

Auszug aus Axis2 Schulung

Dieses Dokument ist ein Auszug aus unserem Skript zur Axis2-Schulung. Es dient lediglich als Beispiel für unsere Kursunterlagen.

Thomas Bayer

Hauptstraße 33
75050 Gemmingen

www.thomas-bayer.de
info@thomas-bayer.de

Mehr zum Kurs finden Sie unter:

<http://www.thomas-bayer.com/axis2-schulung.htm>

SOAP4J, Apache SOAP und AXIS

Architektur 1

IBM SOAP4J

Basis für Apache SOAP

Architektur 2

Apache SOAP

Basiert auf DOM

Architektur 3

Axis1

Neuimplementierung mit SAX

Architektur 4

Axis2

Neuimplementierung mit StAX

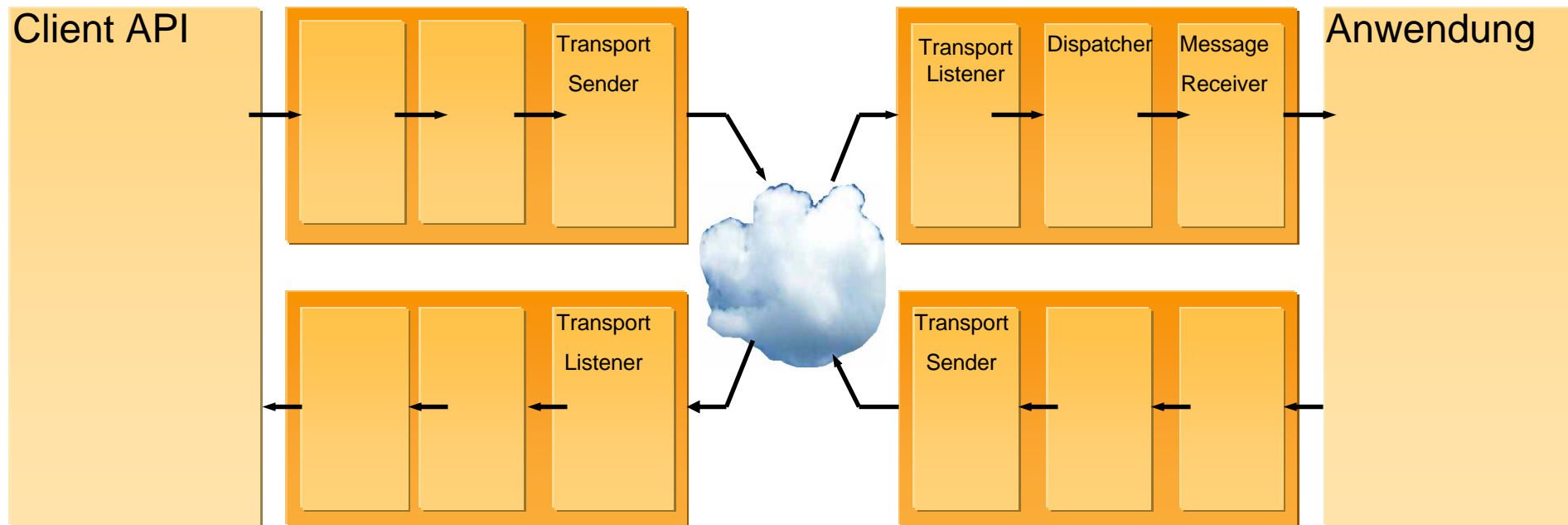
Kleinster Baustein komplexer SOAP Interaktionen



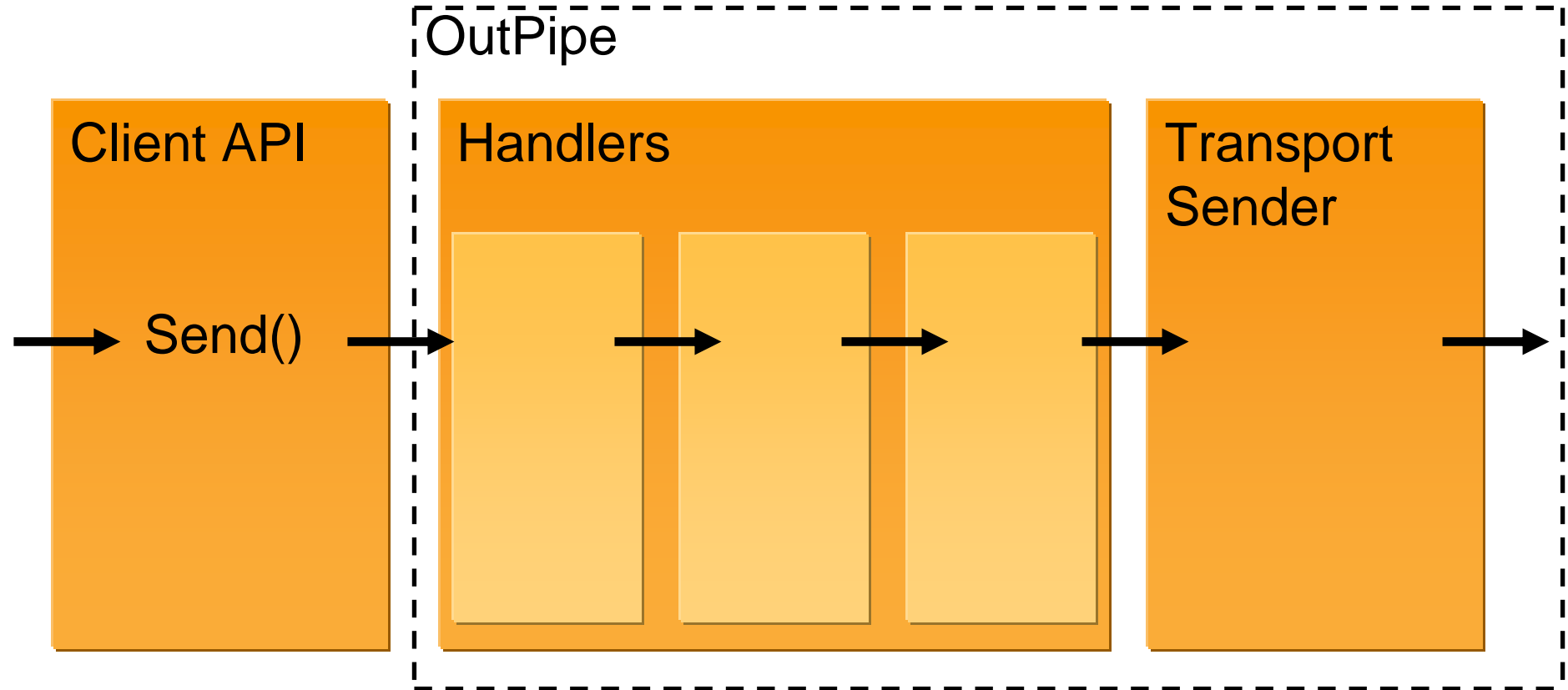
Übung: Axis2 Installation

- 1.) Binary Distribution entpacken
- 2.) `export $AXIS2_HOME=/opt/axis2-1.1.1`
- 3.) `export $PATH=$PATH:$AXIS2_HOME/bin`

Nachrichtenfluss bei Request/Response MEP



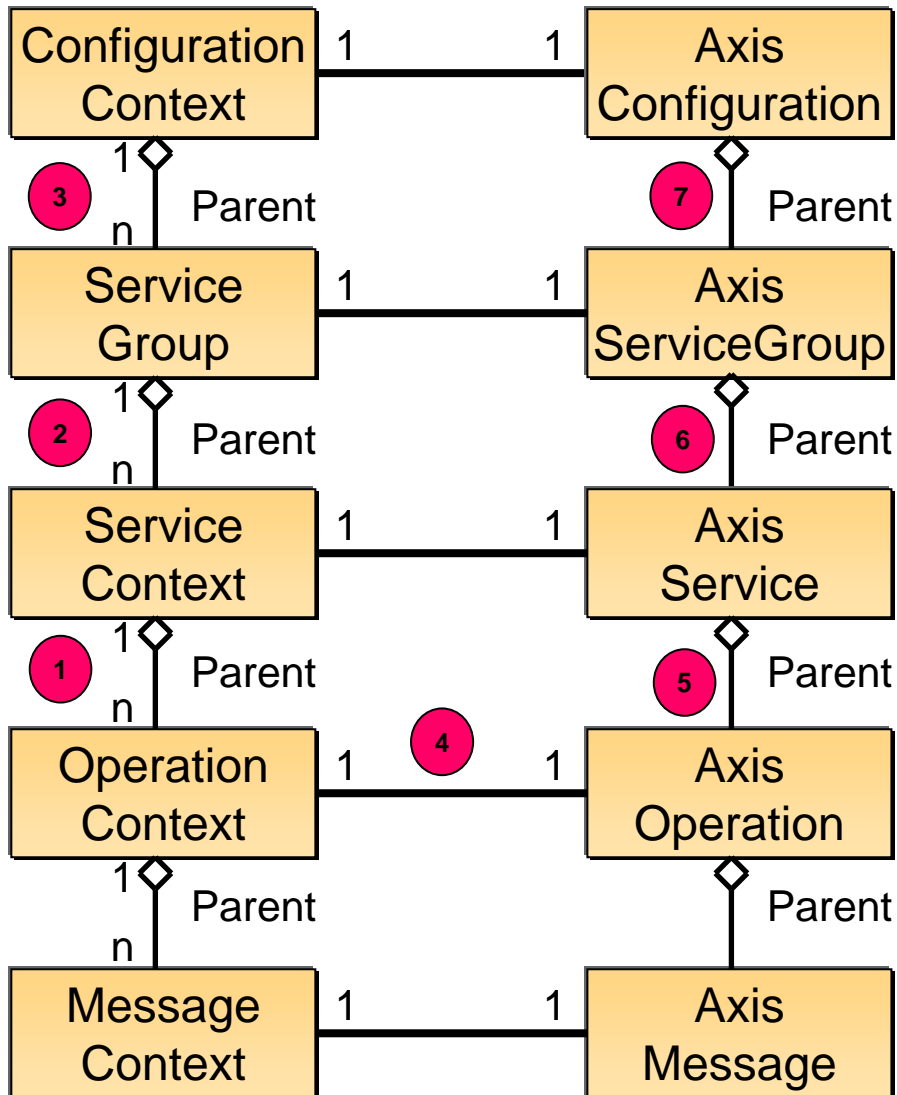
OutPipe auf Client Seite



Axis2 Startup

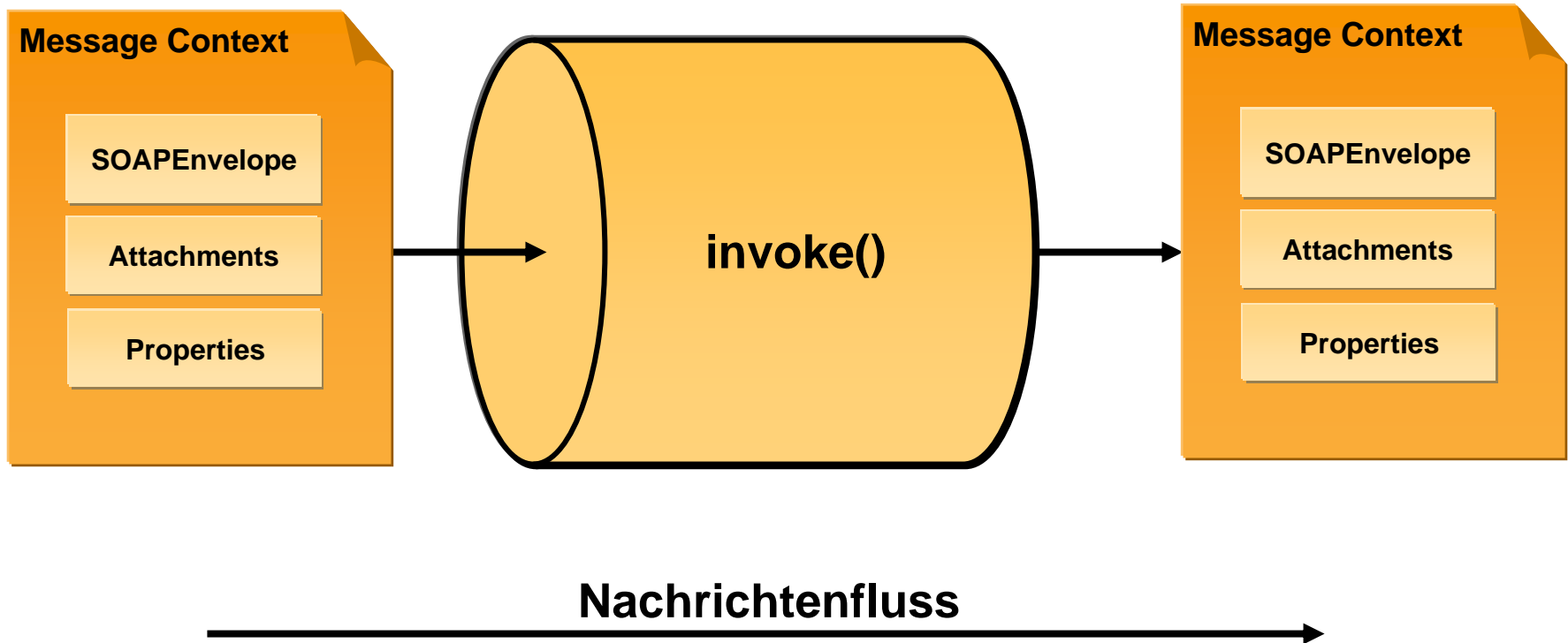
- 1.) Deployment Modul erzeugt Axis Configuration
 - axis2.xml wird ausgelesen
 - Globale Konfiguration wird erzeugt
- 2.) Modul Archive werden gesucht und installiert
- 3.) Service Archive werden gesucht und installiert
- 4.) Module und Services werden der AxisConfiguration hinzugefügt
- 5.) Kontexte werden aus der AxisConfiguration erzeugt

Context und Description Hierarchie

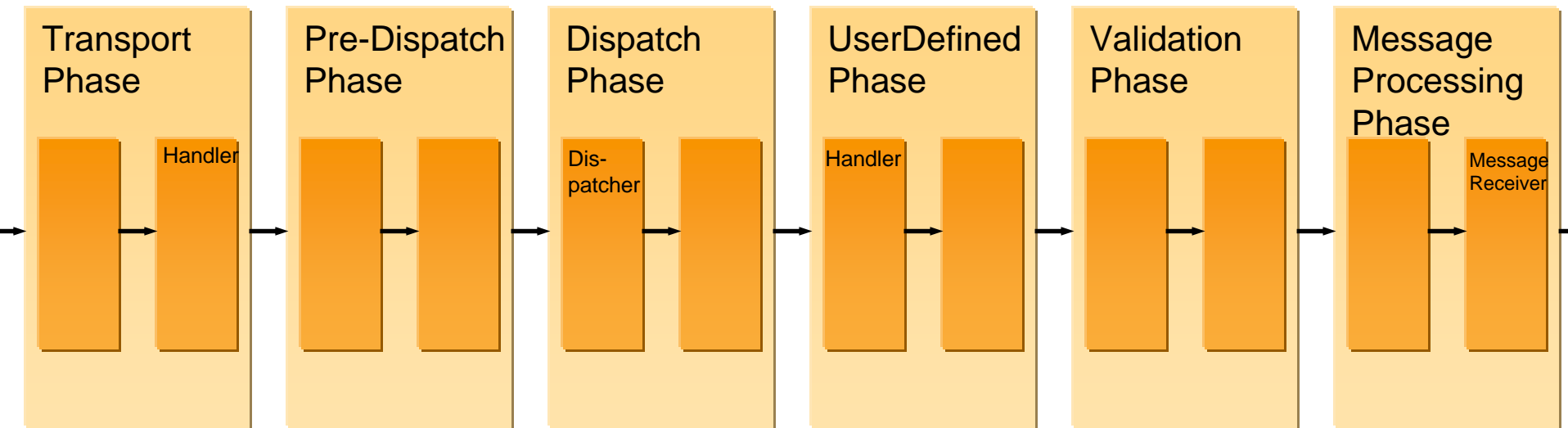


1 ... 7 Lookup Reihenfolge

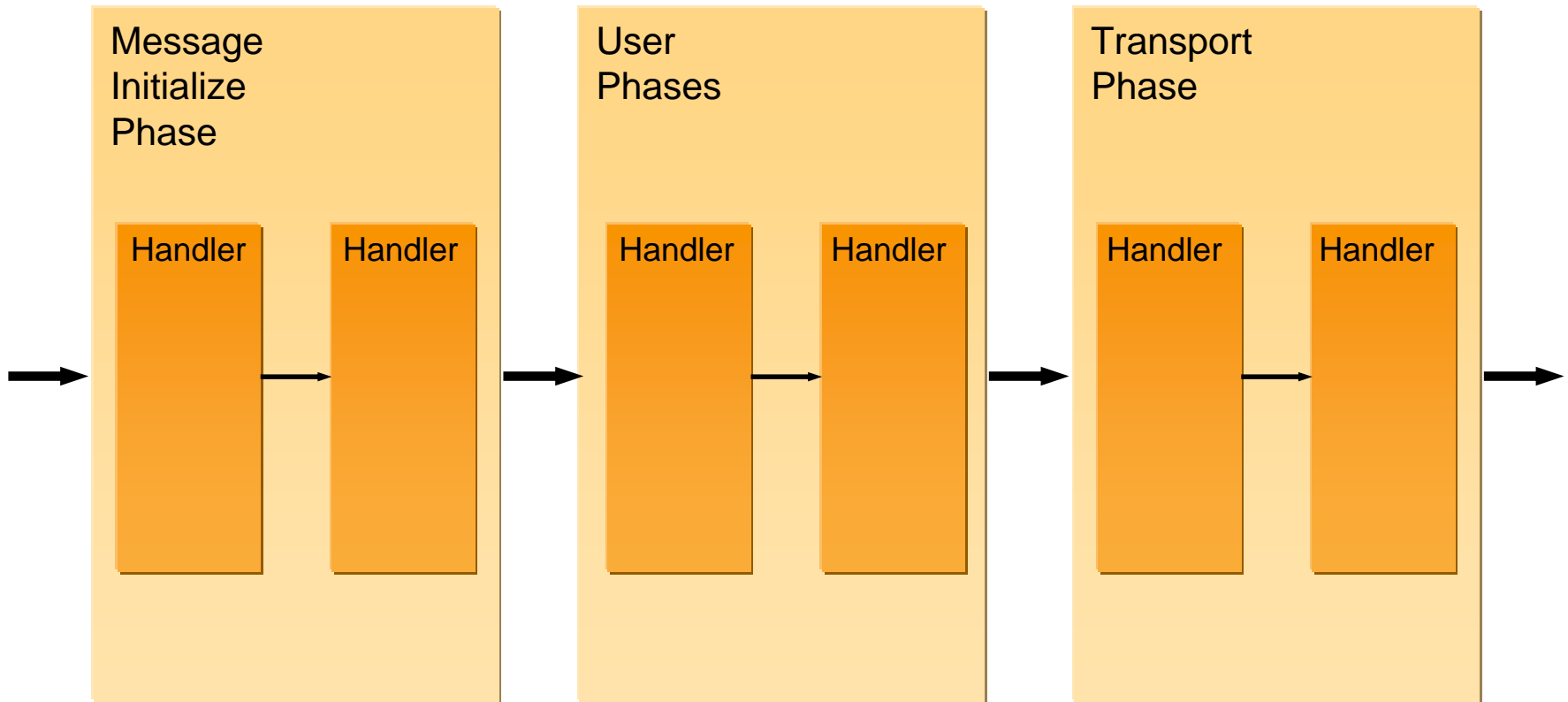
Handler



Phasen einer InPipe



Phasen einer OutPipe



Beispiel eines Axis2 Handlers

```
package com.thomas_bayer.axis2;

import org.apache.axis2.AxisFault;
import org.apache.axis2.context.MessageContext;
import org.apache.axis2.handlers.AbstractHandler;

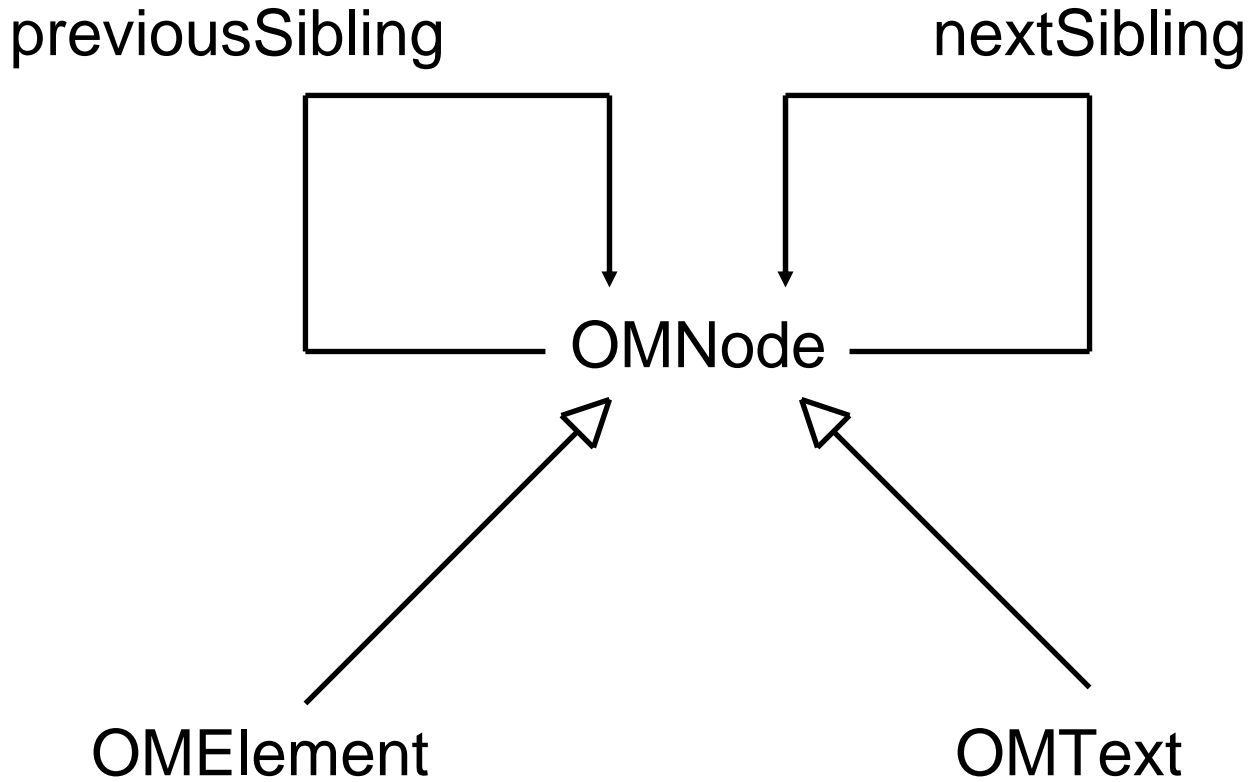
public class LogHandler extends AbstractHandler {

    public InvocationResponse invoke(MessageContext ctx)
        throws AxisFault {

        System.out.println(ctx.getEnvelope().toString());

        return InvocationResponse.CONTINUE;
    }
}
```

AXIOM Model



Parameter und Properties

Parameter	Property
statisch	dynamisch
ausgelesen von axis2.xml, services.xml, module.xml	werden zur Laufzeit ermittelt
<code><parameter name="...">... </...></code>	<code>msgCtx.setProperty("...")</code>
Primitive Typen, String, OMElement	Objekte
Leben in Description Hierarchie	Leben in Context Hierarchie
Parameter <code>p=msgCtx.getParameter("...")</code>	Object <code>o=msgCtx.getProperty("...")</code>

wsdl2java Optionen

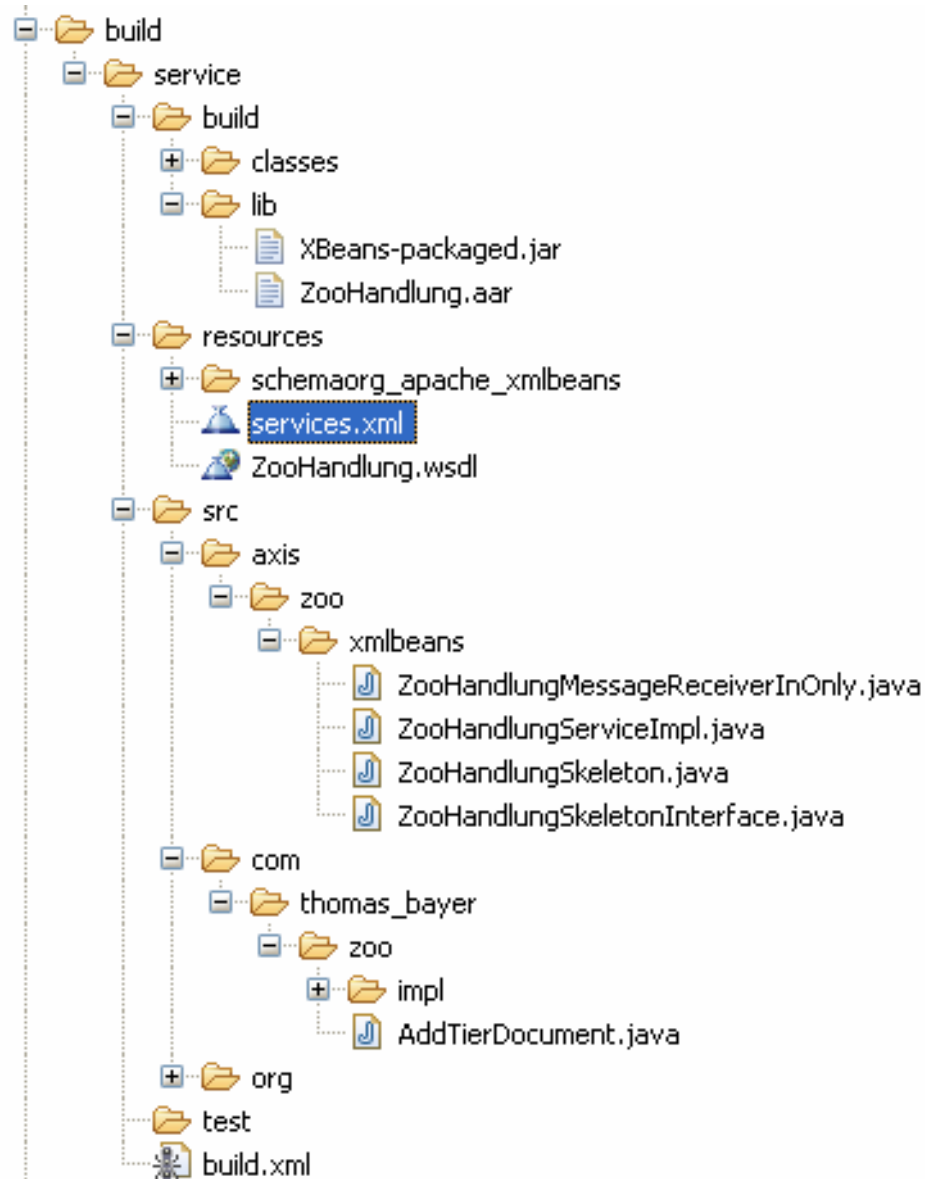
- p Paketnamen für generierten Code
- d Data Binding adb, xmlbeans, jibx, none
- ss Server Side Code
- sd Service Descriptor services.xml (nur mit –ss Option)
- ssi Erzeugt Service Interface
- nw Unwrap
- o Output Verzeichnis
- a Erzeugt rein asynchronen Stub
- s Erzeugt rein synchronen Stub
- t Erzeugt JUnit Test
- g Erzeugt alles
- u Erzeugt Klassen in separaten Dateien anstatt Inner Classes
- sn Servicenamen für den Code generiert werden soll. Falls nicht angegeben wird der erste im WSDL verwendet
- ns2p , getrennte Liste von NS-Paketzuordnungen

java2wsdl Ant Task

```
<taskdef name="java2wsdl"
  classname="org.apache.ws.java2wsdl.Java2WSDLTask"
  classpathref="axis2.classpath" />

<java2wsdl classname="swa.BildService"
  outputfilename="{build.res}/bild.wsdl"
  locationuri="http://localhost:8080/axis2/services/" >
  <classpath refid="server.classpath" />
</java2wsdl>
```


Von WSDL2Java erzeugte Verzeichnisstruktur



Generische HTTP Properties, die von Axis2 unterstützt werden

Property	Bedeutung
CHUNKED	HTTP Chunking
NTLM_AUTHENTICATION	ms Proxy Authentifizierungsinformation Wer: Instanz von NTLMAuthentication
PROXY	HTTP Proxy Wert: Instanz von Proxy Properties
BASIC_AUTHENTICATION	HTTP Basic Authentication Wert: Instanz von Basic Authentication
SO_TIMEOUT	Socket Timeout in Millis
CONNECTION_TIMEOUT	Timeout der Verbindung
USER_AGENT	HTTP User Agent. Default: Axis2
MC.GZIP_REQUEST	Komprimiere Anfrage mit GZIP
MC_ACCEPT_GZIP	Sendet HTTP Accept Header mit GZIP
COOKIE_STRING	HTTP Cookie Header Feld
HTTP_PROTOCOL_VERSION	HTTP Protokoll Version

Options Properties mit getter/setter ohne WS-Addressing

Property	Bedeutung
Action	SOAPAction bzw. WS-Addressing Action
Listener	Transport Listener für eingehende Nachrichten
Parent	Vater Options Objekt mit Default-Werten
SOAPVersionURI	SOAP Version 1.1 oder 1.2
TimeoutInMilliseconds	Timeout Zeit für blockierende Aufrufe
TransportIn	Beschreibung des eingehenden Transports
TransportInProtocol	Protokoll für eingehende Nachrichten
TransportOut	Beschreibung des ausgehenden Transports
ExceptionToBeThrown On SOAPFault	Rethrow Exception mit den Werten aus SOAPFault

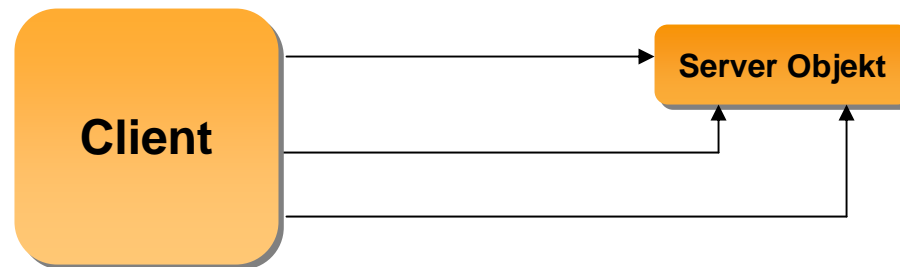
WS-Addressing Properties in Client Options

Property	Bedeutung
Action	SOAPAction bzw. WS-Addressing Option
FaultTo	FaultTo EPR
From	From EPR
MessageID	Id der Nachricht
RelatesTo(String)	RelatesTo Beziehung zu einer anderen Nachricht
Relationships	Array mit allen RelatesTo Hems
ReplyTo	ReplyTo EPR

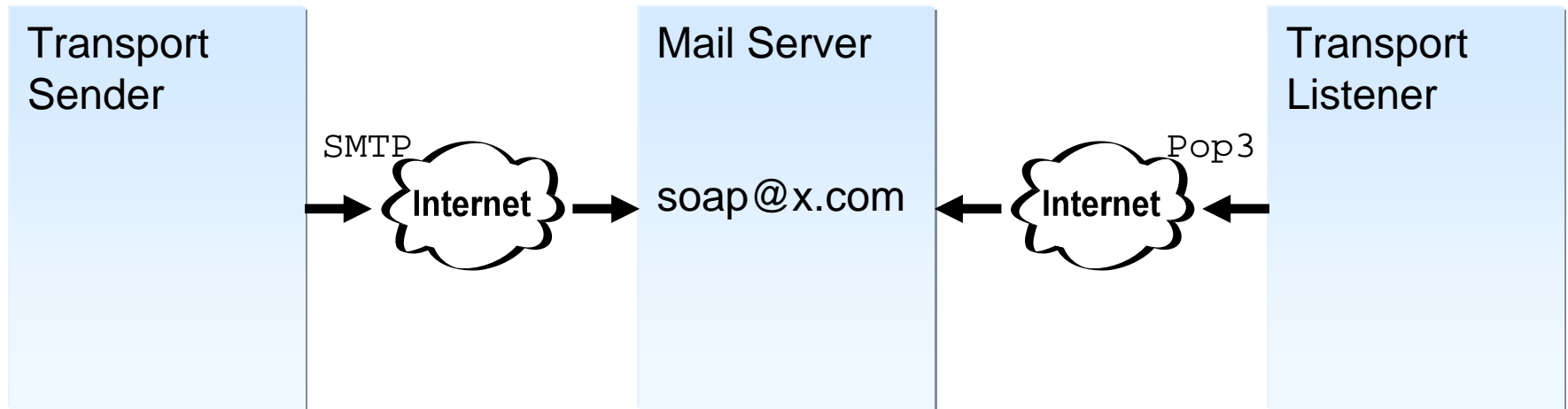
SOAP Session Scope

- Client und Server müssen sich der Session bewußt sein
- Timeout default: 30s
- Gültig in einer Service Group

```
<service name="..." scope="soapession">  
  ...  
</service>
```



Mail Transport



InFlow mit RM User Phase

InFlow

