

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

模式与实践系列课程之一

Web Client Software Factory

张大磊

开发技术经理

平台及开发技术部

<http://msdpe.cnblogs.com>

Ray Zhang

Developer Evangelist

D&PE

Microsoft

本次课程内容包括

- 为什么架构师需要了解软件工厂
- 软件工厂的相关概念
- Web Client Software Factory(WCSF)介绍
- WCSF的主要设计考虑
- 应用WCSF三部曲
- 经验分享： 如何理解争议点、规避风险
- 参考资源

收听本次课程需具备的条件

本系列课程不会涉及下列内容, 假定参加者已具备下列基础:

- 理解软件开发全生命周期与团队协作
- 掌握.NET体系及其相关框架、语言、组件
- 掌握企业库、WF、Service设计
- 了解ESD非功能性需求和复杂性
- 掌握一些基础的设计模式, 有分布式应用概念

本次课程不会涉及下列内容, 假定参加者已具备下列基础:

- 对交易系统有所了解

Level 400

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

为什么架构师需要了解软件工厂

为什么架构师需要了解软件工厂

- 人了解事物的几个层面：
迷惑怀疑-培训学习-自己创造-发掘成熟
- Jack Greenfield, Ketih Short对架构失败的总结：
 - 一锤子买卖式的设计和开发
 - 集成性差，过于自治
 - 抽象层级太低
 - 流程处理不完备
 - 对需求会不断增加的特性估计不足
- 向成熟工业学习，简化与自动化公共任务

为什么架构师需要了解软件工厂

- 流水线式的产品开发
- 健壮的基础架构与框架
- 模式驱动
- 上下文相关的向导

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

软件工厂的相关概念

软件工厂的相关概念

几个较为成熟的软件工厂

- Web Client Software Factory (Browser)
- Smart Client Software Factory (Desktop)
- Web Service Software Factory (SOA)

Front-End (展示层与用户体验):

Winform 最佳实践: SCSF

Web 最佳实践: WCSF

Back-End (服务设计与后台支撑):

SOA 最佳实践: WSSF

软件工厂的相关概念

软件工厂将提供:

- GAX/GAT/GAP/Recipe
- Application Block
- 架构文档
- How-to 系列指南
- 对特定模式的介绍与实现
- 应用参考实例
- 快速入门指南

软件工厂的相关概念

- 其他需要了解的背景知识

CAB/CWAB

企业库

AJAX

SOA

数据访问

SQL Server

今天我们要讨论的话题

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

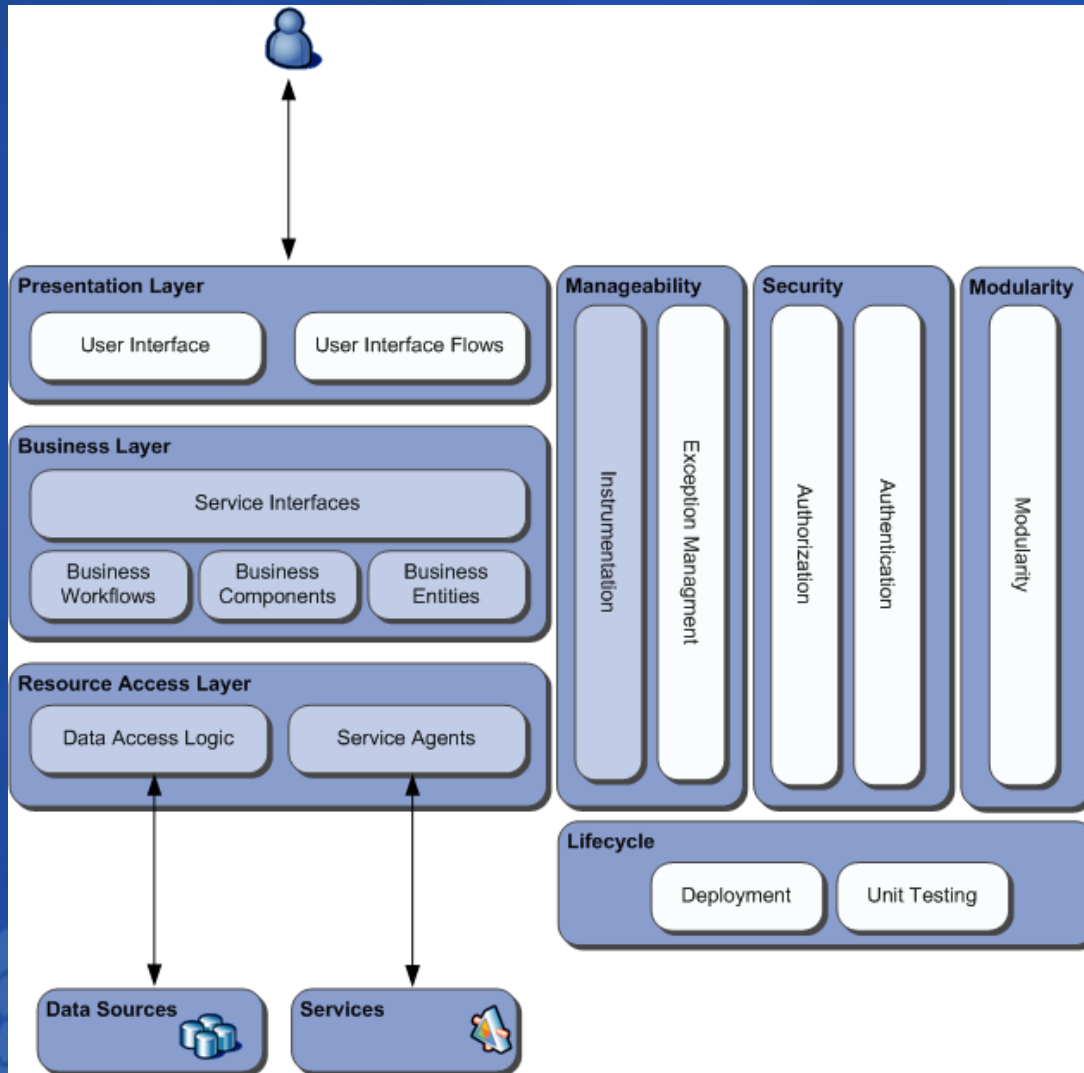
- 使用WCSF制造Web Client（制造网站）

您的潜力, 我们的动力

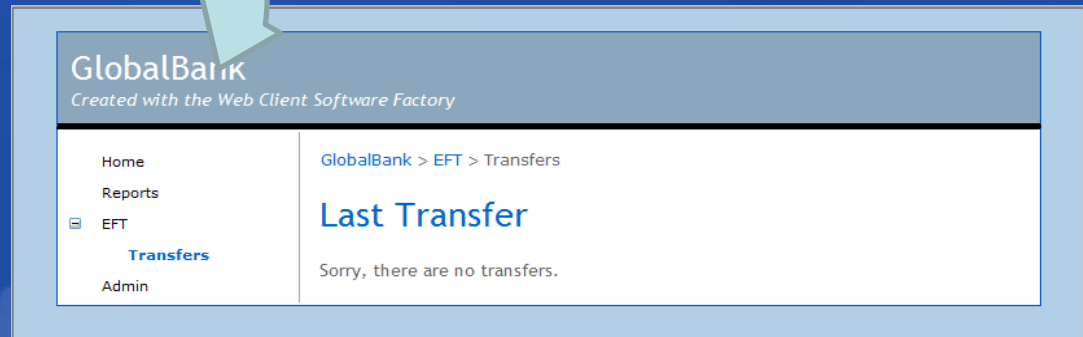
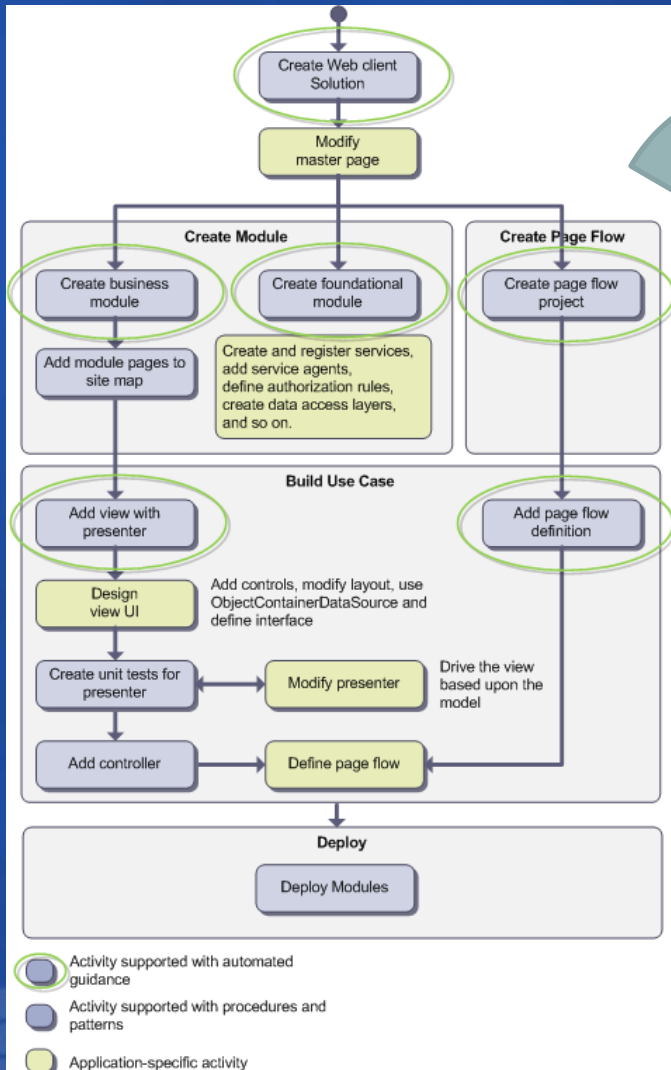
Microsoft
微软(中国)有限公司

Web Client Software Factory介绍

WCSF要解决的问题

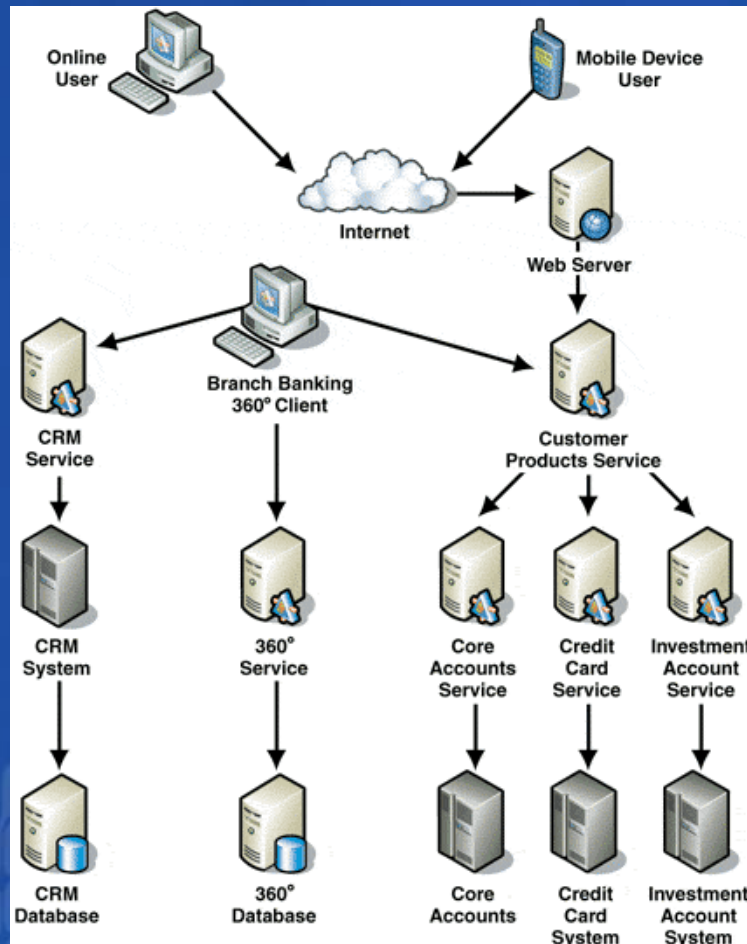


使用WCSF的流程



实际应用场景-Global Bank介绍

Microsoft®
微软(中国)有限公司



您的潜力, 我们的动力

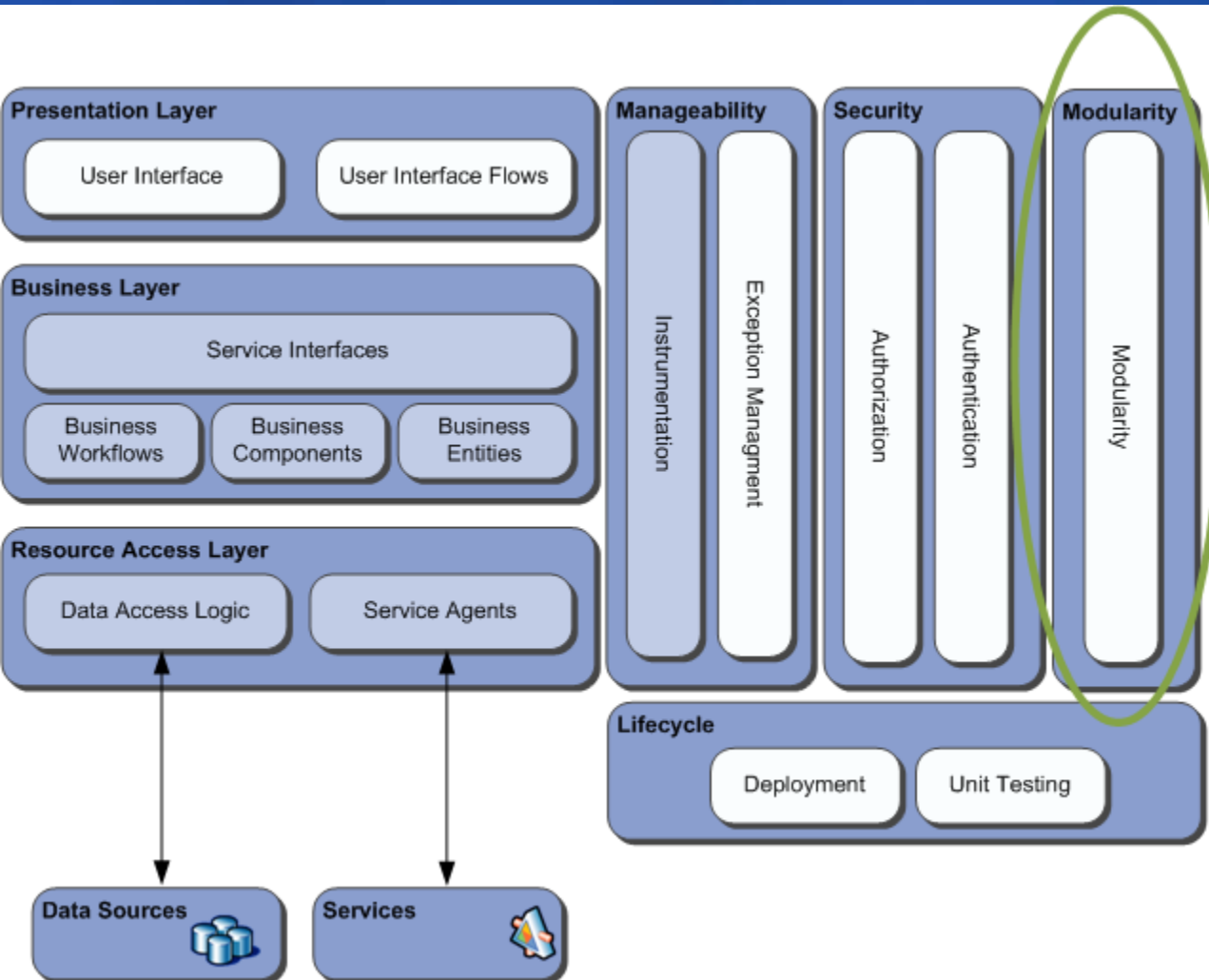
Microsoft
微软(中国)有限公司

WCSF的主要设计考虑

WCSF的主要设计考虑

- ✓ 模块化设计
- ✓ 依赖注入设计
- ✓ 服务与服务定位设计
- ✓ MVP模式的引入
- ✓ 测试驱动开发
- ✓ 企业库的引入
- ✓ 其他: 如ObjectContainerDataSource的引入

模块化设计



封装变化部分
支持多团队开发
便于重用
便于部署

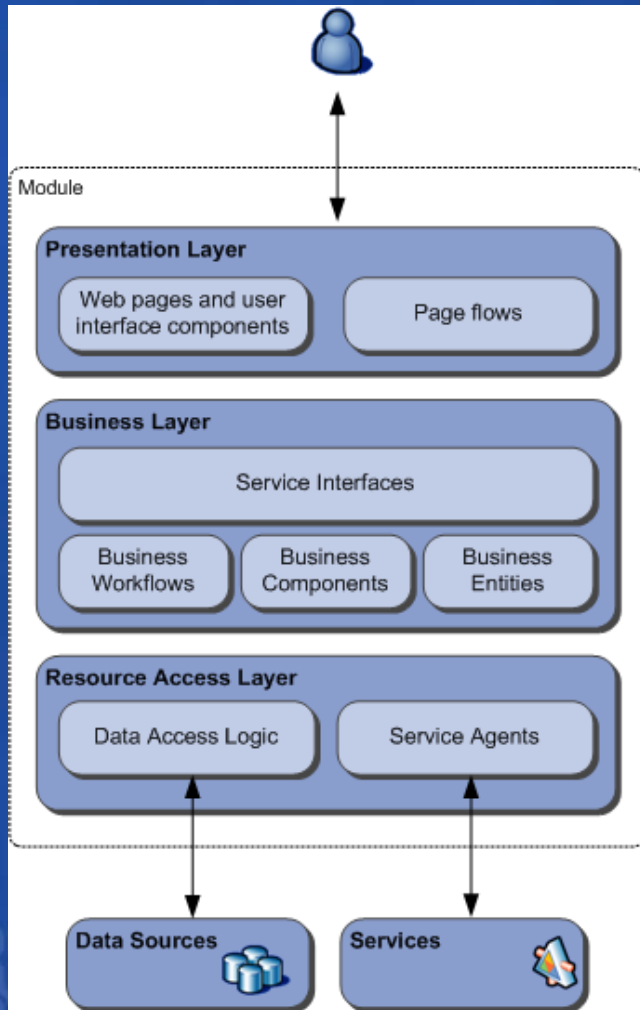
Business Modules
(业务模块)

报表、借贷、帐务

Foundational Modules
(基础模块)

登录、日志、
安全、通用服务

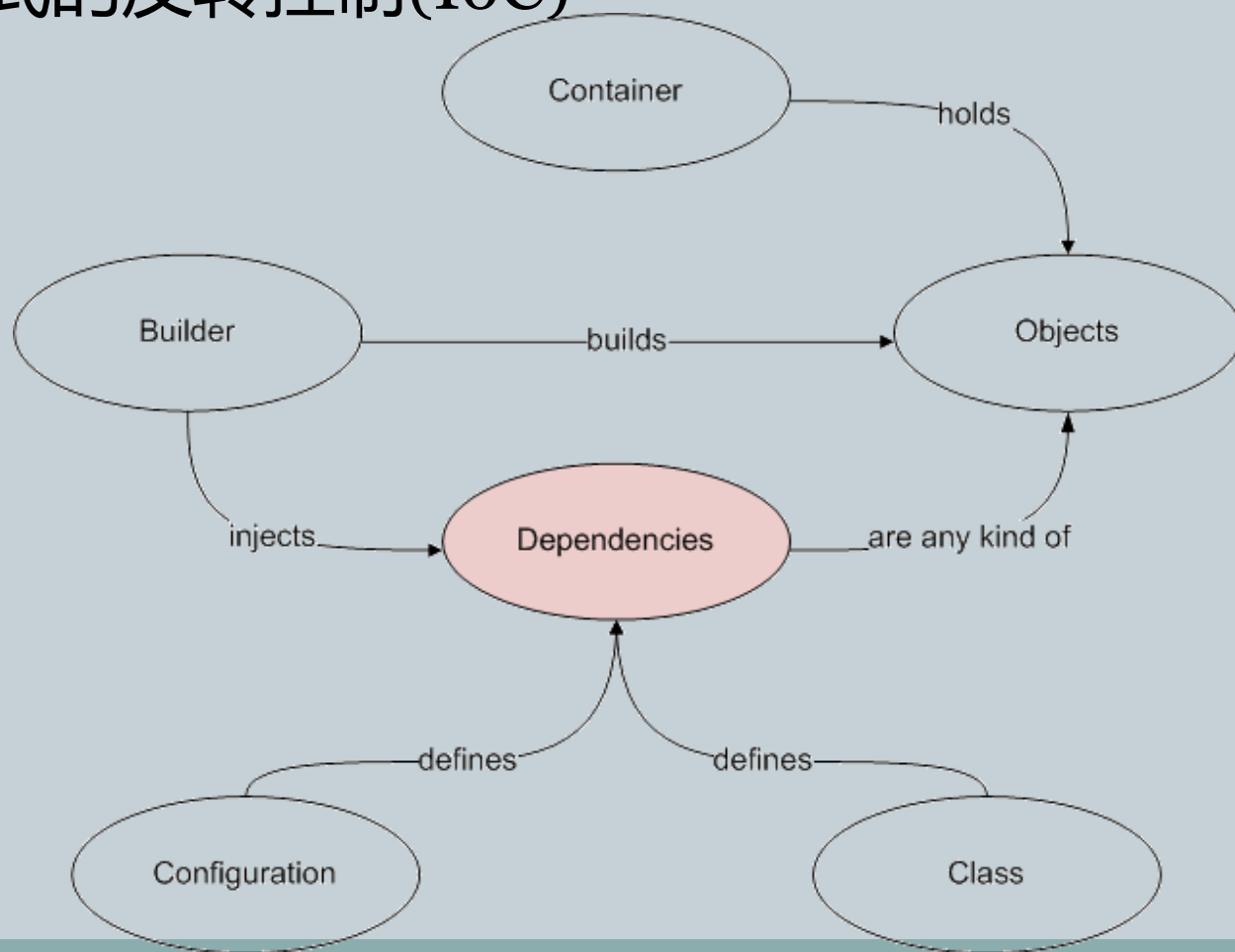
模块化设计



- 模块包含模块加载器(Module_INITIALIZER)
- 模块包含1个或多个控制器(Controllers)
- 模块包含网页、用户控件、逻辑(Presenter)

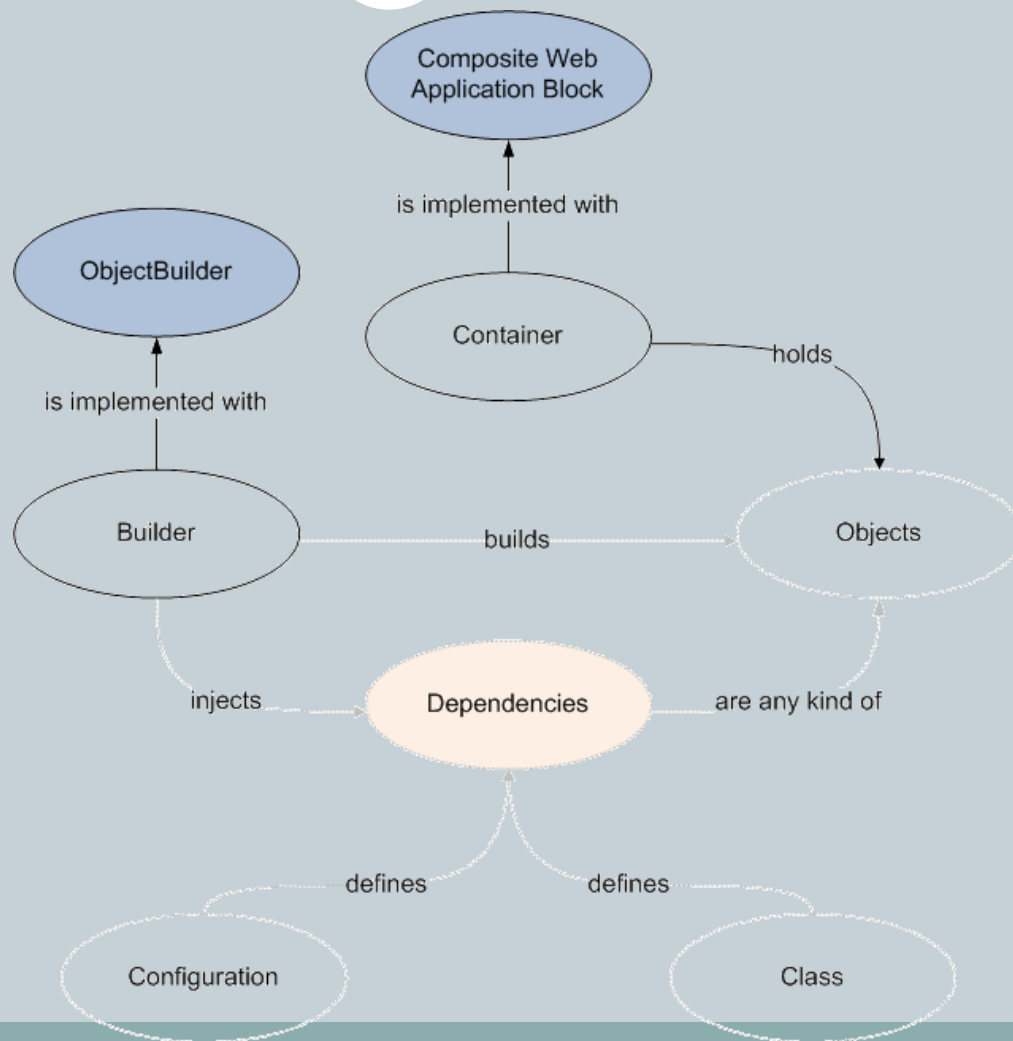
依赖注入设计

- 好莱坞式的反转控制(IoC)



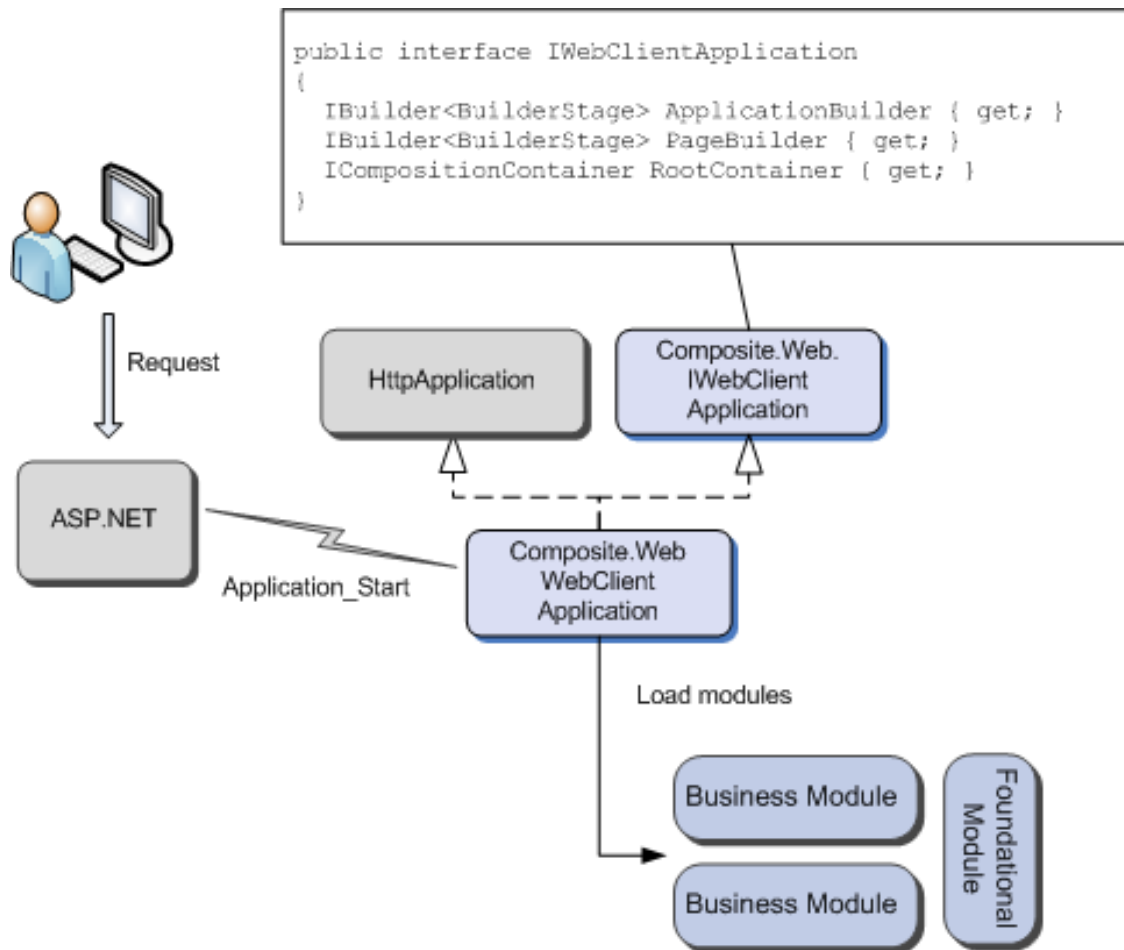
依赖注入设计

• 实现-原理

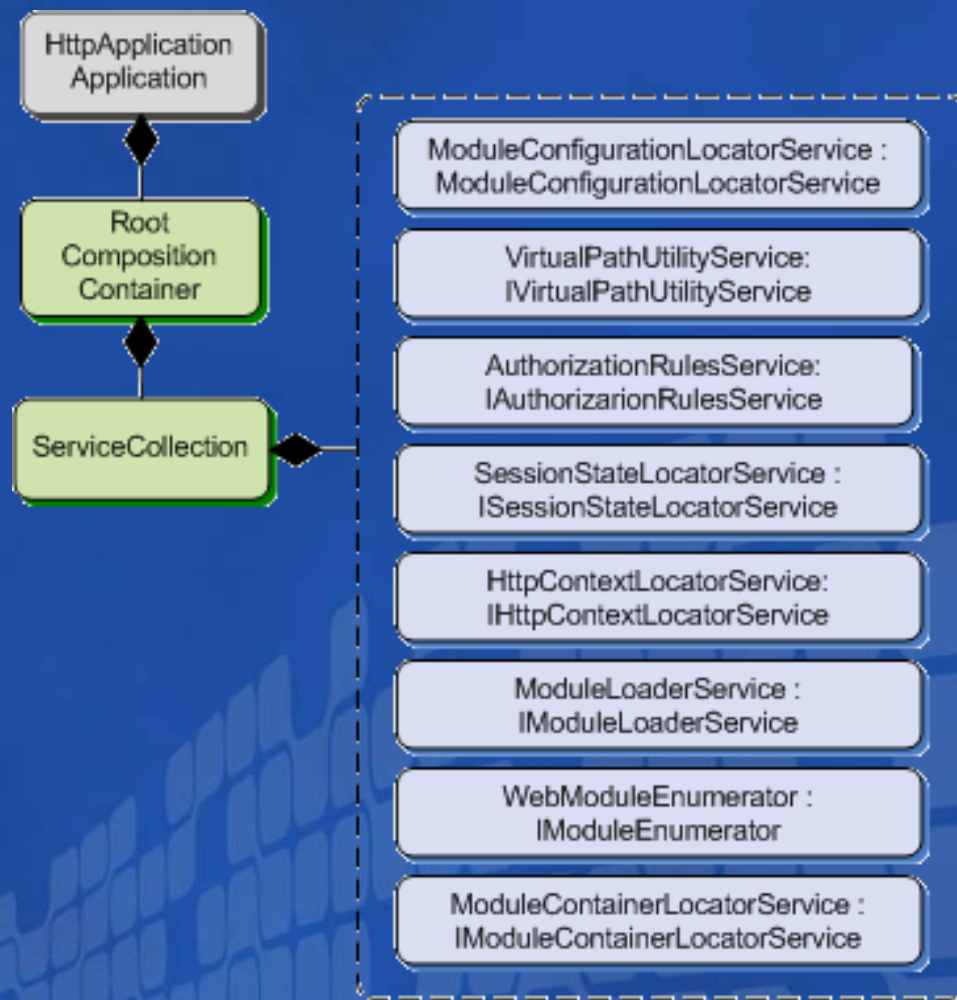


依赖注入设计

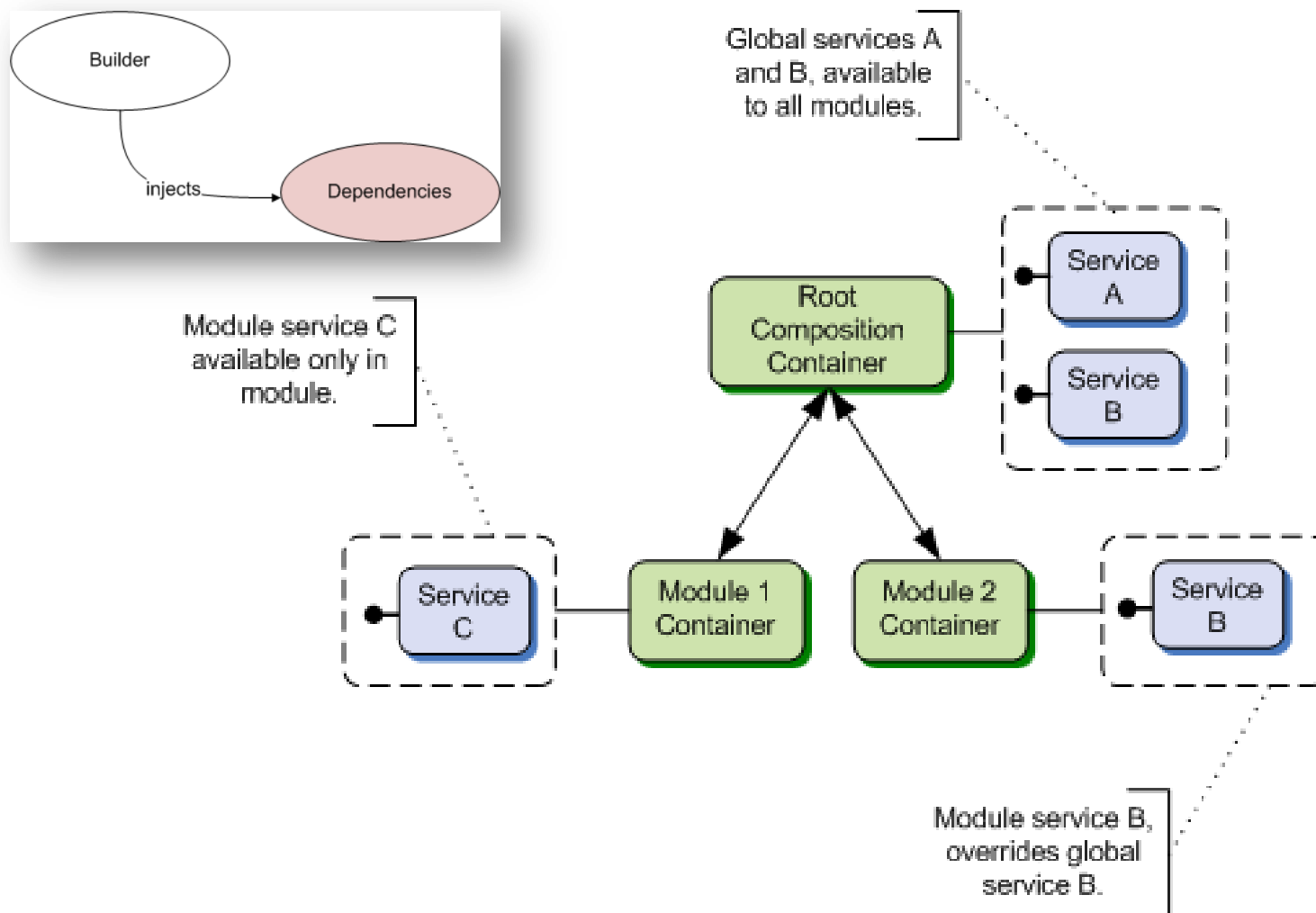
- 实现-结果



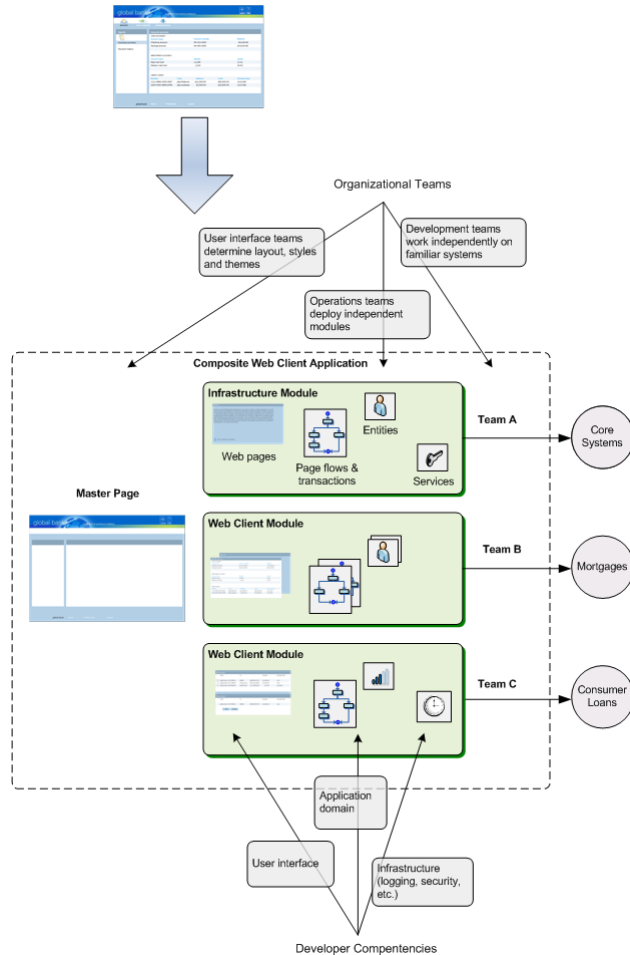
服务与服务定位设计



服务与服务定位设计



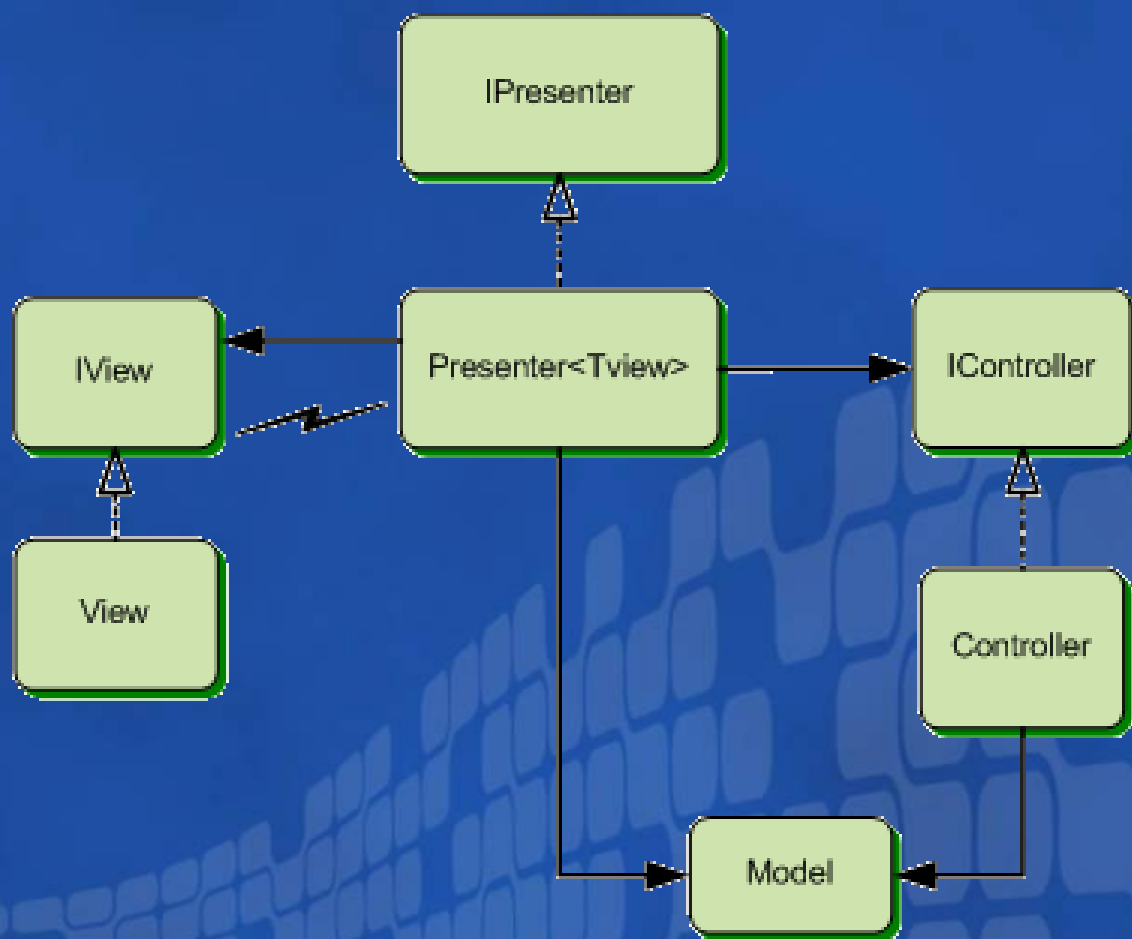
服务与服务定位设计



➤ 责任分离

➤ 解耦

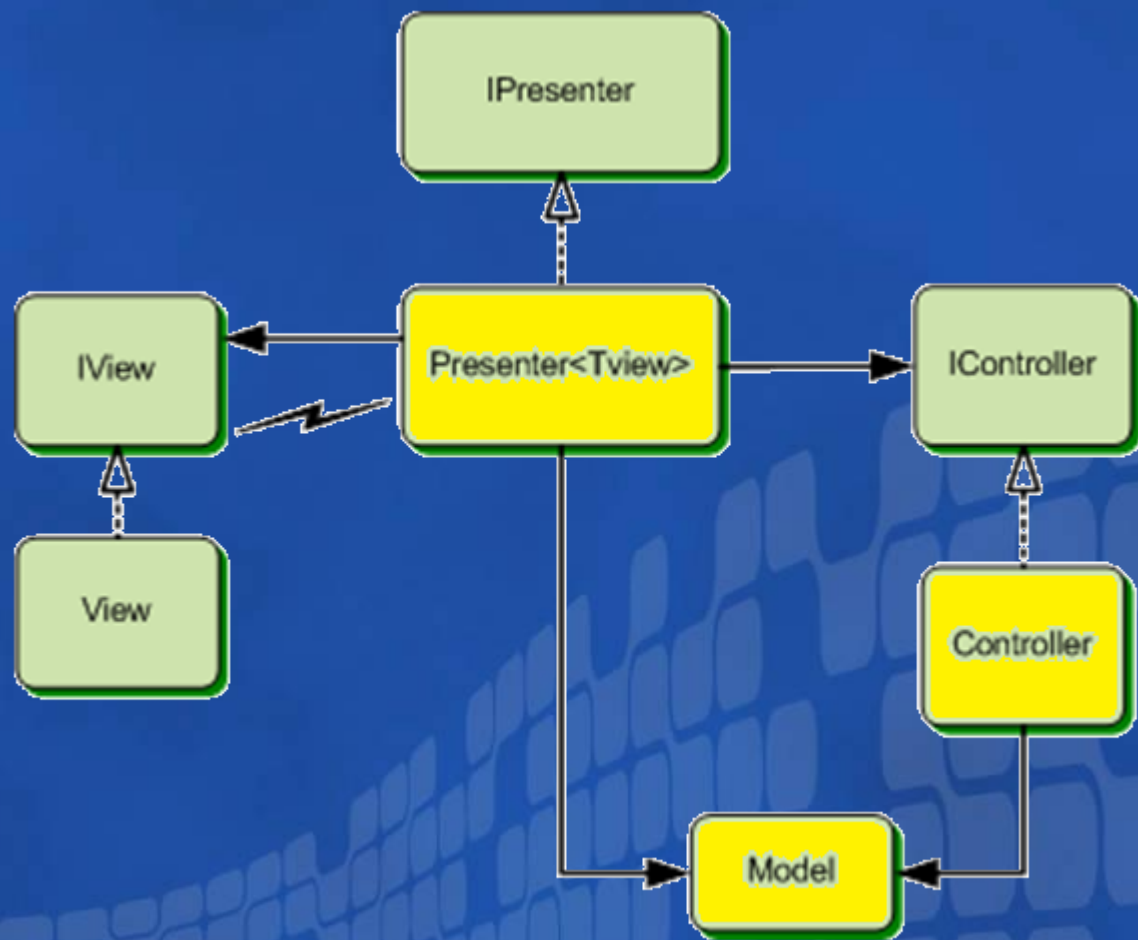
MVP模式的引入



➤ 保证任何操作P层的代码只操作P层，将所有领域代码和数据访问逻辑清晰地分离到程序的相关位置。

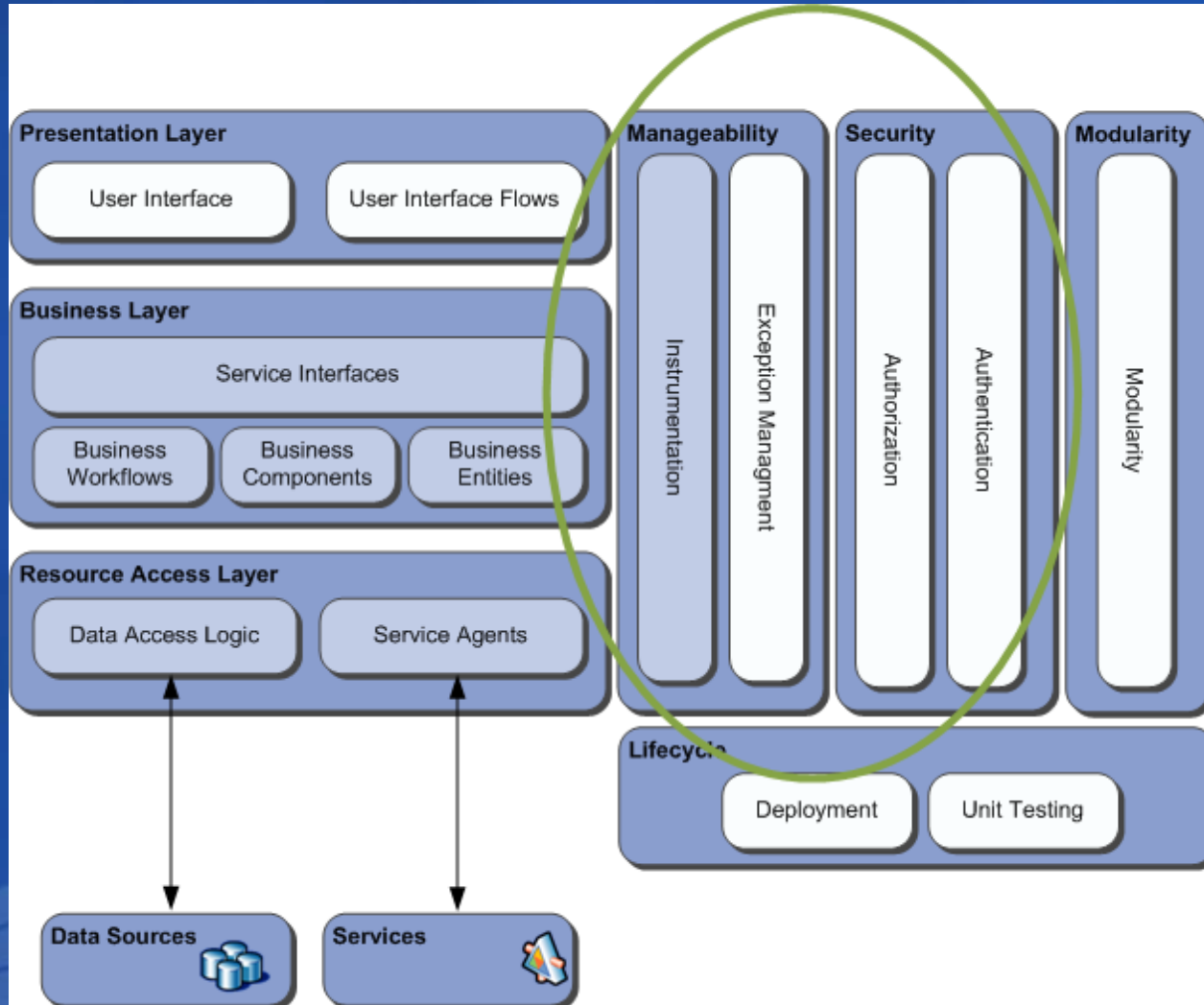
- 避免意大利面代码
- 增加可维护性
- 便于迁移
- 增加测试代码覆盖率

测试驱动开发

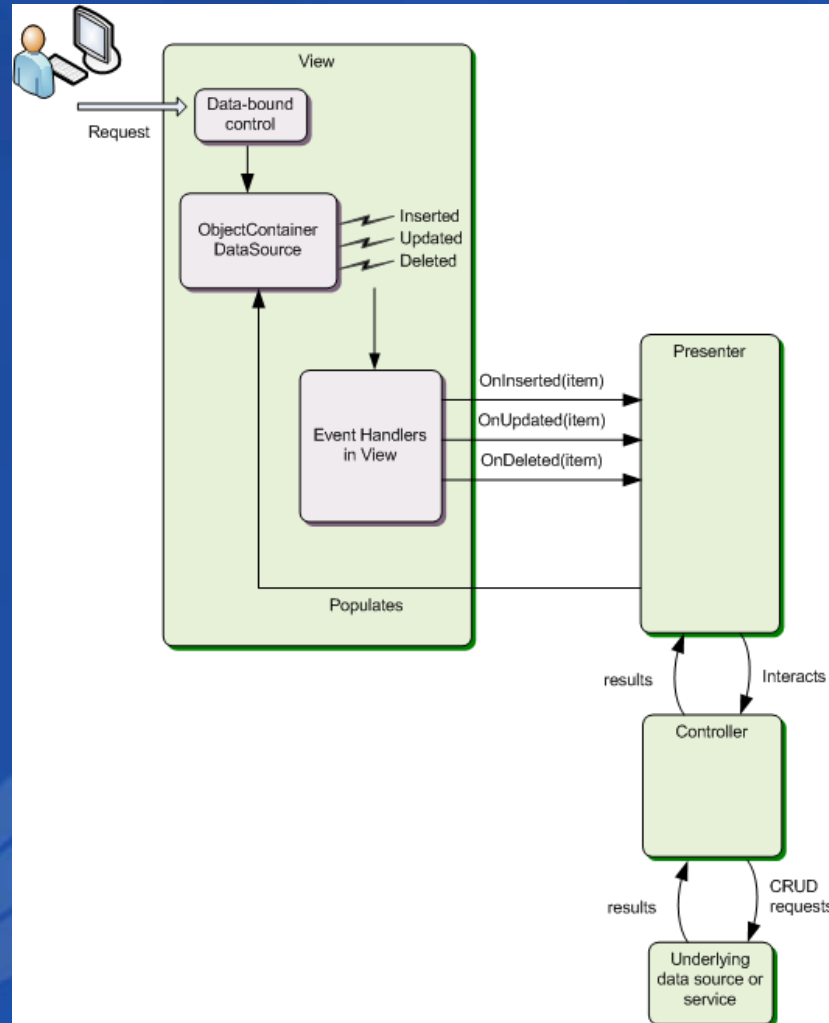


- 先写测试代码
- TDD不只是一个测试工具
- 代码测试可通过产生愉悦效应
- 仅关注需要实现的部分
- Mock模拟
- 辅助其他敏捷实践

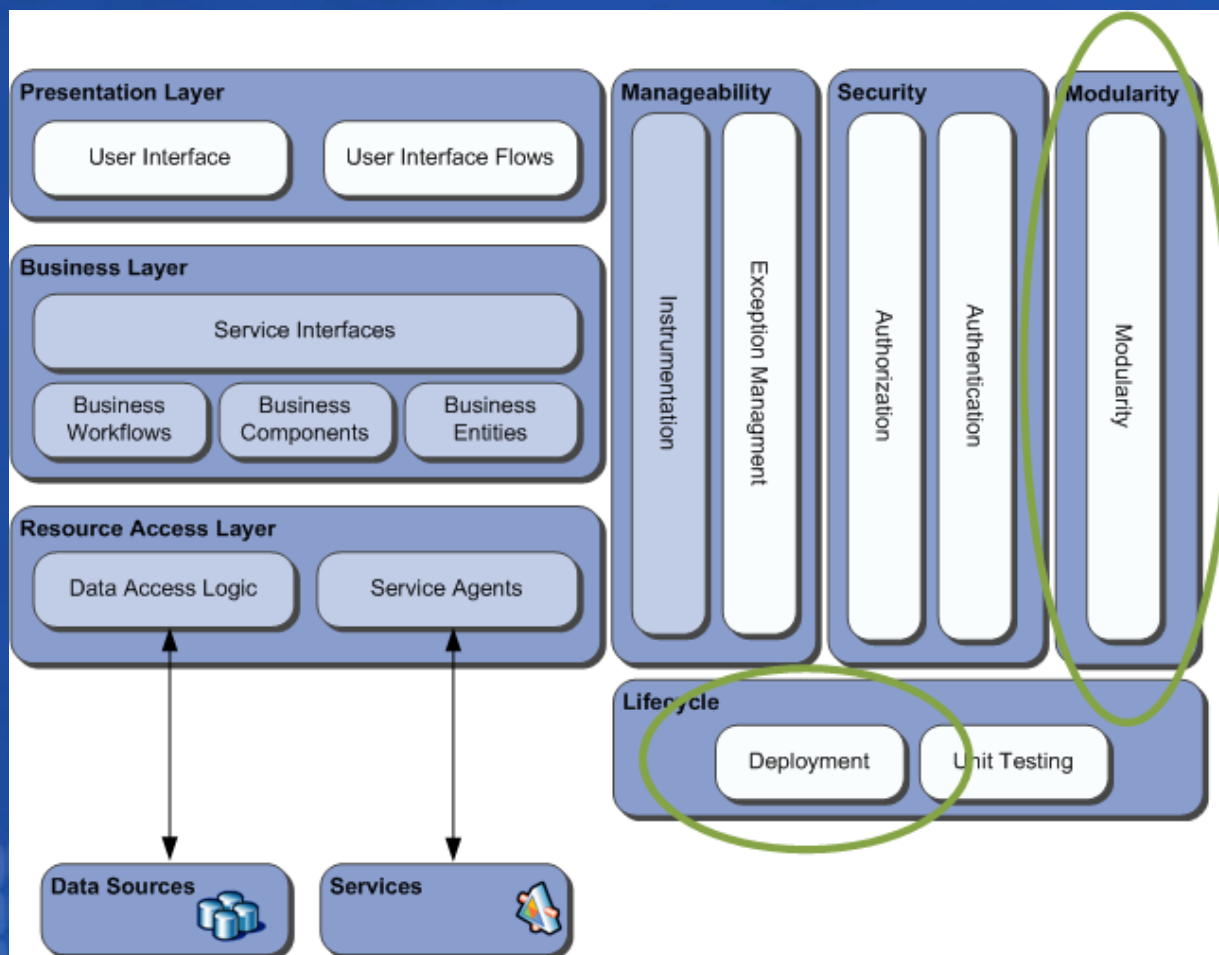
企业库的引入



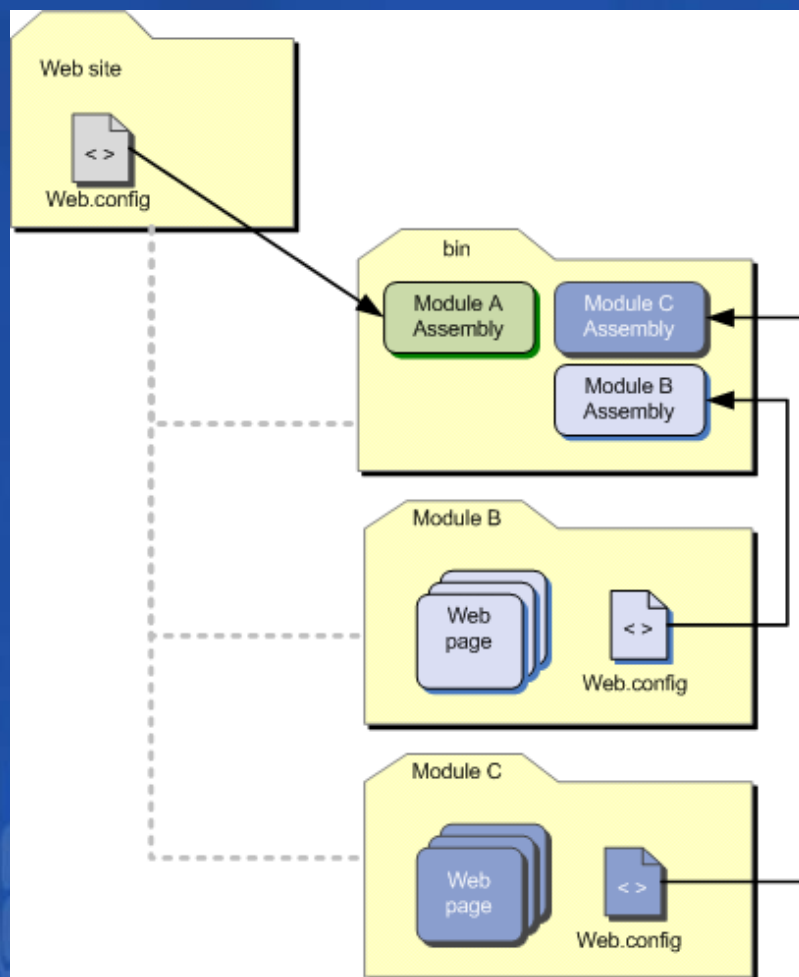
其他-ObjectContainerDataSource



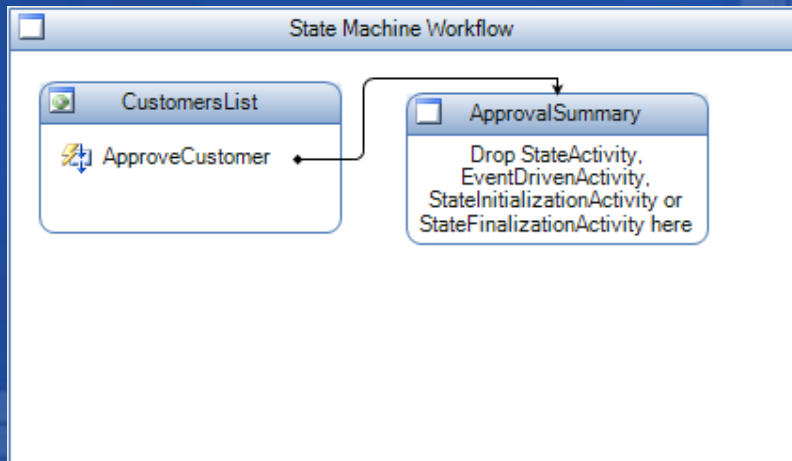
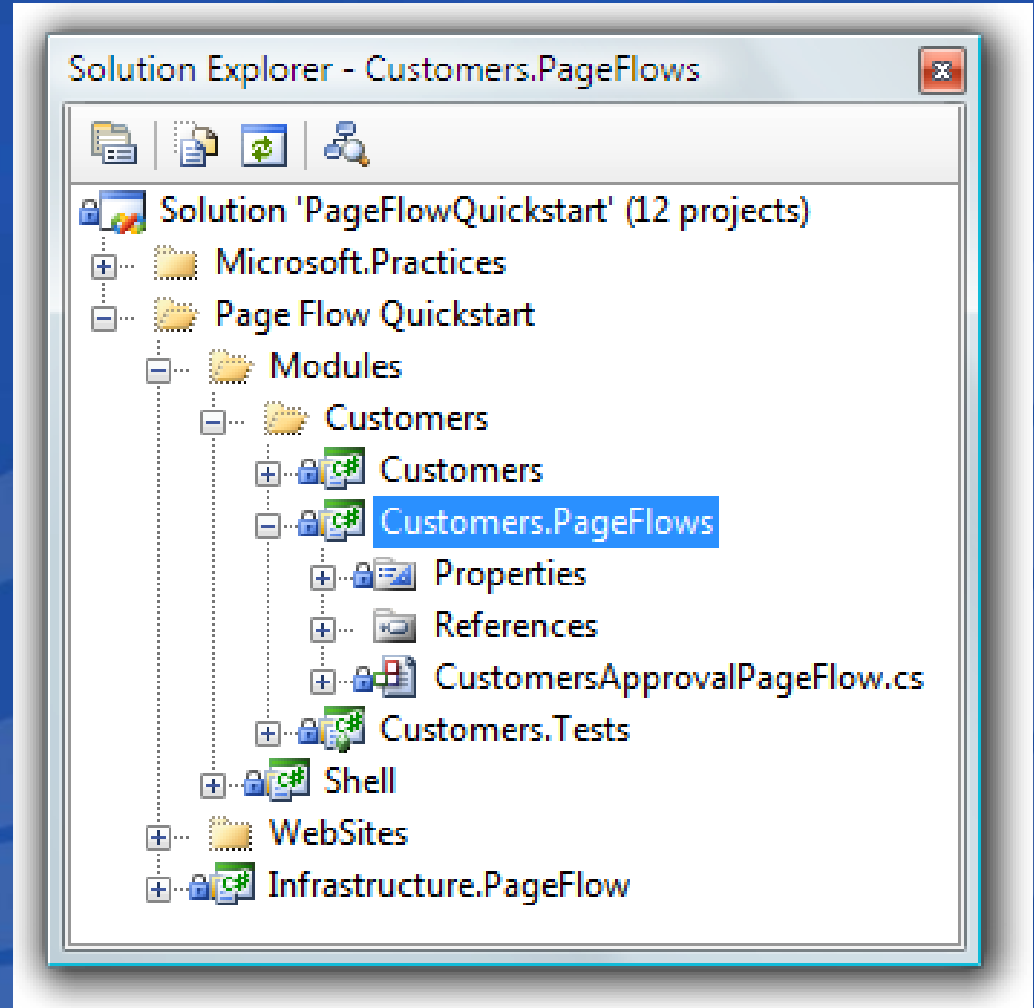
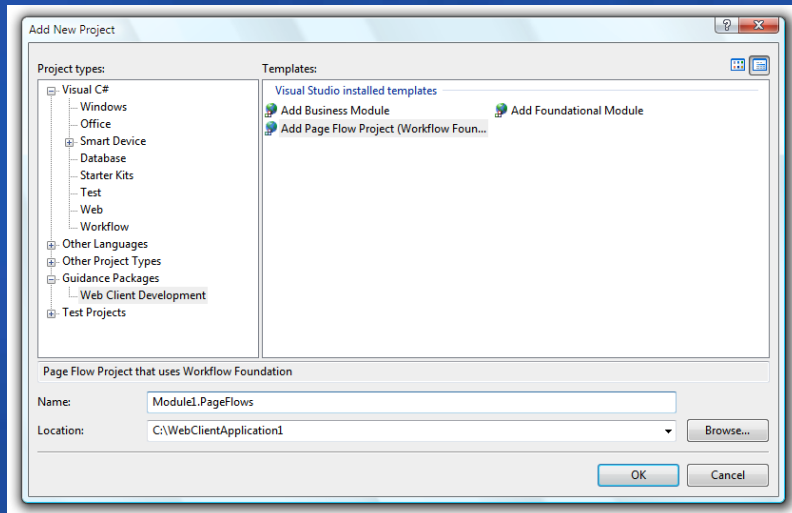
其他-部署



其他-文件夹、程序集、配置文件的关系



其他-Page Flow的引入



您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

应用WCSF三部曲：实施、定制、自动化

实施

- ✓一、创建解决方案与业务模块
- ✓二、创建服务
- ✓三、添加视图与单元测试
- ✓四、V-P数据绑定
- ✓五、创建基础模块
- ✓六、添加认证授权、日志、错误处理模块
- ✓七、使用页面流

演示-略

◆参照WCSF的动手实验进行实施

◆针对阅读英语有障碍的朋友，我会制作一些简化步骤并放到博客上，地址是：

Chinfest Architecture:

<http://msdpe.cnblogs.com>

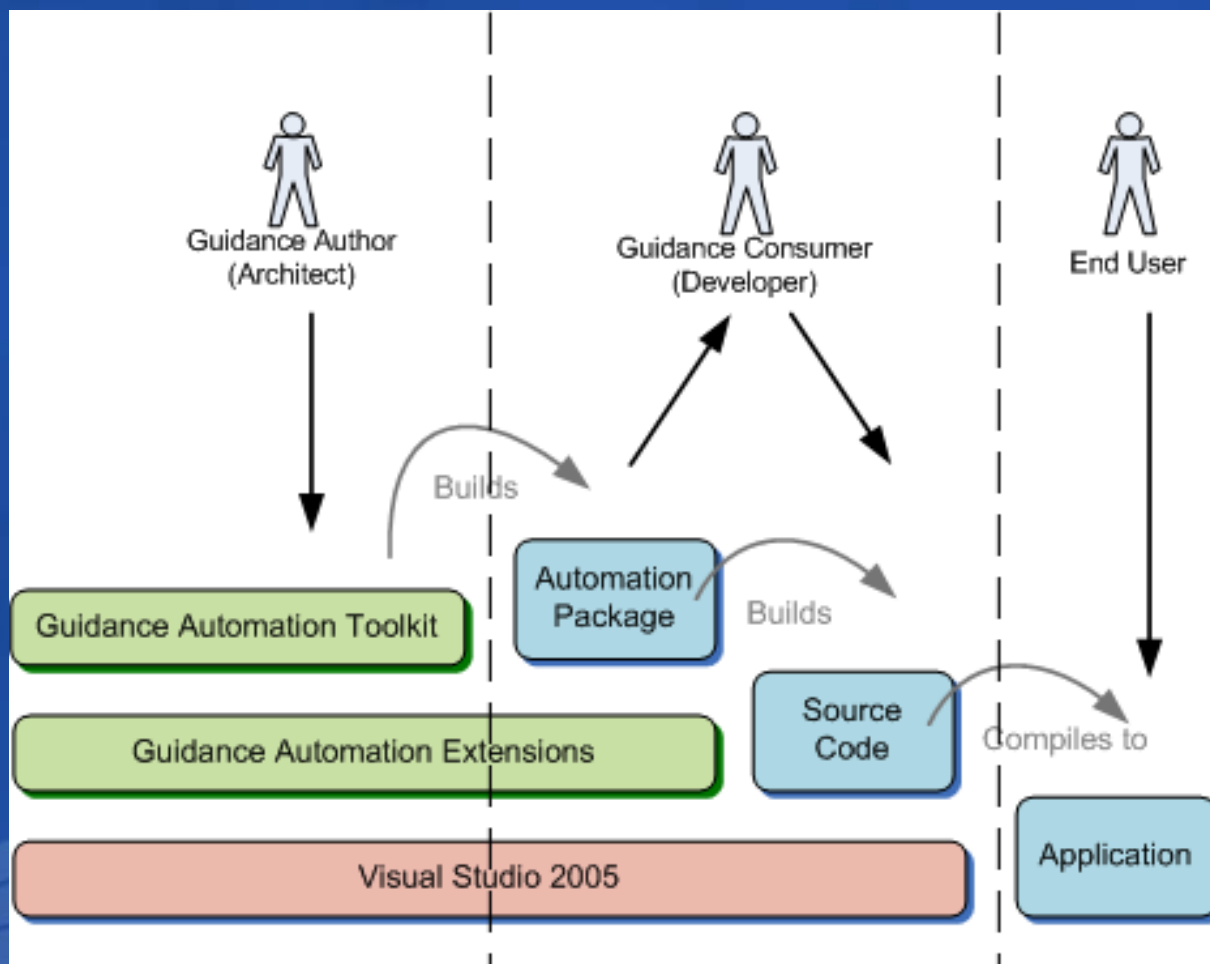
定制

视情况修改和充实三个主要的定制点：
CWAB, ObjectBuilder, PFAB

定制注意事项：

- 1.充分理解前面所讲架构设计点，理解各**AB**的职责
- 2.使用强命名
- 3.选择重新编译解决方案而不是编译
- 4.自定义**AB**后一般需要同时修改**AB**的命名空间，避免冲突，同时定制相关文档

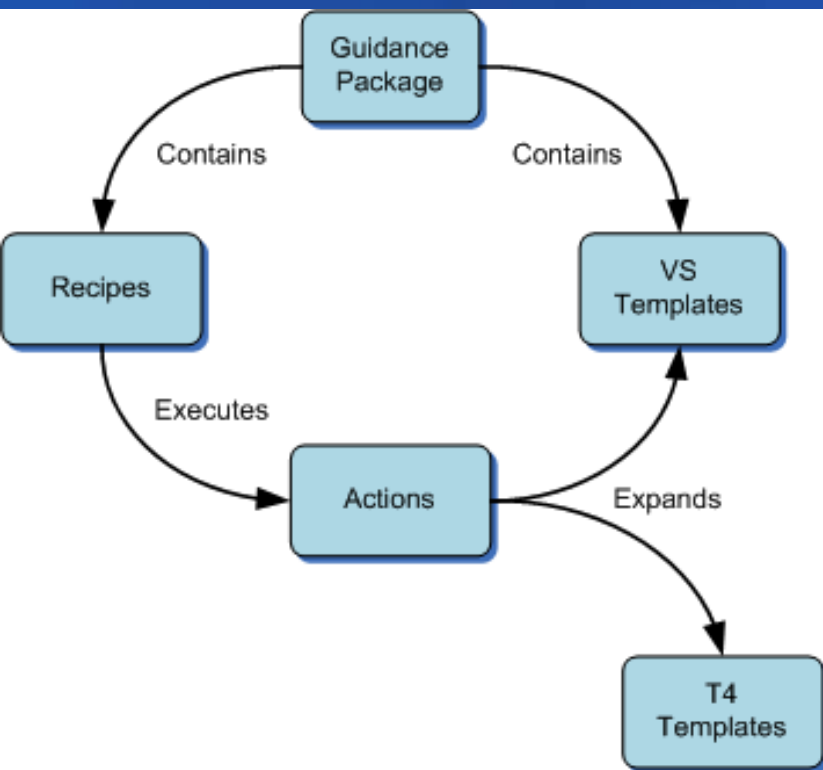
自动化



自动化

您的潜力。我们的动力

Microsoft
微软(中国)有限公司



注意事项:

- 1.使用Register Guidance Package Recipe, 在另一个VS进程中测试结果
- 2.如果没有修改HostData可以用Quick Register Recipe来节省时间
- 3.在配置文件中自定义Guidance Package Identifiers为宜
- 4.使用VS的导入导出设置
- 5.详细步骤见WCSF帮助文件

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

经验分享： 如何理解争议点、规避风险

您的潜力, 我们的动力

Microsoft[®]
微软(中国)有限公司

值吗？

IoC与Service Locator

- 经常可实现相同目的
- 开发复杂度不同
- 人员要求不同
- 区别主要在关系转移

软件工厂的其他考虑

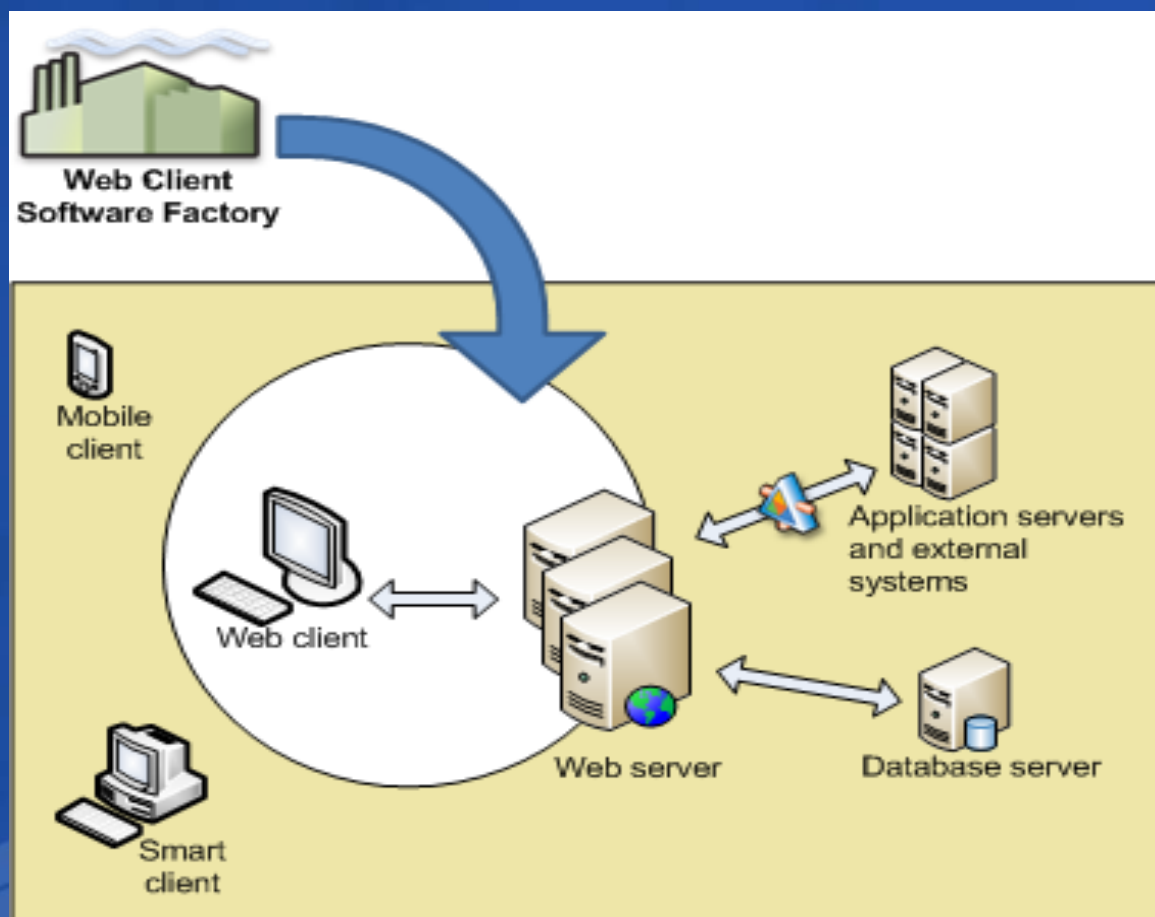
➤与MDA的关系:

MSDN:MDA vs. Software Factories

➤元数据驱动带来的问题

➤与AJAX,ASP.NET MVC,LINQ的集成

多层问题



您的潜力, 我们的动力

Microsoft®
微软(中国)有限公司

参考资料

获取更多WCSF资源

- **MSDN中文网站**

<http://www.microsoft.com/china/msdn>

相关博客

Glenn Block, Matias Woloski

Blaine Wastell, Eugenio Pace




Michael Puleio, Ray Zhang

- **Web Client Software Factory**


<http://www.codeplex.com/websf>

问答交流

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)**  

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问 

提问(A)

删除(D)

问题管理器(Q)

您的潜力, 我们的动力

Microsoft[®]
微软(中国)有限公司

谢谢大家!

备用-Composite Container原理

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

