Securing Web Services Using Microsoft Web Services Enhancements 1.0

Petr PALAS

PortSight Software Architect

petrp@portsight.com

www.PortSight.com

Agenda

- What is WSE and Its Relationship to GXA
- Standards Currently Supported in WSE
- WSE Architecture
- Implementation of WS-Security in WSE
- Demo of Using WSE
- Future Standards

What is WSE and Its Relationship to GXA

- **Microsoft Web Services Enhancements 1.0**
- Replaces Microsoft WSDK
- Free to download and supported by Microsoft
- .NET library for easy use of higher-level WS standards from GXA (that are based on XML, SOAP, XSD and WSDL)
- It also contains samples, documentation and tools
- It will be updated according to GXA standards

Standards currently supported

- **Emerging standards in WSE:**
  - WS-Security (Security Credentials, Digital Signing, Encryption)
  - WS-Routing
  - WS-Attachments and DIME
- **Other functionality:**
  - Writing your own filters
  - Diagnostic features

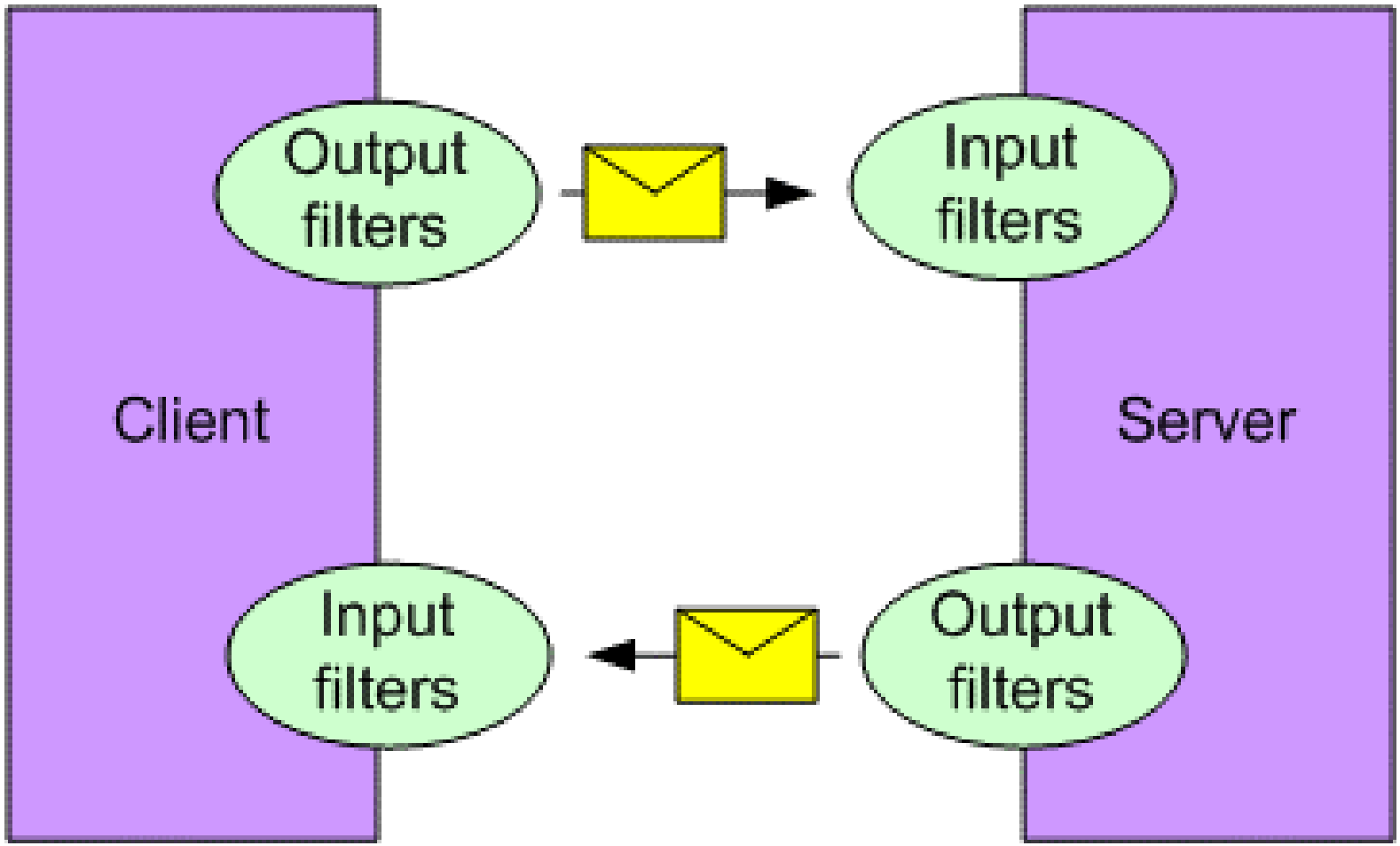
Security Features in WSE

- **Digital signature of a SOAP message**
  - X.509 certificates / username+password / custom binary token
- **Encryption of a SOAP message**
  - X.509 / shared secret / custom binary token
- **Authentication using security credentials**
  - X.509 / username+password / custom binary token

WSE Architecture

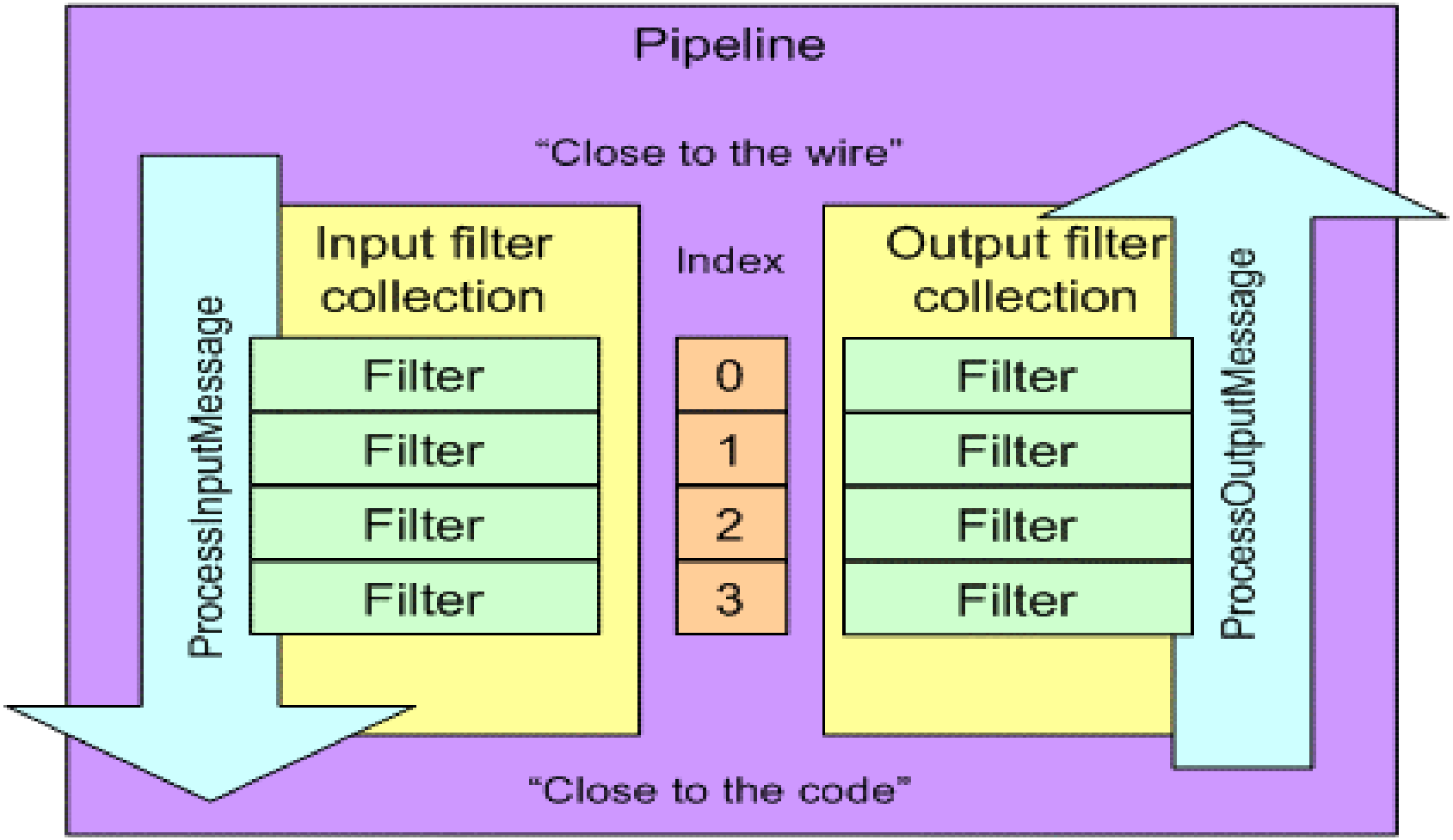
- Additional information is added to the SOAP headers using filters
- Built-in filters:
  - Trace Filter (diagnostics)
  - Security Filter (WS-Security)
  - Timestamp Filter (WS-Security)
  - Referral Filter (WS-Referral)
  - Routing Filter (WS-Routing)
- You can write yours

WSE Filters



Source: http://msdn.microsoft.com

WSE Filter Pipeline



Source: http://msdn.microsoft.com

How To Use WSE (1)

- Create client and server projects in VS.NET
- Add reference to Microsoft.Web.Services.dll library to both server and client
- Add following lines to both server and client code (VB.NET):

  imports Microsoft.Web.Services

  imports Microsoft.Web.Services.Security

How To Use WSE (2)

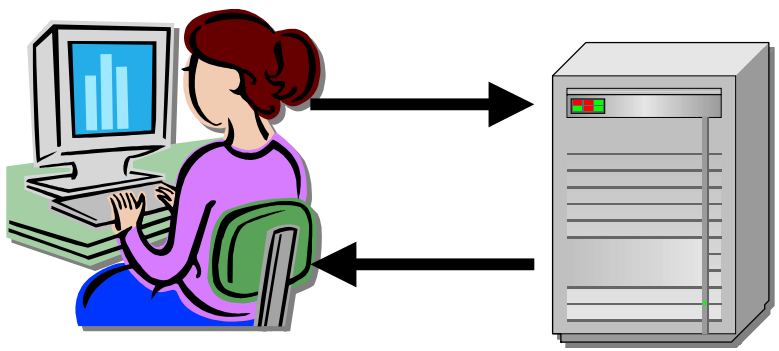
- Edit web.config (see documentation) on server
- Write and compile server code
- Add Web Reference to the client program
- Edit generated proxy class – it must inherit from Microsoft.Web.Services.WebServicesClientProtocol
- Write and run client code
- You can use WSE 1.0 Settings (tool integrated into VS.NET) to edit the configuration files

Adding Security Credentials

- **Without WS-Security:** Use Credentials property of the proxy class (on the transport layer)
  - may be lost at intermediaries ☹
- **With WS-Security:** Add credentials to the SOAP message header
- **Supported types of security credentials:**
  - X.509 Certificate
  - User name and password (none/plain-text/hashed)
  - Custom-built binary token

DEMO – Authentication and Signing a Message

user name, password (user token)
signature (using user name and password)
timestamp



check signature
check timestamp
get user token
return valid password using custom PasswordProvider class
if provided password is the same as the one from the PasswordProvider, then execute the method

DEMO – Server Code (1)

```
<WebMethod()> Public Function getAmount()
  As String
Dim retval As String
Dim requestContext As SoapContext =
  HttpSoapContext.RequestContext
' Verify that a SOAP request was received
If IsNothing(requestContext) Then
            Throw (New
  ApplicationException(“Message is not
  acceptible"))
End If
```

DEMO – Server Code (2)

```
Dim userToken As UsernameToken =
  GetFirstUsernameToken(
      requestContext.Security)

If Not IsNothing(userToken) Then
  ‘return amount
  retval = 1234
End If

Return retval
```

DEMO – Client Code (1)

```
Dim userToken As UsernameToken = New
  UsernameToken(TextBox1.Text,
  TextBox2.Text,
  PasswordOption.SendPlainText)

Dim serviceProxy As localhost.Service1 =
  New localhost.Service1()

Dim requestContext As SoapContext =
  serviceProxy.RequestSoapContext
```

DEMO – Client Code (2)

```
‘add credentials
requestContext.Security.Tokens.Add(userToken)
‘create signature and add it to the message – not secure when sending also password (!) – this is just example
Dim sig As Signature = New Signature(userToken)
requestContext.Security.Elements.Add(sig)
requestContext.Timestamp.Ttl = 10000
‘show result
MsgBox(serviceProxy.getAmount())
```

Advantages & Limitations

- **Advantages:**
  - WS-Security isn't limited to point-to-point security; it supports routing and scaling
  - WSE provides easy to use libraries
- **Limitations:**
  - Since the standards are still under development, various implementations may not be compatible.
  - Recommended only for specific solutions.
  - You have to understand the security protocols.

Future Standards In Next Versions

- **WS-Policy**
- **WS-Trust**
- **WS-Privacy**
- **WS-SecureConversation**
- **WS-Federation**
- **WS-Authorization**

Used and Recommended Sources

- **Demo Source Code:**
  - http://www.portsight.com/technology
- **Microsoft Developer Network (WSE download):**
  - http://msdn.microsoft.com/webservices
- **OASIS:**
  - http://www.oasis-open.org/

Questions & Answers

Thank You for Your Time!