Introduction to Distributed Systems

Scott Seely

http://www.pluralsight.com/

pluralsight
see what you can learn

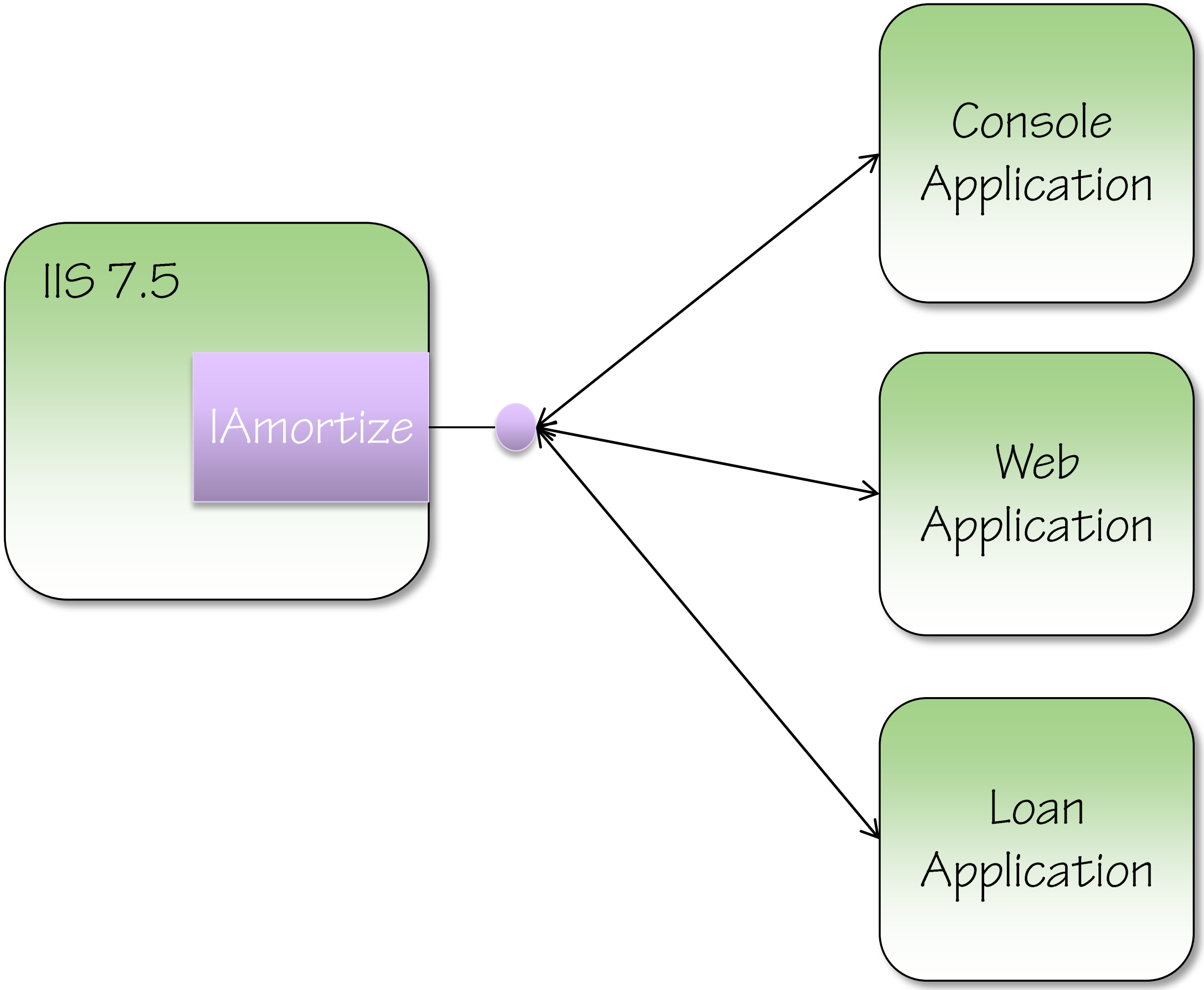
Overview

- What is a distributed system?
- Why do we distribute tasks and responsibilities?
- Look at where the components live.
- An overview of the technologies used by .NET developers (WCF, IIS, MSMQ, WAS, MS-DTC, SQL Server Service Broker, Workflow Foundation, BizTalk, Serialization)

What is a Distributed System?

- Formal definition: requires two or more processes to function at 100%
- Enterprise: multi-tier application. Integration projects
- Partner level: Exposed interfaces
- Web Scale Services: Project Dallas, Facebook Connect

Example Distributed System

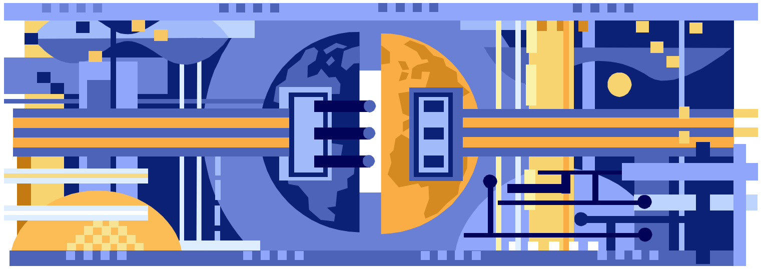


Why do we distribute tasks and responsibilities?

- One source of the ‘truth’
- Enforce uniform rules (documented and undocumented)
- Agility

Components in Distributed Architecture

- Security
- Logging
- Data
- Business Logic
- Connectivity



Where components live

- **Internally**
  - ‘My’ business group
  - ‘Their’ business group
- **Externally**
  - Purchased services
  - Free services
  - Partner services

Security

- Authentication providers maintained centrally.
- Group membership maintained in multiple places.
- Authorization distributed.

Logging

- Debugging distributed applications is hard
- Logs tell you what happened and when

Data

- Ownership
- Synchronization (ideally, one way)

Business Logic

- **Smaller scale items (assembly sharing):**
  - Data element validation
  - Formatting
  - Data relationships
- **Larger scale (distributed app/service sharing):**

Connectivity

- Many consumers
- Provider mechanisms influenced by consumers
- Consumption limited by providers
- Broader mechanisms allow broader use

Technologies Used by Microsoft Developers and Architects

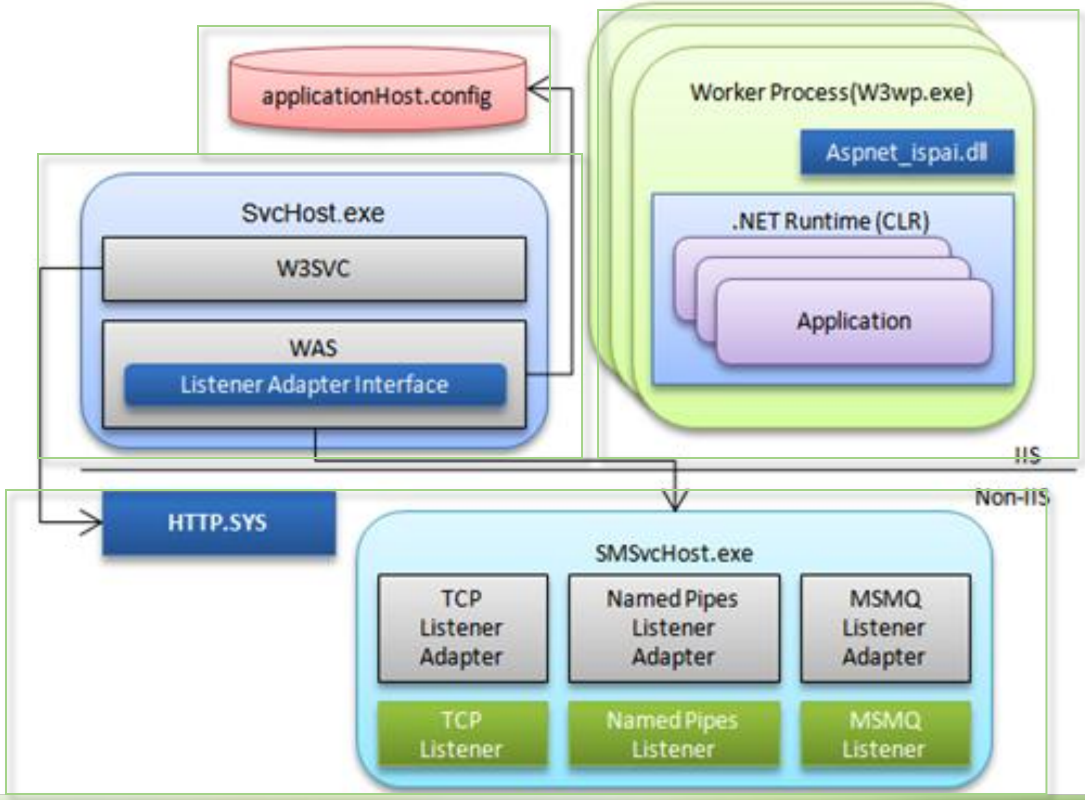
- Internet Information Server (IIS)
- Windows Process Activation Service (WAS)
- Microsoft Message Queue (MSMQ)
- Microsoft Distributed Transaction Coordinator (MS-DTC)
- Windows Communication Foundation (WCF)
- Windows Workflow Foundation (WF)
- SQL Server
- BizTalk

Internet Information Server

- User Interface
- Business Logic
- Version 7.0 makes ASP.NET a peer to ISAPI
- Provides process recycling, rapid fail protection

Windows Process Activation Service (WAS)

- Listen for messages while consuming minimal resources
- Works over TCP, MSMQ, named pipes, HTTP|S.



Microsoft Message Queue (MSMQ)

- One-way messaging
- Durable and volatile queue storage
- Supports sessions, ordering

Microsoft Distributed Transaction Coordinator (MS-DTC)

- **Manages transactions that span processes and machines.**
- **Coordinates commit and rollback of transactions.**
- **Supports ACID properties for database, registry, file system**
    - ACID: Atomic, Consistent, Isolated, Durable
- **Coordinates work with Transaction Managers (database, registry, file system, custom)**

Windows Communication Foundation (WCF)

- General purpose messaging system
- Supports all protocols (either “in the box” or through extensibility mechanisms)
- Supports any serialization (again, “in the box” and through extensibility)

Windows Workflow Foundation (WF)

- XAML based (just like Windows Presentation Foundation)
- Built from Activities and Rules
- Can model logic, loops
- Workflows are mobile.

BizTalk

- An Enterprise Service Bus
- Business Process Automation
- Message Broker

Microsoft® BizTalk® Server 2009

SQL Server

- Manages data
- Handles Security
- Service Broker: async, transactional messaging using queues.
- Reporting Services

Microsoft® SQL Server® 2008

Summary

- Distributed Systems are everywhere
- Built to reduce duplication in data, logic
- Reduce time to market with solutions
- Coordinate with internal and external customers
- .NET developers have a large range of tools to use