Protect the Environment, Love Nature). It lists categories for waste materials (废次材, 废闲置材料, 废设备) and highlights '废钛' (Waste Titanium) and '废钒' (Waste Vanadium) processing.
- 拍卖场 (Auction House):** Displays three auction items:
 - 竞拍: 出售新都约11.94吨报废废钢及低合金钢 ...
当前价: ¥ 27939.60
保证金: ¥ 5000.00
距结束: 00:30:48
[去报价](#)
 - 竞拍: 出售新都约2.19吨报废不锈钢类 ...
当前价: ¥ 13514.49
保证金: ¥ 3000.00
距结束: 00:30:48
[去报价](#)
 - 竞拍: 出售新都约42.12吨报废Cr、Mo钢类 ...
当前价: ¥ 109259.28
保证金: ¥ 20000.00
距结束: 00:30:48
[去报价](#)

4.2ABP的成功案例——要买车(上海)



4.3 ABP的成功案例——健康724(上海)

The screenshot shows a mobile application interface for 'Health724'. At the top, there's a navigation bar with a lock icon indicating security, the URL <https://wx.jk724.com>, and device information (iPhone 6/7... 414 x 736 97% Online). A red banner at the top says '您还没有登录，请先登录。、维生素、美白' (You have not logged in, please log in. Vitamins, Whitening). Below this is a promotional banner for 'VIP入门套餐 免邮免税' (VIP Introduction Package, Free Shipping, Tax-Free) featuring a bottle of calcium supplement. To the left, a sidebar titled 'Visualize / 724电商平台接口统计' (724 E-commerce Platform Interface Statistics) lists various API controllers and their usage statistics. The main content area includes a statistic '【1541873】人, 截至5月3日!' (154,1873 people,截至 May 3rd), three circular icons for '方案' (Solutions), '用户指南' (User Guide), and '商品分类' (Product Categories), and two promotional cards: '坚持锻炼100天' (Persist in exercising for 100 days) and '健康管理师推荐' (Recommended by Health Management师).



4.4 ABP的成功案例——易炬车辆智能运维平台(武汉)

The image displays four screenshots of the Yiju Vehicle Intelligent Maintenance Platform, illustrating its features:

- Top Left:** Home dashboard showing vehicle status (入网车辆, 在线车辆, 故障车辆, 维修车辆, 新增车辆, 置换车辆) and daily data analysis (车辆上线率, 人员出勤率, 任务处理率, 行驶总里程, 车辆总量).
- Top Right:** Department Management screen showing a tree view of departments (北京, 上海, 广州, 深圳, 成都) and a modal for User Settings (用户设置).
- Bottom Left:** Global Map Dashboard showing vehicle locations across China.
- Bottom Right:** Screenshot of the Microsoft Visual Studio IDE showing the code for permission management in the EtochFramework project.

4.5ABP的成功案例——京师在线(北京)

The image shows two screenshots of the JingShiOnline.com website. The left screenshot displays the homepage with a teal header featuring the logo '京师在线' (JINSHI ONLINE) and the tagline '最诚信的法律电商平台' (The most honest legal electricity suppliers platform). It includes a shield icon with '赔付责任' (Compensation Responsibility) and a 100% refund guarantee. Below the header are three navigation tabs: '首页' (Home), '依据业务领域找律师' (Find a lawyer by business field), and '依据审判机关、案由找律师' (Find a lawyer by court and case type). The main content area features a section titled '我们的优势' (Our Advantages) and four circular icons representing different services. The right screenshot shows the 'Lawyer List' page with a search bar and dropdown menus for filtering by region, name, and law firm. The results list includes profiles for several lawyers, such as 刘宏辉 (Liu Honghui), 蔡春雷 (Cai Chunlei), 丁香 (Ding Xiang), and 林福明 (Lin Fuming), each with their profile picture, years of experience, department, and specific practice areas.

4.6ABP的成功案例——WOOW.SITE(岳阳)



4.7 ABP的成功案例——MORE AND MORE



我们下周五上线一个在线教育类的项目。后端也是 abpcore + identity server4 来得及的话到时候也把地址给你

上午10:01:41

你们有正式使用ABP上生产的项目么?



[自动回复] 您好，我现在有事不在，一会再和您联系。

下午3:22:52

我找点成功案例，放PPT上宣传 😊



还没呢~ 最近的一个 还在内测~ 公网上无法访问的



最快 5月底 上线。

ABP是一个很好的起点！ 谢谢大家！

开源
便于学习

设计优良

DDD入门

项目
快速启动

集中精力开发
可维护可扩展
的业务

社区活跃



ABP 架构设计交流群4
扫一扫二维码，加入群聊。