
Getting Started with WCF



Contents

1. WCF and SOA essentials
2. WCF architecture
3. Service hosting and communication



1. WCF and SOA essentials

- What is WCF?
- WCF versions
- What is a service?
- SOA (service-oriented architecture)
- Benefits of SOA

What is WCF?

- WCF is Microsoft's unified API for creating service-oriented applications
 - WCF provides a run-time environment for services
 - Enables you to expose and consume CLR types as services
- WCF implements industry standards, to promote interoperability between services
 - Service interactions
 - Type conversions
 - Marshalling
 - Protocol support

WCF Versions

- WCF 3.* (in .NET Framework 3.*)
 - Hosting services
 - Service instance management
 - Asynchronous calls
 - Reliability
 - Transaction management
 - Disconnected queued calls
 - Security
- WCF 4.0 (in .NET Framework 4.0)
 - Simplified configuration
 - Discovery
 - Routing service
 - REST (representational state transfer) improvements
 - Windows Workflow services

WCF Versions (Continued)

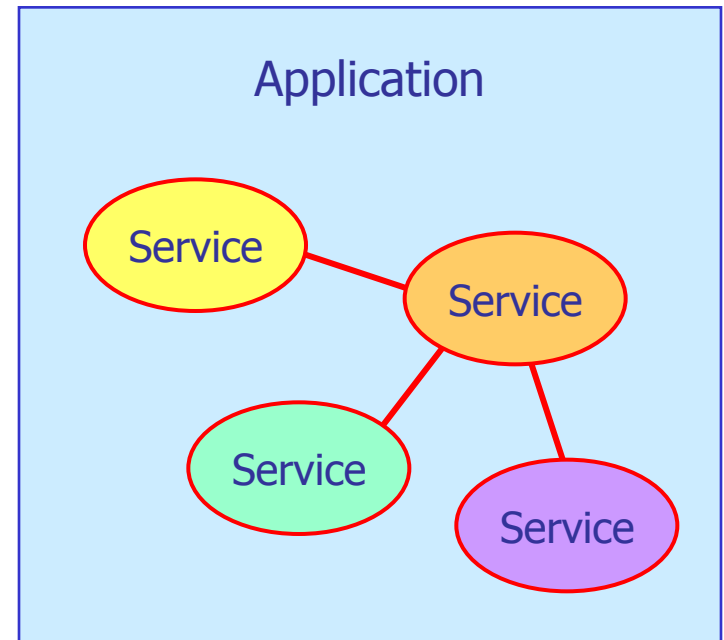
- WCF 4.5 (in .NET Framework 4.5)
 - Task-based asynchronous services
 - Streaming improvements
 - WebSocket support
 - Various configuration simplifications and validation improvements

What is a Service?

- A *service* provides a specific function
 - Typically a business function, such as processing a purchase order
- A service can provide a single discrete function
 - E.g. converting one type of currency into another
- ... or it can perform a set of related business functions
 - E.g. handling the various operations in an airline reservations system

SOA (Service-Oriented Architecture)

- A service-oriented architecture (SOA) is...
 - An information technology approach or strategy in which applications make use of (rely on) services available in a network such as the Web
- In SOA, an application can be...
 - A UI flow that calls services
 - A long-running workflow (e.g. WF) that makes use of services
 - A portal that uses services



Benefits of SOA

- Alignment with business process
 - Define services to implement the business process
 - Services can deliver business benefit quicker
- Reuse and productivity
 - Reuse coarse-grained services in many applications
 - Hopefully achieve developer productivity
- Interoperability and flexibility
 - Services can be interoperable across platform and languages

SOA, Web Services, and WCF

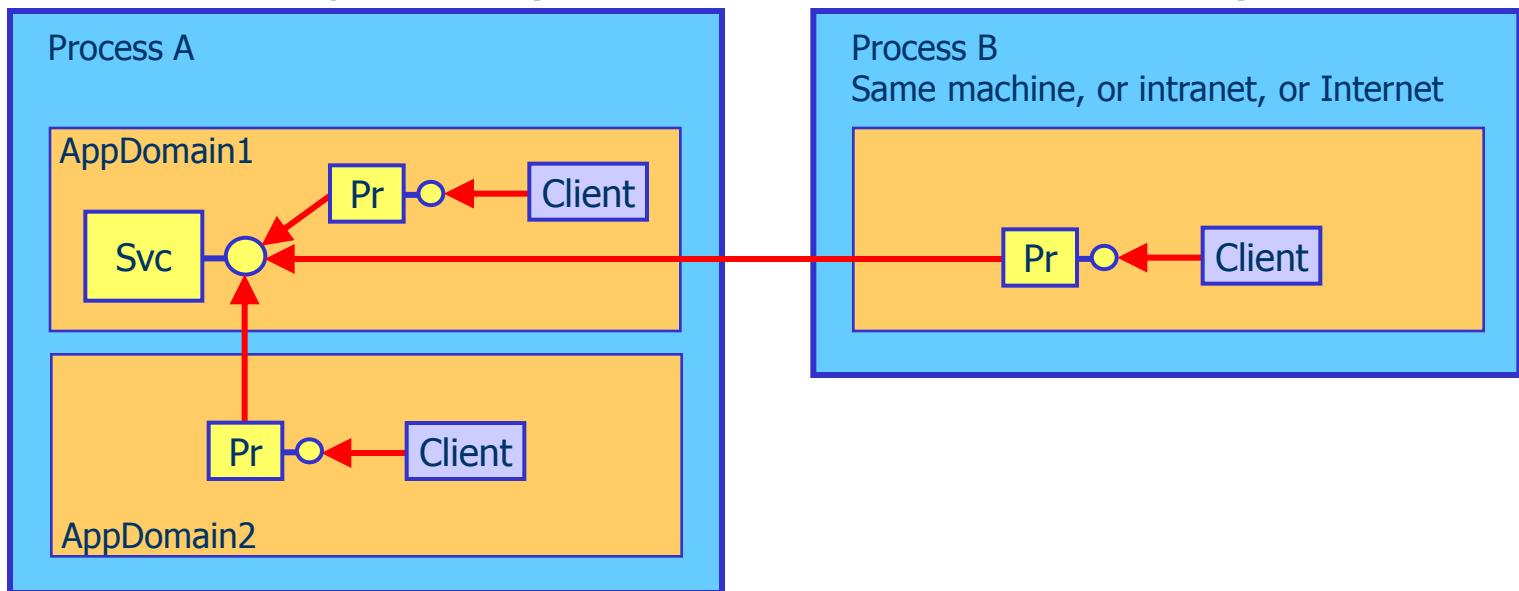
- SOA != Web services
 - SOA is an architectural strategy
 - Web services are an implementation technology, based on HTTP
 - You don't have to use Web services to implement an SOA
- WCF supports SOA
 - Provides tools to enable you to create and consume services
 - Supports many transport protocols (HTTP, TCP, IPC, MSMQ)
 - No need to learn WSDL

2. WCF Architecture

- Service execution boundaries
- Endpoints: A, B, C
- Addresses
- Bindings
- Contracts

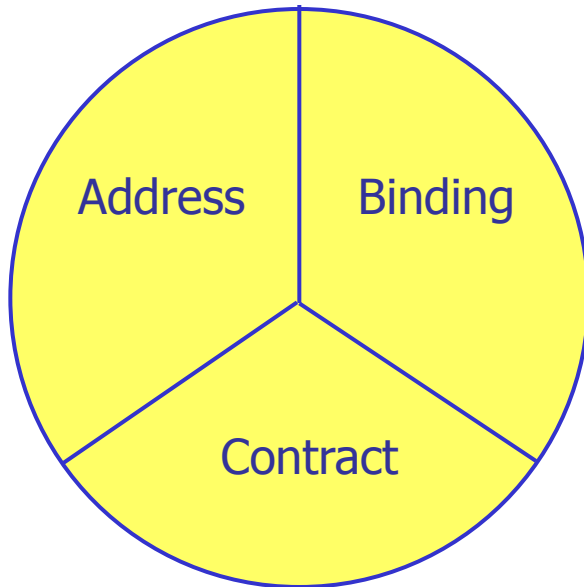
Service Execution Boundaries

- A client never interacts directly with a service
 - The client always uses a proxy to forward calls to the service
 - The proxy exposes the same operations as the service
- WCF allows communication over all execution boundaries:
 - In same app domain
 - In a different app domain
 - In a different process (on same or different machine)



Endpoints: A, B, C

- A locatable service is known as an "endpoint"



Address

Where to find a service (i.e. a URL)

Binding

How to communicate with a service (e.g. HTTP)

Contract

What the service can do (a set of operations)

A is for Address

- Every service is associated with a unique address
 - Address defines location and transport protocol ("schema")

- General format of addresses:

[transport] :// [machine] [:optional port] / [optional URI]


base address

- WCF supports the following transport schemas:

Transport	Example address	Additional notes
HTTP	http://servername:8001/MyService	Default port 80 (443 for HTTPS)
TCP	net.tcp://servername:8002/MyService	Default port 808
IPC	net.pipe://localhost/MyPipe	Must use localhost (or its name)
MSMQ	net.msmq://servername/MyService	Can use private queues too

B is for Binding

- A binding specifies low-level communication details
 - Transport protocol
 - Message encoding
 - Reliability, security, transaction propagation, and interoperability
- WCF has several predefined bindings, for example:

Name	Transport	Encoding	When to use...
BasicHttpBinding	HTTP/HTTPS	Text, MTOM	Expose WS as legacy ASMX
NetTcpBinding	TCP	Binary	WCF client (cross-machine)
NetNamedPipeBinding	IPC	Binary	WCF client (in-process)
NetMsmqBinding	MSMQ	Binary	WCF client (disconnected)
WsHttpBinding	HTTP/HTTPS	Text, MTOM	Non-WCF client
WsDualHttpBinding	HTTP	Text, MTOM	Non-WCF client (bidirectional)

C is for Contract

- A contract defines the callable operations for a service
 - [ServiceContract]
 - Exposes a .NET type (class or interface) as a WCF contract
 - [OperationContract]
 - Exposes a method in a WCF service
 - Note: you can only annotate methods (not properties or indexers!)
 - [DataContract]
 - Designates a type as being usable as a service param/return
- Simple example:

```
[ServiceContract(Namespace="urn:myNamespace")]  
interface ITempConverter  
{  
    [OperationContract]  
    double CtoF(double c);  
  
    [OperationContract]  
    double FtoC(double f);  
}
```


3. Service Hosting and Communication

- Hosting overview
- Hosting options
- Packaging strategies
- Communicating with a service

Hosting Overview

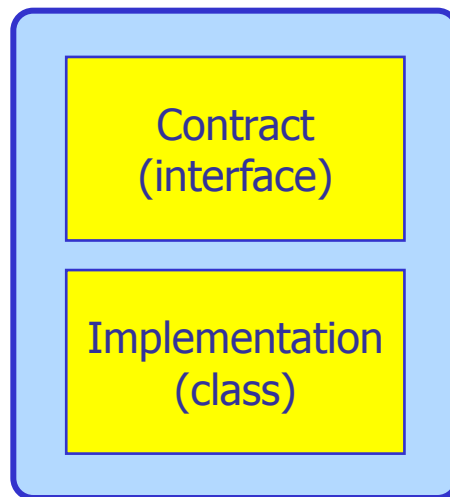
- A service must be hosted in a Windows process
 - A process can host many services
- Technically speaking, to host a service:
 - You must create an instance of the `ServiceHost` class
 - A `ServiceHost` instance is capable of hosting one type of service
 - You configure the `ServiceHost` instance, so it knows what it's hosting 😊

Hosting Options

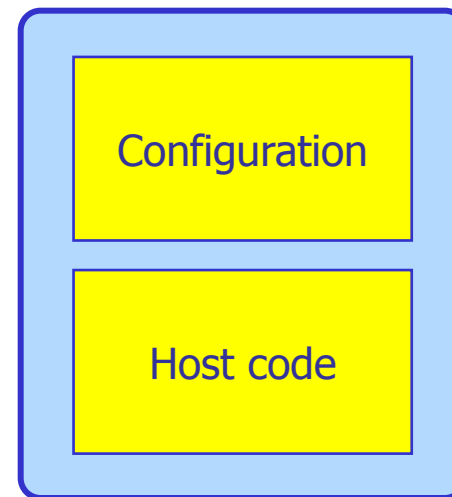
- Options for providing a host process:
 - IIS 5/6
 - Convenient and easy, IIS manages the lifecycle of the service
 - Only supports the HTTP protocol
 - Windows Activation Service (WAS)
 - System service, part of IIS 7+ in Vista/Windows 7/Windows 8 and Windows Server 2008 (can also be installed separately)
 - Not limited to the HTTP protocol
 - Automatic support for pooling, identity management, and isolation
 - Managed application (known as "self-hosted")
 - Console application, Windows service, Windows Forms, or WPF application
 - You manage the lifecycle of the service
 - Flexible but manual

Packaging Strategies

- Simplest packaging strategy:
 - Define a single class for the contract, implementation, and host
 - Not recommended!
- A better packaging strategy:
 - Define interface and implementation as separate types
 - Define service and host as separate assemblies/projects



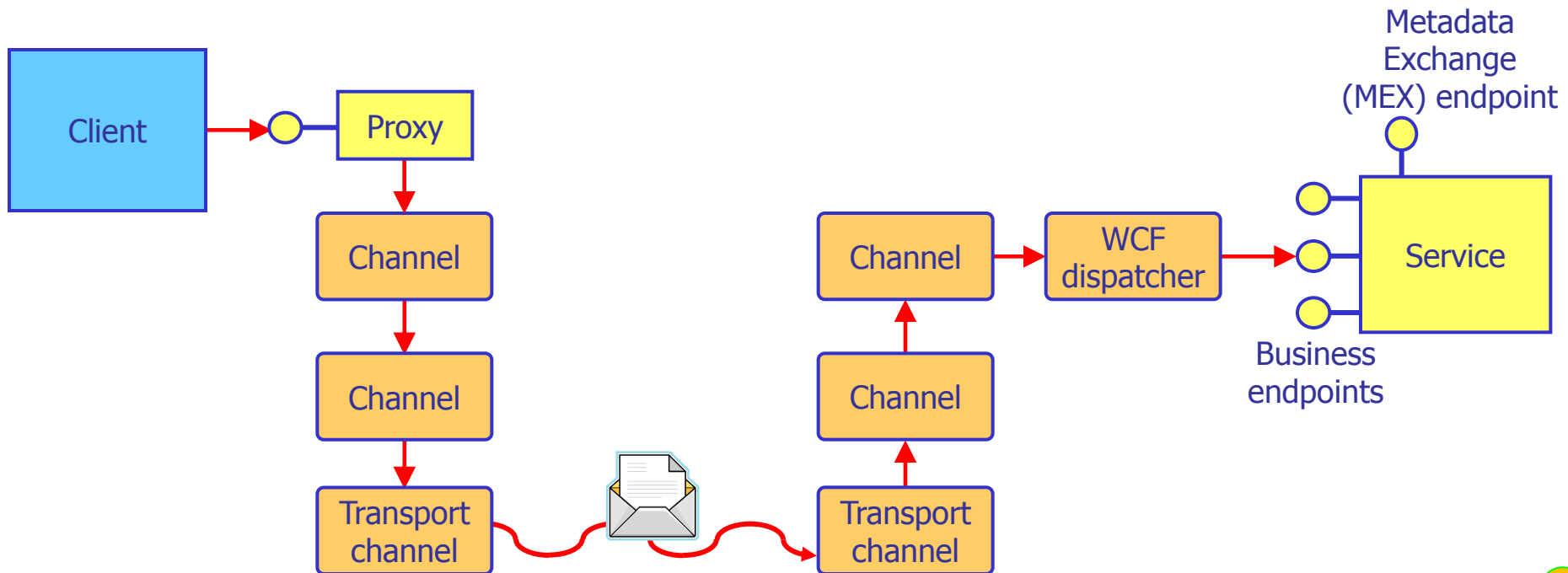
WCF Service Library



WCF Service Host

Communicating with a Service

- Clients communicate with a service via messages
 - Message passes through a series of channels at client and server
 - Channels are interceptors that perform a specific task (encoding, passing security context, propagating transaction context, etc.)
 - The "transport channel" sends/receives message over transport



Summary

- WCF and SOA essentials
- WCF architecture
- Service hosting and communication

Any Questions?

