



Community  
Meetup  
Shanghai



IT 农工会 - 随便聊

跟社区专家一起聊聊混合云、领域驱动、区块链和数字营销



Community  
Meetup  
Shanghai

# ASP.NET Core下领域驱动设计的实践

陈晴阳





Community  
Meetup  
Shanghai

# 自我介绍



- 陈晴阳 (daxnet)
- 系统分析员 (2006)
- 微软最有价值专家 (MVP, 2012-)
- 珀金埃尔默 (PKI) 首席软件工程师
- 专注于Visual Studio与.NET技术
- 关注前沿开发技术
- 爱生活, 爱分享

- ASP.NET Core概览
- 领域驱动设计 (DDD) 简介
- ASP.NET Core实践技巧
- 系统架构设计的思考
- 案例：仓储模式的实现
- 更多阅读
- 问题解答





Community  
Meetup  
Shanghai

# 概览与简介



# ASP.NET Core概览



- 经典ASP.NET框架的完全重写
- 基于.NET Core
- 模块化的Web应用程序开发框架
- 开源
- 跨平台 (Windows, Linux, MacOS)
- 云友好 (Cloud-Friendly)
- 支持多种开发工具



# 各种 “驱动”

- 测试驱动开发 (TDD)
- 行为驱动开发 (BDD)
- 功能驱动开发 (FDD)
- API驱动开发 (ADD)
- 领域驱动**设计** (DDD)



# 领域驱动设计



- 以领域知识（领域模型）为核心
- 建立通用语言为目标，减少沟通成本
  - 实体 (Entities)
  - 值对象 (Value Objects)
  - 聚合/聚合根 (Aggregates/ARs)
  - 规约 (Specification)
  - 工厂 (Factories)
  - 仓储 (Repositories)
  - 服务 (Services)
  - 界定上下文 (Bounded Contexts)
  - ...
- 涵盖系统架构、设计和实现的方方面面







Community  
Meetup  
Shanghai

# ASP.NET Core实践技巧



# 为何选择ASP.NET Core



- 开源与跨平台
- 模块化的设计，框架丰富
- 契合微服务的设计理念
- 支持docker容器部署，能方便地运行于云环境
- 强大的开发工具和开发语言支持



# 数据传输对象 (DTO)



Community

- 多进程间数据传递的载体
- 序列化与反序列化操作
- 与值对象的区别
- “对象爆炸” 问题
- 解决方案：
  - dynamic关键字
  - 匿名类型与JSON对象

```
[HttpPost("calculate")]
0 references | 0 changes | 0 authors, 0 changes | 0 requests | 0 exceptions
public IActionResult CalculateValue([FromBody] dynamic model)
{
    var x = (int)model.x;
    var y = (int)model.y;

    return Ok(new
    {
        add = x + y,
        sub = x - y
    });
}
```

```
PS C:\Users\daxne> $request = @{x=10; y=5;}
PS C:\Users\daxne> $json = $request | ConvertTo-Json
PS C:\Users\daxne> $response = Invoke-RestMethod 'http://localhost:51689/api/customers/calculate' -Method Post -Body $json -ContentType 'application/json'
PS C:\Users\daxne> echo $response | ConvertTo-Json
{
  "add": 15,
  "sub": 5
}
PS C:\Users\daxne>
```



# 异步方法async/await的支持


```
[HttpPost]
0 references | daxnet, 5 days ago | 1 author, 3 changes | 0 requests | 0 exceptions
public async Task<IActionResult> Create([FromBody] dynamic model)
{
    var name = (string)model.Name;
    if (string.IsNullOrEmpty(name))
    {
        return BadRequest();
    }

    using (var connection = new SqlConnection(connectionString))
    {
        var customer = new Model.Customer(name);
        await connection.ExecuteAsync(InsertSQL, customer);
        await this.eventBus.PublishAsync(new CustomerCreatedEvent(name));

        return Created(Url.Action("Get", new { id = customer.Id }), customer.Id);
    }
}
```



# 应用程序配置信息

- appsettings.json
  - appsettings.<EnvironmentName>.json
  - 环境变量 (Environment Variables)
- 



# ASP.NET Core Identity与界定上下文



- 区分User的概念
  - Identity User
  - 领域模型中User的概念
- 以应用层服务的形式隔离界定上下文





Community  
Meetup  
Shanghai

# 系统架构设计的思考



# 面向对象分析与设计的五大原则



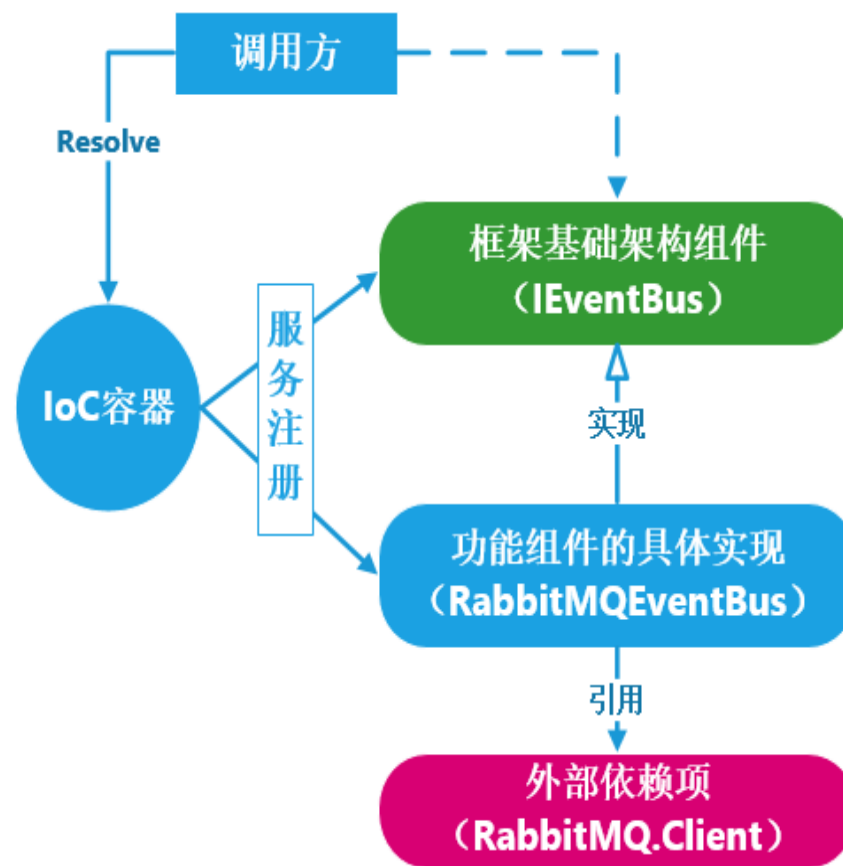
- S: 单一职责原则 (Single Responsibility Principle)
- O: 开-闭原则 (Open-Closed Principle)
- L: 里氏替换原则 (Liskov Substitution Principle)
- I: 接口分离原则 (Interface Segregation Principle)
- D: 依赖反转原则 (Dependency Inversion Principle)





# 合理设计组件的依赖关系

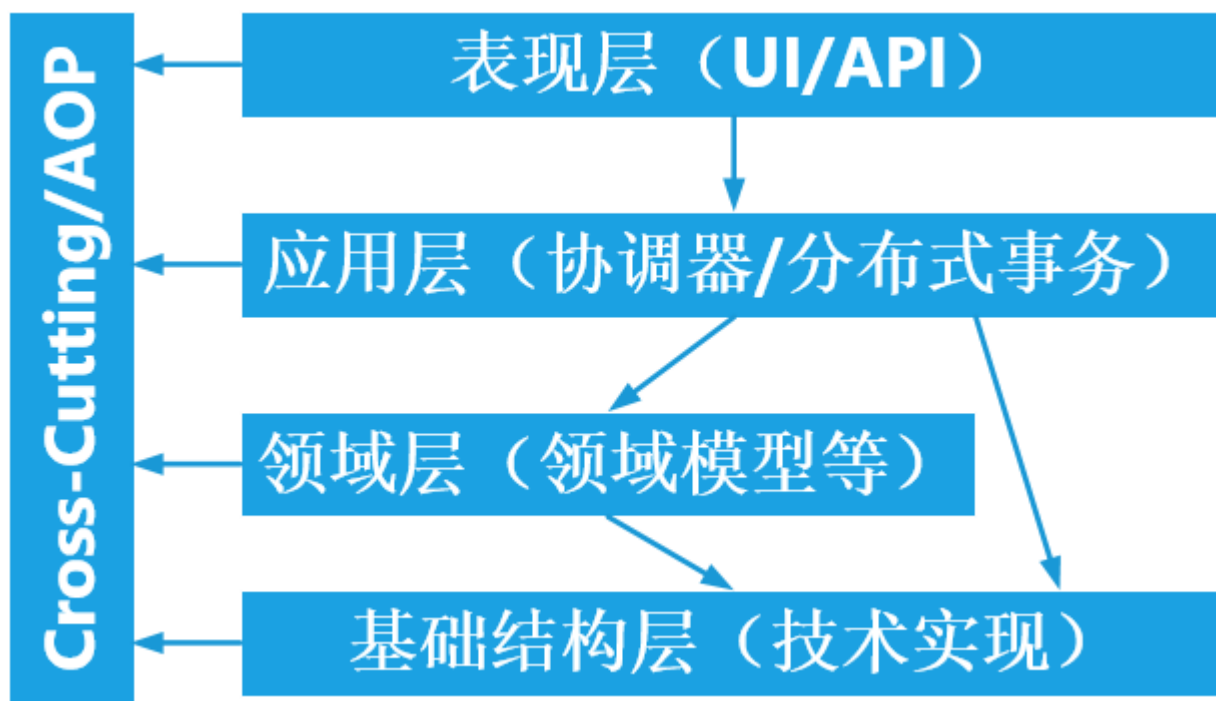
- 框架基础架构组件
  - 定义框架接口及对象关系
  - 仅依赖于编程工具SDK
  - 保持框架设计的纯净度
- 功能组件的具体实现
  - 实现“框架基础架构”中的接口
  - 实现过程依赖于外部依赖项
- 外部依赖项
  - 提供基础设施相关的SDK API
  - 通常由第三方厂商提供（通过NuGet下载）
- IoC容器
  - 服务实例的注册
  - 管理实例的依赖关系与生命周期
- 调用方
  - 通过IoC容器获得所需要的服务实例
  - 原则上仅需依赖于“框架基础架构组件”程序集



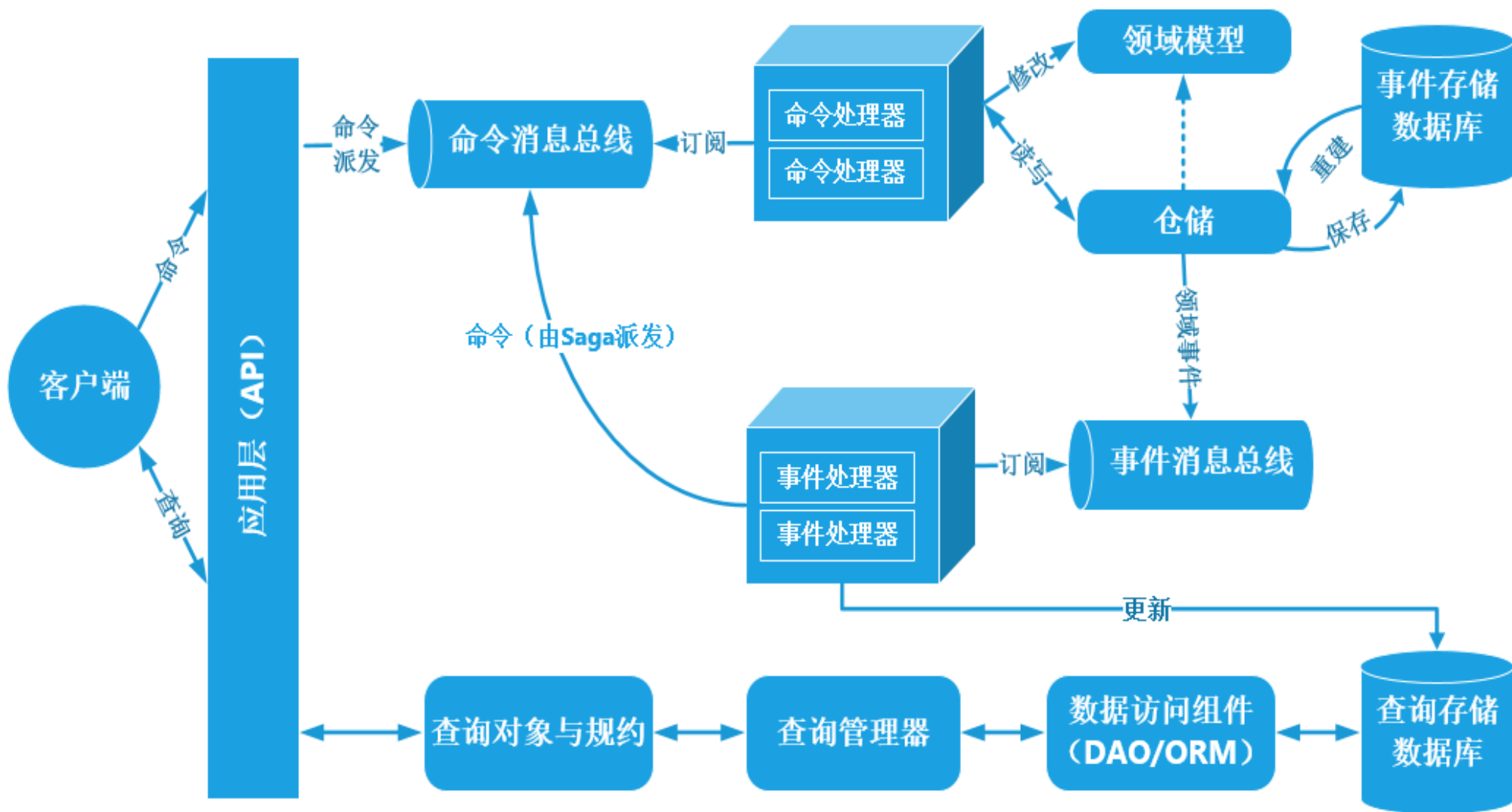
# 经典分层架构



Community



# 事件驱动的命令查询职责分离 (CQRS)



# 微服务与云友好的设计



- 基于界定上下文（BC）划分微服务边界
- 设计合理的通信模式
  - 同步：RESTful API
  - 异步：事件消息
- 支持云端的部署策略
  - 构建与发布的脚本化
  - 遵循云端部署的最佳实践
  - 容器化（dockerize）
- 避免单点失败



# 演示

## ASP.NET Core下仓储模式的实现



Community  
Meetup  
Shanghai





Community  
Meetup  
Shanghai

更多阅读



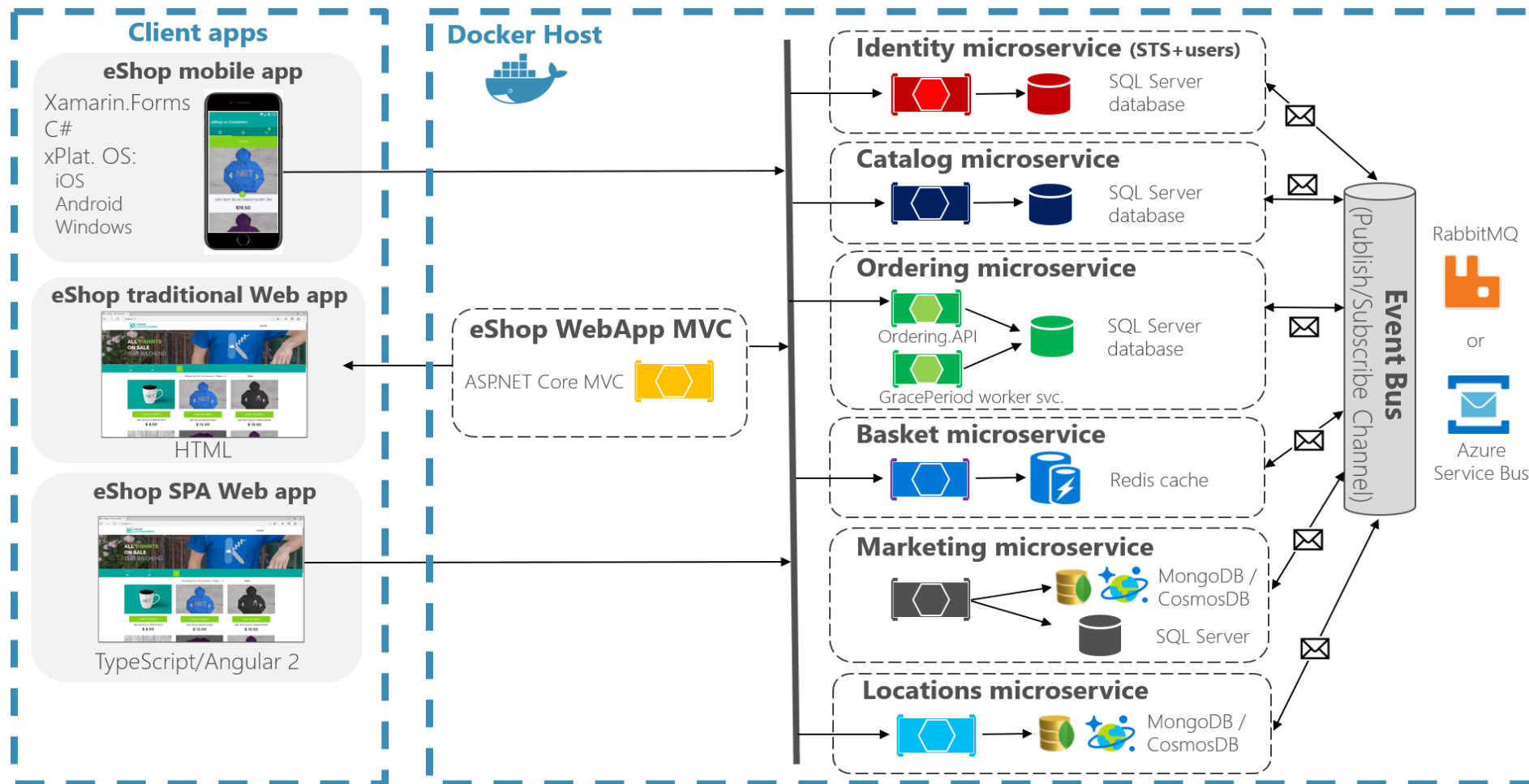
# 微软官方案例：eShopOnContainers

<https://github.com/dotnet-architecture/eShopOnContainers>



## eShopOnContainers reference application

(Development environment architecture)

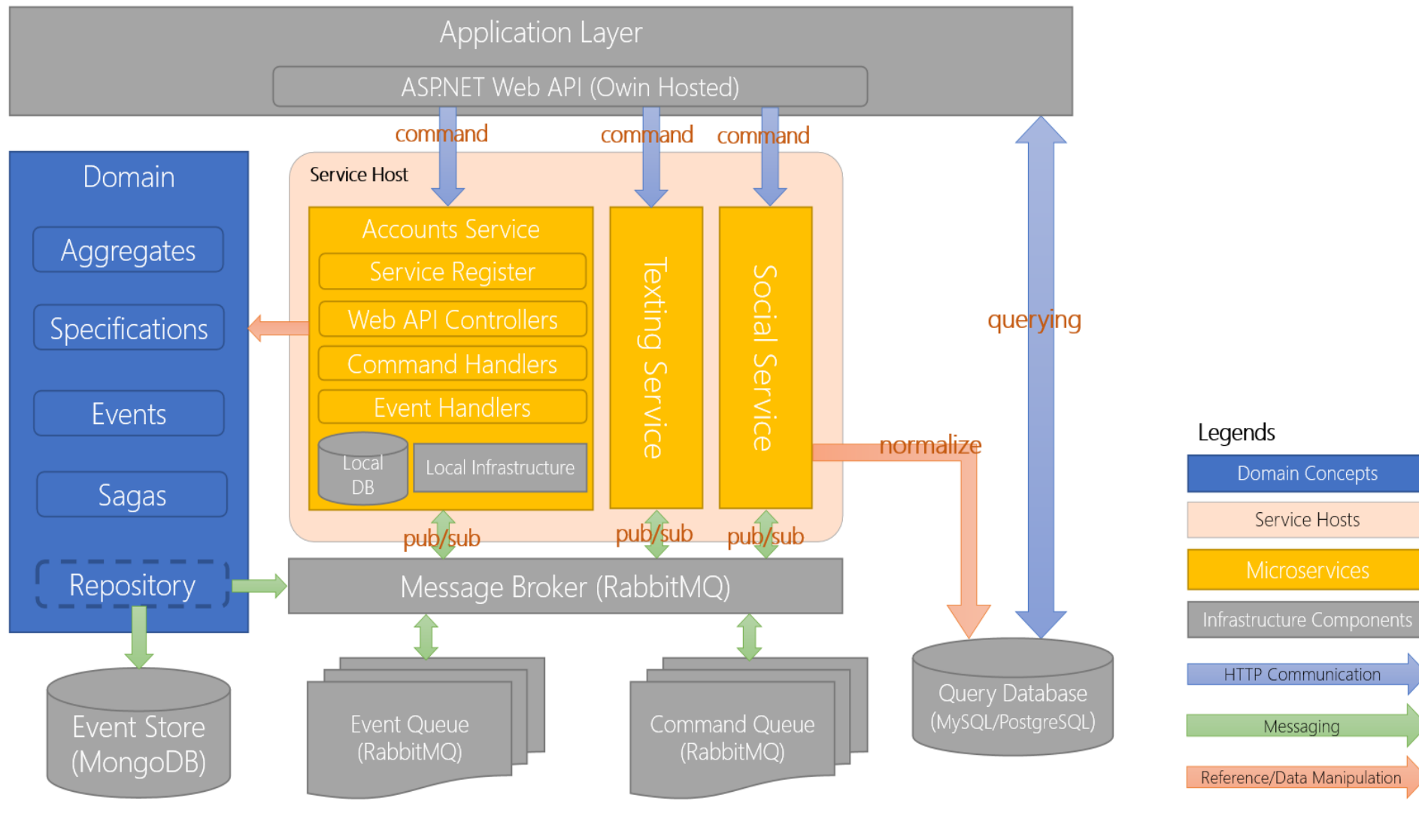


# CQRS与微服务的架构案例：WeText

<https://github.com/daxnet/we-text>



Community





# .NET领域驱动设计开发框架：Apworks

<https://github.com/daxnet/apworks-core>



Community

- 开始于2009年 (原: [github.com/daxnet/Apworks](https://github.com/daxnet/Apworks))
- 2016年基于.NET Core完全重写
- 支持.NET Core 1.1与2.0
- 基于ASP.NET Core提供RESTful API快速构建方案
  - Data Services
  - Hypertext Application Language
  - Query Specification Parsers
- 为.NET下领域驱动设计实践提供框架与SDK
  - 领域驱动设计基础元素的实现
  - 消息派发、订阅与处理框架
  - CQRS事件存储与快照框架
  - More...





Community  
Meetup  
Shanghai

# 问题解答

技术博客：



开源代码库：





Community  
Meetup  
Shanghai

**vă mulțumesc**

*Salamat sa iyo*

**Grazie**

**Спасибо**

*Teşekkür ederiz*

*Jamädi*

*Gracias*

**Merci**

*Aitäh Dziękuje Mauruuru ia oe*

**Danke** *Obrigado* **dioloh**

*Terima kasih* *Cảm ơn bạn* **תודה** **አዳደሩ**

*Vinaka vakalevu*

**Thank you**

**谢谢**

**Tak**

**Благодаря** *köszönöm*

*Ευχαριστούμε* *Hvala tí*

**ขอบคุณ** **謝謝** *Děkuju*

*Dankie* *Gràcies*

*köszönöm*

*감사합니다*

*hvala* **Dank u** **धन्यवाद**

**شكرا**

**ありがとう**

**Дякую**