Architecting Enterprise Application on .NET

在.NET上构架企业级应用程序

Peng Huang

peng.huang@acm.org

课程设计

- **Session 1:**
  - 软件构架的基础知识
- **Session 2:**
  - **.NET**软件构架的基础知识
- **Session 3:**
  - 企业级应用程序的构架
- **Session 4:**
  - 案例学习: **PetShop .NET**
- **Session 5:**
  - **Teamwork**与**Bug Tracking** 概念工具与实践（可选）

Session 1
软件构架的基础知识

- 系统构架师
- 软件构架概述
- 构架框架

系统构架师

- “理想的建筑师应该既是文学家又是数字家，他还应通晓历史，热衷于哲学研究，精通音乐，懂得医药知识，具有法学造诣，深谙天文学及天文计算。”

  --Vitruvius（古罗马建筑师），约公元前 25 年

所以开发良好的企业应用程序的
第一步就是寻找优秀的

系统构架师

软件构架概述

- 构架与设计的关系：
  - 构架属于设计的一方面，它集中于某些具体的特征
- 构架的定义：
  - **IEEE Working Group on Architecture** 把其定义为“系统在其环境中的最高层概念”
  - **David Garlan** 和 **Mary Shaw** 认为软件构架是有关如下问题的设计层次：

    “在计算的算法和数据结构之外，设计并确定系统整体结构成为了新的问题。

    结构问题包括总体组织结构和全局控制结构；通信、同步和数据访问的协议；设计元素的功能分配；物理分布；设计元素的组成；定标与性能；备选设计的选择。”

软件构架概述

- 构架重点
  - 构架只同以下几个具体方面相关：
    - 模型的结构，即组织模式，例如分层。
    - 基本元素，即关键用例、主类、常用机制等，它们与模型中的各元素相对。
    - 几个关键场景，它们表示了整个系统的主要控制流程。
    - 可选特征、产品线状况的服务。

什么是构架框架?

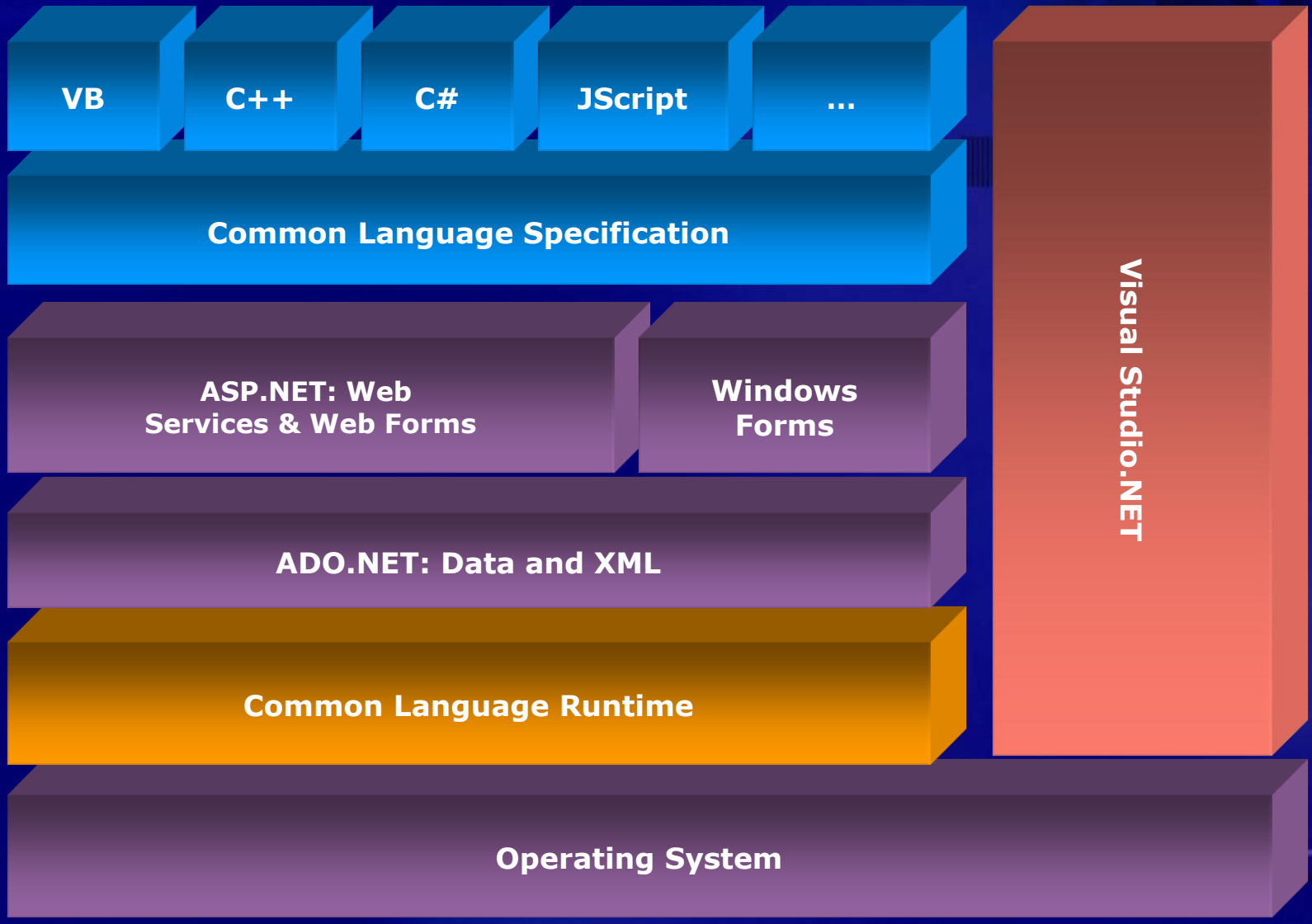
- 构架框架或构架基础设施（中间件）
  - 可以在其上构建某种构架的构件集。许多主要的构架困难应在框架或基础设施中进行解决，而且通常针对于特定的领域：命令和控制、MIS、控制系统等等。
- 目前比较优秀，总拥有成本低的企业应用构架框架

Microsoft .NET Framework

Session 2
.NET软件构架的基础知识

- .NET Framework
- .NET Framework 类库
- VS.NET中的企业应用程序模板
- 演示
  - Duwamish 7 C#中的层次结构

.NET Framework与VS.NET



VS 6与VS.NET

Visual Studio 6.0	Visual Studio .NET
ASP	ASP .NET
HTML generated using XML and XSL	HTML generated using ASP .NET Web controls and ASP .NET Data Binding
Small amount of VB Script	VB .NET code behind
Proprietary cache	ASP .NET cache
Cookie / session / db state	ASP .NET session state
COM+ components	.Net Assemblies
ADO disconnected recordsets	ADO.NET DataSets
SQL Server with stored procedures	SQL Server with stored procedures

J2EE and .NET Framework

- **Java：**
  - 一种语言
  - 发展为一种平台
  - **Packages**
  - **J2EE**
- **Windows：**
  - 一个平台
  - 支持多种语言
  - **Frameworks**
  - **.NET Framework**

.NET平台

海容百川

有容乃大

.NET Framework支持Java语言:
Visual J# .NET

.NET Framework基本概念

- 一个基于Internet高度分布式计算环境的以简化应用程序开发为目的的全新计算平台
  - Common Language Runtime (CLR)
    - 一个在运行时管理代码的代理，提供核心服务，如：内存管理、线程管理、 remoting，强制保证代码的安全和正确。
  - .NET Framework 类库
    - 一个全面的、面对对象的可重用类集合，可以用于开发包括传统的命令行、GUI应用程序，还可以开发基于ASP.NET和。
- code management
  - managed code
  - unmanaged code

.NET Framework类库

Component model	Configuration	Data	Framework services
Globalization and localization	Net	Common tasks	Reflection
Rich, client-side GUI	Runtime infrastructure services	Web Services	

The .NET Framework Class Library

分类	Namespace	功能
Common tasks	System.Collections	集合对象，包括队列、数组、哈希表、链表等。
	System.IO	简单数据流访问与管理，包括文件I/O、内存I/O等。
	System.Text	字符编码、转换和字符串处理。
	System.Text.RegularExpressions	全面支持正规表达式。
	System.Threading	多线程支持，包括锁定和同步。
Rich, client-side GUI	System.Drawing	丰富的2-D功能和GDI+支持。
	System.Windows.Forms	Windows传统应用程序的丰富界面特性支持 。

.NET Framework类库

分类	Namespace	功能
Web Services	System.Web	支持Web服务器和 client管理、通信与设计。提供ASP.NET的核心支持，包括Web Forms。
	System.Web.Services	基于SOAP的Web Service的客户与服务器端支持。
.NET Framework security	System.Security	访问.NET Framework安全系统的基本机制。
	System.Security.Cryptography	编码及解码服务，包括数据的编码、解码、随机数生成、消息认证、数字化签名的支持。
Data	System.Data	访问、管理数据和数据源。
	System.Xml	处理XML支持。
	System.Xml.Serialization	对象到XML的双向映射。

.NET Framework类库

分类	Namespace	功能
Framework services	**System.Diagnostics**	**跟踪调试代码支持，Debug and Trace**
	System.DirectoryServices	**访问活动目录。**
	System.Management	**服务与应用程序管理工具**
	System.Messaging	**微软消息队列 (MSMQ) 的访问与管理，消息的接收与发送。**
	System.ServiceProcess	**安装、执行基于Windows的服务程序，不能访问特定服务，诸如Active Directory、Web Services。**
	System.Timers	**定时器、其他更复杂的应用程序时间调度**

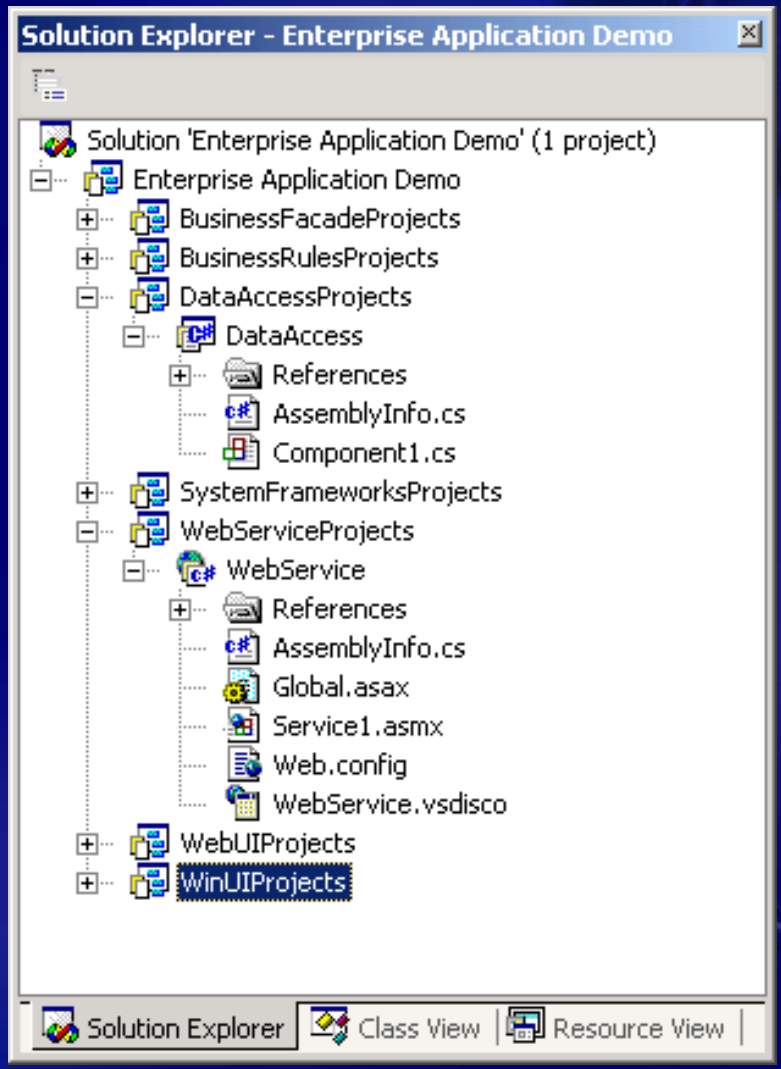
.NET Framework的优势

- 提高生产力
- 整合性
- 完全面向对象
  - 易于重用
  - 提供构架企业应用程序的全面支持
- 部署
  - 操作系统独立性
  - 平台独立性
- 高性能
- 总拥有成本低

VS.NET中的企业应用程序模板

提供多层结构模板

- Business Façade
- Business Rules
- Data Access
- System Framework
- Web Service
- Web UI
- Win UI



演示

- Visual Studio .NET中的企业应用程序模板
  - Duwamish 7 C#中的层次结构

Session 3
企业级应用程序的构架

- 如何编写好的需求
- **Design Goals**
- **.NET Framework对设计目标的支持**
- 企业级应用程序的部署

msdn

如何编写好的需求

- 功能需求必须是可测试的
- 功能需求必须避免实现细节
- 好的功能需求
  - **Administrator must be able to create, read, update, and delete (CRUD) customer accounts.**
    - **Administrator must be able to configure the max outstanding balance for a customer**
    - **Administrator must be able to change the name of a customer**
- 不好的功能需求
  - 该站点必须易于使用（功能需求不可测试）
  - 该站点必须用**VB.NET**语言编写（涉及实现细节）

如何编写好的需求

- An Internet customer will be able to browse the PetShop .NET pets catalog by category.
- An Internet customer will be able to search for specific pets by keyword:
- An Internet customer will be able to select one or more pets and place them in a shopping cart for purchase.
- …

.NET technology and Design Goals

- 可用性（Availability Goal）
- 可维护性（Maintainability Design Goal）
- 可管理性（Manageability Design Goals）
- 高性能（Performance Design Goal）
- 可靠性（Reliability Design Goal）
- 可缩放性（Scalability Design Goal）
- 安全性（Security Design Goal）

Maintainability Design Goal

- 可维护性
  - 代码自然的映射到设计文档
  - 代码合理划分，易于多个开发组维护
- .NET Framework的支持
  - Use Cases implemented directly in Business Façade component
  - Code segmented into many Visual Studio projects that can be modified together, or independently

Availability Goal

- 可用性
  - 100% 正常运行
- .NET Framework的支持
  - Web Farm
  - ASP.NET Availability Enhancements
    - Session State Stored Externally in Session Server
    - Automatic Problem Detection and Web Server Restart Without Interruption of Service
    - Replace DLLs Used By Site, Without Interruption of Service

Manageability Design Goals

- 可管理性
  - 变更系统配不需要重启动系统
  - 系统跟踪和系统日志功能，系统性能监视
- .NET Framework的支持
  - Store configuration info in Web.config
  - Trace and log to the event log using the CLR EventLog class
  - Trace to a text file using the CLR FileInfo and StreamWriter classes
  - ASP.NET provides performance counters for each web application
    - Requests and response statistics
    - Cache statistics
    - Error statistics
    - Transaction statistics

Performance Design Goal

- 高性能
  - 超越J2EE上的PetShop
- .NET Framework的支持
  - ASP.NET compiles pages into executables
  - Utilize ASP.NET page output caching
  - Full Web Farm support through external session state, and stateless classes

Reliability Design Goal

- 可靠性
  - “温柔”的处理错误
    - 系统内部
  - “友好”的处理错误
    - 系统外部
- .NET Framework的支持
  - Exception based error handling
  - ASP.NET custom error page support

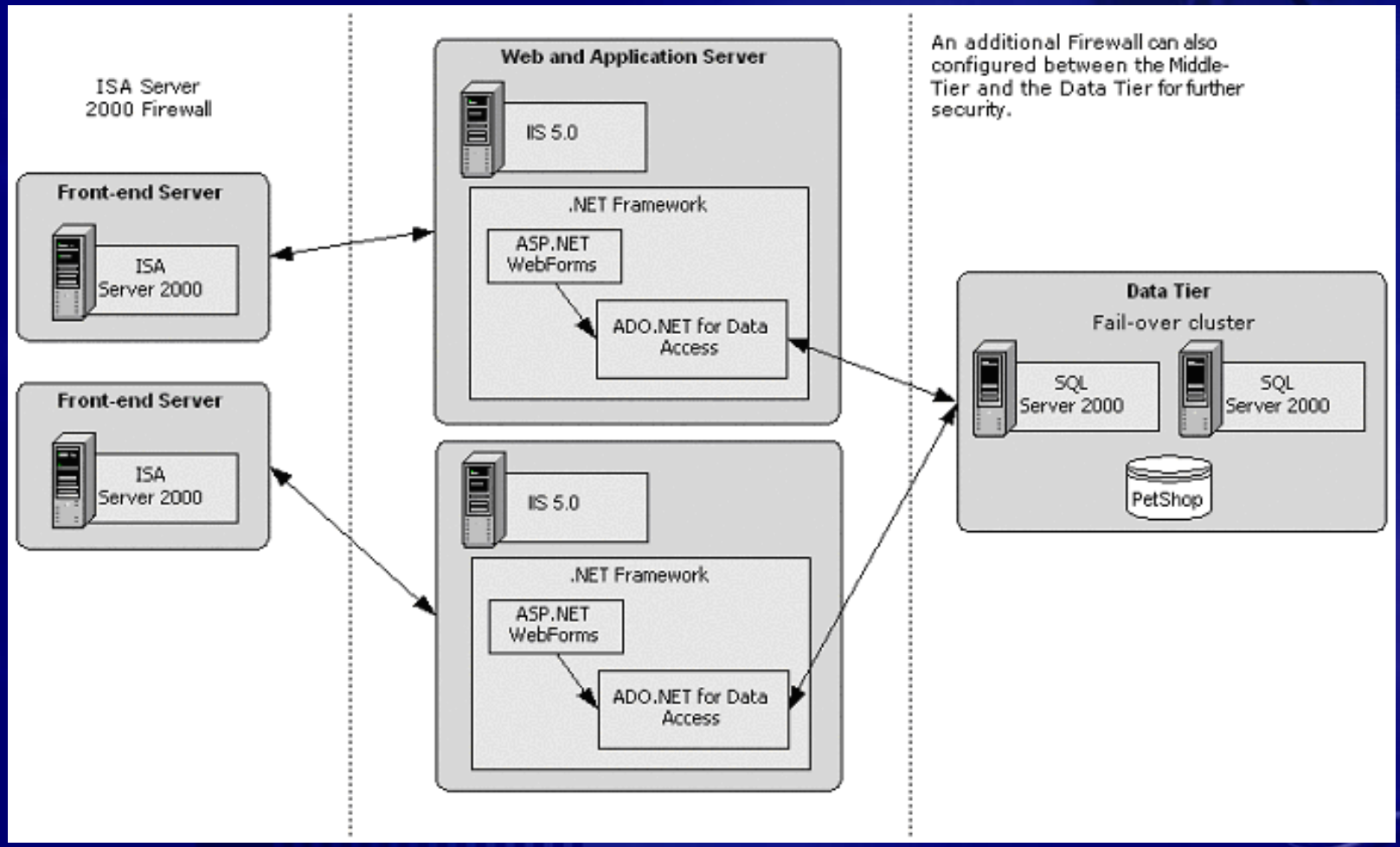
Scalability Design Goal

- 缩放性
  - Must scale up and out
- .NET Framework的支持
  - Scale up through ASP.NET Web Garden support
  - Scale out through external session state.
  - Access components locally or through .NET remoting.

Security Design Goal

- 安全性
  - Require authentication to prevent URL spoofing
- .NET Framework的支持
  - Utilize ASP.NET built-in form authentication

企业级应用程序的简单部署



Session 4
案例研究: PetShop .NET

- PetShop .NET概述
  - PetShop使用演示
- PetShop .NET与PetShop J2EE
- PetShop .NET系统模型
- PetShop .NET中的Web Service
- 演示:
  - Visio与PetShop .NET软件构架
  - PetShop .NET中的Web Service

Overview of PetShop .NET

- PetShop .NET是一个简单的宠物销售网站系统
- 主要业务：在线宠物销售
- B2C的商业模式
- 基本功能:
  - 会员管理、帐号管理、购物车、搜索、结帐
- 构架于Microsoft .NET平台之上

Overview of PetShop .NET

J2EE体系结构的经典之作

看看它在.NET上是什么样

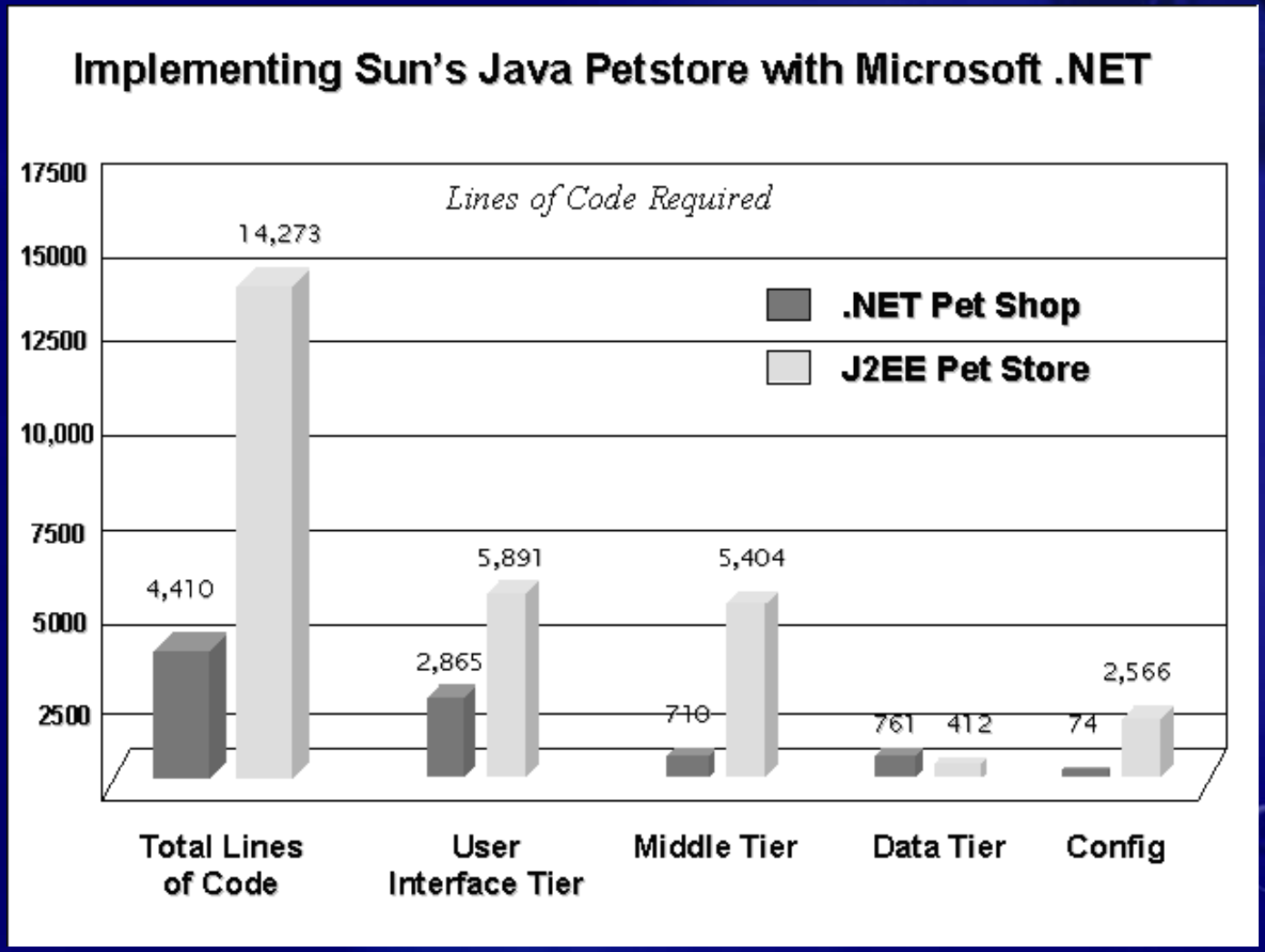
Overview of PetShop .NET

- 演示PetShop .NET Solution

PetShop .NET与PetShop J2EE

- 同样的应用在.NET上重写后表现如何？
  - 代码编写量比较
  - 性能与可缩放性比较
  - 占用CPU百分比比较
- 比较的本质：
  - .NET Framework与J2EE两个软件基础框架在构架企业应用程序上的优劣

代码编写量比较

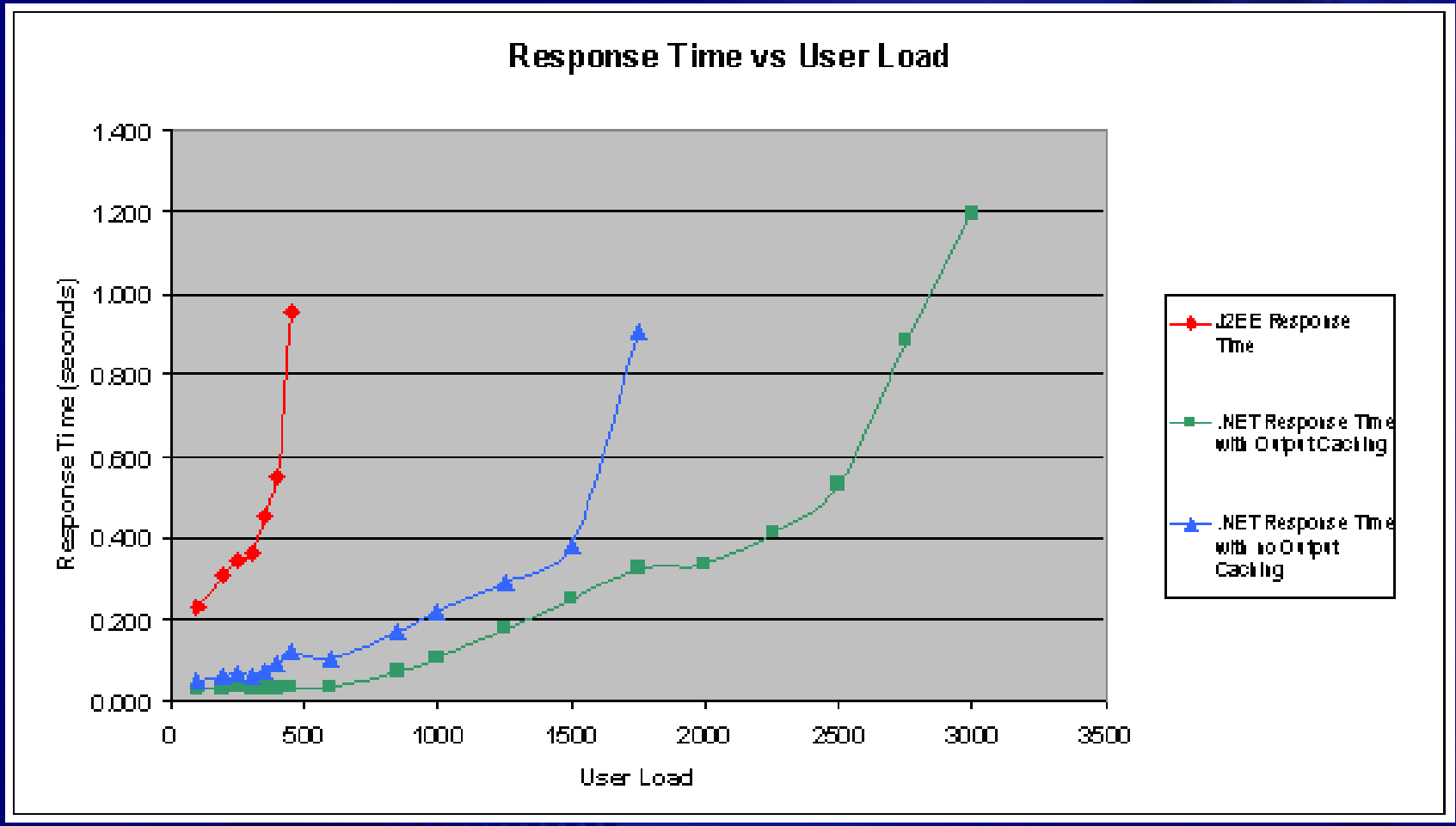

Implementing Sun's Java Petstore with Microsoft .NET
Lines of Code Required
17500
15000
12500
10,000
7500
5000
2500
14,273
4,410
5,891
2,865
5,404
710
761
412
74
2,566
.NET Pet Shop
J2EE Pet Store
Total Lines of Code
User Interface Tier
Middle Tier
Data Tier
Config

代码编写量比较

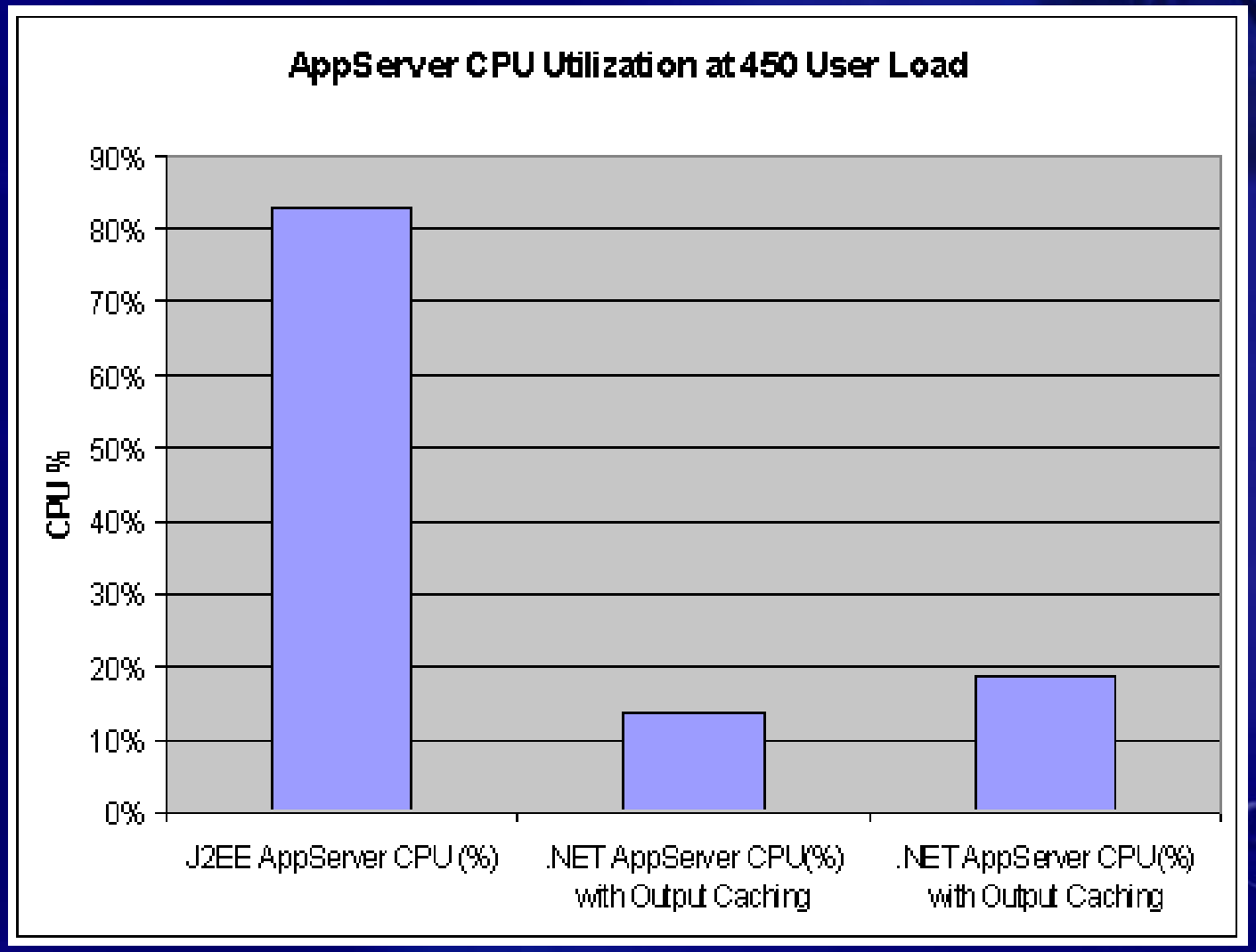
简单就是美

- 实现同样的功能需求：三分之一的代码编写量
  - .NET 4410行，J2EE 14273行
- 配置减少：
  - .NET 74行，J2EE 2566行
- 中间层减少：
  - .NET 710行， J2EE 5404行
- 代码减少意味着什么：
  - 拥有成本低
  - 更易于维护

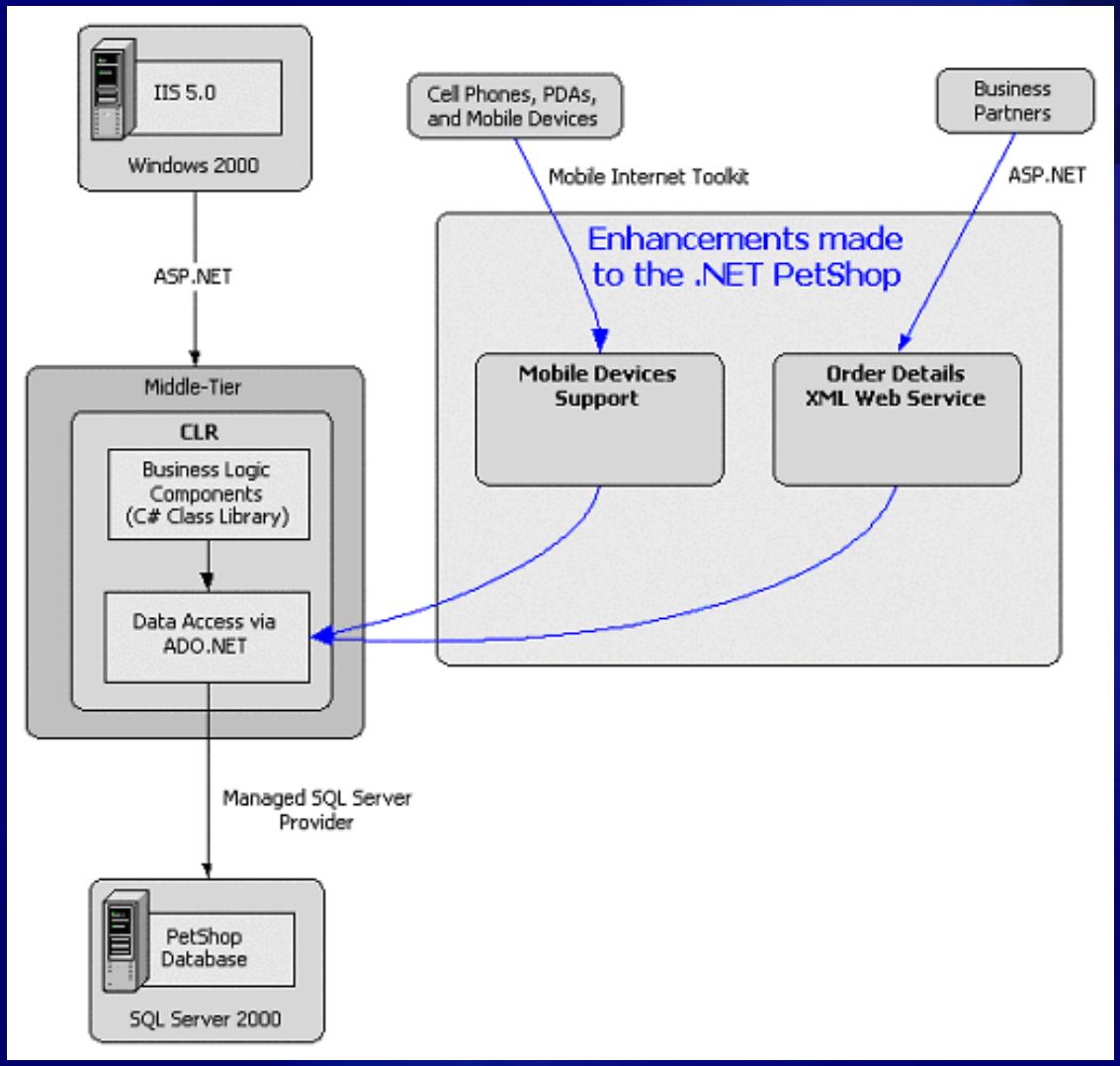
性能与可缩放性比较


Response Time vs User Load
Response Time (seconds)
1.400
1.200
1.000
0.800
0.600
0.400
0.200
0.000
0
500
1000
1500
2000
2500
3000
3500
User Load
J2EE Response Time
.NET Response Time with OutputCaching
.NET Response Time with no Output Caching

占用CPU百分比比较

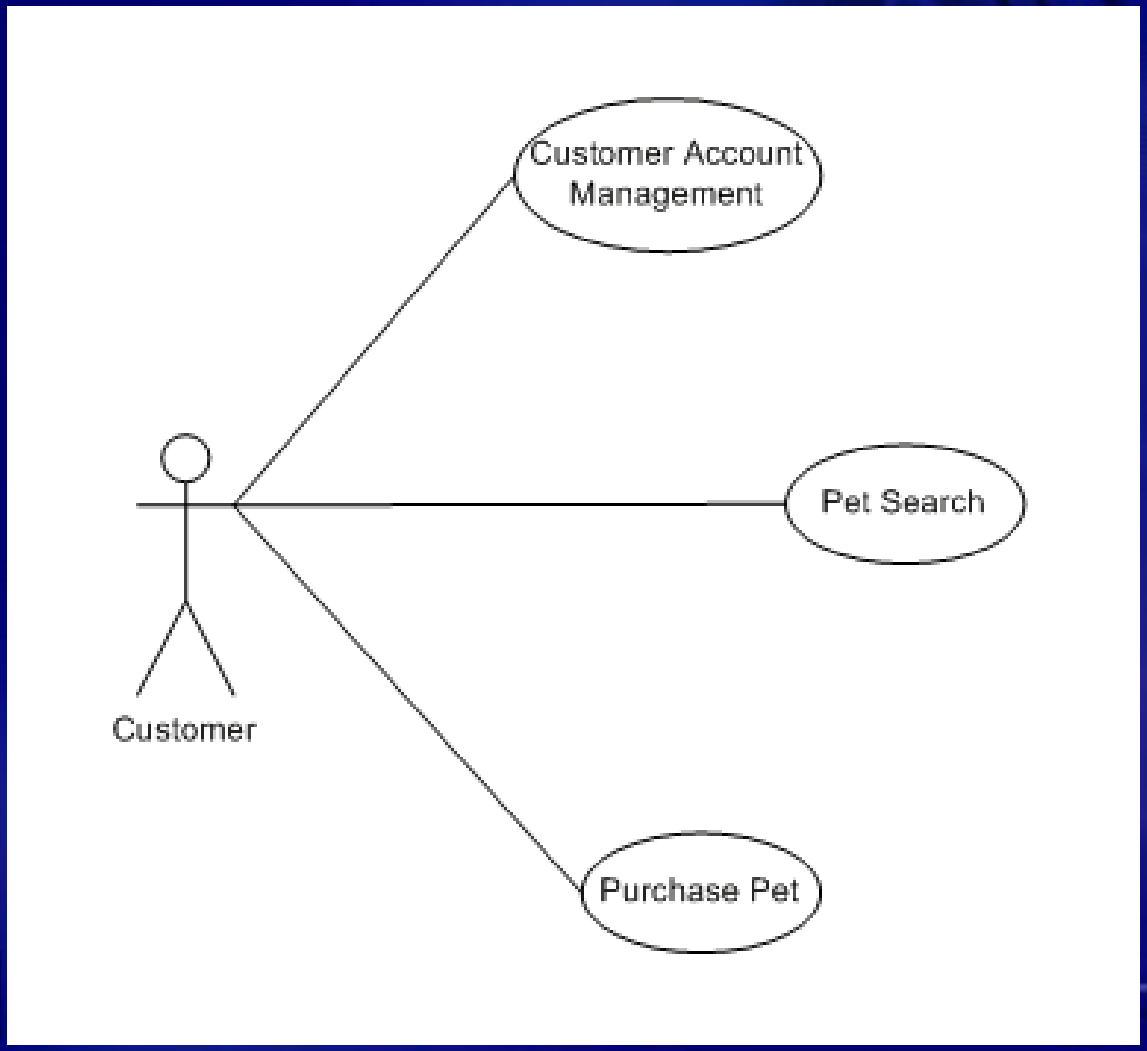

AppServer CPU Utilization at 450 User Load
CPU %
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
J2EE AppServer CPU (%)
.NET AppServer CPU(%) with Output Caching
.NET AppServer CPU(%) with Output Caching

Logical Layer Architecture



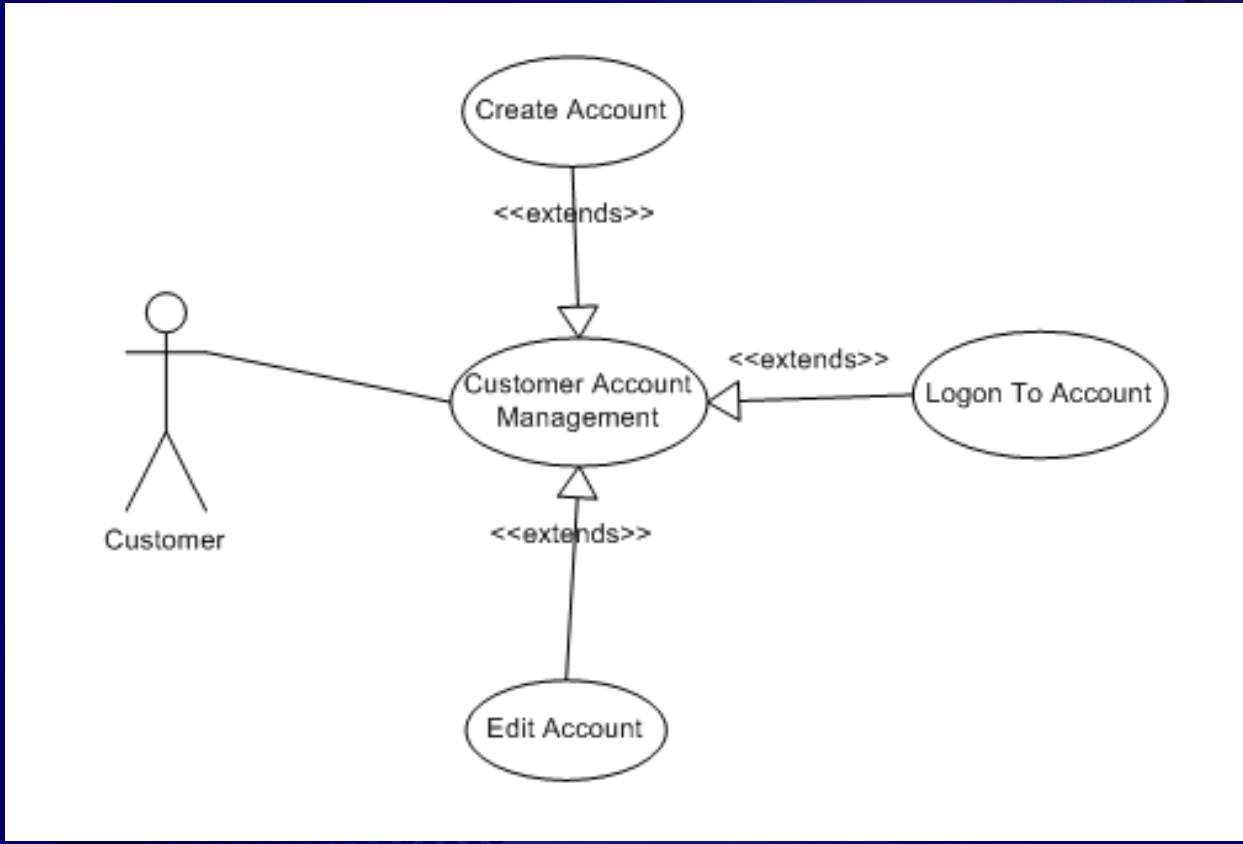
关键的Use Case Diagram

- **Use Case Diagram** 从用户通过系统能做什么角度描述系统
- **Software Model**



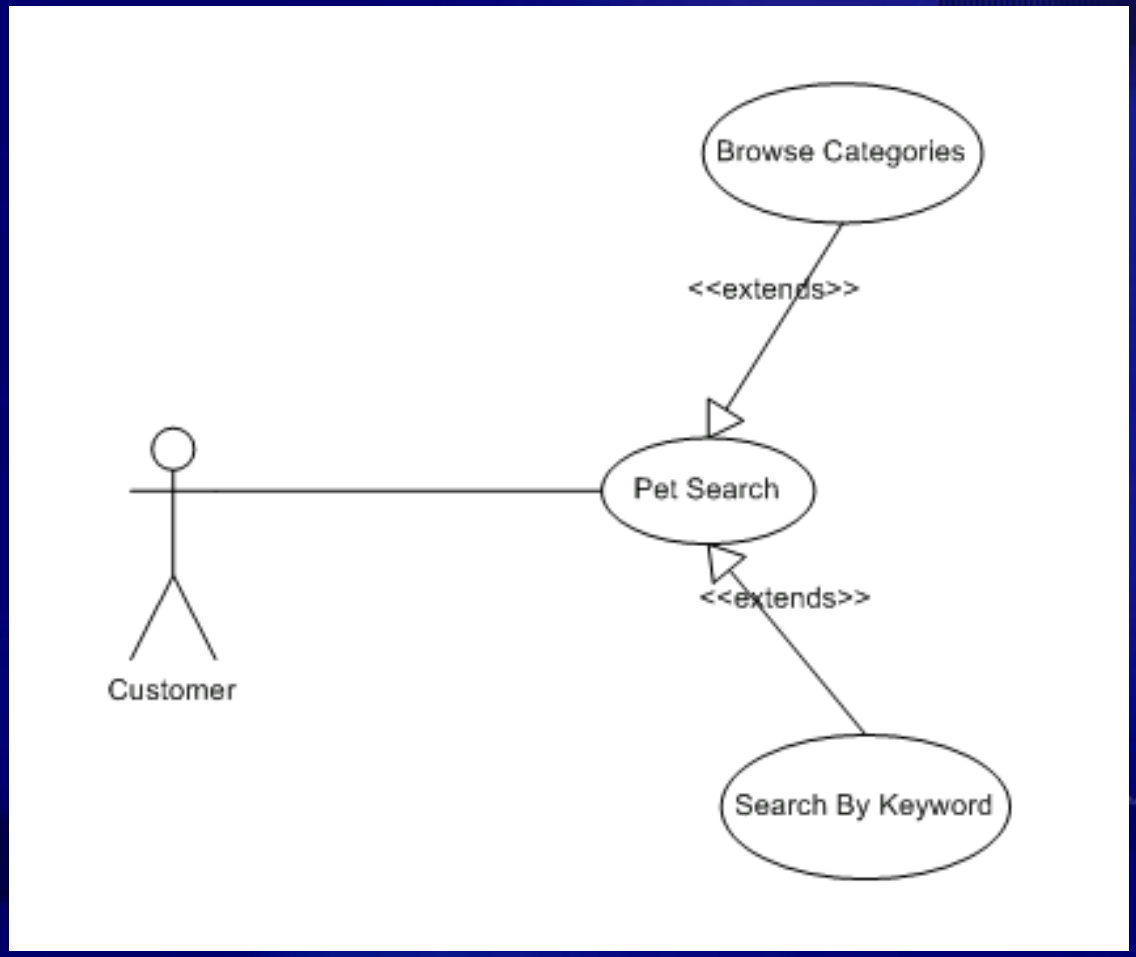
关键的Use Case Diagram

- Customer Account Management



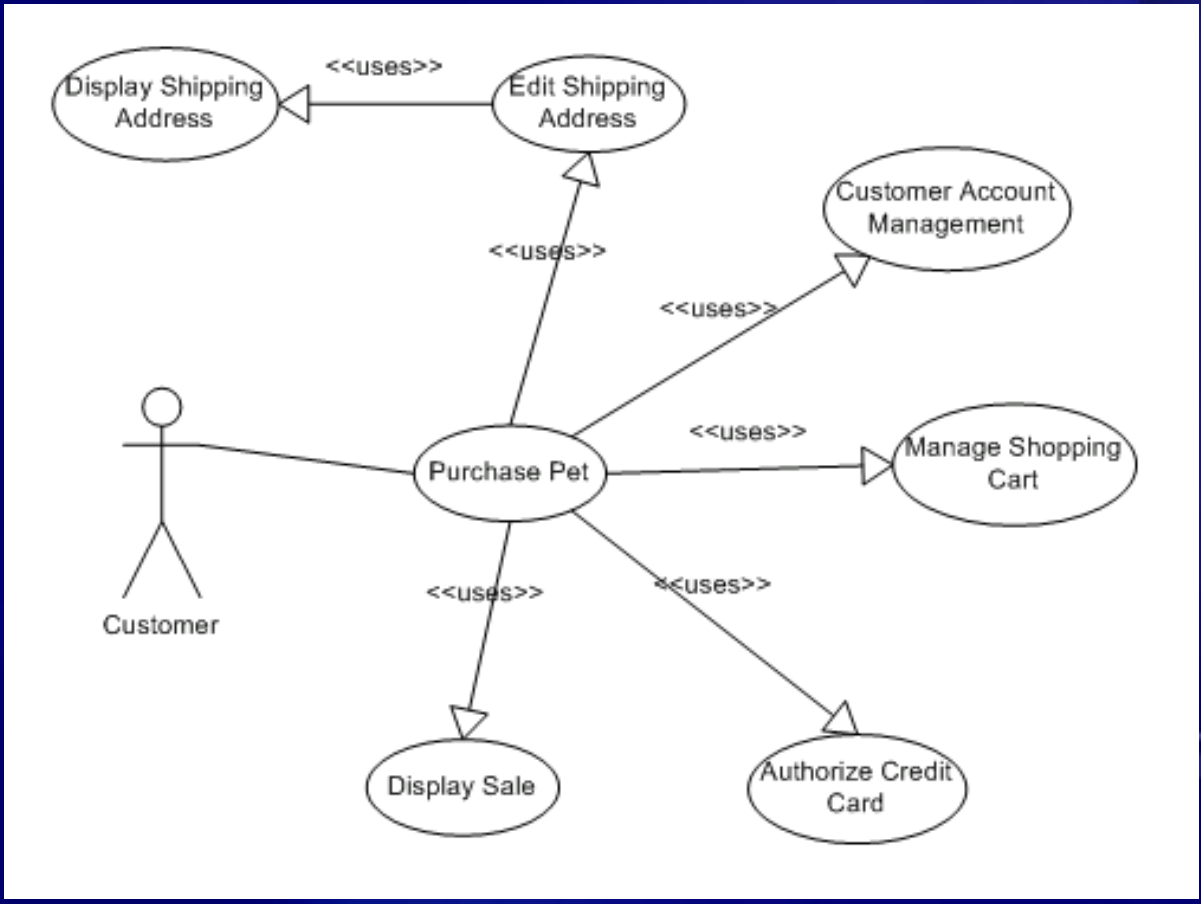
关键的Use Case Diagram

- Pet Search



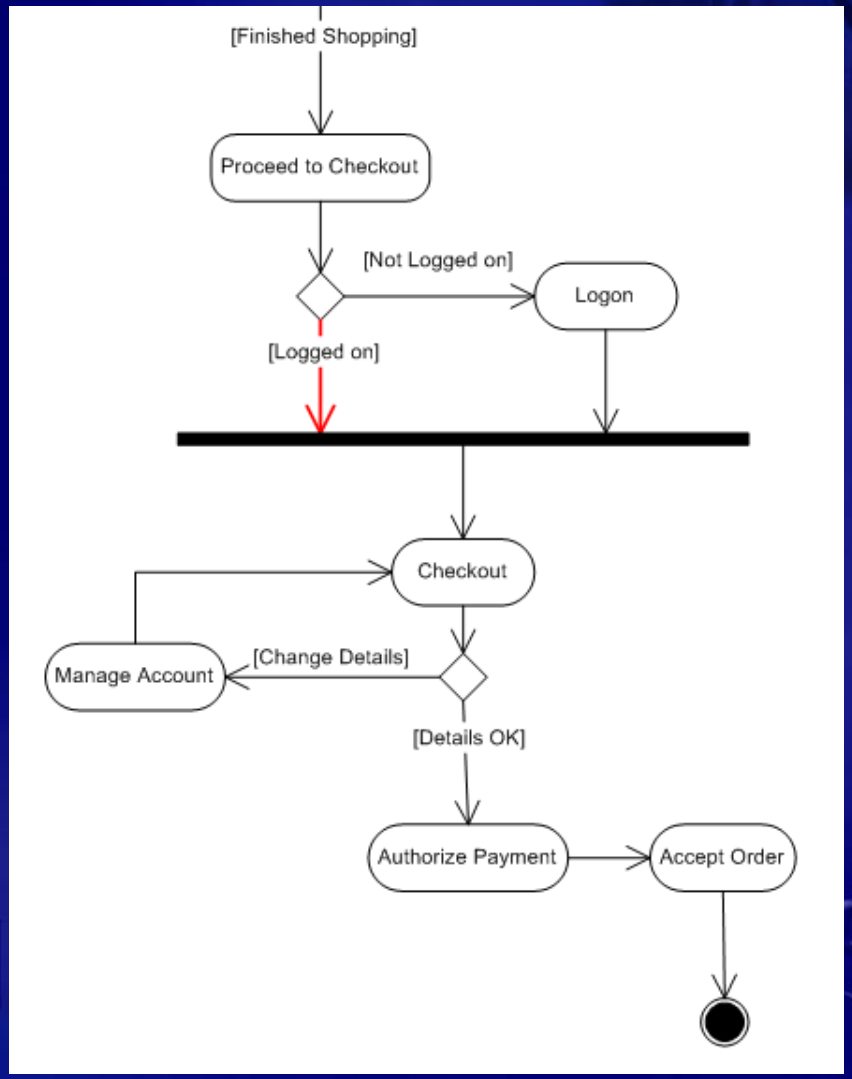
关键的Use Case Diagram

- **Purchase Pet**



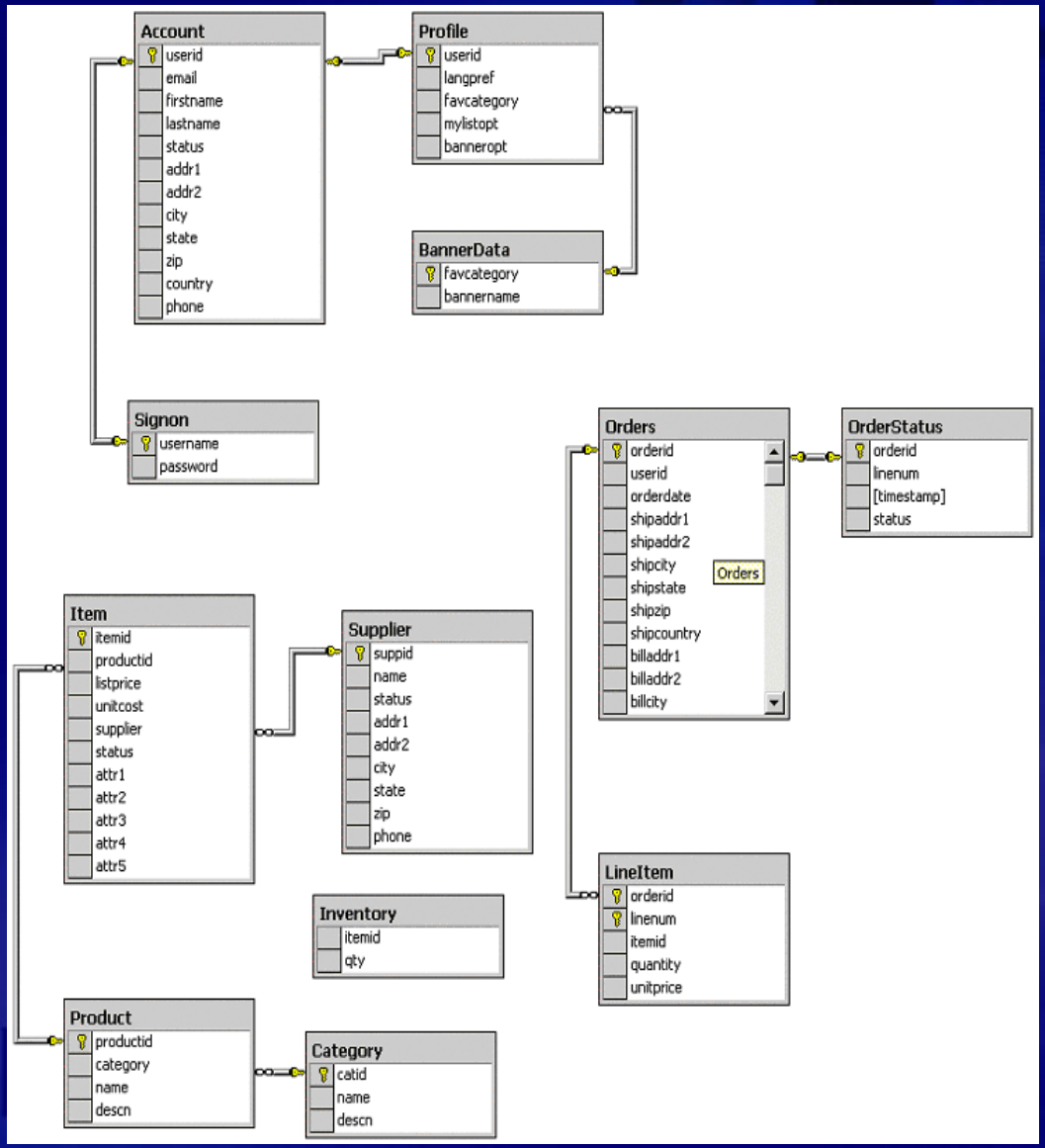
Activity Diagram

- **Activity Diagram** 描述用户如何做某事



数据模型

- 在设计中使用Visio对数据库建模
- 利用Visio分析原有数据库
- ER vs. ORM



Web Service

Web Service Definition

Web services are loosely coupled software components delivered over Internet standard technologies.

Daryl Plummer, Gartner

Web Service Characteristics

- Programmatic interface
- Using standard web protocols
- Loosely-coupled connections

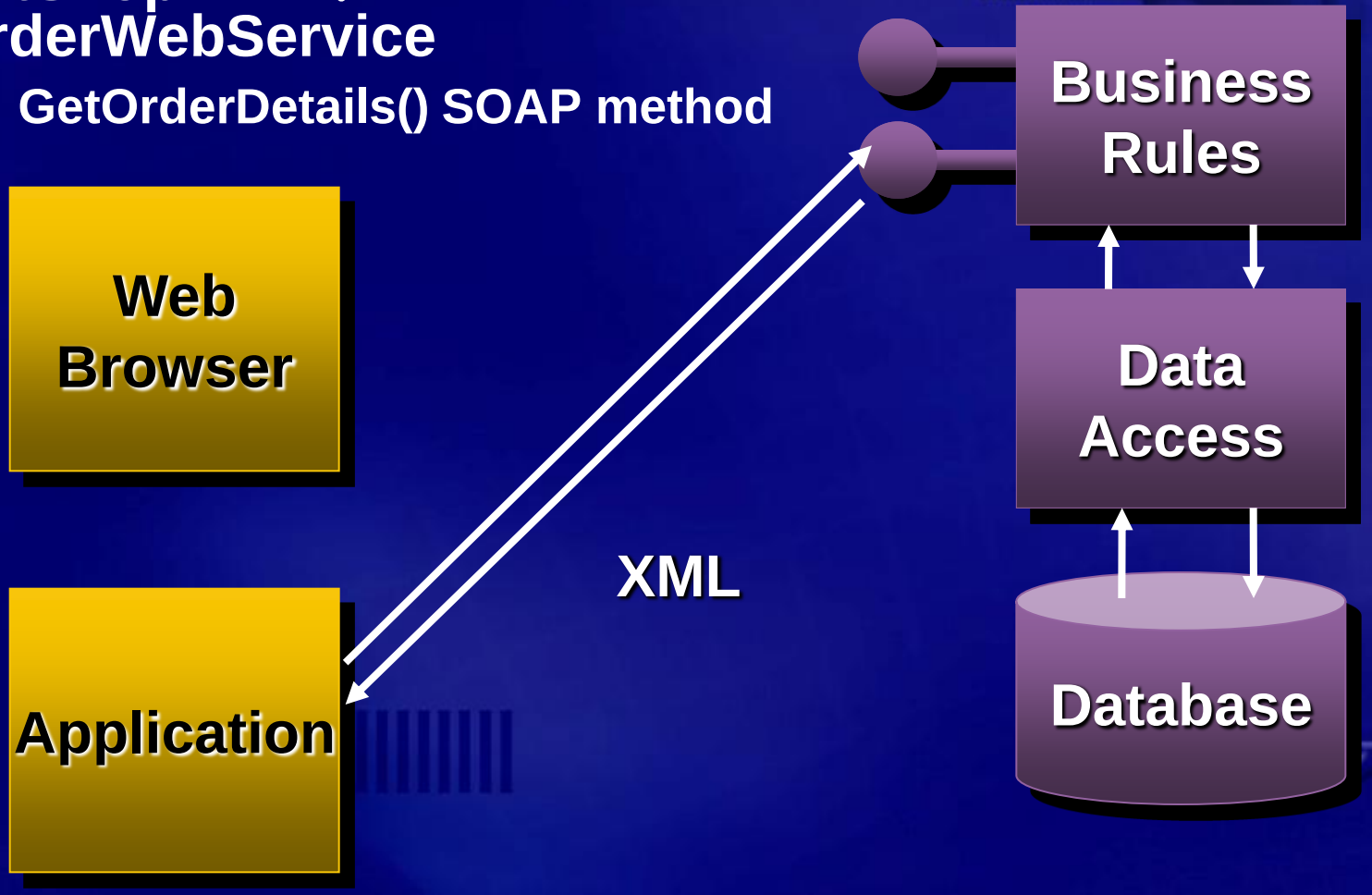
Web Service

- Where is the web service ?
  - NG Web Application
  - Web-accessible XML-based services
  - Internet, Intranet, Extranet
  - B2B, B2C, P2P
- Application Service
  - Schedule, Email, stock quotes
- System Service
  - Directory, messaging, database, security

Hailstorm → .NET My Services

Web Service

- B2B 和 EAI的最佳选择
- PetShop .NET：OrderWebService
  - GetOrderDetails() SOAP method



Demo

- Visio与PetShop .NET建模
  - PetShop Solution
  - 软件建模
  - 数据库建模
- PetShop .NET中的Web Service
  - GetOrderDetails() SOAP 方法

Session 5
Teamwork与Bug Tracking工具与实践

- Teamwork的基本知识
- 演示： VSS and VS.NET演示
- Bug Tracking的基本知识
- 演示：Bug Tracking Tool

Teamwork的基本知识

- 目标: On time and on budget
- 企业软件开发的需求需要Teamwork
- 什么样的Teamwork是合理的
  - Teamwork的机制
    - Check in, Check out, Get last version
    - 每日构建如何实现
  - Teamwork工具：Visual Source Safe
- 经理的任务: push and check，not coding

VSS and VS.NET演示

- 在Visual Studio .NET中使用Visual Source Safe 6.0C
  - PetShop .NET项目

Bug Tracking的基本知识

- 目标：交付满足客户需求的高质量软件
- 管理的对象：不仅仅是Bug
  - Bugs and Features
- Bug Tracking的机制
  - 保护程序员
  - 有计划的Debug
- 衡量Bug的两个指标
  - 出现频率和严重程度

Bug Tracking Tool演示

- 小工具: **Bug Tracking System**
  - 一个基于**Access**的中小项目**Bug Tracking**工具

The Road to .NET Architect

- 充分了解.NET Framework功能结构
- 充分了解OOA/D
- 在Framework上为系统建模
- 充分利用Framework的体系结构优势，实现强大的企业级解决方案
- 构架满足客户需求的完美应用软件结构
- 成为MSDN Universal用户

.NET Architect Resource

- MSDN Online:
  - http://msdn.microsoft.com
- Rational:
  - http://www.rational.com
- Cetus Links
  - http://www.cetus-links.org
- MSDN .NET Resource
  - http://msdn.microsoft.com/net

感谢

- MSDN Universal and MSDN Online
  - 此次讲座的所有素材及演示程序均来自：MSDN Universal和MSDN Online
- MSDN Online China
  - http://www.microsoft.com/china/msdn
  - MSDN Online China大量的技术参考文献提供了对.NET Platform更深刻的理解

Q&A