infogain

Windows Communication Foundation

Ankur Pathak

August 3, 2018

WCF:Day1

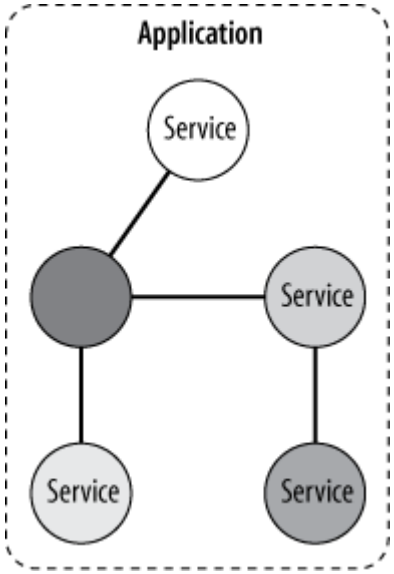
Agenda

- Service
- SOA
- WCF – Conceptual Overview
- WCF Buildings Block
- Defining a WCF Service
- Configuring WCF Services with endpoints
- Hosting Service
  - IIS
  - WAS
  - Self Host
- Implementing Clients
- FaultContract

day2

Service

- A service is a unit of functionality exposed to the world.
- Service orientation (SO) is an abstract set of principles and best practices for building service-oriented applications.
- It is the next evolutionary step in the long journey from functions to objects to components to services.
- A service-oriented application aggregates services into a single logical application, similar to the way a component-oriented application aggregates components and an object-oriented application aggregates objects.



What is SOA ?

- **Service - oriented architecture** is a style of programming, an architectural approach in software development, where an application is organized in functional units of code with a given behavior called services.

- **Services** are a group of methods that share a common set of requirements and functional goals.

- The service operations are invisible — there is no direct interaction with a user and the work is executed as instructed by the given input parameters.

Four Tenets Of SOA

- **Boundaries are explicit :**
  - Service should be self contained.
  - A **contract** is considered metadata for the service being a black box with only this well – described interface.
- **Services are autonomous :**
  - Service should be independent of other service.
  - Services are considered standalone pieces of code that do not rely on the behavior of other services.
- **Services share schema and contract, not class :**
  - Service should be able to define themselves.
  - A schema is the definition of a service operation and describes the signature in a platform -neutral way: the name of the functions, types of parameters, and the type of return value.
- **Service compatibility is based on policy :**
  - A policy is used to negotiate elements in the communication, such as message format and security requirements.

Technologies Behind SOA

- **SOAP :**
  - Simple Object Access Protocol (SOAP) is an XML specification for exchanging data as structured information in messages.
  - SOAP standardizes how data is exchanged on the wire. As it's based on XML, it is platform agnostic.
  - A SOAP message simply carries the data as a message.
  - A SOAP envelope contains a (optional) header and a (required) body element.

- **WSDL :** WSDL is a XML - formatted definition of the contract.
  - It contains all metadata for the interface of the service including function names, parameter names, and their types and the types of return values.
  - The purpose of a WSDL file is to define this contract in a cross - platform way as the types are expressed in XML types.

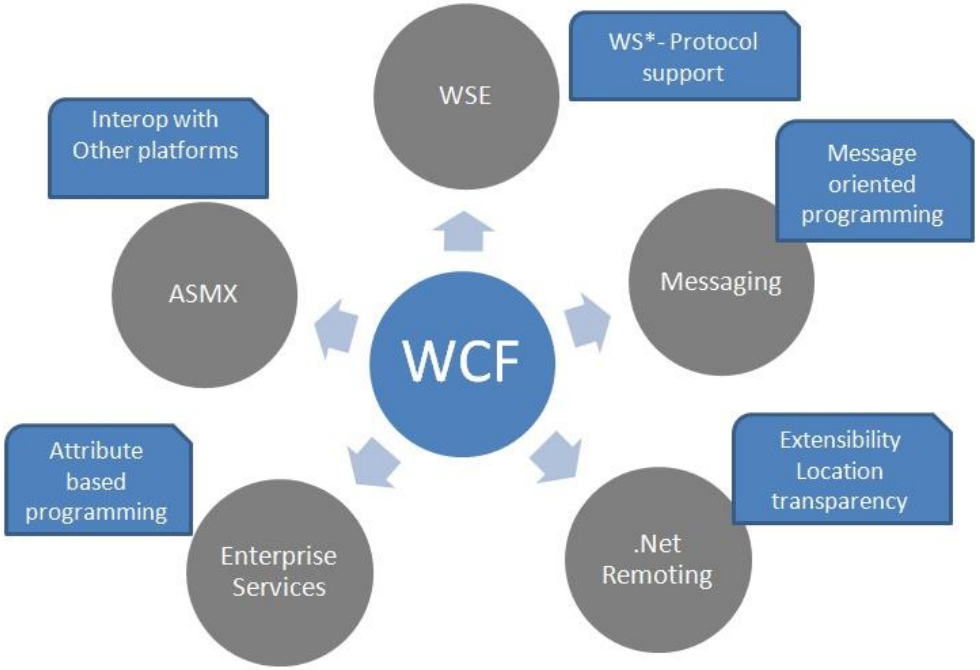
- **CONTRACT - FIRST PRINCIPLE :**
  - The first thing that should be clearly defined is the contract of a service.

WCF Overview

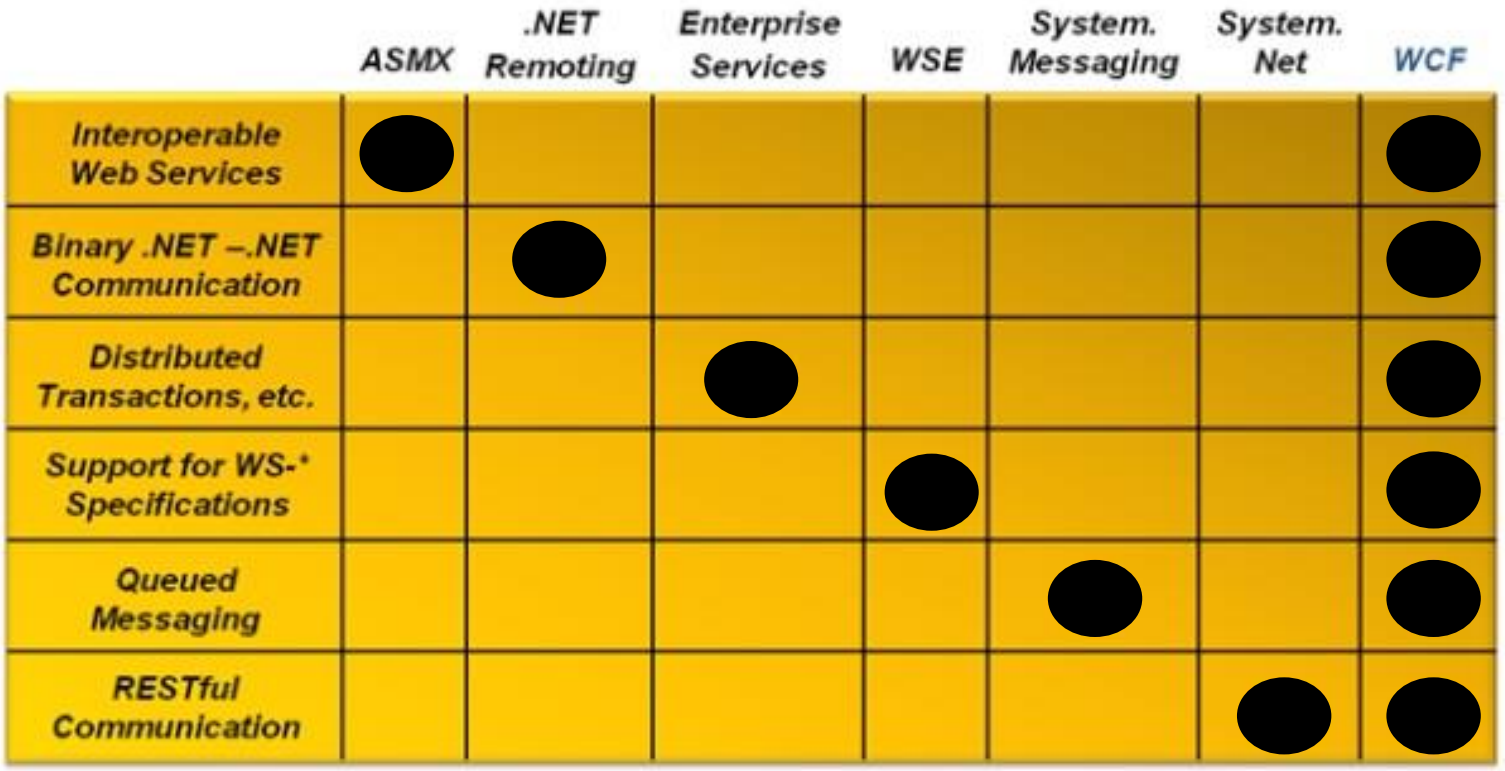
- Windows Communication Foundation
- Platform for building distributed, service-oriented applications
  - Define services and hosts for those services
  - Define clients to connect to services
- SOAP Messaging
- Make it easy for .NET, object-oriented programmers to build distributed, service-oriented applications
  - Secure
  - Scalable
  - Flexible

WCF : Unified Model

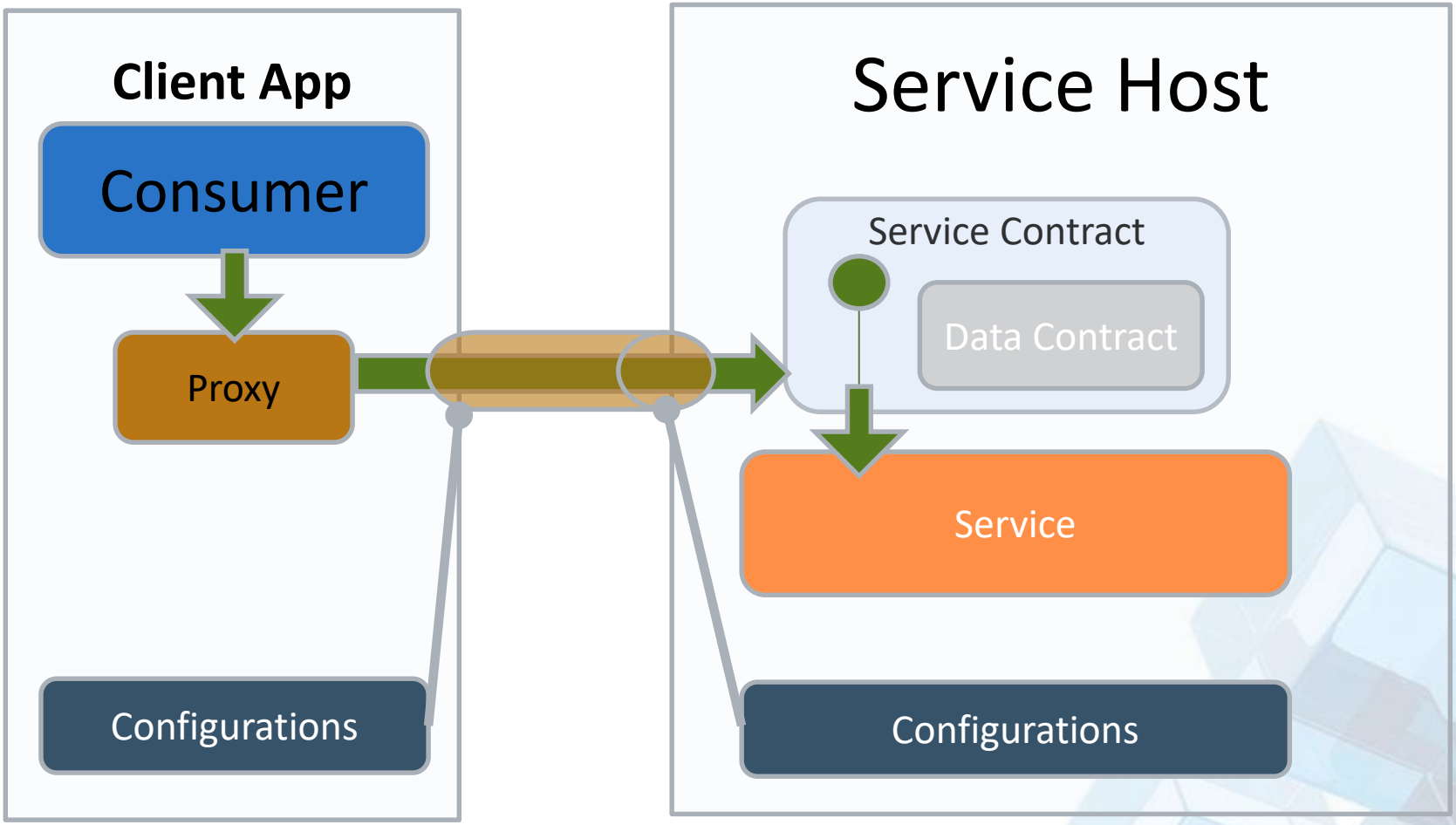
- Windows Communication Foundation (WCF) is Microsoft's unified programming model for building service-oriented applications.
- It enables developers to build secure, reliable, transacted solutions that integrate across platforms and interoperate with existing investments.



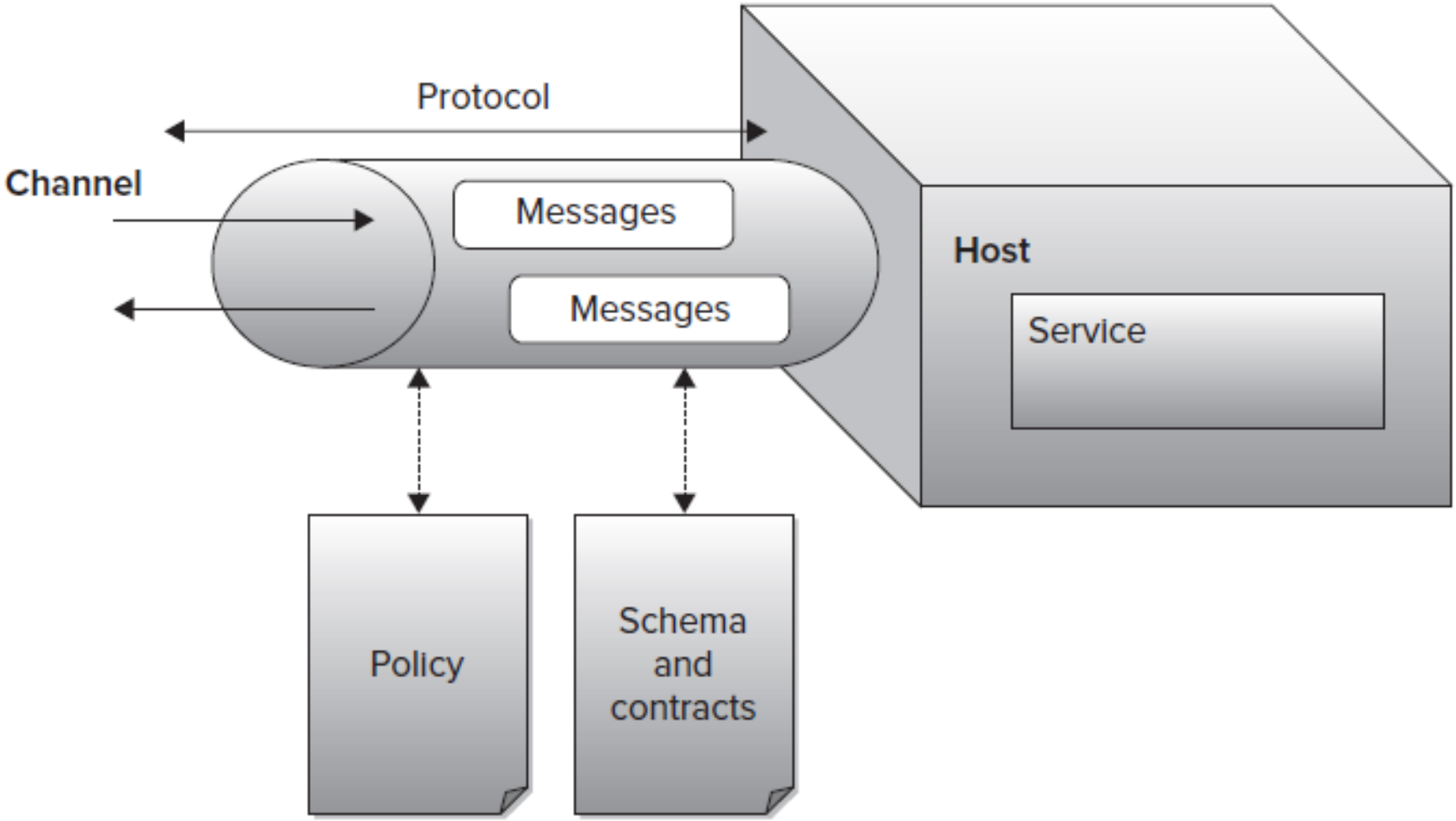
Unification of Microsoft's Distributed Computing Technologies



WCF Buildings Blocks



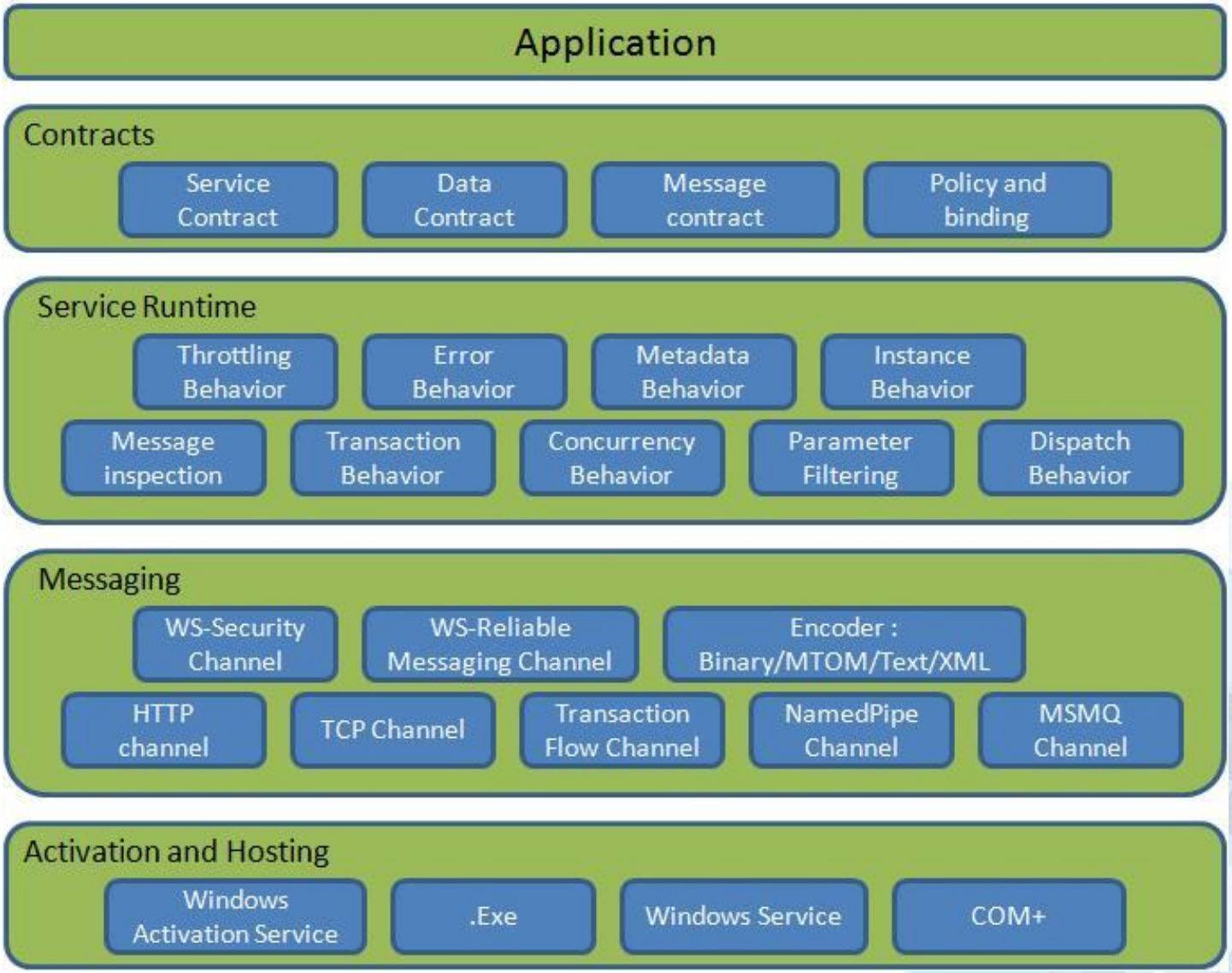
ANATOMY OF A SERVICE



WCF Version History

WCF Version	Introduced with .NET & IDE	Features Detail
3.0	3.0 and Visual Studio 2005	Introduced first version of WCF with many features like Address, Binding, Contract, Sessions, Instancing, and Concurrency management
3.5	3.5 and Visual Studio 2008	UriTemplates Support ,Support for REST Style Services, Asp.NET Ajax Integration and JSON support, Added WS* Specification Support, Support for RSS and Atom feeds.
4.0	4.0 and Visual Studio 2010	Simple Configuration, Serialization Enhancements , Support for WS-Discovery Routing Service, Standard Endpoints, Workflow Service
4.5	4.5 and Visual Studio 2012	Task-based Async Support ,Contract-First Development, WCF Configuration Validation ,Web Socket Support, UDP Endpoint Support ,New Https protocol mapping on IIS, Streaming Improvements ,Multiple Auth support for single endpoint

WCF Architecture



Components of WCF Service

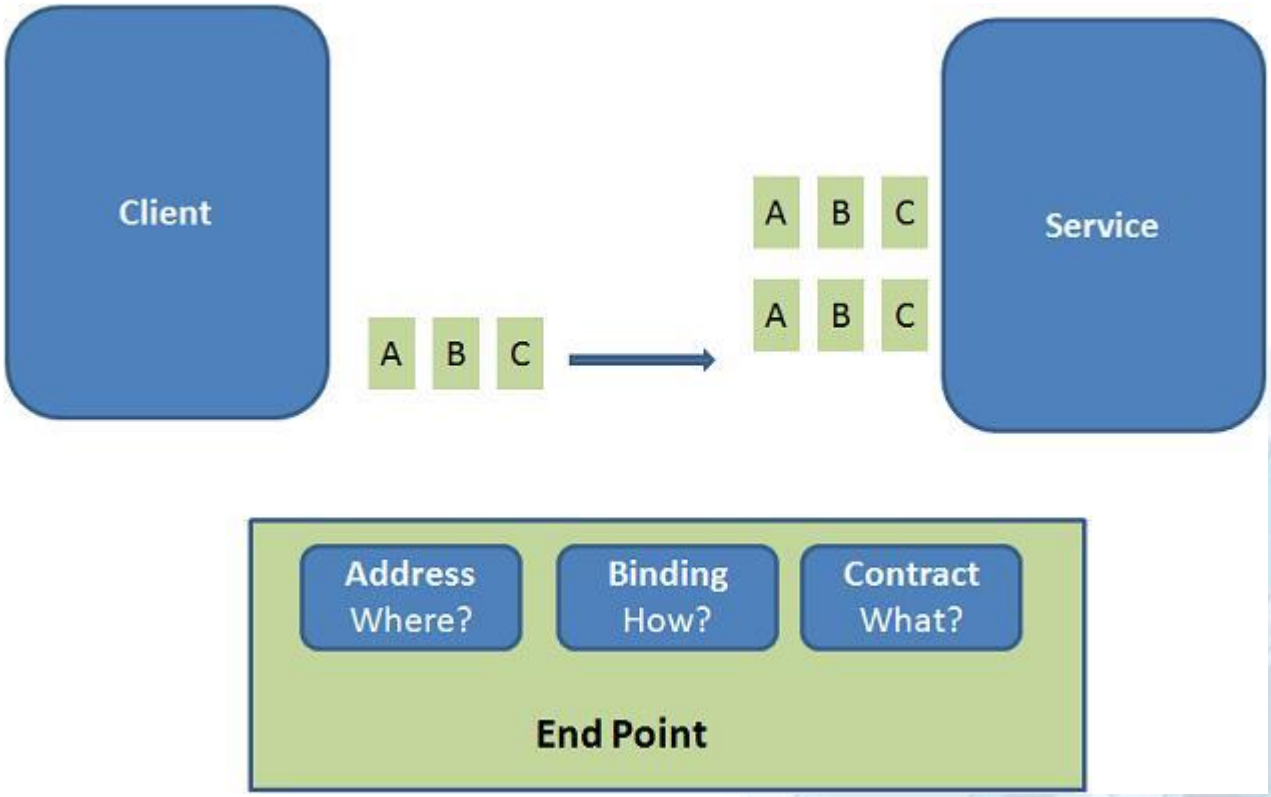
- **Service**
  - Service Contract : describe the operation that service can provide.
  - Data Contract : describes the custom data type which is exposed to the client.
  - Message Contract : we can create our own message format.
  - Fault Contract : This helps us to easy identity, what error has occurred.
- **Endpoints**
  - Where : Address
  - How :Binding
  - What :Contract
- **Hosting Environment**
  - Self Hosting
  - IIS
  - Windows Activation Service (WAS)

Components of WCF Service

- **Proxies**
  - For clients only
  - Create channel factory
- **Channels**
  - Facilitating communication between clients and services.
    - Transport channel
    - Communication channel
    - Message Encoding
- **Behaviors**
  - How message are processed
  - Local to service or client
  - Authentication, authorization, transactions and throttling etc.

Endpoint

- WCF Service is a program that exposes a collection of Endpoints. Each endpoint is a portal for communicating with the world.
- All the WCF communications are take place through end point. End point consists of three components.
  - **Address**
  - **Binding**
  - **Contract**



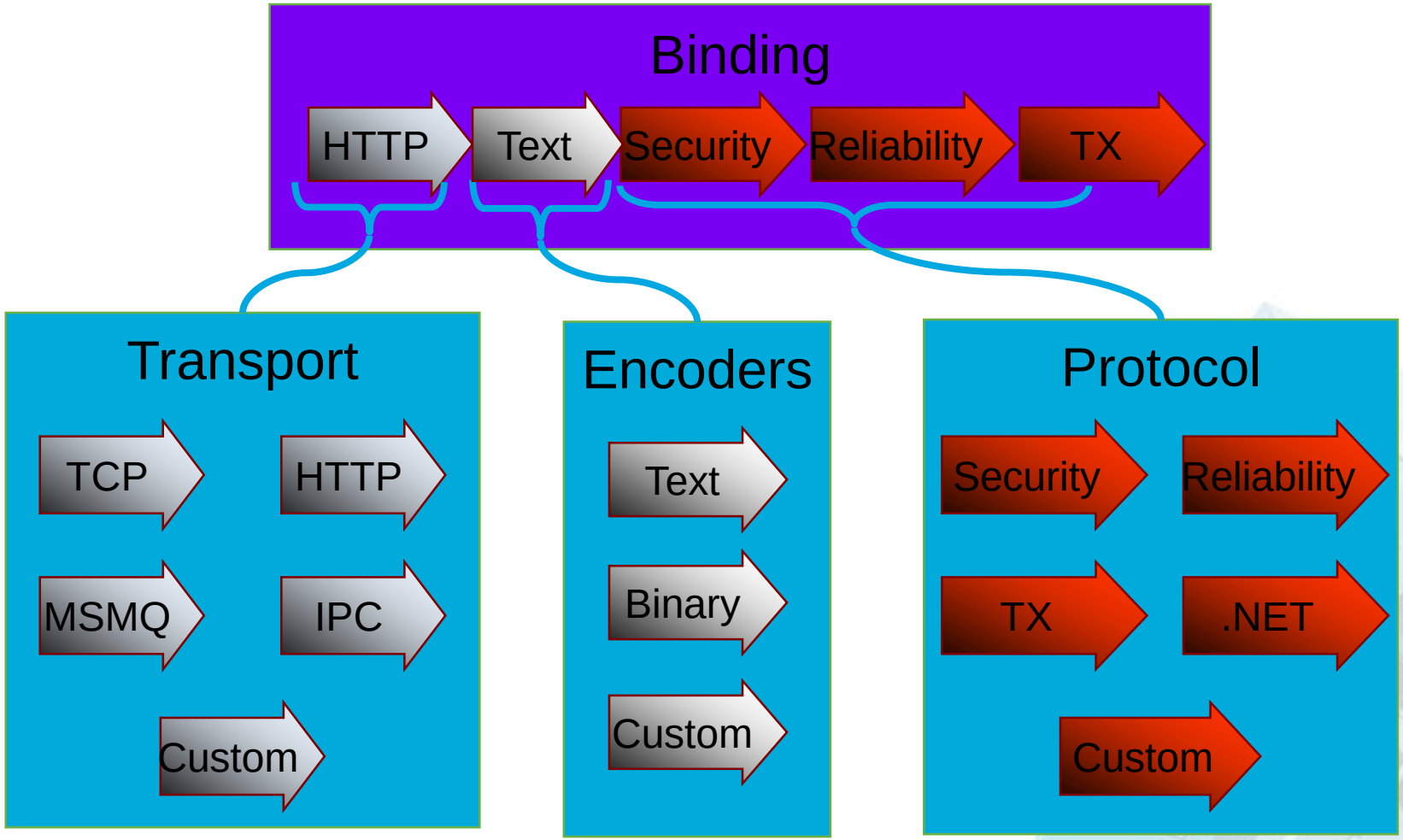
Address

- Basically URL, specifies where this WCF service is hosted .Client will use this url to connect to the service.

e.g http://localhost:8090/MyService/SimpleCalculator.svc

Transport Protocol	Example Address
Http	http://localhost:8001 http://localhost:8001/Service1
Http(Secure)	https://localhost:8001
TCP Peer network	net.tcp://localhost:8001 net.p2p://localhost/
IPC(Inter-process communication over named pipes)	net.pipe://localhost/PipeService1
MSMQ(Microsoft Message Queue)	net.msmq://localhost

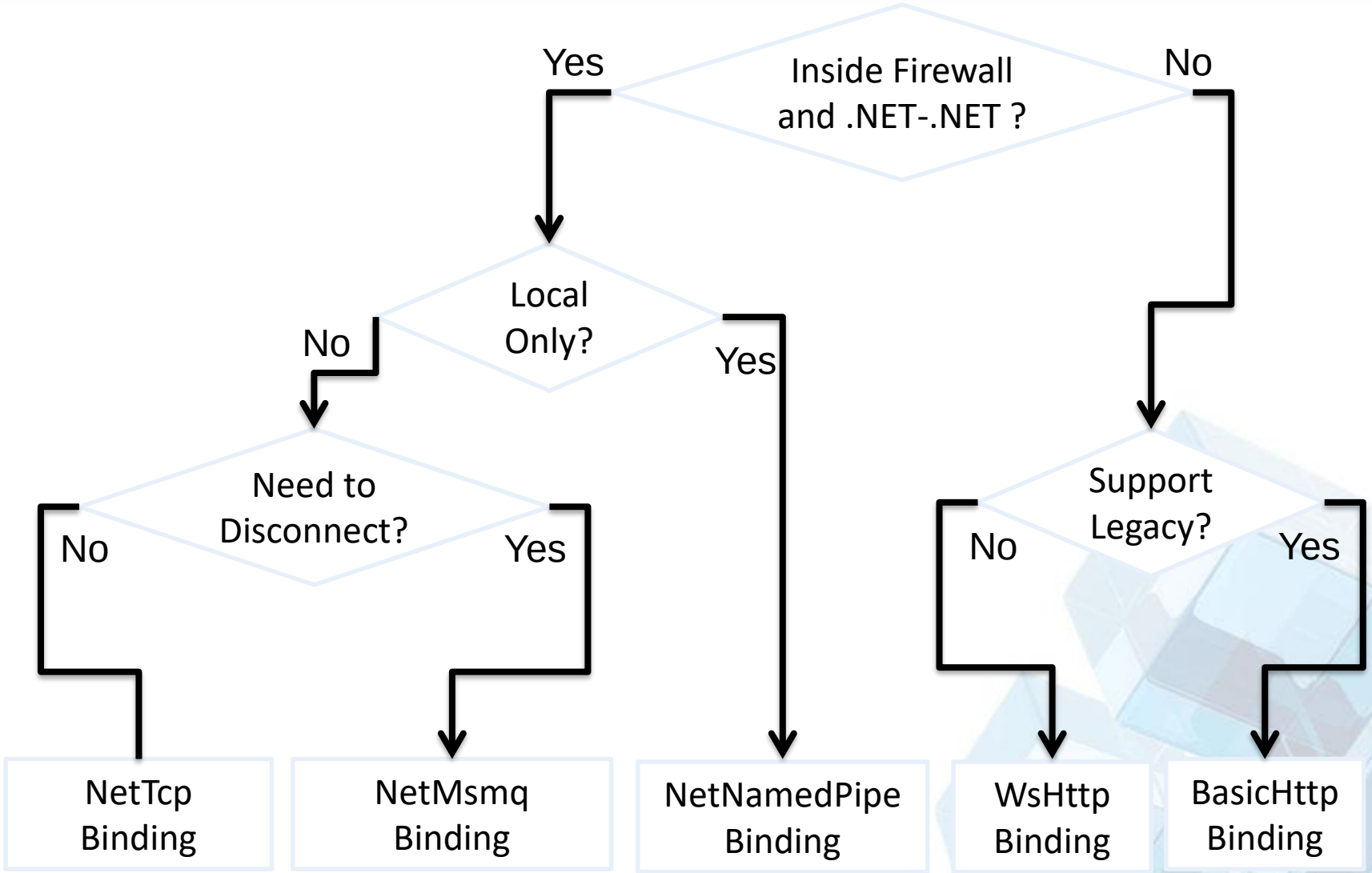
Bindings & Binding Elements



Binding

Name	Transport	Encoding	Interoperable
BasicHttpBinding	HTTP/HTTPS	Text, MTOM	Yes
NetTcpBinding	TCP	Binary	No
NetNamedPipeBinding	IPC	Binary	No
WSHttpBinding	HTTP/HTTPS	Text, MTOM	Yes
NetMsmqBinding	MSMQ	Binary	No

Choosing a Binding



Binding configuration

- Binding can be configured either through configuration file or Programming.
  - Administrative (Configuration file):

  - Programming Model:

Contract

- Usually name of the Interface will be mentioned in the Contract, so the client application will be aware of the operations which are exposed to the client.

- Each operation is a simple exchange pattern such as one-way, duplex and request/reply.
  - Default exchange pattern is request/reply.

- E.g
- **<endpoint address="http://localhost:8090/MyService/MathService.svc" contract="IMathService" binding="wsHttpBinding"/>**

Instancing

The following instancing modes are available:

- PerCall: A new InstanceContext (and therefore service object) is created for each client request.
- PerSession: A new InstanceContext (and therefore service object) is created for each new client session and maintained for the lifetime of that session (this requires a binding that supports sessions).
- Single: A single InstanceContext (and therefore service object) handles all client requests for the lifetime of the application

Lets Feel it

- **Create a Service Type**
  - Define Service Contract.
- **Create a Host**
  - **Expose End Points**
    - **Address**
    - **Binding**
    - **Contracts**
- **Create a Client**
    - **Create a client proxy**
    - **Where , How, What (Address, Protocol and Contract)**

WCF Hosting

- Multiple hosting and protocols supported by WCF.
- Microsoft has introduced the WCF concept in order to make distributed application development and deployment simple.

Hosting Environment	Supported protocol
Windows console and form application	HTTP,net.tcp,net.pipe,net.msmq
Windows service application (formerly known as NT services)	HTTP,net.tcp,net.pipe,net.msmq
Web server IIS6	http, wshttp
Web server IIS7 - Windows Process Activation Service (WAS)	HTTP,net.tcp,net.pipe,net.msmq

Windows Activation Service

- Windows Activation service is a system service available with Windows vista and windows server 2008.
- It is available with IIS 7.0 and it is more powerful compared to IIS 6.0 because it supports Http, TCP and named pipes were IIS 6.0 supports only Http.
- Hosting WCF in Activation service takes many advantages such as process recycling, isolation, idle time management and common configuration system. WAS hosted service can be created using following steps
  - Create WAS hosted service
  - Enable different binding to the hosted service
  - Enable WCF for non-http protocols

Rest Services with WCF:Day2

Agenda

- Understanding REST
- Resource Oriented Architecture
- REST support in WCF 3.5
- Implementing a RESTful Service
- Using WebServiceHost/Factory
- Calling a RESTful service
- WebOperationContext
- Ajax integration via JSON

Back

REST

- Representational State Transfer
- Architecture for building systems (introduced by Roy Fielding)
- Based on the advantages of the Web
  - URIs
  - Uniforms Interface
  - Stateless
  - Hypermedia-driven (i.e. links)
  - Cache-ability

It's all about URIs

- A RESTful services models its resources as URIs
  - Builds on the success of the web
- Everything is addressable via a URI
- You interact with a resources by using the Uniform Interface
  - Start with URI, add well-known HTTP verbs

Uniform Interface

GET

- No side-effects (safe)
- Idempotent (Calling a million times has the same as one request)
- Retrieves resources
- Cacheable

POST

- Creates a new resource
- Same as SOAP – still unsafe

PUT

- Updates a new Resources
- Also idempotent

DELETE

- Removes a resources
- Also idempotent

WCF programming styles

- **Most of the built-in binding use SOAP & WS-* by default**
  - You have to configure them to disable SOAP
- **WCF 3.5 comes with a new Web (REST) programming model**
  - Found in System.Service.Model.Web.dll
  - Allows you to map HTTP request to method via URI templates
- **You enable the Web model with new binding/behavior**
  - Apply to messaging layer using WebHttpBinding
  - Apply to dispatcher using WebHttpBehavior
- **WebHttpBinding produces an appropriate HTTP-based channel**
  - You can customize certain settings (cookies, proxy, security ,etc.)
  - Security modeled by WebHttpSecurity (HTTP vs HTTPS)
  - Produces a WebMessageEncoder (supports XML & JSON)
- **WebHttpBehavior customizes the HTTP-based dispatching logic**

Wiring-up a Web-based service

add the Web binding →
add the Web behavior →

```
...
ServiceHost host = new ServiceHost(typeof(EvalService),
new Uri("http://localhost:8080/evals"));

host.AddServiceEndpoint(typeof(IEvals),
    new WebHttpBinding(), "");
host.Description.Endpoints[0].Behaviors.Add(
    new WebHttpBehavior());
host.Open();  // service is up and running

Console.ReadLine(); // hold process open
...
```

WebServiceHost/Factory

- **WebServiceHost simplifies hosting Web-based services**
  - Drives from ServiceHost
  - Automatically adds endpoints for the base address using WebHttpBinding
  - Automatically adds WebHttpBehavior to the endpoint

```
...
WebServiceHost host = new WebServiceHost(
    typeof(EvalService),
    new Uri("http://localhost:8080/evals"));
host.Open();  // service is up and running
Console.ReadLine(); // hold process open
...
```

WebOperationContext

- **Use WebOperationContext to access HTTP specific within methods**
  - To retrieved the current context use WebOperationContext.Current
  - Provides properties for incoming/outgoing request/response context
- **Each context object surfaces most common HTTP details**
  - URI, Content Length, Content Type,Etags, etc.

WebMessageFormat

- **WCF provides support for two primary Web formats : XML & JSON**
  - You control the format via RequestFormat and ResponseFormat

```
[ServiceContract]
public interface IEvals
{
    [WebGet(UriTemplate = "/evals?name={nameFilter}",
        ResponseFormat = WebMessageFormat.Json)]
    [OperationContract]
    List<Eval> GetCurrentEvals(string nameFilter);
...
```

Enabling Ajax integration

- **Use the WebScriptEbalingBehavior to Ajax-enable a WCF service**
  - Make JSON the default message format
  - Enables Ajax-style invocations for each [Operation Contract]
  - You don't need to use [WebGet] or [WebInvoke]
  - Produces a JavaScript proxy for Ajax clients (via base address + "/js")
- **Use WebScriptHostFactory to simplify AjaxHosting**
  - Automatically adds the WebScriptEnablingBehavior

```
<%@ ServiceHost Language="C#" Service="EvalService" Factory=
    "System.ServiceModel.Activation.WebScriptServiceHostFactory" %>
```

infogain

Thank You

For additional information, visit us at:

www.infogain.com

August 3, 2018