Clemens Vasters
Technischer Vorstand
clemensv@newtelligence.com

The XML Web Services X-Files
Episode II: Tales from the Labs

ASP.NET Bigotry – The Good

- Very simple: You just write classes and methods
  - Simple enough for everybody
  - Write, compile, done
- The WSDL contract is generated on the fly
- Wire format is rendered on the fly
- Proxies are automatically code-generated

ASP.NET Bigotry – The Bad

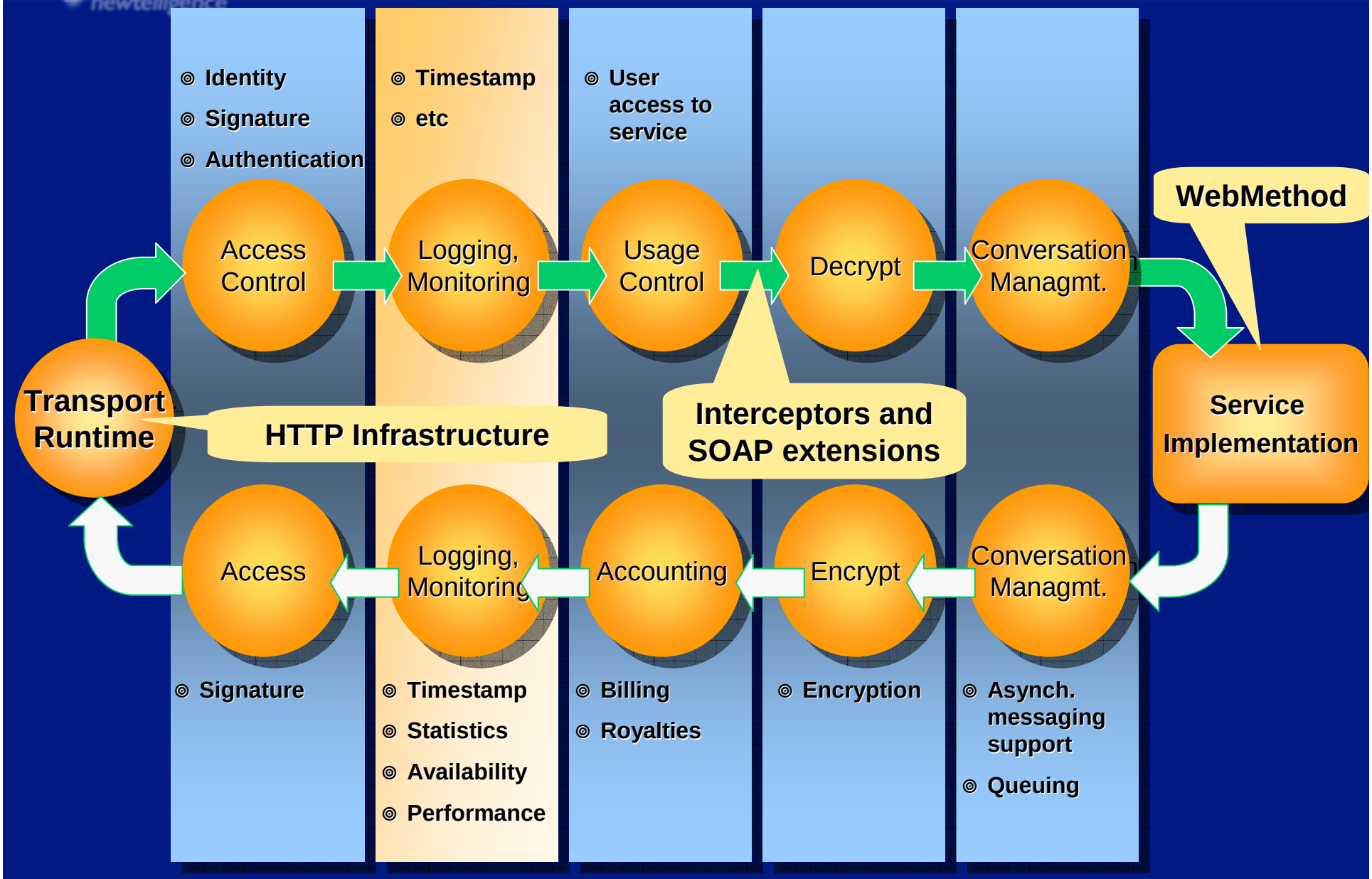
- Too simple: You just write classes and methods
  - Suggests that a Web Service is just like anything else
- The WSDL contract is generated on the fly
  - Contract should dominate the programming model
- Wire format is rendered on the fly
  - Somewhere in deep inside the infrastructure
- Proxies are automatically code-generated
  - Hard to maintain customized versions

The Middle Ground

- Many developers need simplicity
  - They have no time to deal with XML philosophy
  - They have no time to carefully craft Schema and WSDL
  - They have no time to even learn Schema or WSDL
  - They have even less time to deal with GXA, etc.
- Whether we like it or not ...
  - ... the programming model must be kept easy
  - ... it's hard to move from RPC to messages
- Bridging the gap is an infrastructure job.


```
[CheckArguments]
[TipTransaction]
[KerberosAuthentication][PrincipalPermission]
[ExceptionMonitor][MethodStatistics]
[WebMethod]
public string MakeReservation(
  [MinLength(8), MaxLength(8)] string FlightCycleId,
            [Between(1,100)] int Row,
          [Between('A','K')] char Seat,
           [Match(@"[A-Z]")] string Fare,
  [MinLength(1),MaxLength(80)] string FirstName,
  [MinLength(1),MaxLength(80)] string LastName,
        [OneOf("MR","MRS")] string Title,
[Match(@"[0-9]*"),MaxLength(20)] string CustomerCode)
```

Building A Pipeline



ASP.NET Runtime Environment

Host environment:
IIS with authentication infrastructure and additional ISAPI filters for compression, etc.

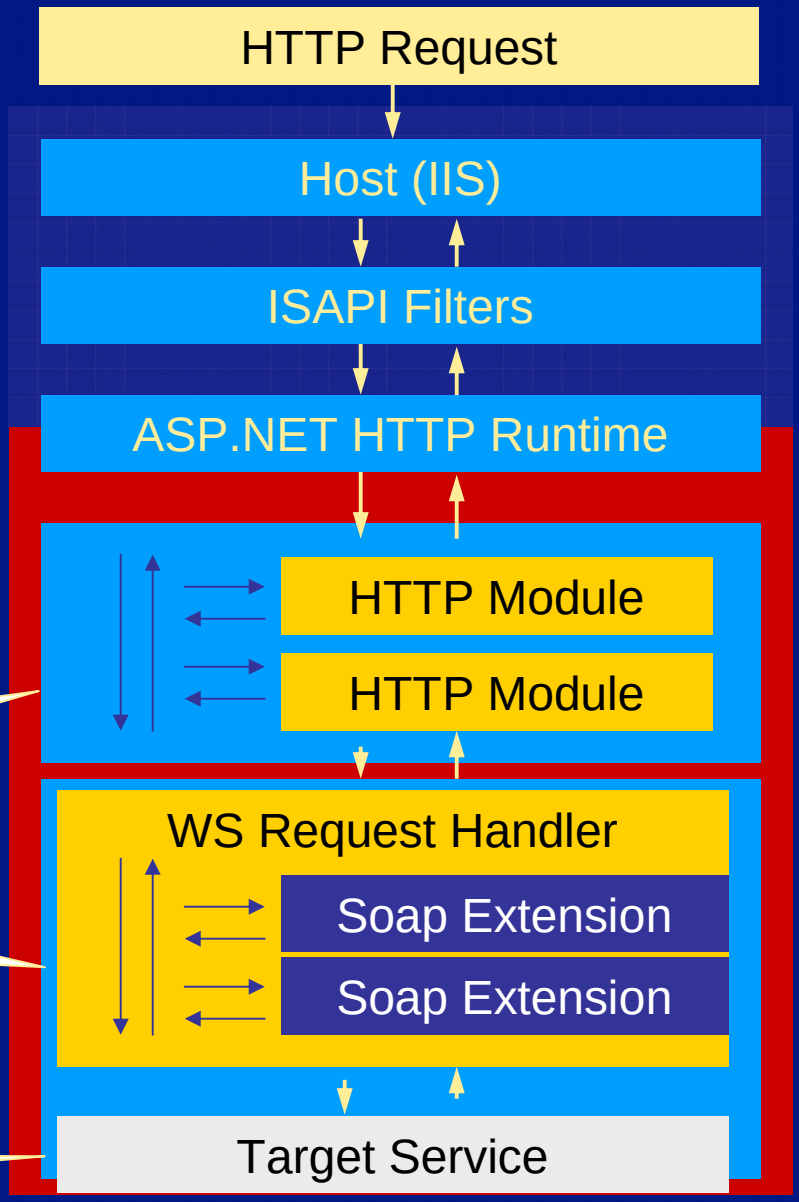
ASP.NET hooked into IIS via ISAPI Filter and Extension for .asmx extension (customizable)

ASP.NET: Managed runtime running as separate process. Processes IIS requests asynchronously. Doesn't tie up IIS threads.

HTTP Module chain allows for raw data filtering and processing. Modules are hooked in as callbacks.

Web Service Request Handler maps request to WebService class and method. SoapExtensions hooked in as callbacks.

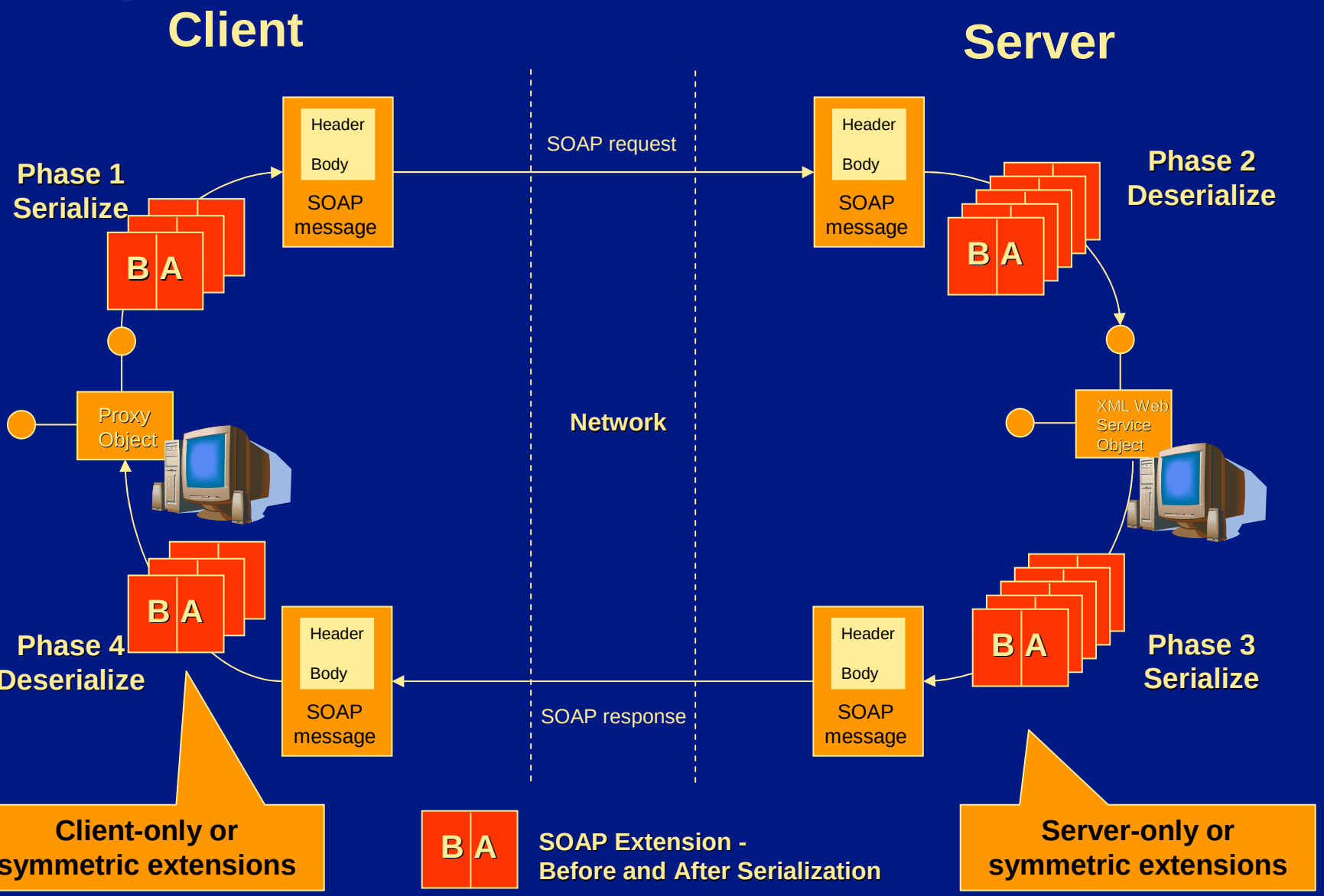
[SoapMethod]



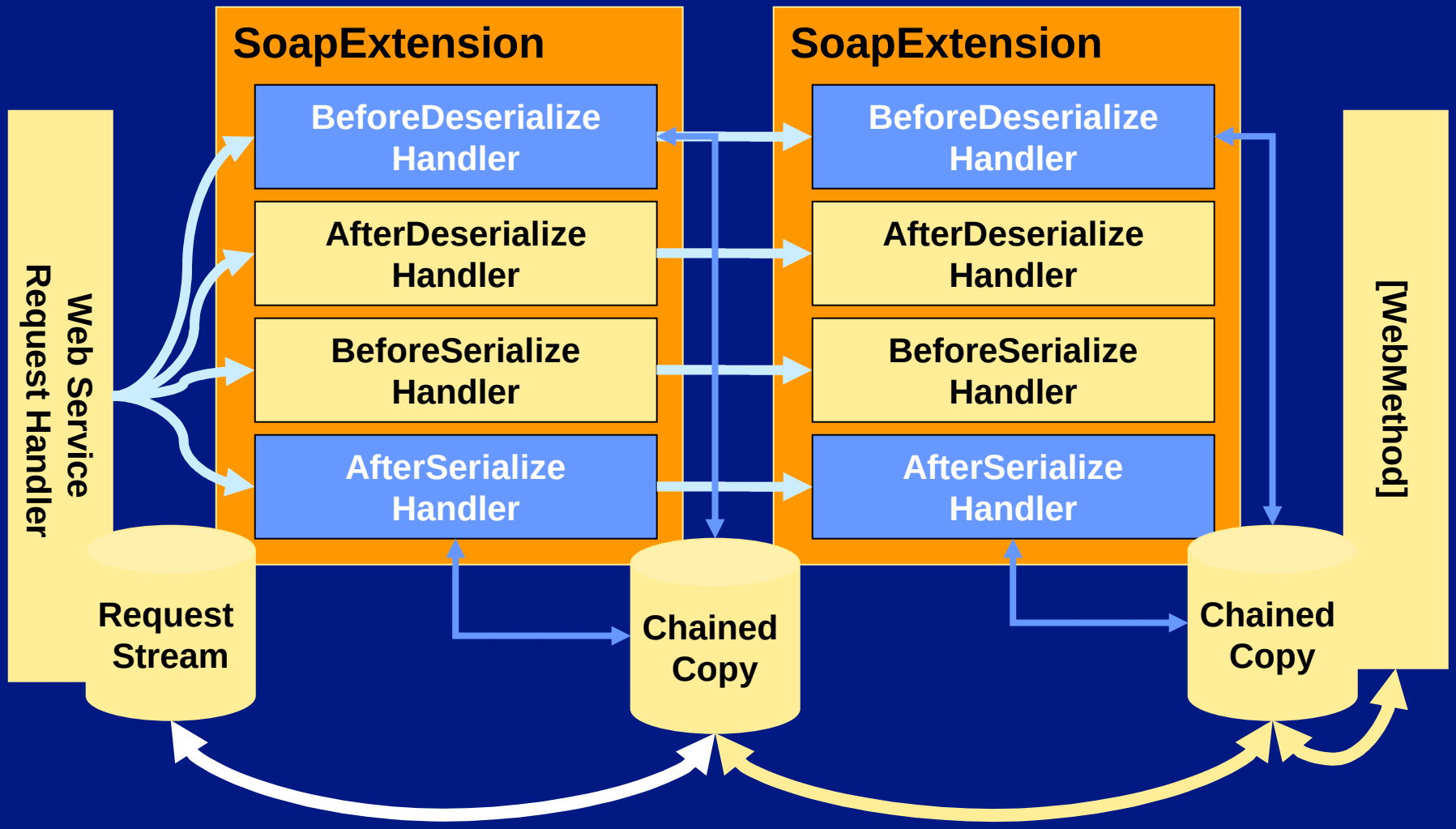
ASP.NET „SOAP Extensions“

- Plug-in architecture for ASP.NET Web Services
  - Declarative and configurable plug-ins for Web Services
- Can intercept and manipulate payloads and headers
  - Interception-driven functionality
- Can extend WSDL and inject code into proxies
  - ServiceDescriptionFormatExtension
    - Injects attributes into WSDL
  - ServiceDescriptionReflector reads metadata
    - Injects "operation binding" into WSDL
  - ServiceDescriptionImporter reads WSDL
    - Injects code into CodeDOM at proxy creation time

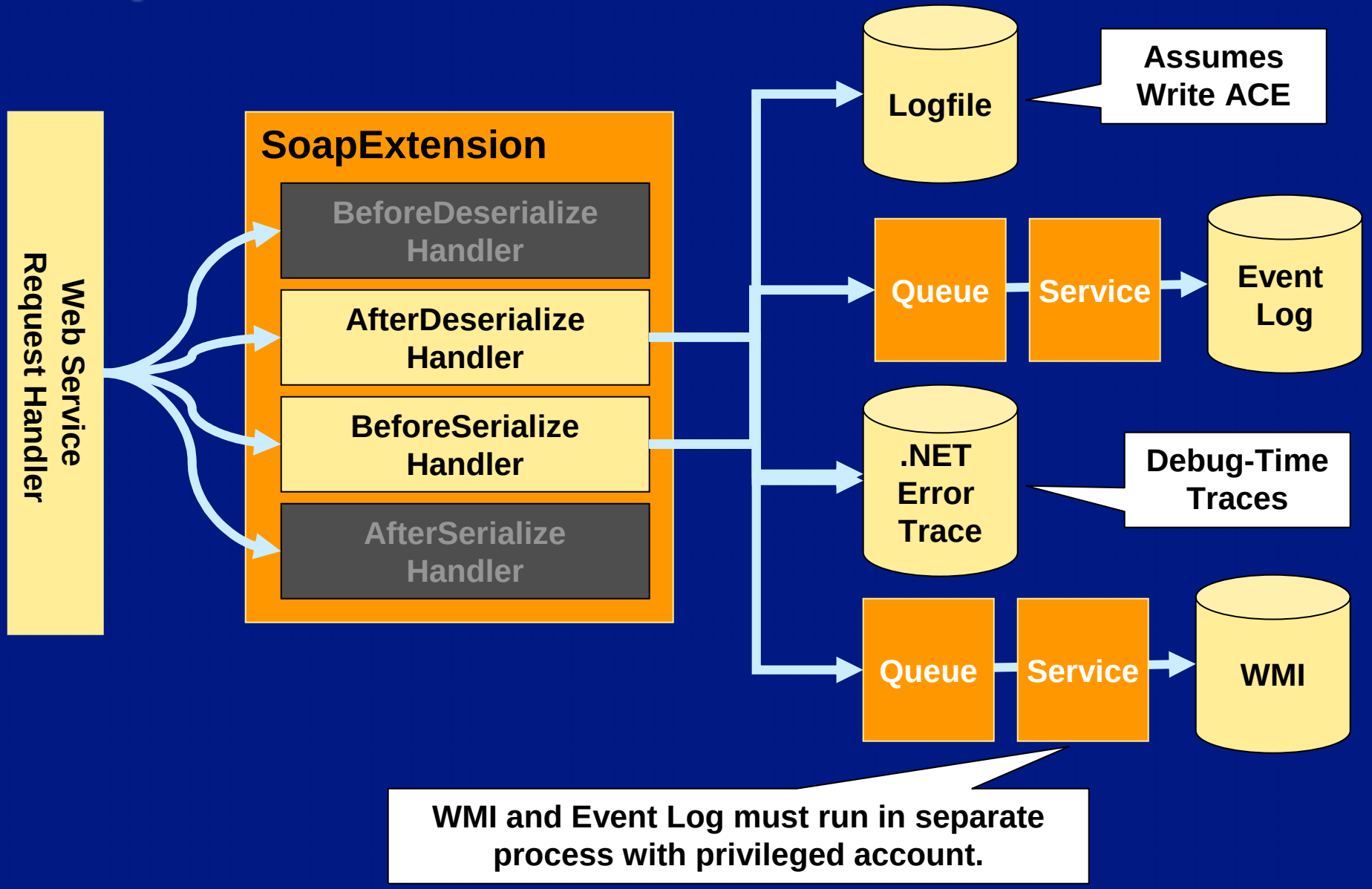
ASP.NET Pipelines



The Pipeline Call Chain



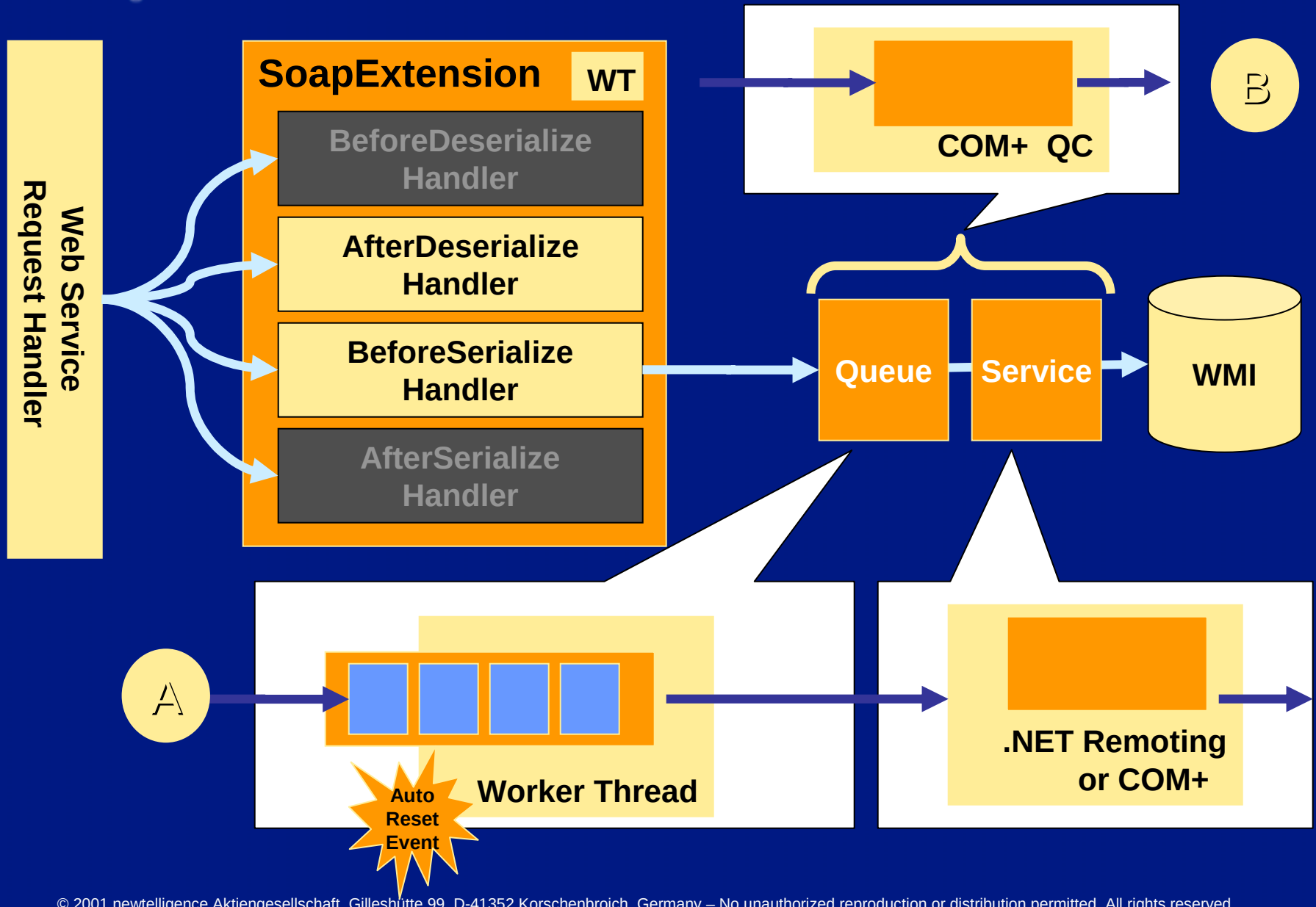
Implementing Tracing



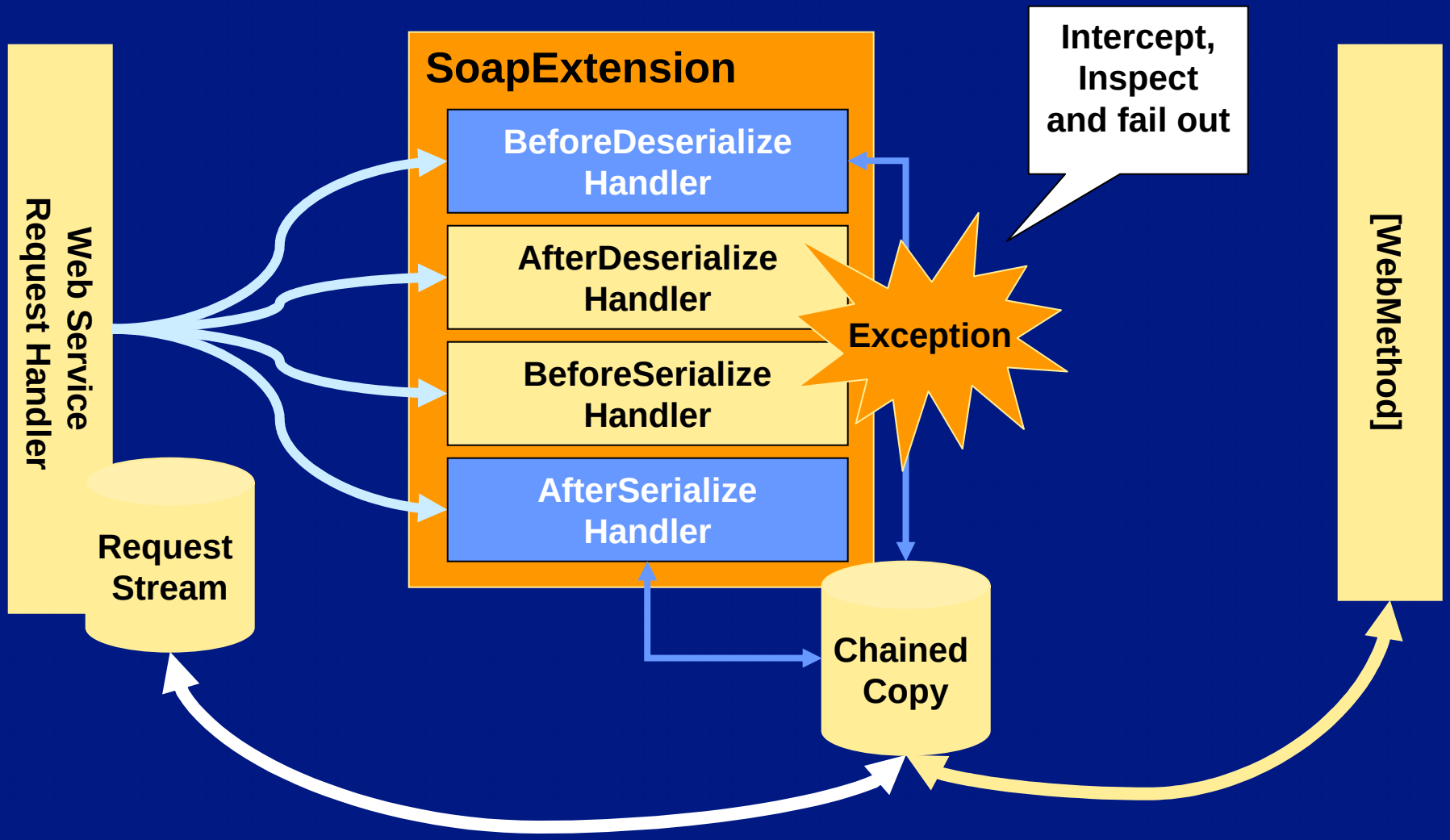
From the labs: MethodStatistics

- Demo: Method Statistics Sample
- Usage:
  - [WebMethod, MethodStatistics]
    void m();
- Functionality:
  - Logs method execution times to WMI
- Shows the following pattern:
  - Interception
  - Tracing
  - Queueing

Tracing is better with queues



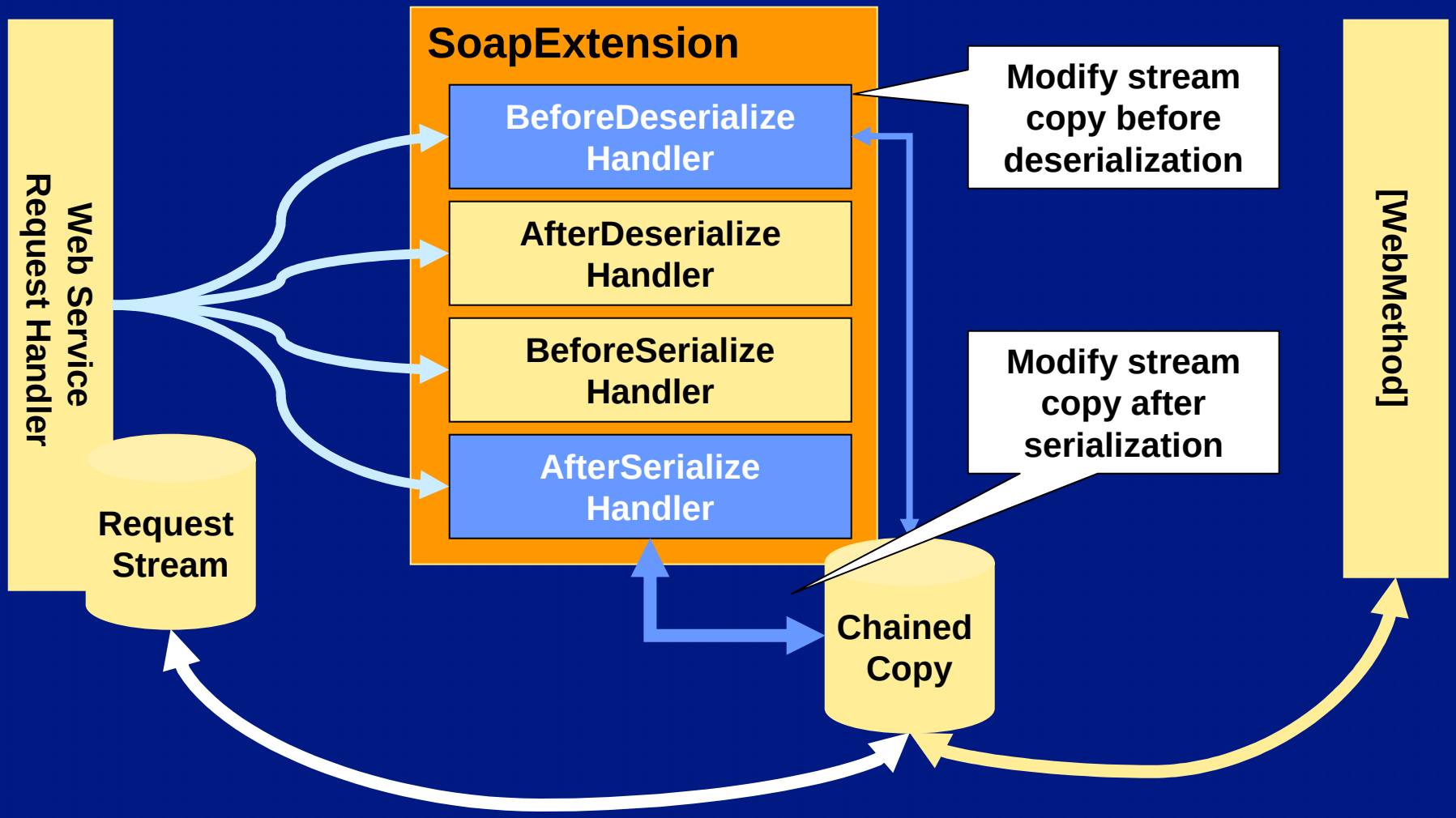
Implementing Interception



From the labs: Argument Validation

- Demo: Argument Validation Sample
- Usage:
  - [WebMethod, CheckArguments]
    void m([MaxLength(20)] string arg);
- Functionality:
  - Checks arguments before method entry
- Shows the following pattern:
  - Interception

Implementing XML Injection



From the labs: Kerberos

- Demo: Kerberos Authentication Sample
- Usage:
  - [WebMethod, KerberosAuthentication] void m();
- Functionality:
  - Mutual authentication and encryption with Kerberos
- Shows the following patterns:
  - Interception
  - Injection
  - Extra Roundtrips
  - Extra Metadata

Format Extensions

- Extensibility mechanism for WSDL
  - Perfectly legal through use of separate namespaces
- Inject your own tags into „operation bindings“
  - Operating Binding:
    - Pair of input and output messages
    - Equivalent to a function signature
    - Also defined out-of-band data (headers)
- Allows WSDL-mapping of method-level attributes
  - Metadata-enhanced WSDL

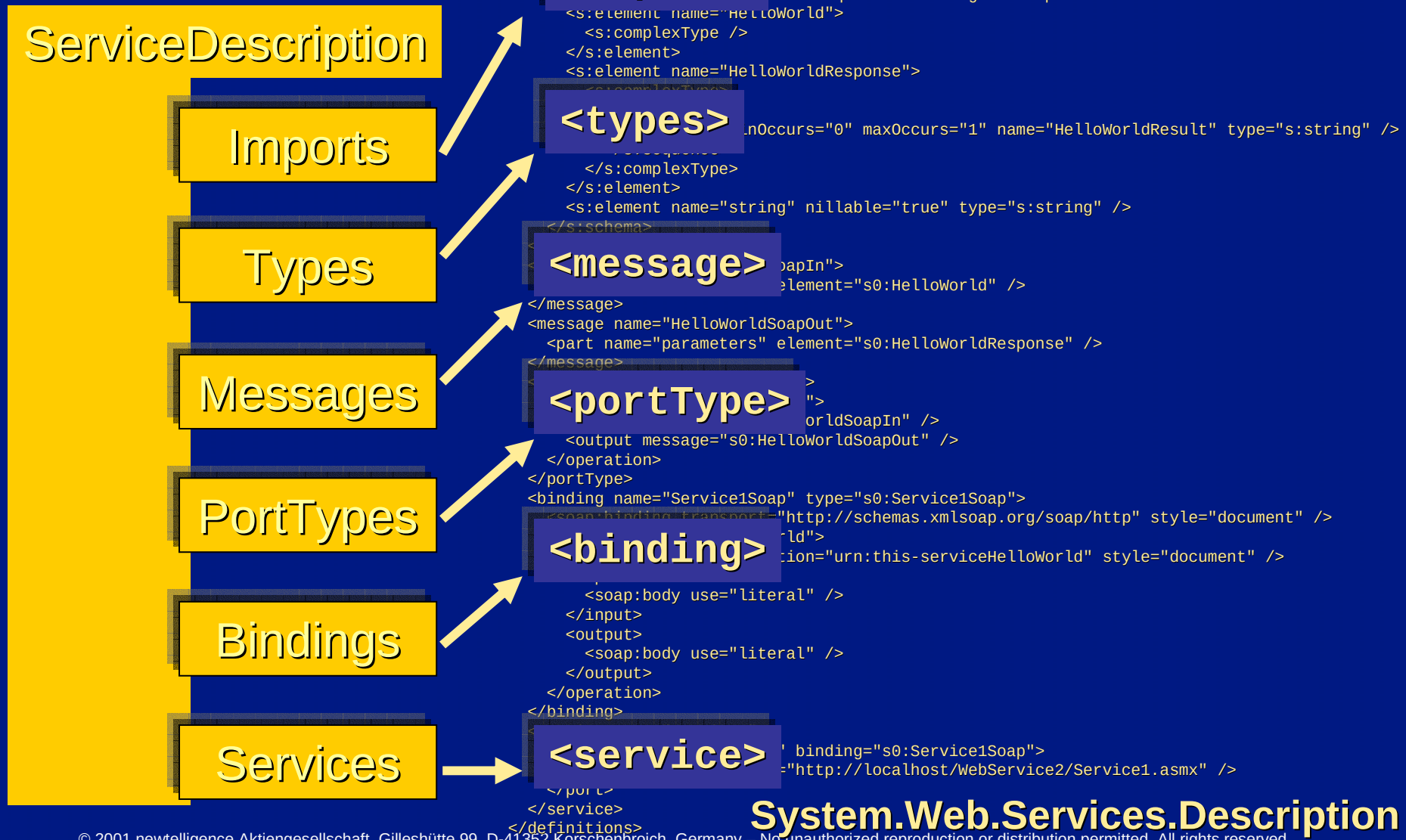
Format Extensions Applied

```
<operation name="SampleKerberos">
      <soap:operation soapAction="urn:schemas-newtelligence-com:SampleKerberos"
                      style="document" />
      <wsse-kerb-ext:wssecurityKerberosExtension>
         <wsse-kerb-ext:ServiceClass>
              host
         </wsse-kerb-ext:ServiceClass>
         <wsse-kerb-ext:ServiceHost>
              ambassador.embassy.newtelligence.com
         </wsse-kerb-ext:ServiceHost>
       </wsse-kerb-ext:wssecurityKerberosExtension>
      <input>
        <soap:body use="literal" />
        <soap:header d5p1:required="true"
                     message="s0:SampleKerberosWSSecurityKerberos"
                     part="WSSecurityKerberos" use="literal"
                     xmlns:d5p1="http://schemas.xmlsoap.org/wsdl/" />
      </input>
      <output>
        <soap:body use="literal" />
        <soap:header d5p1:required="true"
                     message="s0:SampleKerberosWSSecurityKerberos"
                     part="WSSecurityKerberos" use="literal"
                     xmlns:d5p1="http://schemas.xmlsoap.org/wsdl/" />
      </output>
    </operation>
```

Reflectors

- Reflectors are locally or globally installed classes
  - <soapExtensionReflectorTypes> in machine.config or web.config
- Derived from ServiceDescriptionReflector
- Called for every [WebMethod] on "?WSDL"
- Allow you to:
  - Modify WSDL using ServiceDescription object model
    - Add Schemas, Imports
    - Modify Messages
    - Add Headers
  - Inject format extensions into operation bindings

WSDL in the Framework



System.Web.Services.Description

Importers

- Importers are globally installed classes
  - <soapExtensionImporterTypes> in machine.config
- Derived from ServiceDescriptionImporter
- Called for every [WebMethod]
  - "Add Web Reference" or wsdl.exe (extends VS.NET)
- Allow you to:
  - Modify the proxy code (!) in any .NET language
    - Uses the CodeDOM
  - Add properties, methods to the proxy
  - Add attributes to the proxy methods
    - Allows to automatically hook in client-side SoapExtensions

Summary

- ASP.NET promotes the pipeline model
- Patterns ease development of pipelines
- Extensions with full docs now at
  http://www.newtelligence.net/wsextensions