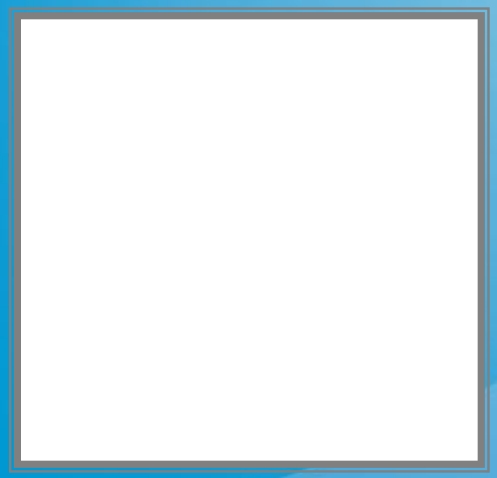
개발자분들의 꿈과 함께 합니다. 당신의 열정에 투자합니다.
Realize your Potential @ MSDN

msdn

WCF (Windows Communication Foundation) 제대로 알기



WCF 등장 배경 및
프로그래밍 기초 소개

드원 테크놀로지
수석 컨설턴트
유경상

Microsoft

Agenda

- Understanding Web Service
  - History of Communication
  - Problem of Legacy Communication Infra
  - Requirement of New Communication Infra
- Understanding WCF
  - What's WCF?
  - WCF Features
  - Basic Programming Model of WCF
- Conclusion

msdn

UNDERSTANDING WEB SERVICES

Microsoft Silverlight

Microsoft

Why Web Service?

- Understanding Background of Web Services
  - Web Service의 등장 의의를 이해할 수 있음
  - Web Service를 왜 사용해야 하는가에 대한 답을 찾을 수 있음
  - SOA (Service Oriented Architecture) 를 이해하는데 도움이 됨
  - Web Services의 다양한 표준에 대한 이해에 도움이 됨
  - WCF의 등장 배경을 이해하는데 도움이 됨
- Communication between Applications
  - 오랫동안 어플리케이션 사이에 통신 방법이 변화됨
  - 다양한 문제점들을 해결하는 방향으로 발전
  - 보다 편리한 개발을 위한 방향으로 발전
  - 새로운 프로토콜, 제품 들이 등장
  - 가장 최근에는 XML Web Service로 통합되고 있음

History of Communication (1)

- Stone Age
  - 메인프레임과 더미 터미널 통신
  - 문자 기반의 통신
- Age of Down sizing
  - 워크스테이션 서버들을 이용한 분산 처리
  - TCP/IP, Socket 프로그래밍 보급
  - 전문 방식 메시지 통신
- Age of Middleware
  - 메시지 처리의 자동화
    - 데이터 마샬링, 메쏘드 디스패치 자동화
    - 프로그래밍 언어에 의존적
  - 데이터 패킷 기반 미들웨어: Tuxedo, Tmax, ...
  - RPC 기반 미들웨어: Entera, ...

History of Communication (2)

- Age of Distributed Components
  - 프로그래밍 언어와 무관한 통신 방법 필요
  - 객체 지향적인 프로그래밍 방식 필요
  - 객체 지향 프로그래밍의 한계
    - 컴파일러 의존적, 낮은 코드 재 사용성 등
  - 분산 컴포넌트 등장
    - Component Based Design (CBD)
  - 다양한 분산 객체 기술/프로토콜 도입
    - COM/DCOM, CORBA/IIOP, Java/RMI (Remote Method Invocation)
  - 분산 객체 기반의 미들웨어
    - COM+ (DCOM)
    - Orbix (IIOP)
    - EJB (RMI)

Problem of Distributed Components

- Interoperability Problem
  - 내 것이 네 것보다 낫다 !
    - CORBA, DCOM, RMI 가 각각의 바이너리 포맷을 가짐
    - 각 표준의 벤더들이 자신 기술이 컴포넌트 기술의 인프라가 되길 원함
    - 상호운영성(interoperability)이 현저히 떨어짐
  - 프로그래밍 언어 장벽
    - CORBA – C/C++ 및 COBOL 등 몇몇 언어 지원
    - RMI (java) – Java 언어만을 지원
    - DCOM – C/C++, VB, Delphi, PowerBuilder 등 언어 지원
  - 인터넷 비호환
    - 분산 객체 프로토콜들은 가변 포트 사용
      - 다수의 클라이언트 지원을 위해 다수의 TCP/IP 포트 사용
    - 방화벽을 통과하기 매우 어려움
      - 보안상 다수의 포트를 개방할 수 없음

Engaging Web Services

- Request for New Communication Infra
  - 표준 준수
    - 국제 표준 기구에서 제정하는 국제적 IT 표준을 사용할 것
  - 상호운영성/독립성
    - 특정 벤더에 종속적이지 않으며, 구현 기술, 구현 언어에 독립적일 것
    - 다양한 운영체제, 개발 플랫폼에서 모두 쉽게 지원할 수 있을 것
  - 인터넷 적응성
    - 인터넷 환경에 적응 하기 쉬운 인터넷 기술을 사용할 것
  - 낮은 복잡도
    - 간단명료하여 복잡한 처리를 요구하지 않을 것
- Web Service Concept
  - 세부 구현에 관계 없이 소프트웨어 컴포넌트(모듈, 함수)를 인터넷 상의 하나의 서비스(기능 집합)으로서 제공
  - 높은 상호운영성을 가지며 다양한 H/W, 운영체제, 개발 플랫폼 상에서 서비스를 개발하거나 클라이언트를 개발

SOAP

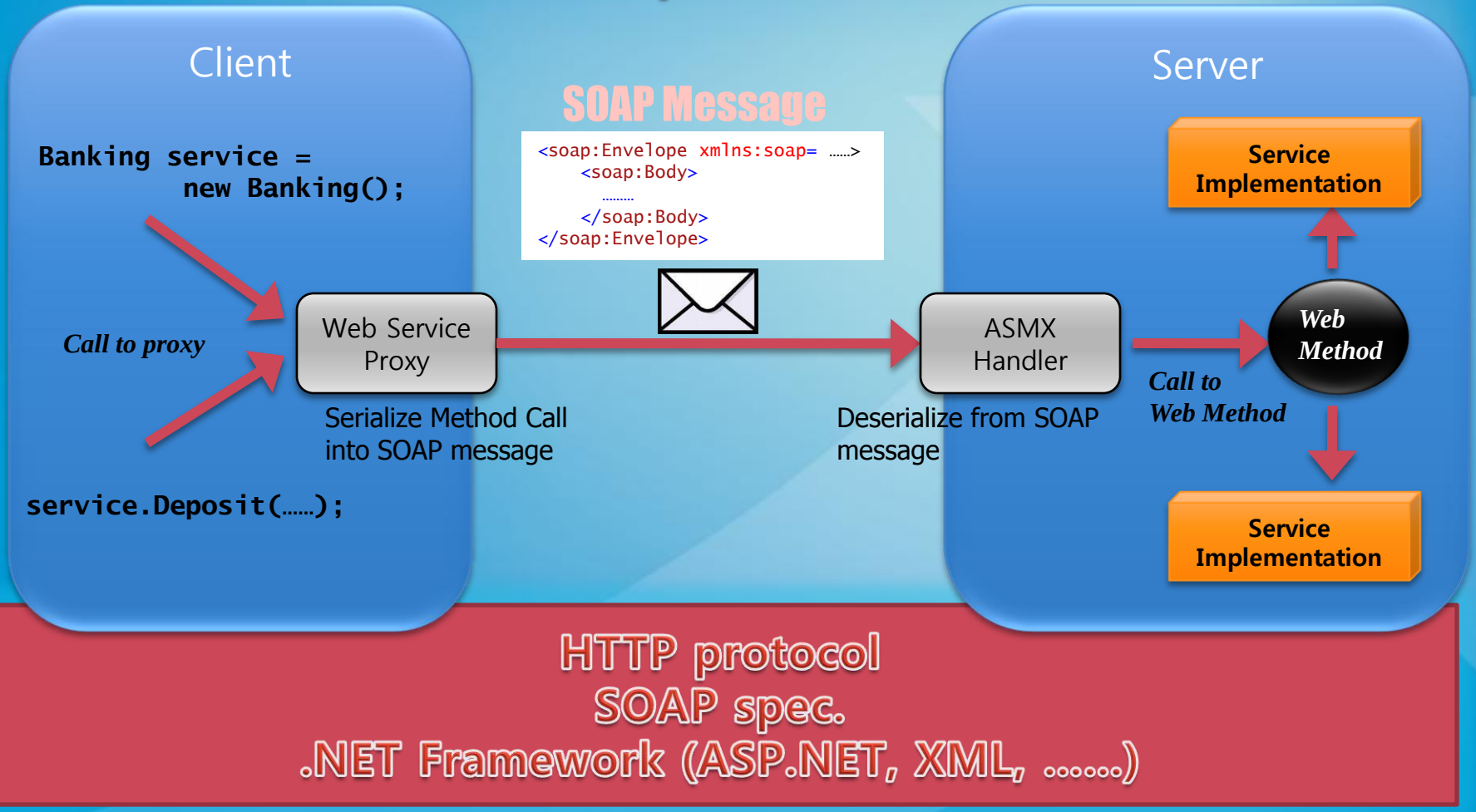
- SOAP (Simple Object Access Protocol)
  - 웹 서비스에 사용하는 프로토콜
  - 메시지 기반
  - W3C 표준 : Microsoft, IBM, BEA 등 주요 벤더들 주도
  - XML 기반이므로 다양한 플랫폼에서 모두 지원될 수 있음
  - HTTP, SMTP 등 인터넷 기반 프로토콜을 하부 구조로 사용함
  - 간단하지만 확장 가능한 명세로 구성됨

```
POST /url HTTP/1.1
Host: HostServerName
……
SoapAction: http://banking.sample/Deposit
……

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <soap:Header>
        <!-- addtional header information -->
    </soap:Header>
    <soap:Body>
        <Deposit xmlns="http://banking.sample/">
            <AccountID>123-45-678-1234</AccountID>
            <Amount>20.0</Amount>
        </Deposit>
    </soap:Body>
</soap:Envelope>
```

SOAP Message Driven Comm.

ASP.NET Web Service Example



Microsoft Silverlight

Microsoft

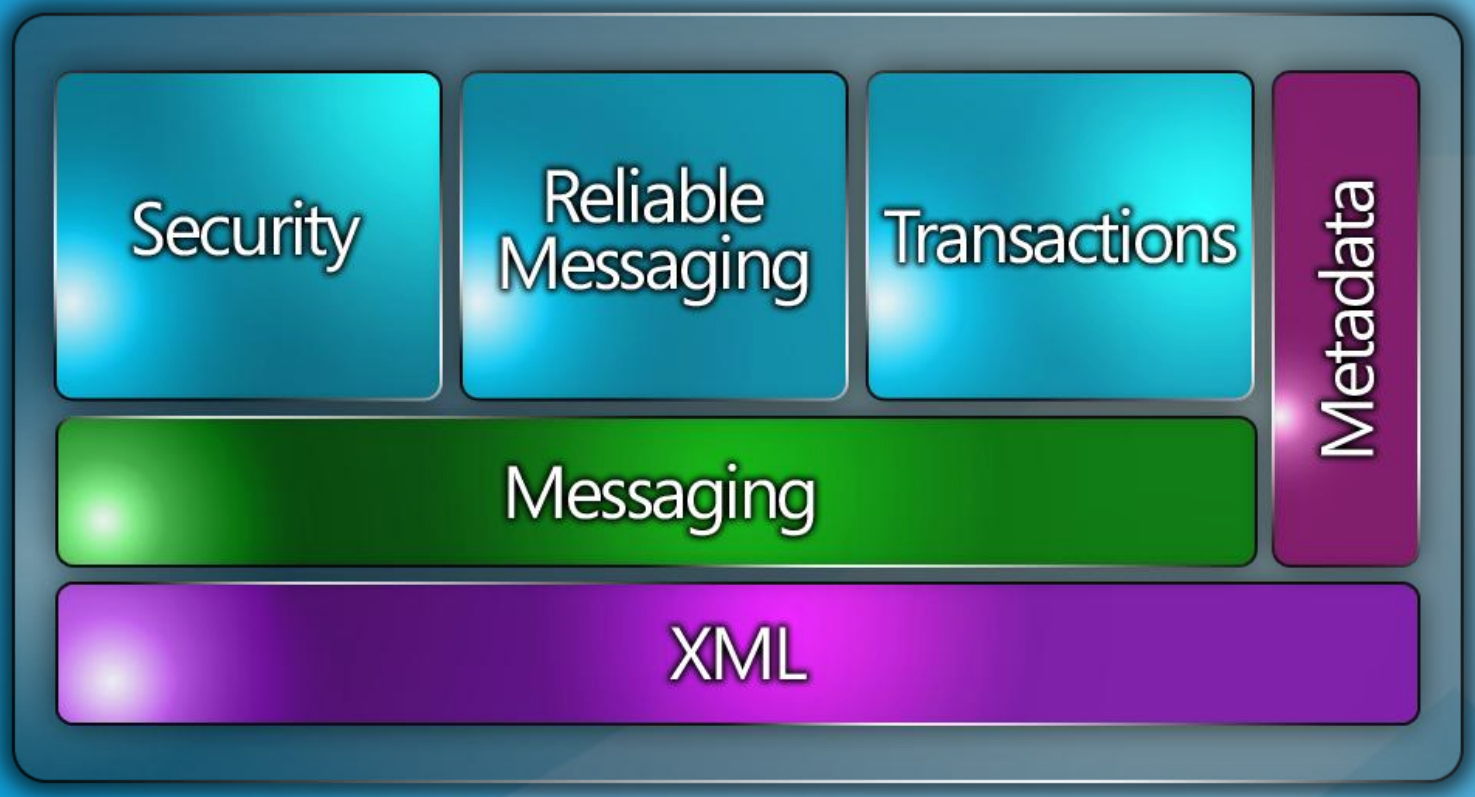
We Need More……

- Problems of Web Service & SOAP
  - 구현 마다 조금씩 상이한 SOAP 메시지 구현
  - 진보된 SOAP 메시지 보안 필요
    - SSL 만으로는 라우팅 등 다양한 보안 요구사항을 만족하기 어려움
    - 다양한 인증 서비스를 만족할 수 있어야 함
  - 신뢰할 수 있는 SOAP 메시지 배달 필요
    - Reliable Messaging
  - Long-running, Loosely-coupled 트랜잭션 지원 요구
    - 인터넷 상에 연결된 웹 서비스들 사이의 트랜잭션 메커니즘 필요
  - 웹 서비스를 통해 대용량 바이너리 데이터 통신 필요
  - 이러한 새로운 요구 사항들이 상호운영성을 해쳐서는 안됨
- Lack of functionality with SOAP 1.x

New Web Service Standards (1)

- WS-* Specifications
  - 새로이 요구되는 다양한 기능을 위한 웹 서비스 스펙
    - 호환성 관련 : WS-I Profile
    - 메시징 관련 : SOAP 1.2, MTOM, WS-Addressing, …
    - 보안 관련 : WS-Security, WS-Trust, WS-Federation, …
    - 트랜잭션 관련 : WS-AtomicTransaction, …
    - 신뢰성 관련 : WS-ReliableMessaging
    - 메타데이터(WSDL) 관련 : WSDL, WS-Policy, …
  - SOAP 메시지의 확장
    - SOAP 메시지의 Header 요소를 통해 기능 확장
    - 기존 웹 서비스 근간을 그대로 유지
  - W3C에 의해 표준으로 유지 (혹은 일부 진행 중)
    - 상호운영성
    - 벤더에 종속적이지 않음
  - 다양한 벤더들이 지원 (모든 스펙을 지원하는 것은 아님)
    - Microsoft, BEA, IBM, ……

New Web Service Standards (2)



Need a new communication infra

Communication of Windows World (1)

- WinSock
  - Windows Socket 라이브러리
  - TCP/IP 등 저 수준의 네트워킹
- WinINet
  - HTTP, FTP 등 인터넷 프로토콜 구현
  - 클라이언트 용
- RPC
  - Windows 서비스들이 다수 사용
  - TCP/IP, Named Pipe, HTTP 등 다수 Transport 하부 프로토콜 사용 가능
  - C/C++ 전용
- DCOM
  - RPC 기반
  - Windows 운영체제와 COM+에서 폭넓게 사용
- MSMQ
  - 비동기 메시지 전송 및 메시지 큐 기능 제공

Communication of Windows World (2)

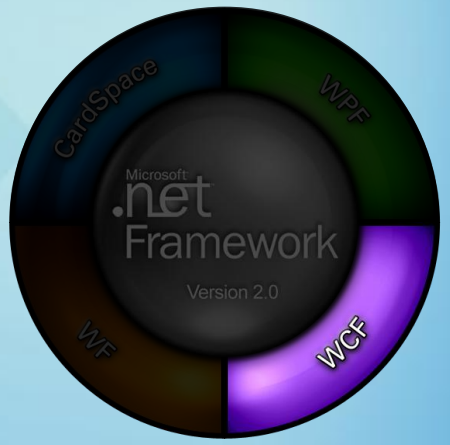
- In Managed World
  - System.Net namespace (HTTP, FTP, SMTP, Socket)
  - .NET Remoting
  - ASP.NET Web Service
  - WSE 3.0
    - WS-Addressing, WS-Security, MTOM 등 일부 WS-* 스펙 지원
    - ASP.NET Web Service의 에드온 성격
  - Enterprise Services (COM+)
  - MSMQ (System.Messaging)
- Problems
  - 각기 다른 프로그래밍 모델을 가짐
  - 각기 다른 configuration 설정을 가짐
  - 서비스의 구현 로직은 재사용 가능하지만 통신에 필요한 코드는 재사용할 수 없음

Requirement of New Comm. Infra

- Service Oriented Architecture
  - 서비스 기반 아키텍처
- Web Service based
  - XML 기반
  - 상호운영성
  - 다양한 웹 서비스 스펙들을 지원
    - 메시징, 보안, 트랜잭션, 신뢰성, 메타데이터 등
- Unified Programming Model
  - 단일 코드로 다양한 프로토콜, 성능 요구 사항에 대처할 수 있어야 함
  - HTTP, TCP, MSMQ 등 하부 프로토콜에 관계 없이 단일 코드 사용
- Compatibility
  - ASP.NET Web Service 등 기존 서비스/클라이언트와 호환되어야 함
- Support New Trend
  - P2P communication

Windows Communication Foundation

- What's WCF ?
  - 차세대 닷넷 Communication 인프라
  - .NET Framework 3.0의 기능
  - Service Oriented Architecture 기반
  - Web Service (XML + SOAP) 기반
  - 다양한 Communication 프로토콜 지원
    - HTTP
    - TCP/IP
    - Named Pipe
    - MSMQ
    - P2P
  - 다양한 웹 서비스 스펙 지원
    - WS-Security
    - WS-AtomicTransaction
    - WS-ReliableMessaging



Why WCF ?

- Future Communication Infra
  - 웹 서비스 기반(XML + HTTP)
  - 서비스 지향 아키텍처(SOA)
  - Interoperability
    - 표준 WS-* 스펙 구현
  - Next generation communication infra of Microsoft
- Common Programming Model
  - Web Service, .NET Remoting, LRPC, MSMQ 프로토콜에 관계 없이 하나의 프로그래밍 모델 사용 가능
  - 단일 코드 베이스 사용 가능
- New feature
  - 트랜잭션, 신뢰할 수 있는 메시징, P2P, ……
  - 다양한 바인딩(프로토콜 지원)

WCF Features

- Web Service Based
  - XML, HTTP, SOAP, WSDL 등 웹 서비스 스펙 준수
- Security
  - WS-Security, WS-Trust 등 메시지 기반 보안 제공
  - HTTPS, 인증 기반의 TCP/IP 등 트랜스포트 수준의 보안 제공
- Transaction
  - 분산 트랜잭션 지원
    - TCP 바인딩 사용 시 OLE Transaction 직접 사용
    - HTTP 바인딩 사용 시 WS-AT 를 통해 트랜잭션 전파
- Reliability
  - 신뢰도 높은 메시징을 위해 WS-RM 프로토콜 구현
  - 세션 기능 제공 (ASP.NET의 세션과는 다른 개념)
- P2P Networking
- Rich Built-in Bindings
  - BasicHttp, WSHttp, WSDualHttp, NetTcp, NetMsmq, NetNamedPipe 등

msdn

PROGRAMMING WCF

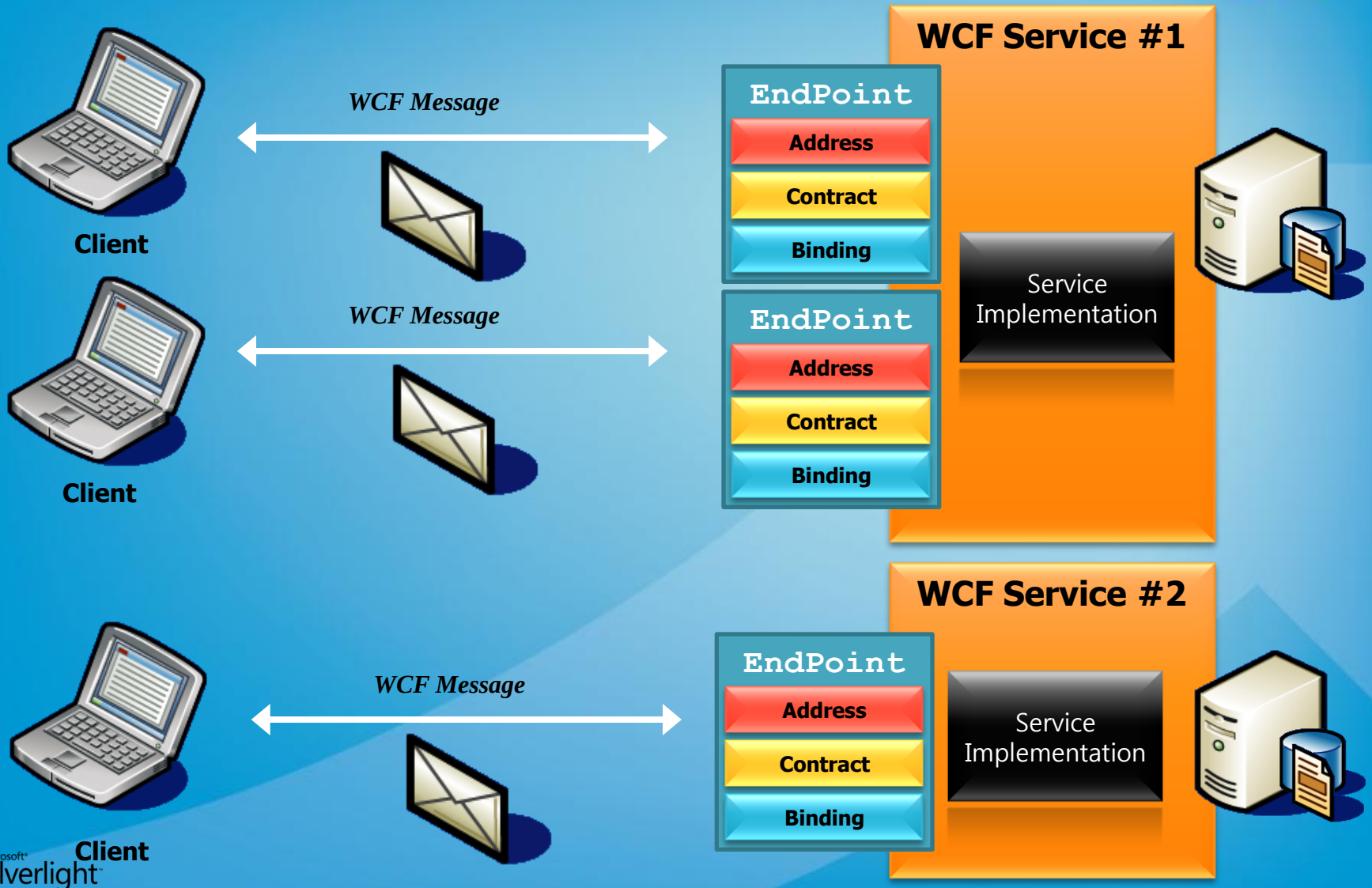
Microsoft Silverlight

Microsoft

WCF Programming Model (1)

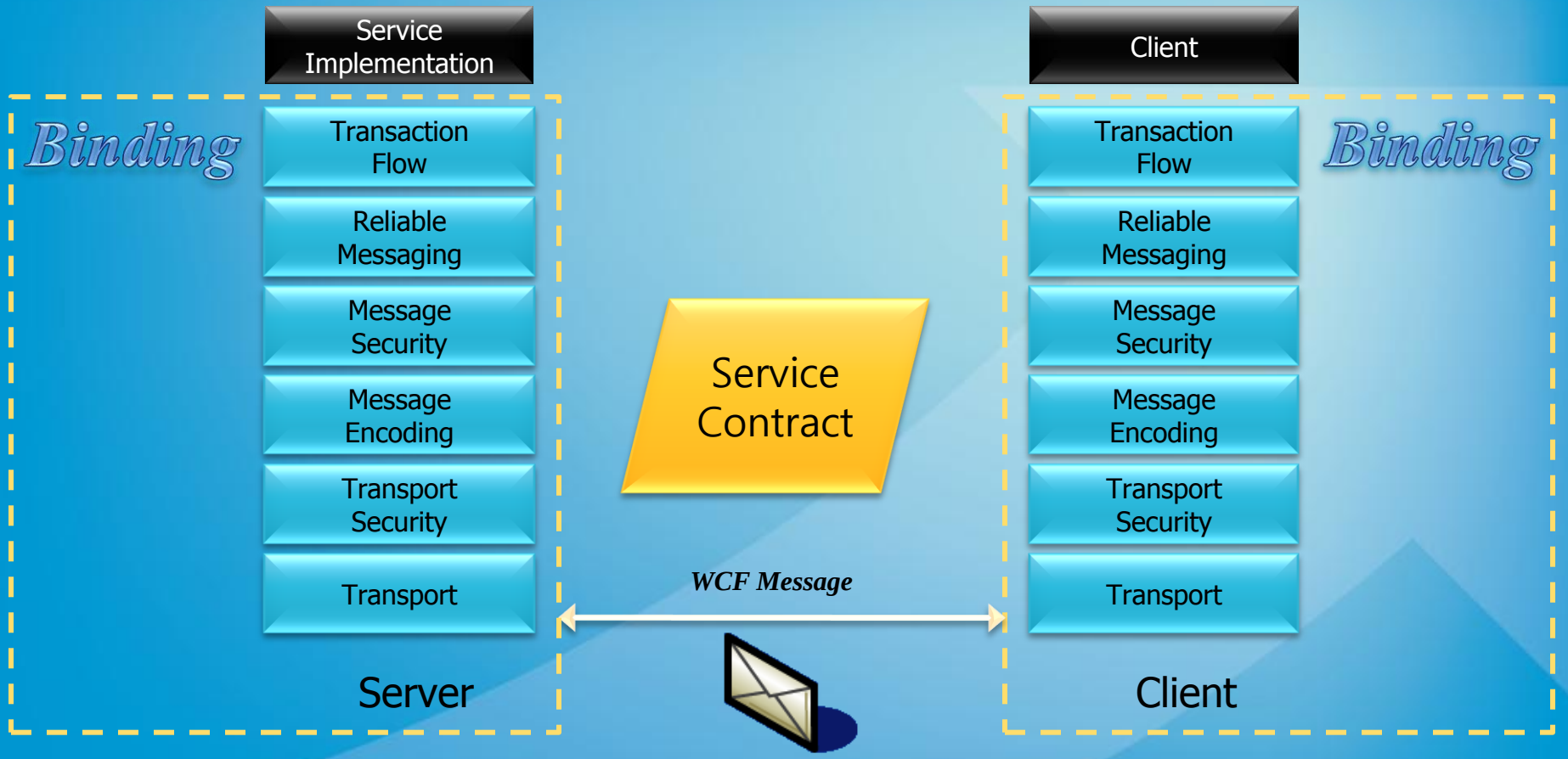
- Basic Concept
  - WCF Endpoint (ABC of WCF Programming)
    - **A**ddress
      - 서비스를 액세스하기 위한 주소
      - Transport에 따라 달라짐
        http://....., **net.tcp**://....., **net.pipe**://....., **net.msmq**://....., **net.p2p**://.....
    - **B**inding
      - 서비스 호출에 사용되는 Transport 프로토콜, 인증, 암호화, 메시지 인코딩, 세션 여부 등
      - Binding에 의해 Address 형식이 결정되곤 함
      - Custom Binding 제공도 가능
    - **C**ontract
      - 서비스에 대한 인터페이스
      - 서비스의 메쏘드(Operation Contract)
      - 관련 데이터 타입들(Data Contract)
  - 하나의 WCF 서비스는 다수의 Endpoint를 가질 수 있음

WCF Programming Model (2)



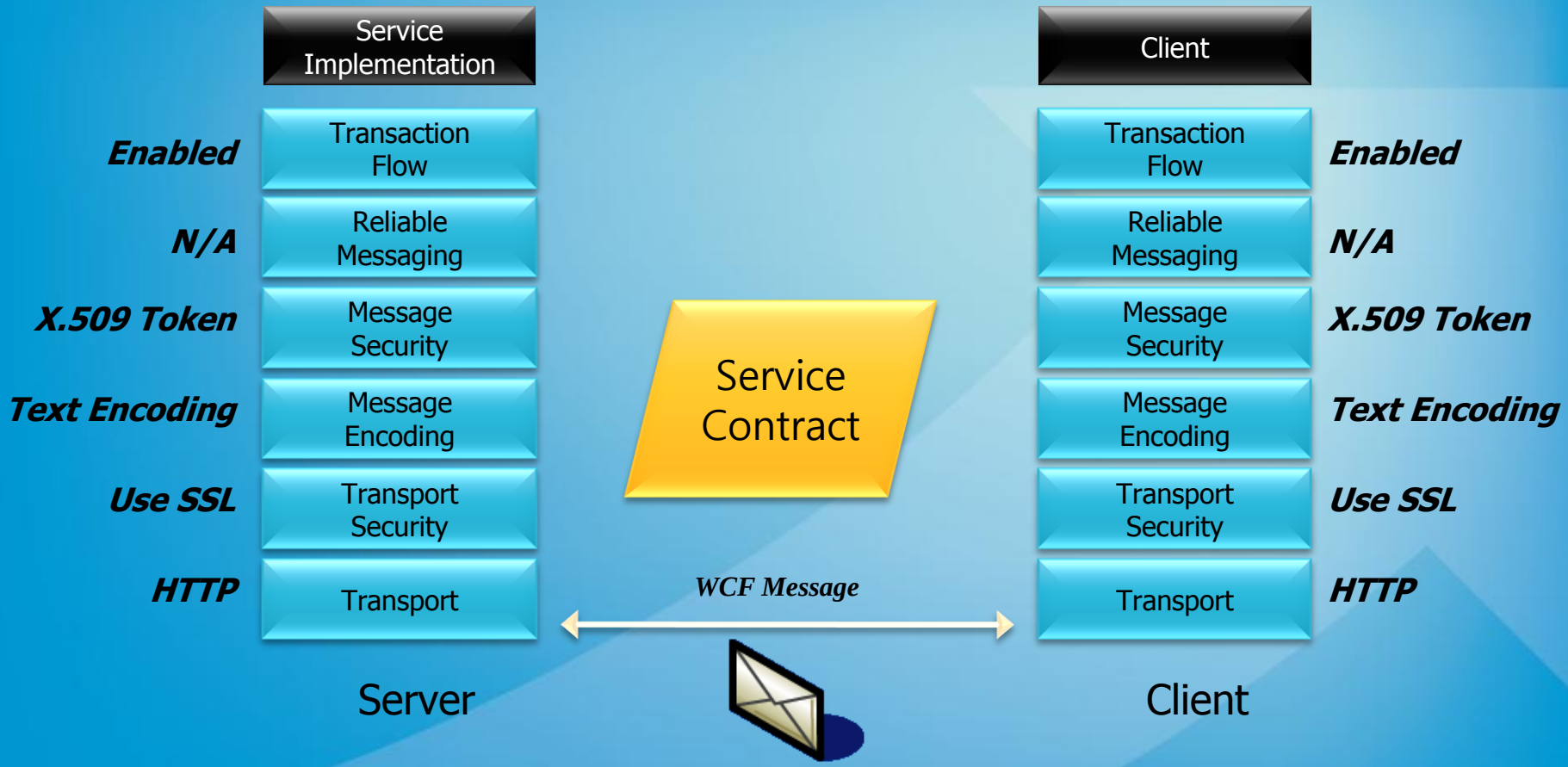
WCF Programming Model (3)

Concept of Binding



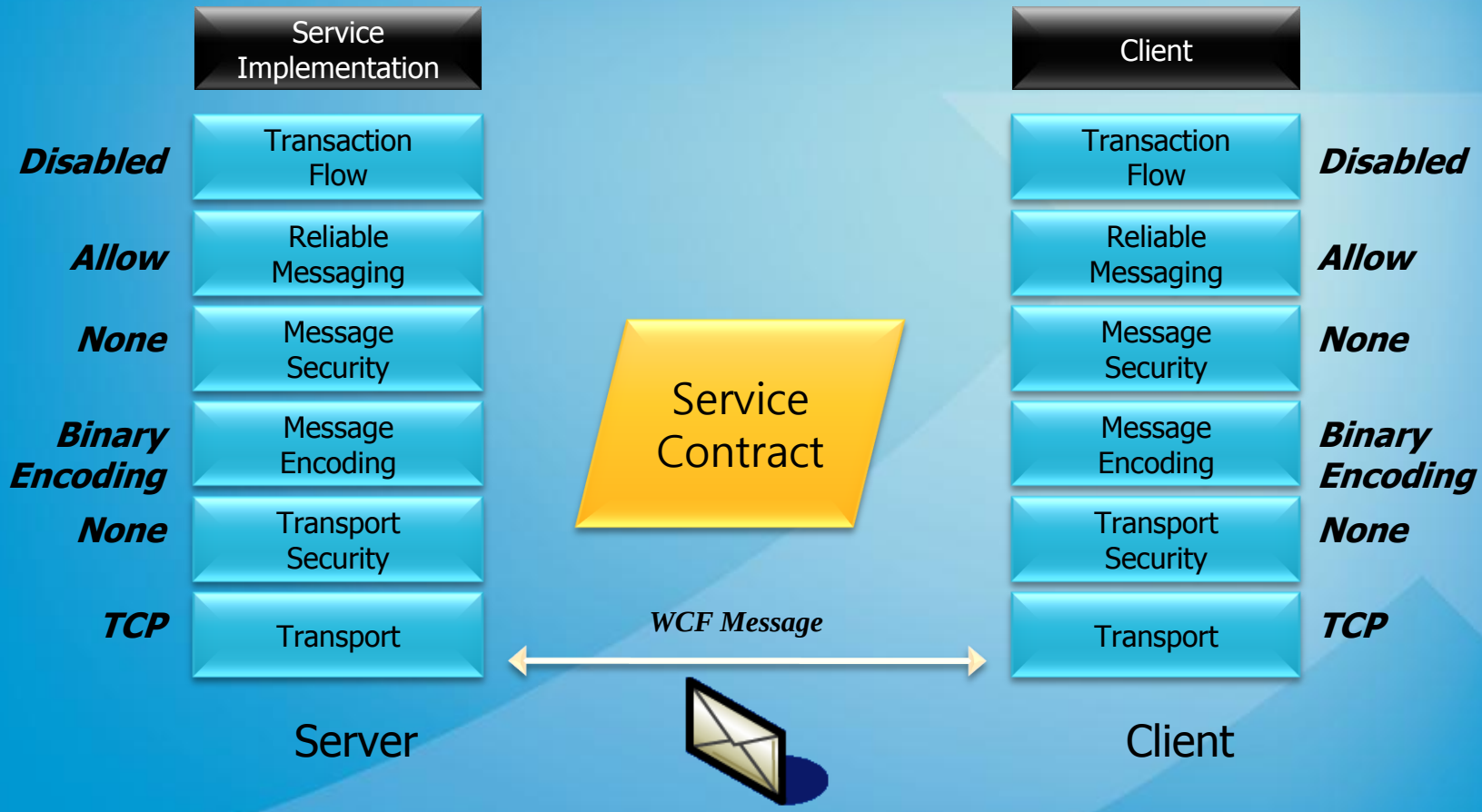
WCF Programming Model (4)

Example of WSHttpBinding



WCF Programming Model (5)

Example of NetTcpBinding



Writing WCF Service (1)

- 서비스 계약 정의
  - WCF 서비스의 인터페이스 정의
  - 닷넷 인터페이스에 ServiceContract 특성 명시
  - 인터페이스 메쏘드 마다 OperationContract 특성 명시
- 데이터 계약 정의(optional)
  - Operation에 의해 주고 받을 데이터 타입에 대한 정의
  - 데이터 타입에 DataContract, DataMember 특성 명시

```
[ServiceContract]
```

```
[DataContract]
class TransferInfo {
    string from_account_id;
    . . . . .

    [DataMember]
    public string From
    {
        get { return from_account_id; }
        set { from_account_id = value; }
    }

    . . . . . .
}
```

Writing WCF Service (2)

- 서비스 타입 구현
  - 서비스 인터페이스에 대한 구현
  - 서비스 인터페이스를 구현하는 임의의 클래스
- 서비스 호스트 구현
  - 서비스에 대한 호스트 객체
  - 서비스 호출을 리스닝하고 메시지를 서비스에 전달

```
class BankingImpl : IBanking
{
```

```
string address = "http://workman/wcf/example/Banking.svc";

// 서비스 호스트 객체 생성 (서비스 타입, 서비스 기본 주소를 매개변수로 전달)
ServiceHost host = new ServiceHost(typeof(BankingImpl), new Uri(address));
host.AddServiceEndpoint(
    typeof(IBanking),                   // service contract
    new BasicHttpBinding(),             // binding (http & WS-I profile)
    "");                                // address (use base address)

host.Open();        // 서비스 리스닝 시작 ! (비동기적으로 작동함)

// 다른 작업을 수행하는 등의 작업 수행

host.Close();
```

Writing WCF Client (1)

- ServiceEndpoint 구성`
  - 호출하고자 하는 WCF 서비스의 Contract, Binding, Address 를 명시
- Channel 객체 구성
  - ChannelFactory 객체 생성 및 Channel 생성
- 서비스 호출!

```
string address = "http://workman/wcf/stepbystep/banking.svc";
ServiceEndpoint ep = new ServiceEndpoint(
    ContractDescription.GetContract(typeof(IBanking)),    // service contract
    new BasicHttpBinding(),                               // binding
    new EndpointAddress(address));                        // address

ChannelFactory<IBanking> factory = new ChannelFactory<IBanking>(ep); // get channel factory

IBanking proxy = factory.CreateChannel();  // get service interface
proxy.Deposit("123-45-678-1234", 3000.0);

TransferInfo info = new TransferInfo();
info.From = "123-45-678-1234";
info.To = "987-65-432-1234";
info.Amount = 2000.0;
proxy.Transfer(info);

factory.Close();
```

Writing WCF Client (2)

- Proxy 코드 자동 생성
  - Using svcutil.exe
    - WCF 클라이언트 프록시 생성 유틸리티 (MSDN 참조)
  - Using Visual Studio 2005/2008
    - Add Service Reference 메뉴 사용
  - 반드시 proxy를 Dispose 해주어야 함 (using 사용 권장)
- Configuration에 의해 서비스 종점(Endpoint) 정보 생성
  - Web Service 프록시와 매우 유사
  - Binding에 대한 다양한 설정 포함
- Tip
  - 기본적으로 WCF는 HTTP에 의해 WSDL을 조회를 금지하고 있음 (보안 문제)
  - 서버에서 명시적으로 이를 해제 해야 함.
    - 코드 혹은 Configuration 이용

Using Service/Client Configuration

- Using Configuration
  - Service Endpoint에 대한 설정을 configuration으로 설정 가능
  - <system.ServiceModel> 섹션 사용
  - Service 관련 설정 및 Client 관련 설정

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <system.serviceModel>
        <bindings>
            <wsHttpBinding>
                <binding name="WSHttpBinding_IBanking"  . . . . . >
                    . . . . . .
                </binding>
            </wsHttpBinding>
        </bindings>
        <client>
            <endpoint address="http://workman/wcf/example/Banking.svc"
                binding="wsHttpBinding" bindingConfiguration="WSHttpBinding_IBanking"
                contract="WCFClient.BankingService.IBanking" name="WSHttpBinding_IBanking">
                <identity>
                    <userPrincipalName value="Workman\ksyu33" />
                </identity>
            </endpoint>
        </client>
    </system.serviceModel>
</configuration>
```

Programming WCF

msdn

DEMO

Microsoft Silverlight

Microsoft

Conclusion

- Evolution of Network Communication
  - 새로운 요구사항을 수용하면서 보다 편리한 개발 환경을 제공하는 방향으로 진화되어 왔음
  - 최근에는 XML 기반의 상호운영성이 뛰어난 통신 방법 선호
- Windows Communication Foundation
  - XML 웹 서비스 기반의 뛰어난 상호운영성
  - 진보된 메시징, 보안, 트랜잭션, 신뢰성, P2P 등 새로운 기능
  - 프로토콜에 무관한 통합된 단일 프로그래밍 모델
    - 기존 ASMX, WSE, .NET Remoting, COM+ 프로그래밍 모델의 장점들 모두 수용
    - 간결한 구현 방식
  - 다양하고 풍부한 프로그래밍 환경
- WCF : Microsoft의 차세대 커뮤니케이션 인프라

Q&A

- 참고 자료
  - MSDN
  - MSDN Magazine
  - Windows Vista Developer Story
  - 월간 마이크로소프트웨어 2006년 10월호 Inside Developer 칼럼
  - 월간 마이크로소프트웨어 2006년 11월호 Inside Developer 칼럼
  - 월간 마이크로소프트웨어 2006년 12월호 Inside Developer 칼럼
  - 월간 마이크로소프트웨어 2007년 2월호 Inside Developer 칼럼
  - Windows SDK v6.0 Samples
- Contact information
  - http://www.simpleisbest.net
  - ksyu33 at theonetech.co.kr
  - ksyu33 at korea.com