Windows Communication Foundation

Manu Cohen-Yashar
CTO Applisec
Sela Group
It is not only a client server issue

More !!!
Distributed Systems Challenges

- Distributed Transactions
- Security
  - B2B
  - Enterprise
  - Web ...
- Versioning – The possibility to change!
- Interoperability
- Performance
- Separation between logic and distribution technology.
- Extensibility
WCF

The unified framework for rapidly building service-oriented applications
Unified Programming Model

- **ASMX**: Interop with other platforms
- **Attribute-Based Programming**: Enterprise Services
- **WS-* Protocol Support**: WSE
- **.NET Remoting**: Extensibility Location transparency
- **Message-Oriented Programming**: System.Messaging
WS-* Protocol Support

Security
Reliable Messaging
Transactions
Metadata
Messaging
XML
Services and Clients

Client

Message

Message

Service
Endpoints

Client

Message

Service

Endpoint

Endpoint
Address, Binding, Contract

Client

Service

Message

Address (Where)

Binding (How)

Contract (What)
Elements of Binding

- **Transport**
  - TCP
  - HTTP
  - TCP
  - Pipes
  - MSMQ
  - Custom

- **Encoder**
  - Binary
  - Text

- **Security**
  - WS-*

- **Reliability**
  - WS-*

- **Protocol**
  - WS-*

- **Custom**
  - MTOM
  - Custom
  - Custom
  - HTTP
  - Binary
  - Binary
  - MTOM
  - Custom
  - Custom
  - Custom
  - Custom
The bigger picture
WCF Security Model
Messaging Security Requirements

- Confidentiality
- Integrity
- Authentication
- Authorization
- Auditing
WCF Security Model

- Based on credentials and claims
- Can satisfy security requirements
- Secure by default
- Consistent across bindings
- Consistent across credentials
Authorization

- Normal .Net Authorization using existing CLR constructs
- Claims-based model known as *Identity Model*.

![Security Token Diagram]

- Claim 1
- Claim 2
- ...
- Claim n
The Identity model

- Claims-based system
- Claims describe the capabilities associated with some entity in the system.
- Claims are used to gain access to resources. (Like a key)
- WCF Create claims from incoming messages.
- Example: a claim of type "File", with right "Read" over the value "Biography.doc"
Claims are always issued by some entity in the system.

Claims are grouped together as a set and each set has an issuer.

An issuer is just a set of claims.
Authorization Policies

Claims are generated as part of the process of evaluating the authorization policy.

Choose to add additional claims based on the claims already present.

For example: If you have a claim identifying you as a student, the policy will give you the claim permitting you to use the library.

A given authorization policy may need to be evaluated multiple times.
Authorization Context

- An authorization manager evaluates the various authorization policies
- The result is an authorization context
- The authorization context can be examined to determine what claims are present in that context.
WCF Security out of the box
Transport Security

- Security requirements satisfied at transport layer

Advantages
- Performance benefits
- Common implementation

Disadvantages
- Restricted claim types
- No security off the wire
Transport Security

<endpoint address="https://localhost/calculator"
    binding="basicHttpBinding"
    bindingConfiguration="Binding1"
    contract="ICalculator" />

Only the server certificate will be used (server authentication)

<basicHttpBinding>
    <binding Name="Binding1">
        <security mode="Transport">
            <transport clientCredentialType="None"/>
        </security>
    </binding>
</basicProfileBinding>
Transport Security technology

- Depend on the binding and transport being used
  - WsHttpBinding – Https (Default)
  - NetTcpBinding – TLS (Default)
  - BasicHttpBinding – None (Default)
    - Can be configured to:
      - Basic
      - Certificate
      - Digest
      - NTLM
      - Windows
Message Security

- Security requirements satisfied at message layer

Advantages
- More credential types
- Extensible
- End-to-end security

Disadvantages
- Standards and usage still solidifying
- Performance impact
Message Security technology

- Depend on binding
  - `wsHttpBinding` for example is using Windows Kerberos token as a default token.
  - You can set the token type (next slides)
  - You can set encryption and digital signatures order
<endpoint address="http://localhost/calculator"
    binding="wsHttpBinding"
    bindingConfiguration="Binding1"
    contract="ICalculator" />

<wsWithHttpBinding>
    <binding Name="Binding1">
        <security mode="Message">
            <message clientCredentialType="Windows"/>
        </security>
    </binding>
</wsHttpBinding>
Mixed Mode

- Compromise between Transport and Message Security
- Transport layer satisfies integrity and confidentiality requirements
  - Performance benefits
- Message layer carries claims
  - Rich credentials, extensibility
<endpoint address="https://localhost/calculator"
    binding="wsHttpBinding"
    bindingConfiguration="Binding1"
    contract="ICalculator" />

<wsHttpBinding>
    <binding Name="Binding1">
        <security mode="TransportWithMessageCredential">
            <message clientCredentialType="Windows"/>
        </security>
    </binding>
</wsHttpBinding>
You can use different credentials types:
- Windows
- Username Password
- Certificate
- Issued Token (CardSpace…)

Credential types
Authentication Modes

You can use different authentication technologies:
- Windows
- Membership provider (ASP.NET)
- Custom
<bindings>
  <wsHttpBinding>
    <binding name="WSHttpBinding_manuCalc" >
      <security mode="Message">
        <message clientCredentialType="UserName" />
      </security>
    </binding>
  </wsHttpBinding>
</bindings>
<behaviors>
  <serviceBehaviors>
    <behavior name="MyBehavior">
      <serviceAuthorization
        principalPermissionMode="UseAspNetRoles"/>
      <serviceCredentials>
        <userNameAuthentication
          userNamePasswordValidationMode="MembershipProvider"/>
        <serviceCertificate
          storeLocation="LocalMachine"
          storeName="My"
          findValue="CN=WSE2QuickStartServer"
          x509FindType="FindBySubjectDistinguishedName"/>
      </serviceCredentials>
    </behavior>
  </serviceBehaviors>
</behaviors>
Service Certificate

Service Certificate must be set to enable server authentication and the safe transfer of client credentials.

The automatic proxy created in the client will include (in the config file) a reference to this certificate so the client will be able to encrypt its credentials using the public key.
Console.WriteLine("Enter username[domain\user]:");
string username = Console.ReadLine();
Console.WriteLine("Enter password:");
string password = Console.ReadLine();

CalculatorProxy proxy = new CalculatorProxy();
proxy.Credentials.UserName.UserName = username;
proxy.Credentials.UserName.Password = password;

//When using channel factory
ChannelFactory<ICalc> chf = new ChannelFactory<ICalc>(binding,RemoteAdd);
proxy = chf.CreateChannel();
proxy.ChannelFactory.Credentials.UserName.UserName = username;
wsHttpBinding
1. Windows Authentication
2. Authentication using ASP.NET membership provider
3. ClientCredentialType="Certificate"
questions?