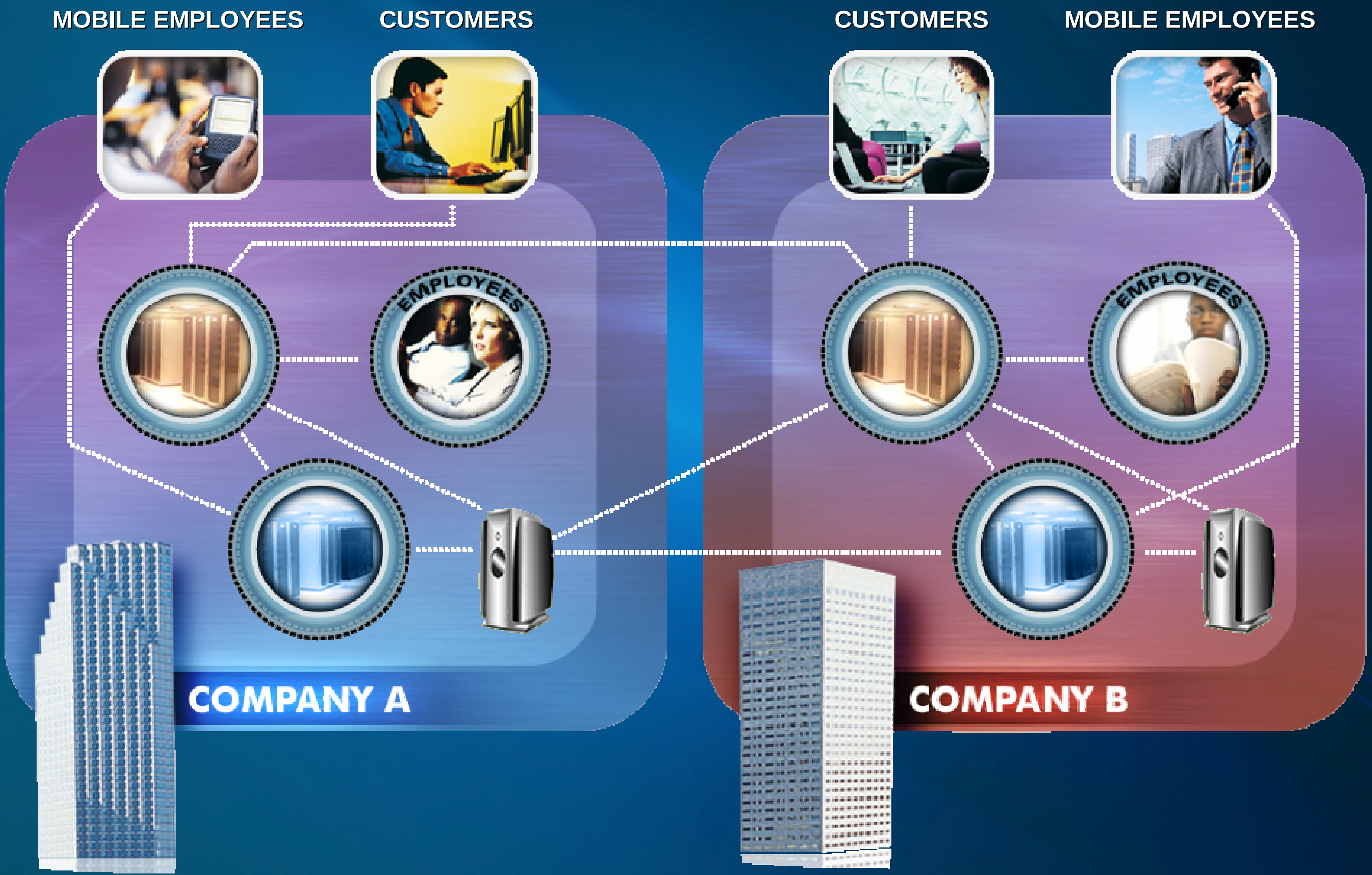
Windows Communication Foundation

Microsoft .net Framework

Version 3.0

Manu Cohen-Yashar
CTO Applisec
Sela Group

The Imperative to Connect



It is not only a client server issue

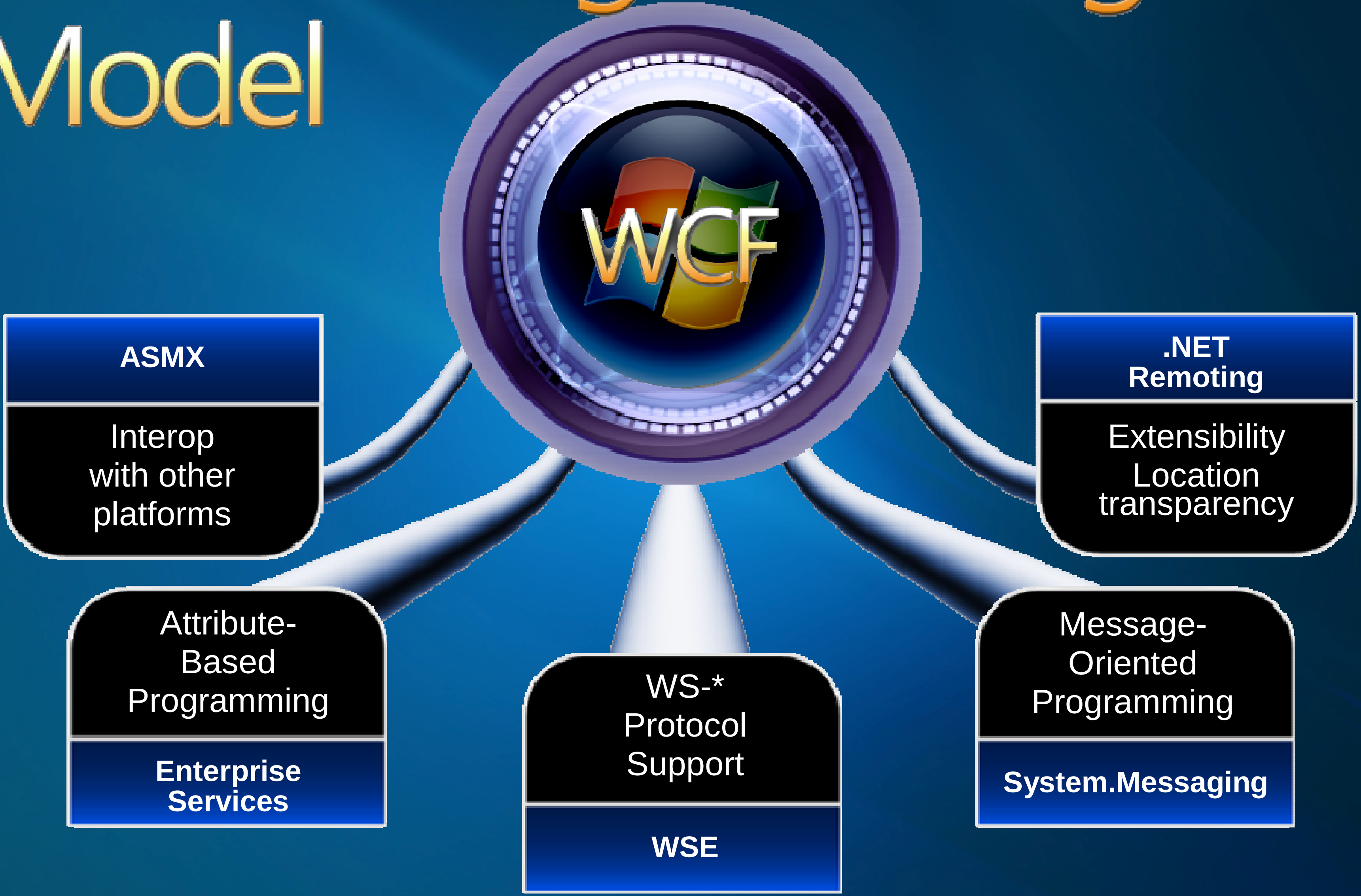
Distributed Systems Challenges

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

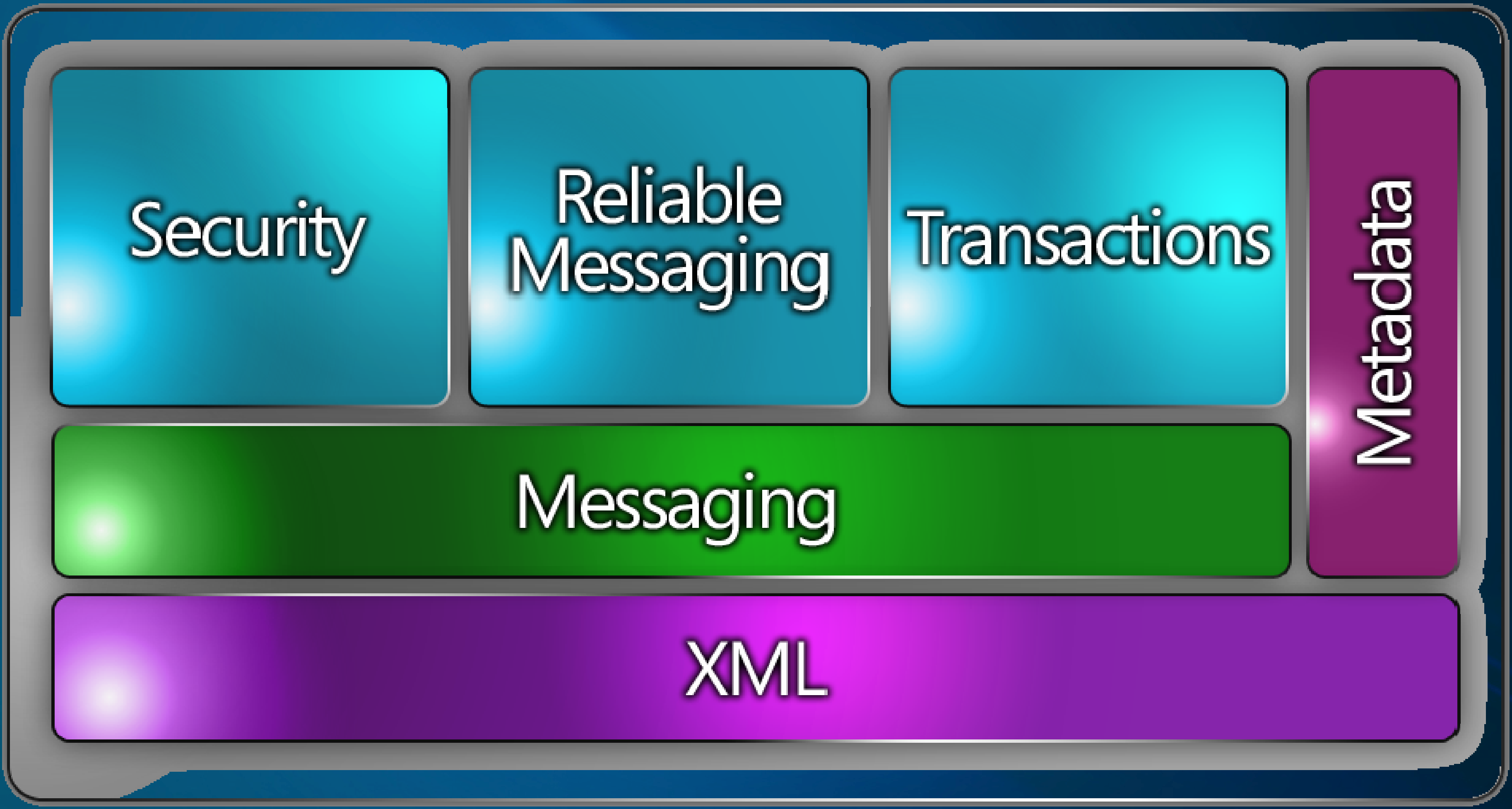
WCF

The unified framework
for rapidly building
service-oriented applications

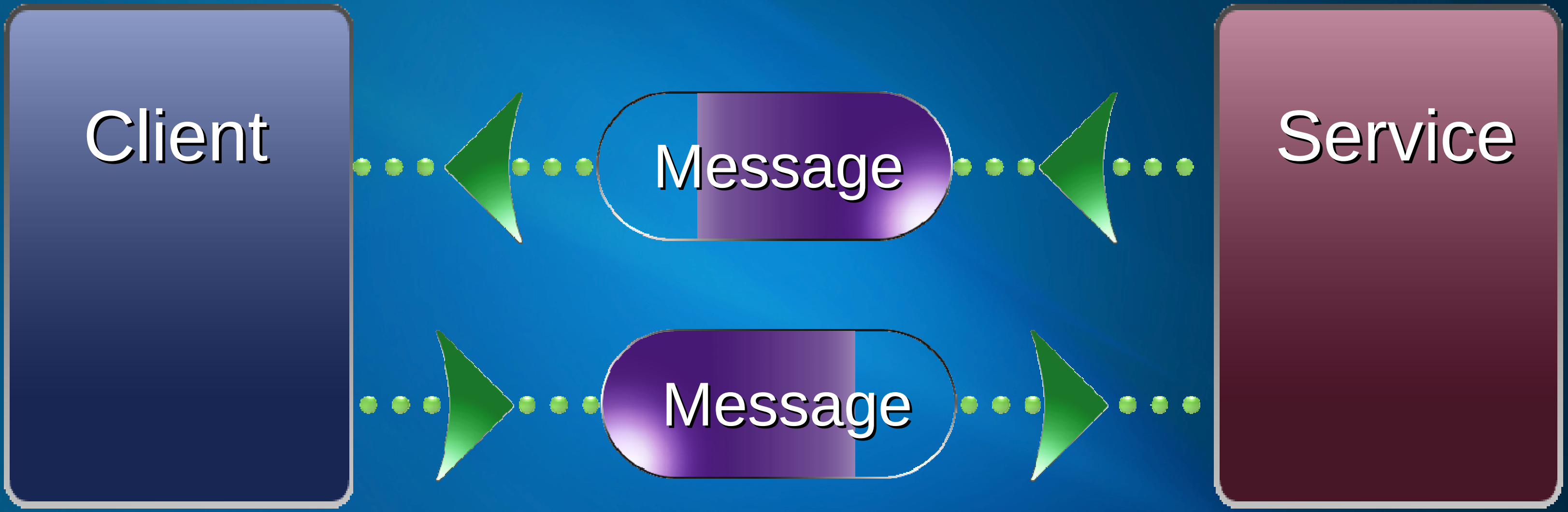
Unified Programming Model


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

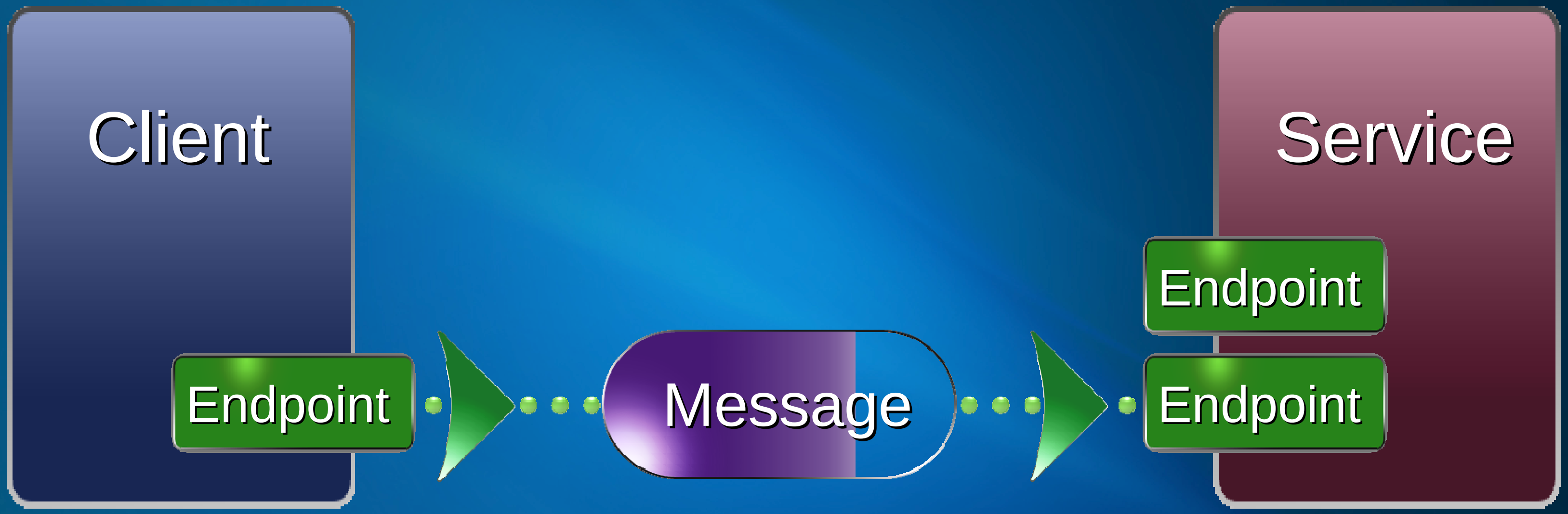
WS-* Protocol Support



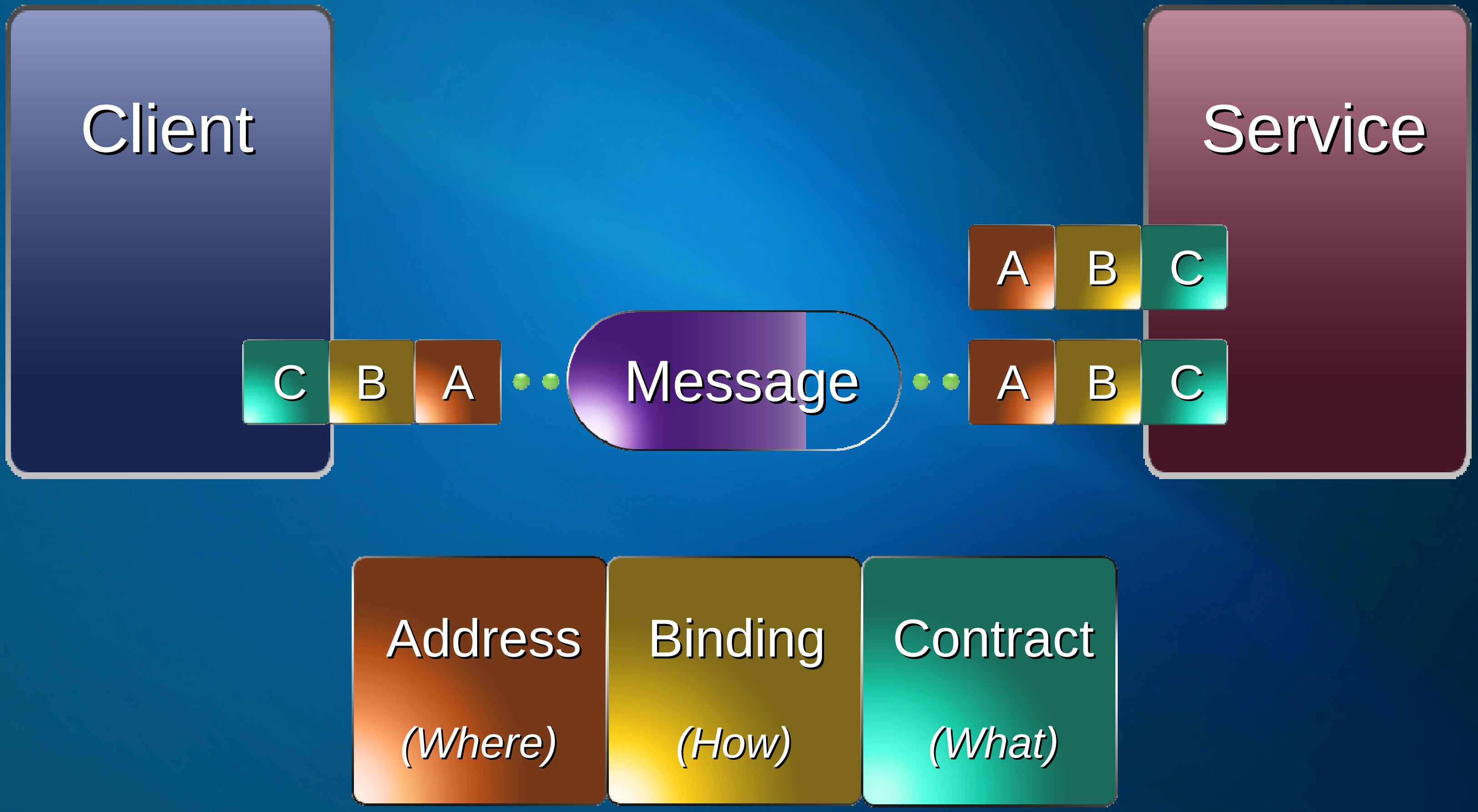
Services and Clients



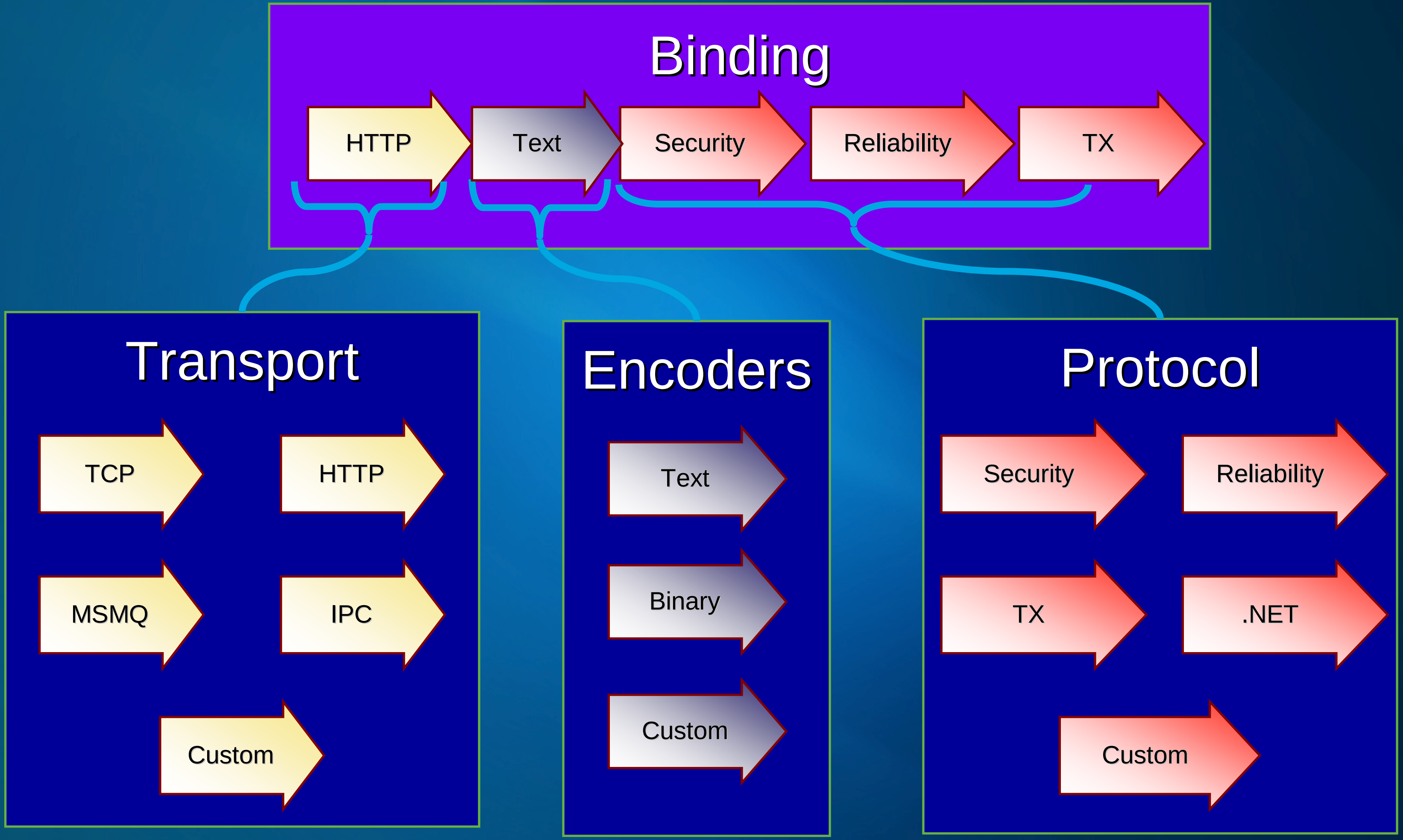
Endpoints



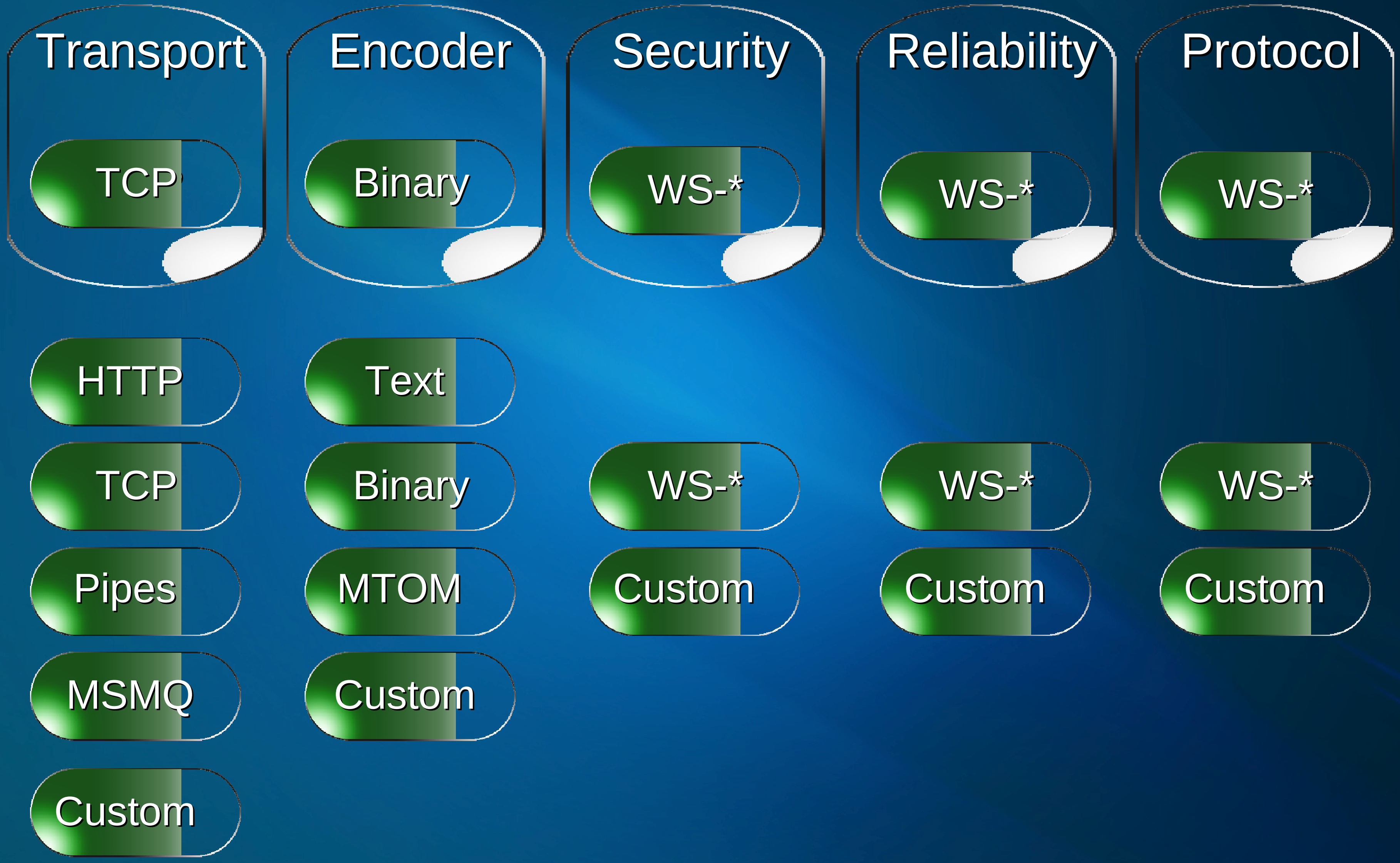
Address, Binding, Contract



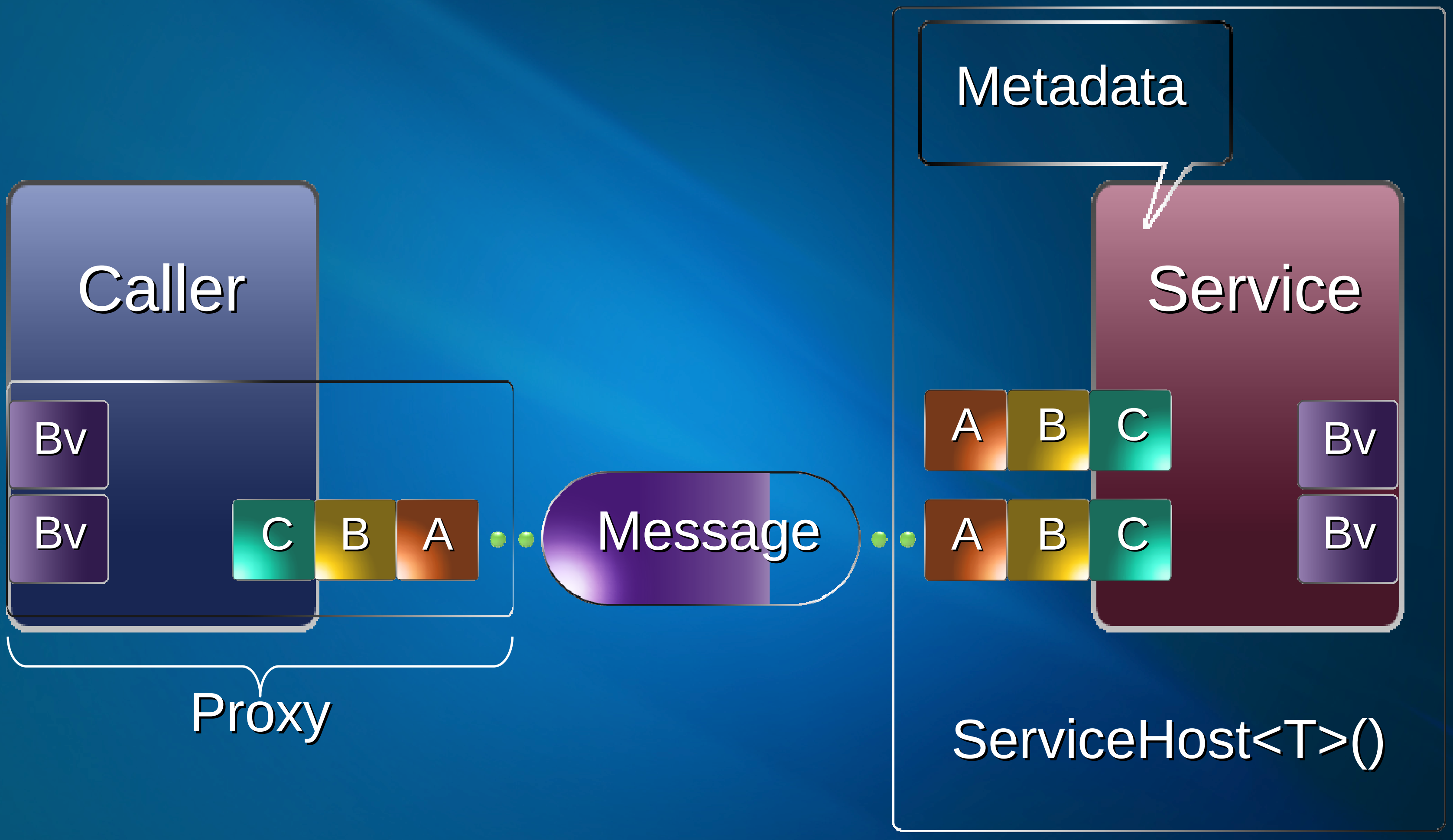
Bindings & Binding Elements



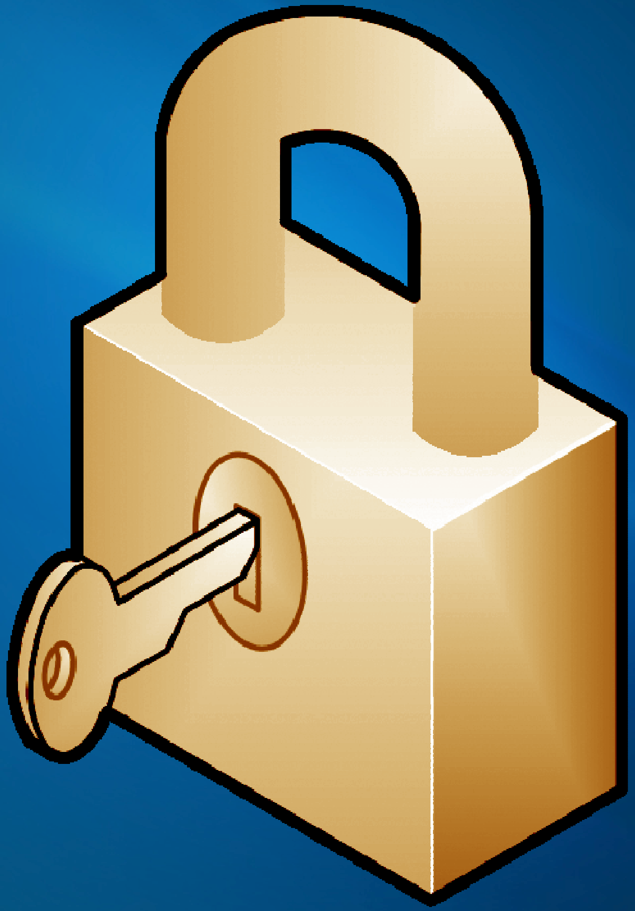
Elements of Binding



The bigger picture

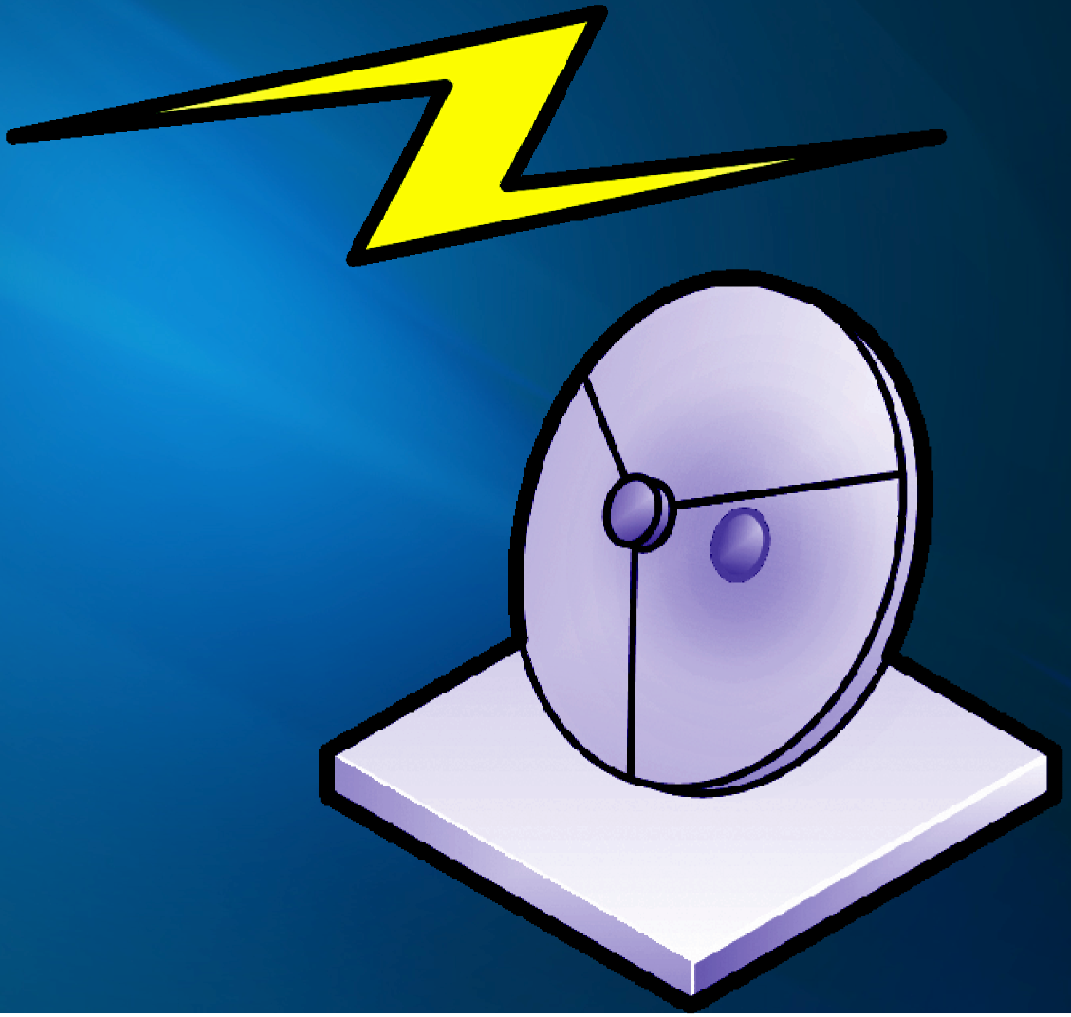


WCF Security Model



Messaging Security Requirements

- Confidentiality
- Integrity
- Authentication
- Authorization
- Auditing

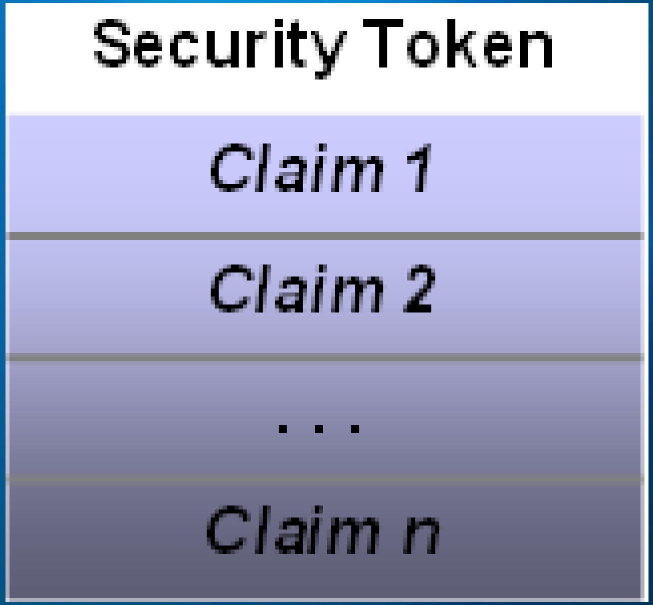


WCF Security Model

- 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*.

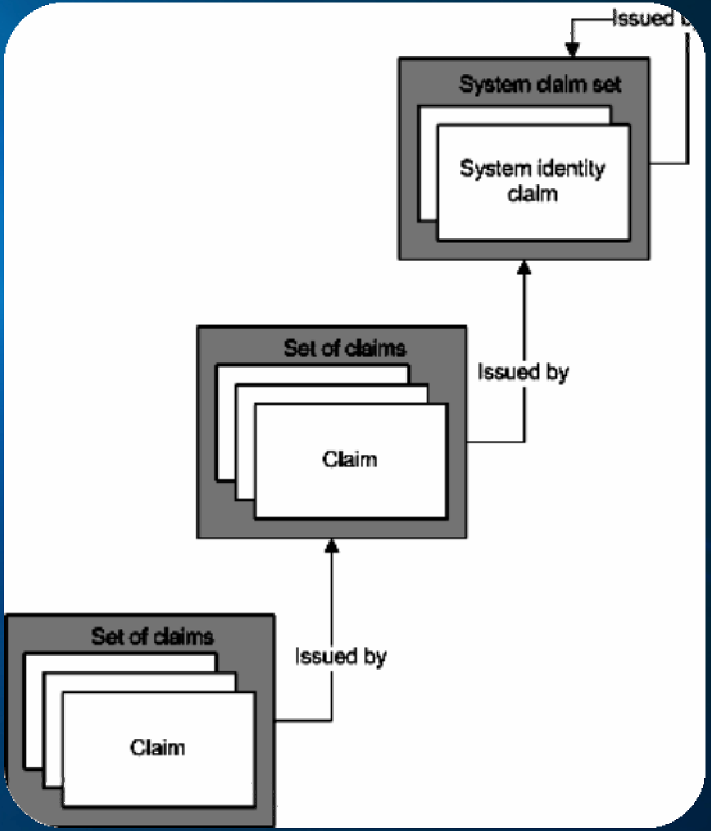


The Identity model

- 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 issuer

- 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.

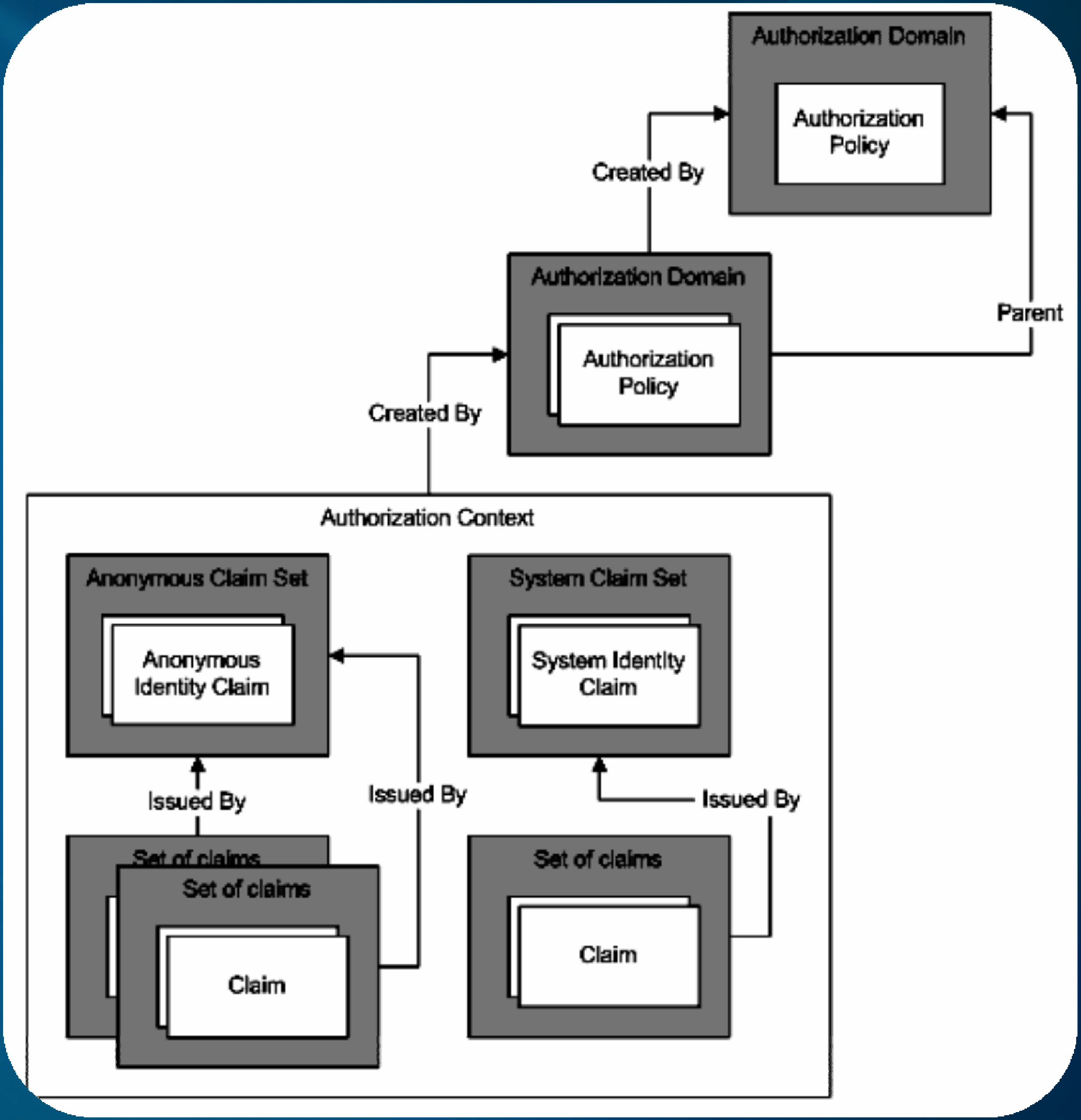


Authorization Policies

- 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.


Authorization Domain
Authorization Policy
Created By
Authorization Domain
Authorization Policy
Parent
Created By
Authorization Context
Anonymous Claim Set
Anonymous Identity Claim
System Claim Set
System Identity Claim
Issued By
Issued By
Issued By
Set of claims
Set of claims
Claim
Set of claims
Claim

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

Message Security

```
<endpoint address=“http://localhost/calculator"
                  binding=“wsHttpBinding“
                  bindingConfiguration=“Binding1”
                  contract="ICalculator" />



<wsHttpBinding>
  <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

Mixed Mode Security

```
<endpoint address=“https://localhost/calculator"
                  binding=“wsHttpBinding“
                  bindingConfiguration=“Binding1”
                  contract="ICalculator" />



<wsHttpBinding>
  <binding Name="Binding1">
      <security mode="TransportWithMessageCredential">
          <message clientCredentialType=“Windows"/>
      </security>
  </binding>
</wsHttpBinding>
```

Credential types

- You can use different credentials types:
  - Windows
  - Username Password
  - Certificate
  - Issued Token (CardSpace…)

Authentication Modes

- You can use different authentication technologies:
  - Windows
  - Membership provider (ASP.NET)
  - Custom

Credentials Type

```
<bindings>
      <wsHttpBinding>
        <binding name="WSHttpBinding_manuCalc" >
          <security mode="Message">
            <message clientCredentialType="UserName" />
          </security>
        </binding>
      </wsHttpBinding>
</bindings>
```

Authentication Mode

```
<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.

Username/Password

```
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;
proxy.ChannelFactory.Credentials.UserName.Password = password;
```

demo

wsHttpBinding

1. Windows Authentication
2. Authentication using ASP.NET membership provider
3. ClientCredentialType="Certificate"

questions?