



IBM Systems and Technology Group

## S07 - Web Services generieren und andere Goodies im z/VSE



**ON** DEMAND BUSINESS™

# Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and / or other countries.

|                            |                       |                        |
|----------------------------|-----------------------|------------------------|
| AIX*                       | IBM logo*             | SQL/DS                 |
| CICS*                      | IMS                   | Virtual Image Facility |
| CICS/VSE*                  | Intelligent           | VisualAge*             |
| C/370                      | Language Environment* | VisualGen*             |
| DB2*                       | Miner                 | VM/ESA*                |
| DB2 Connect                | MQSeries*             | VSE/ESA                |
| DB2 Universal Database     | Multiprise*           | VTAM*                  |
| DFSORT                     | MVS                   | WebSphere*             |
| e-business logo*           | OS/2*                 | xSeries*               |
| eServer                    | OS/390*               | z/Architecture         |
| Enterprise Storage Server* | OS/400*               | z/OS*                  |
| HiperSockets               | Rational*             | z/VM                   |
| IBM*                       | S/390*                | z/VSE                  |
|                            | SNAP/SHOT*            | zSeries*               |

\* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

LINUX is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Tivoli is a trademark of Tivoli Systems Inc.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows the Windows 95 logo, and Windows NT, are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

Intel is a registered trademark of Intel Corporation.

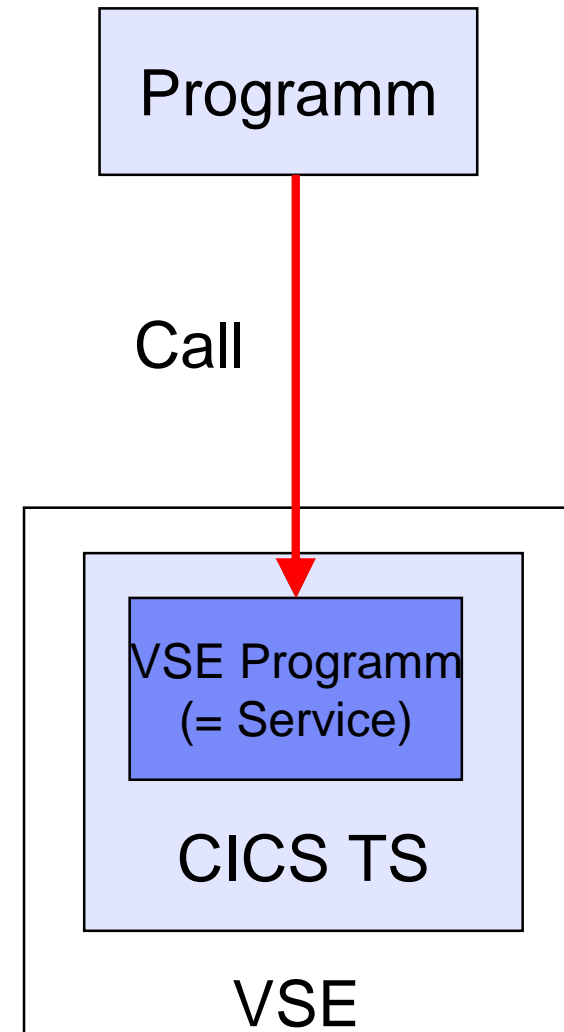
Other company, product, and service names, may be trademarks or service marks of others.

# Agenda

- § **Was ist ein Web Service?**
- § **Begriffe und Theorie**
  - SOA, SOAP, WSDL, SOAP RPC, XML
- § **Web Services mit Java/.Net verwenden**
  - Proxy-Code
- § **Web Services mit VSE verwenden**
  - VSE SOAP Engine, Programming Interfaces
- § **CICS2WS Tool**
  - Live Demo
- § **Andere Goodies**

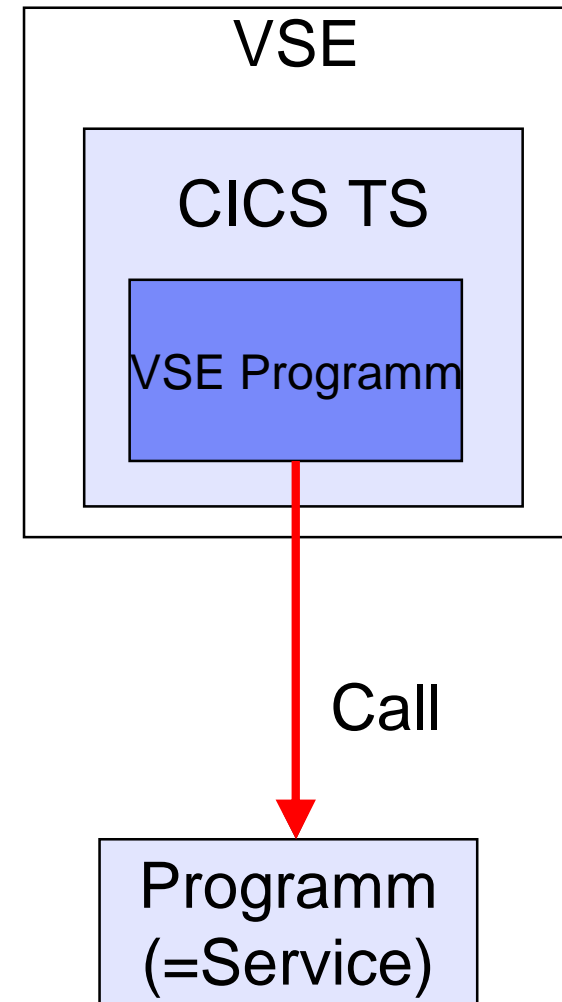
## Was ist ein Web Service?

- § **Sie haben ein VSE Programm** das eine **wichtige Funktion (Business Logik)** implementiert
- § **Jemand anders (außerhalb VSE)** möchte dieses Programm auch benutzen
  - 1. Möglichkeit: Neuschreiben des Programms
    - Benötigt evt. Zugriff auf VSE Daten
    - Änderungen/Fixes im VSE Code müssen im neuen Code nachgezogen werden
  - 2. Möglichkeit: Das VSE Programm aufrufen
    - Das VSE Programm kann als **Web Service** betrachtet werden
    - VSE ist der Web Service Provider



## Was ist ein Web Service?

- § **Jemand anderes hat ein Programm** das eine wichtige Funktion (Business Logik) implementiert
- § **Sie wollen dieses Programm in Ihrer VSE Anwendung verwenden**
  - 1. Möglichkeit: Neuschreiben des Programms
    - Benötigt evt. Zugriff auf entfernte Daten
    - Änderungen/Fixes müssen im VSE Code nachgezogen werden
  - 2. Möglichkeit: Das externe Programm aufrufen
    - Das externe Programm kann als **Web Service** angesehen werden
    - VSE ist der Web Service Requestor



# Agenda

- § **Was ist ein Web Service?**
- § **Begriffe und Theorie**
  - SOA, SOAP, WSDL, SOAP RPC, XML
- § **Web Services mit Java/.Net verwenden**
  - Proxy-Code
- § **Web Services mit VSE verwenden**
  - VSE SOAP Engine, Programming Interfaces
- § **CICS2WS Tool**
  - Live Demo
- § **Andere Goodies**

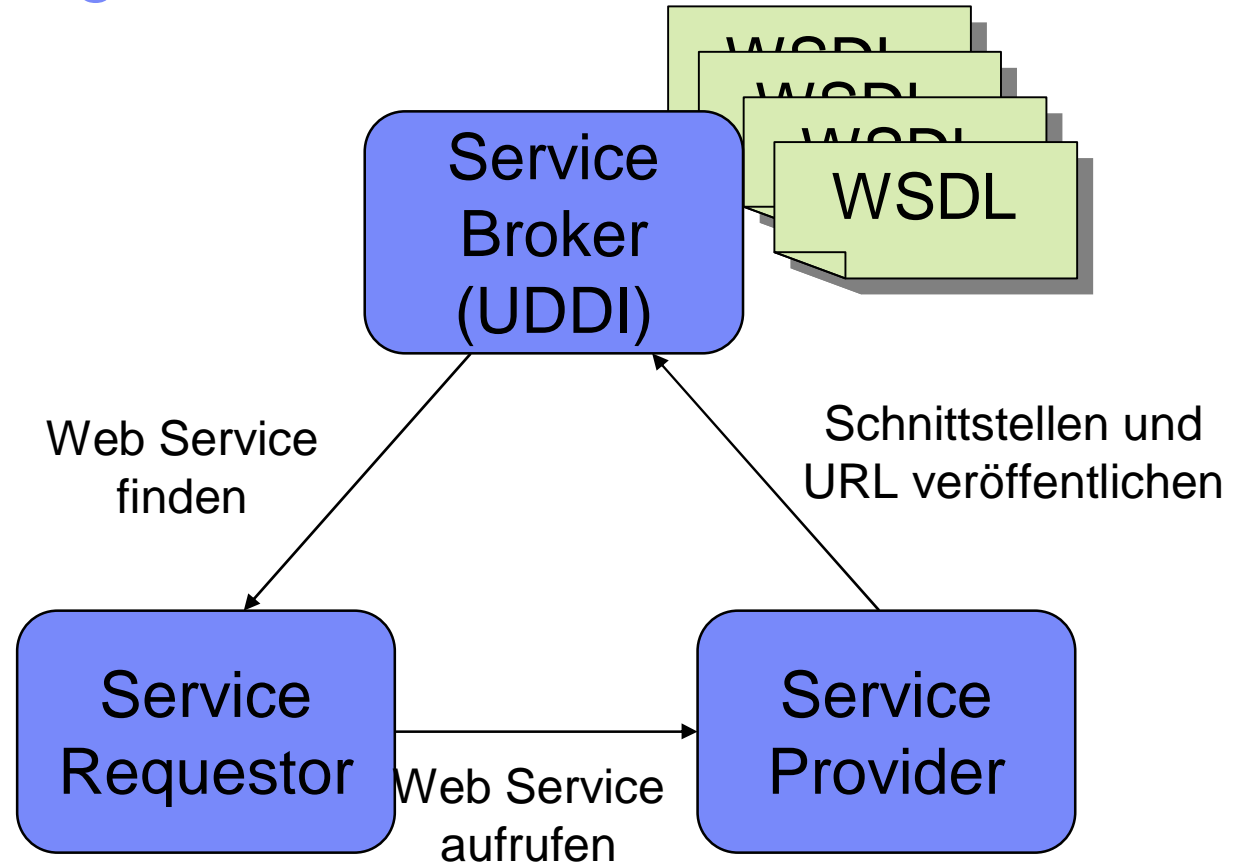
# Web Services - Begriffe

**XML**

**SOAP**

**HTTP - Transport Service**

**TCP/IP - Street**



## Web Services - Begriffe

### Request:

```
<soap:Envelope>  
  <soap:Body xmlns:m="http://www.stock.org/stock" />  
    <m:GetStockPrice>  
      <m:StockName>IBM</m:StockName>  
    </m:GetStockPrice>  
  </soap:Body>  
</soap:Envelope>
```

### Response

```
<soap:Envelope>  
  <soap:Body xmlns:m="http://www.stock.org/stock" />  
    <m:GetStockPriceResponse>  
      <m:Price>34.5</m:Price>  
    </m:GetStockPriceResponse>  
  </soap:Body>  
</soap:Envelope>
```



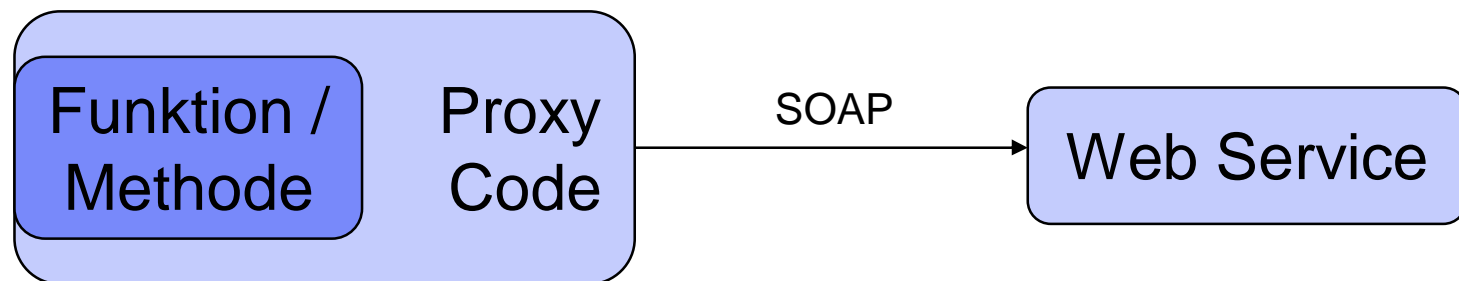
# Agenda

- § **Was ist ein Web Service?**
- § **Begriffe und Theorie**
  - SOA, SOAP, WSDL, SOAP RPC, XML
- § **Web Services mit Java/.Net verwenden**
  - Proxy-Code
- § **Web Services mit VSE verwenden**
  - VSE SOAP Engine, Programming Interfaces
- § **CICS2WS Tool**
  - Live Demo
- § **Andere Goodies**

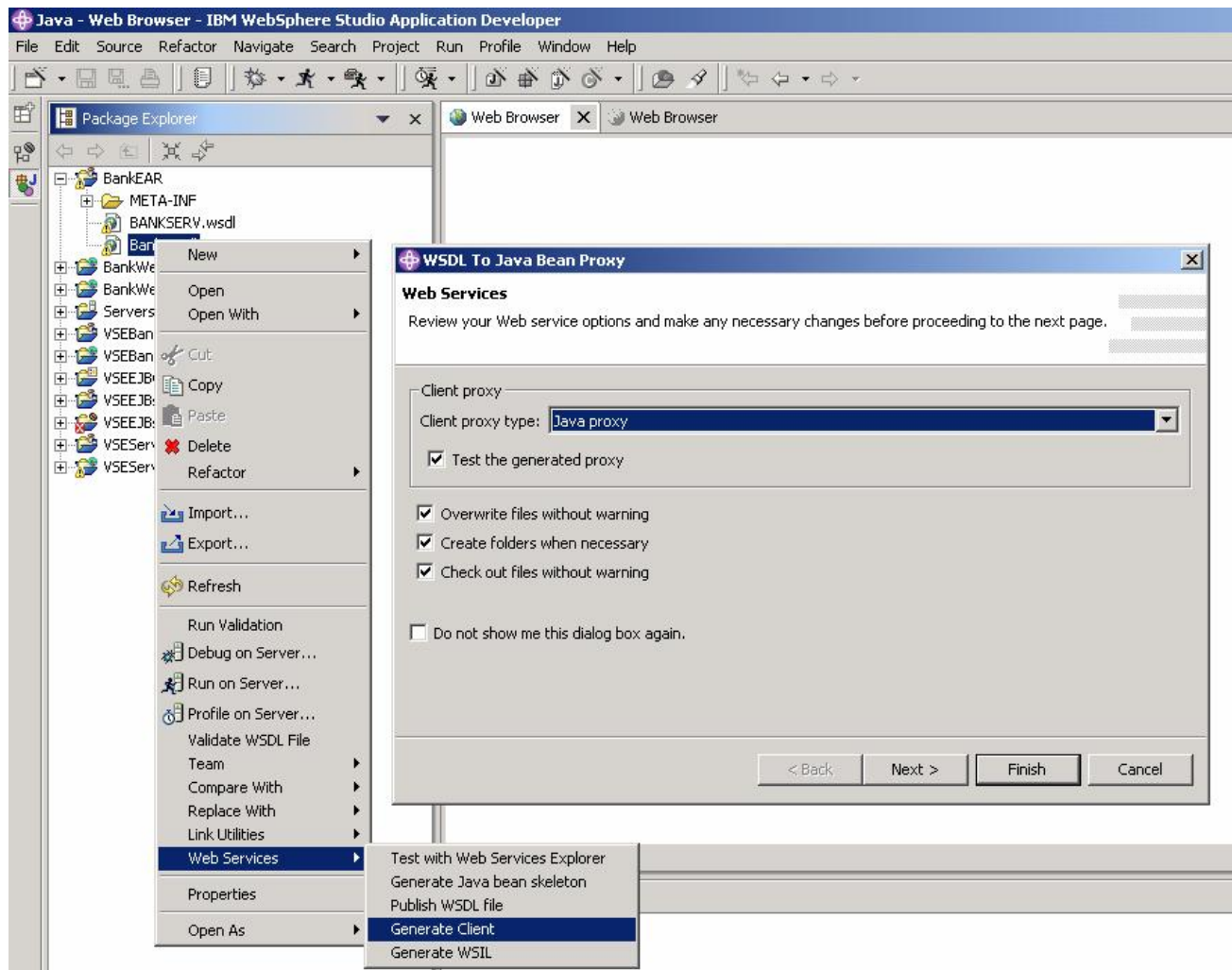
## Web Services mit Java oder MS .Net verwenden

### § Aufrufen eines bestehenden Web Service

- Sie wissen dann ein bestimmter Web Service existiert
- Die Web Service Description (WSDL) lokalisieren
- In einem Tool wie Rational Application Developer (RAD/WSAD) oder Microsoft Visual Studio wird die WSDL importiert
  - Generieren des “Proxy-Codes” welcher alle speziellen Dinge implementiert die benötigt werden um den Web Service aufzurufen
  - Anwendungen rufen eine Methode/Funktion des Proxy-Codes auf also ob es eine lokale Methode/Funktion wäre



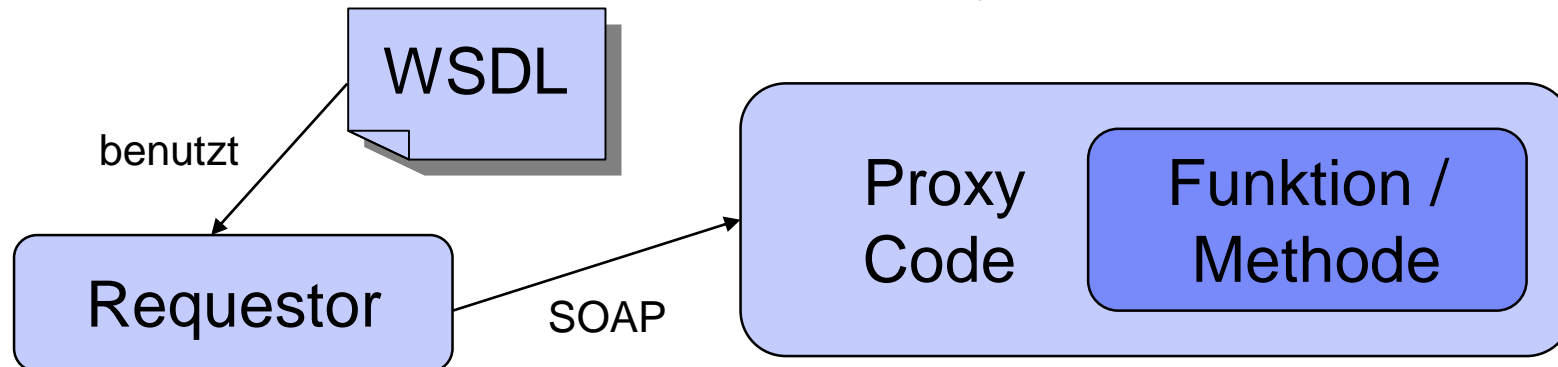
# Web Services mit Java oder MS .Net verwenden



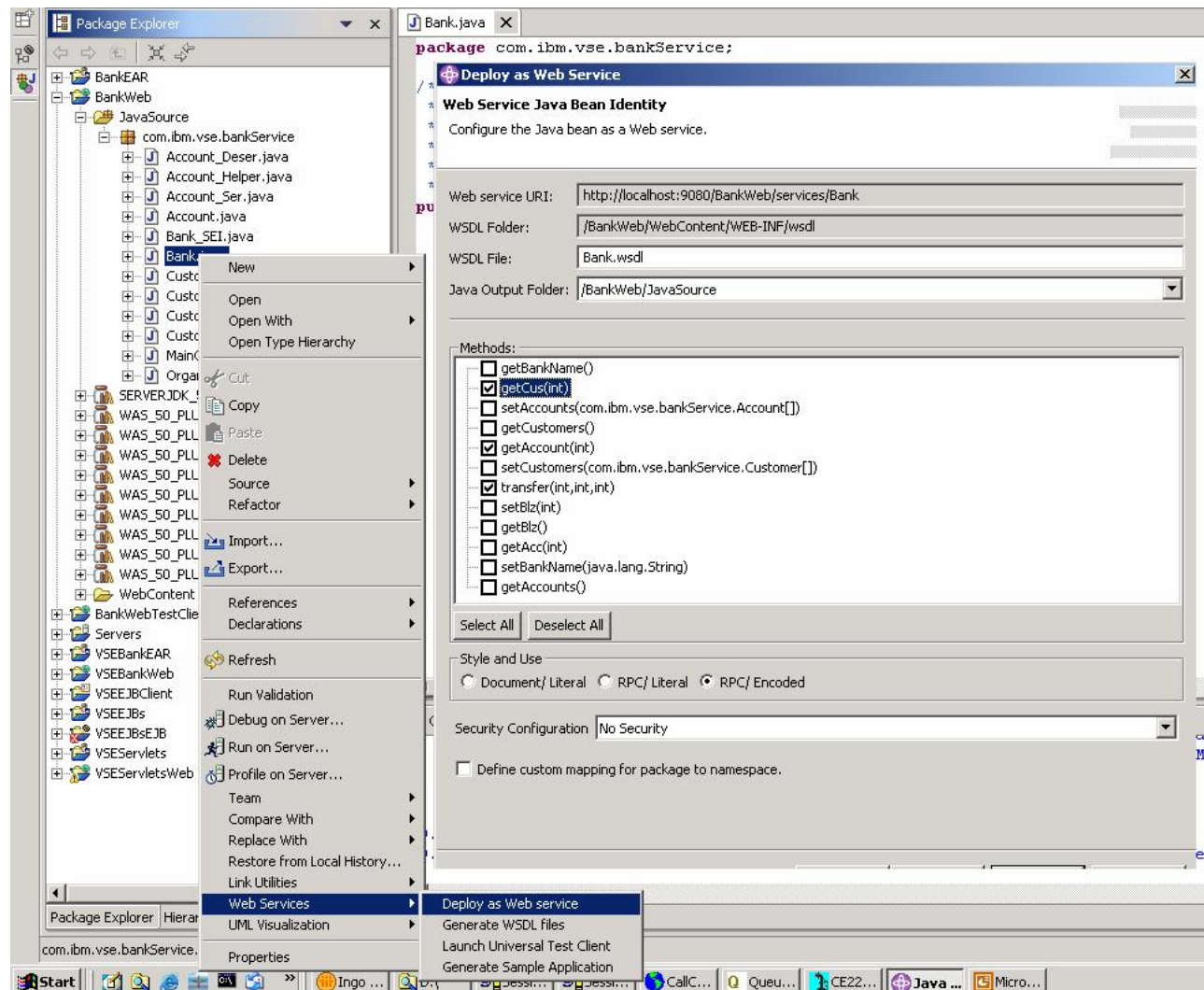
## Web Services mit Java oder MS .Net verwenden

### § Erstellen eines neuen Web Services

- Sie haben eine Funktion der Methode, die einen Service implementiert den Sie zur Verfügung stellen wollen.
- In einem Tool wie Rational Application Developer (RAD/WSAD) oder Microsoft Visual Studio wird der Web Service modelliert
  - Erzeugen der Web Service Description (WSDL) und veröffentlichen
  - Generieren des “Proxy-Codes” welcher die Funktion oder Methode aufrufbar über SOAP als Web Service macht
  - Den Code In einen Application Server deployen



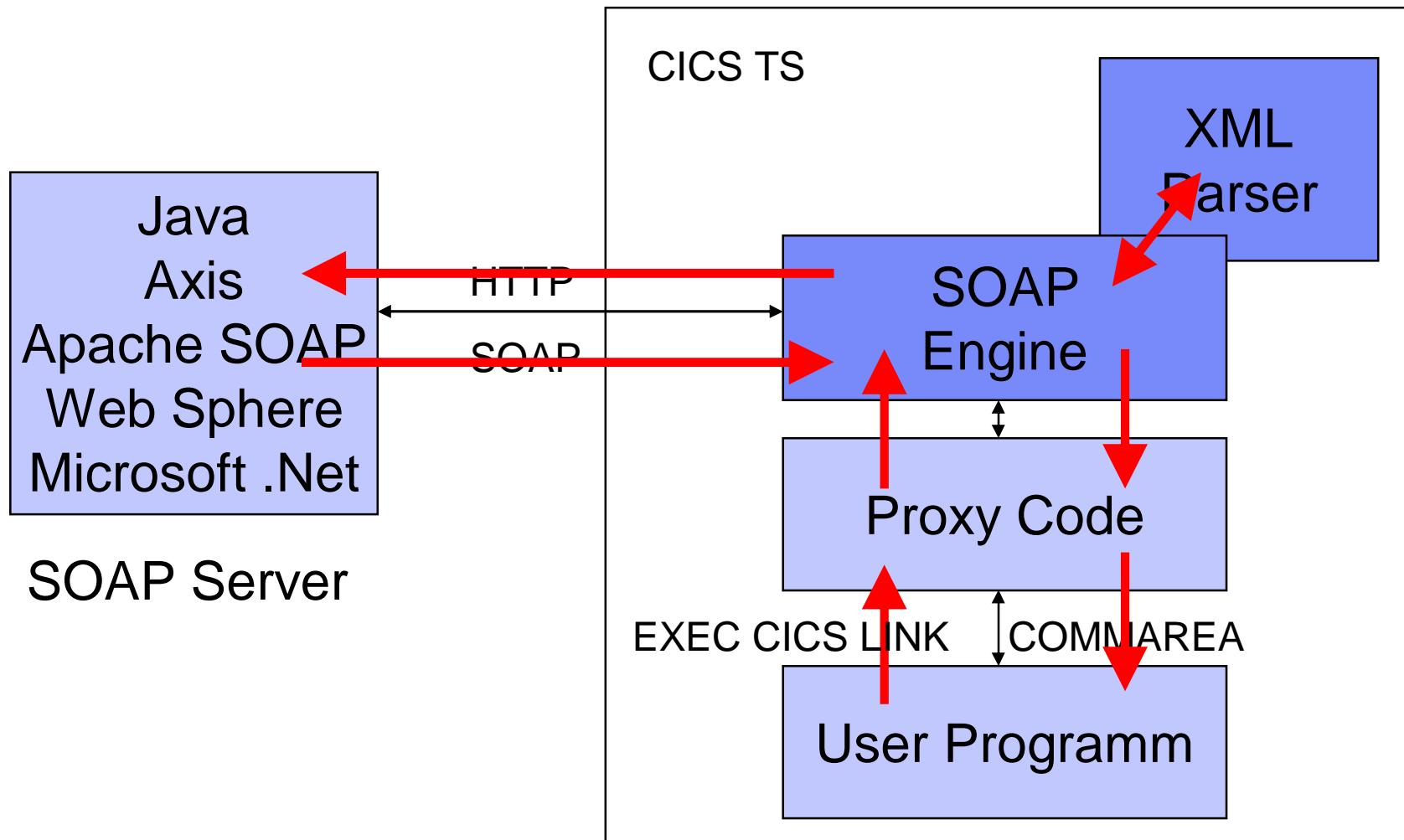
# Web Services mit Java oder MS .Net verwenden



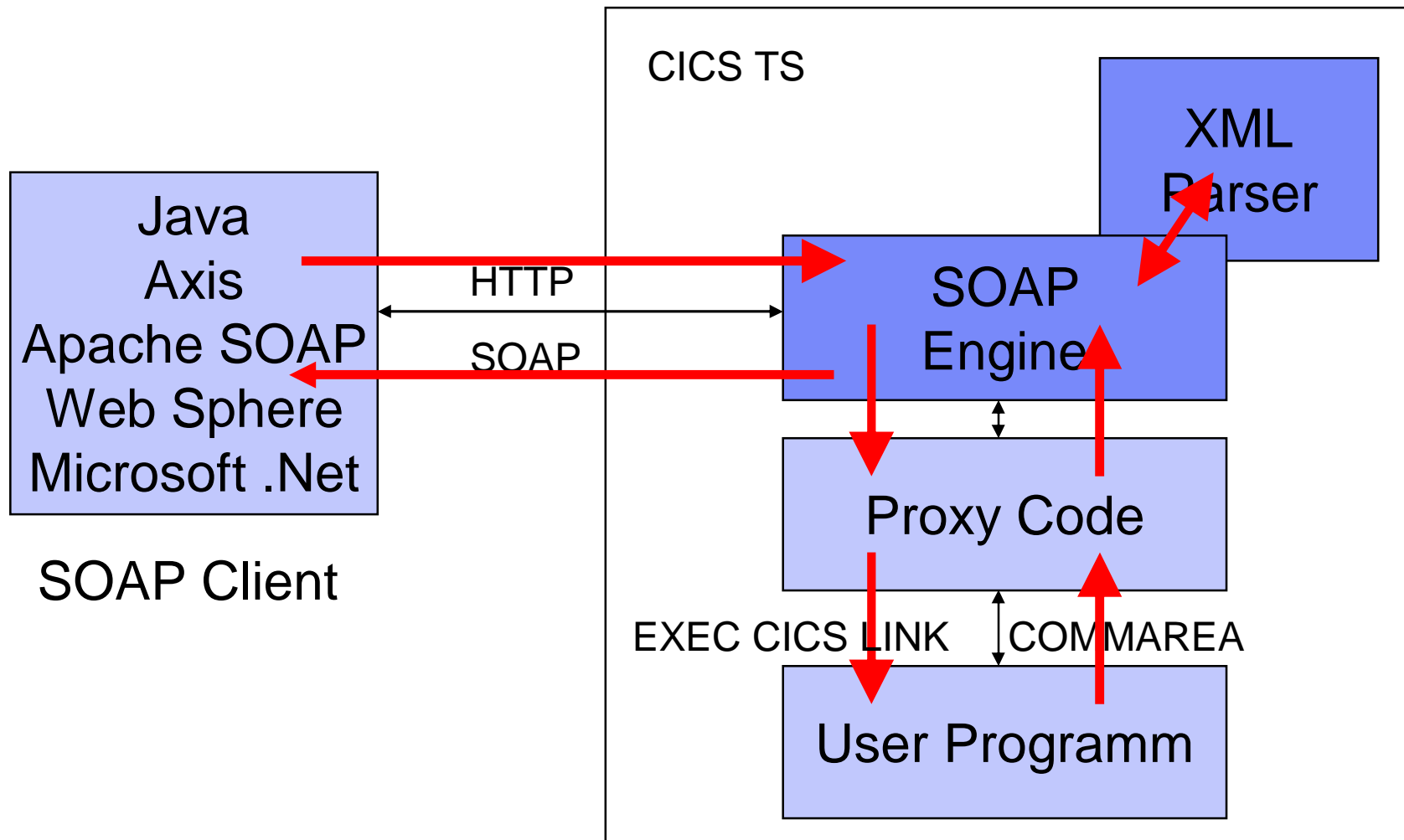
# Agenda

- § **Was ist ein Web Service?**
- § **Begriffe und Theorie**
  - SOA, SOAP, WSDL, SOAP RPC, XML
- § **Web Services mit Java/.Net verwenden**
  - Proxy-Code
- § **Web Services mit VSE verwenden**
  - VSE SOAP Engine, Programming Interfaces
- § **CICS2WS Tool**
  - Live Demo
- § **Andere Goodies**

# Web Services mit VSE verwenden – SOAP Client

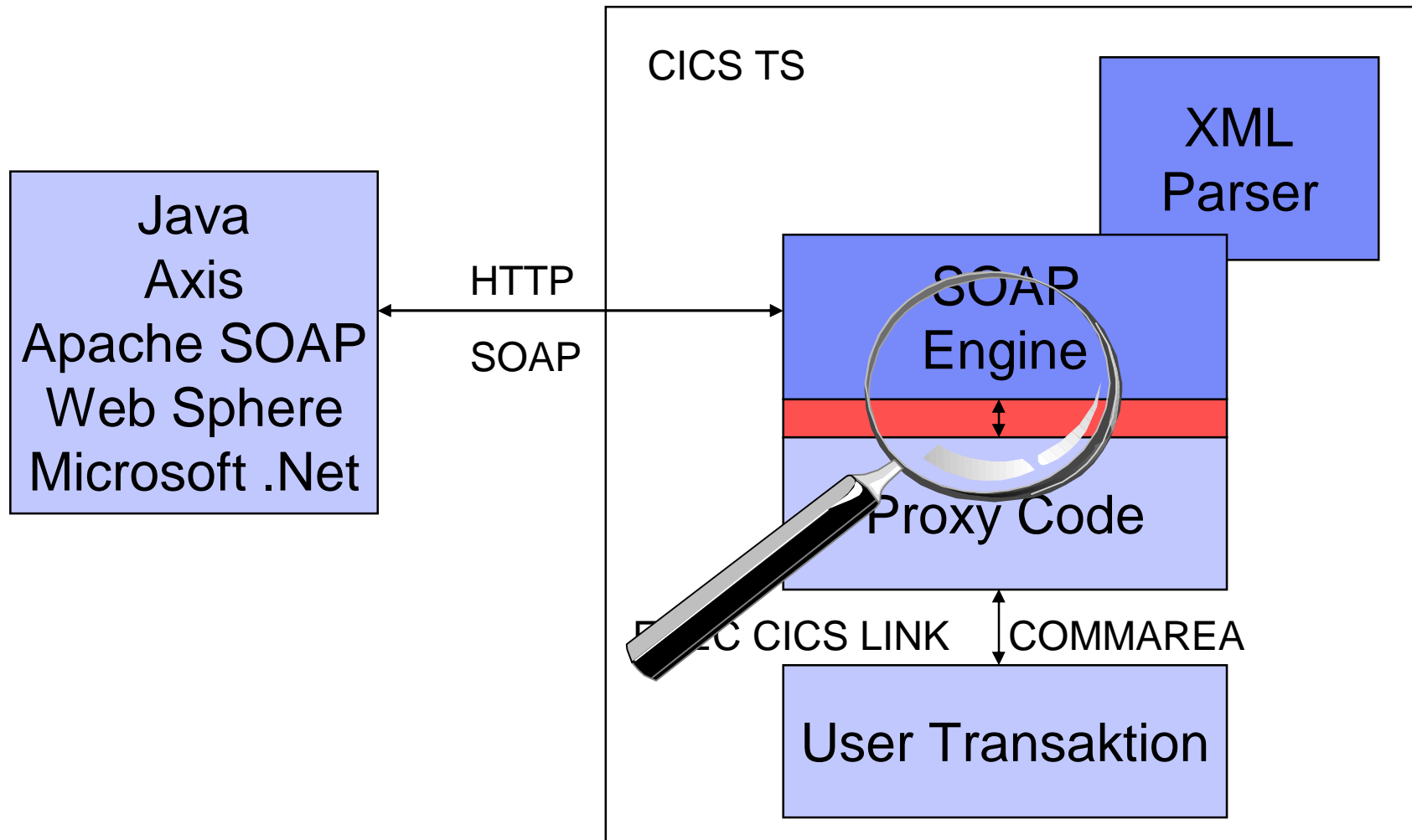


# Web Services mit VSE verwenden – SOAP Server





# Web Services mit VSE verwenden



## VSE SOAP Engine

- § **Teil der VSE Connectors**
- § **Übersetzt SOAP Nachrichten in ein/aus einem natives Format**
  - Empfängt/verschickt die SOAP Nachricht über HTTP
  - Parset/generiert XML
  - Interpretiert/erzeugt die SOAP Nachricht
  - Serialisiert/de-serialisiert die Input und Output Werte
    - Übersetzt sie von textuellem Format in ein natives Format und umgekehrt
  - Ruft das Service-Provider-Programm auf
  - Wird aufgerufen wenn externer Web Services benutzt werden sollen

# VSE SOAP Engine

## § Input/Output Parameter

- Jeder Parameter wird durch einen TS-Queue-Entry repräsentiert
  - Parameter Name (z.B. “StockPrice”)
  - Parameter Type (z.B. “String”)
  - Parameter Wert (z.B. “34.5”)
  - Länge der Daten
- Input Parameters werden in die Input-Queue gestellt
- Output Parameters werden von der Output-Queue gelesen

```
01 SOAP-PARAM-HDR .
   05 NAME                PIC X(16) .
   05 TYPENAME            PIC X(16) .
   05 LENGTH              PIC 9(8)  COMP .
   05 TYPECODE            PIC 9(8)  COMP .
   05 VALUE               PIC X(20) .

EXEC CICS WRITEQ TS QUEUE(OUTQUEUE)
          FROM(SOAP-PARAM-HDR)
          LENGTH(TS-QUEUE-LENGTH-OUT)
          RESP(COMMAND-RESPONSE)
          END-EXEC .
```

## Warum braucht man den Proxy-Code?

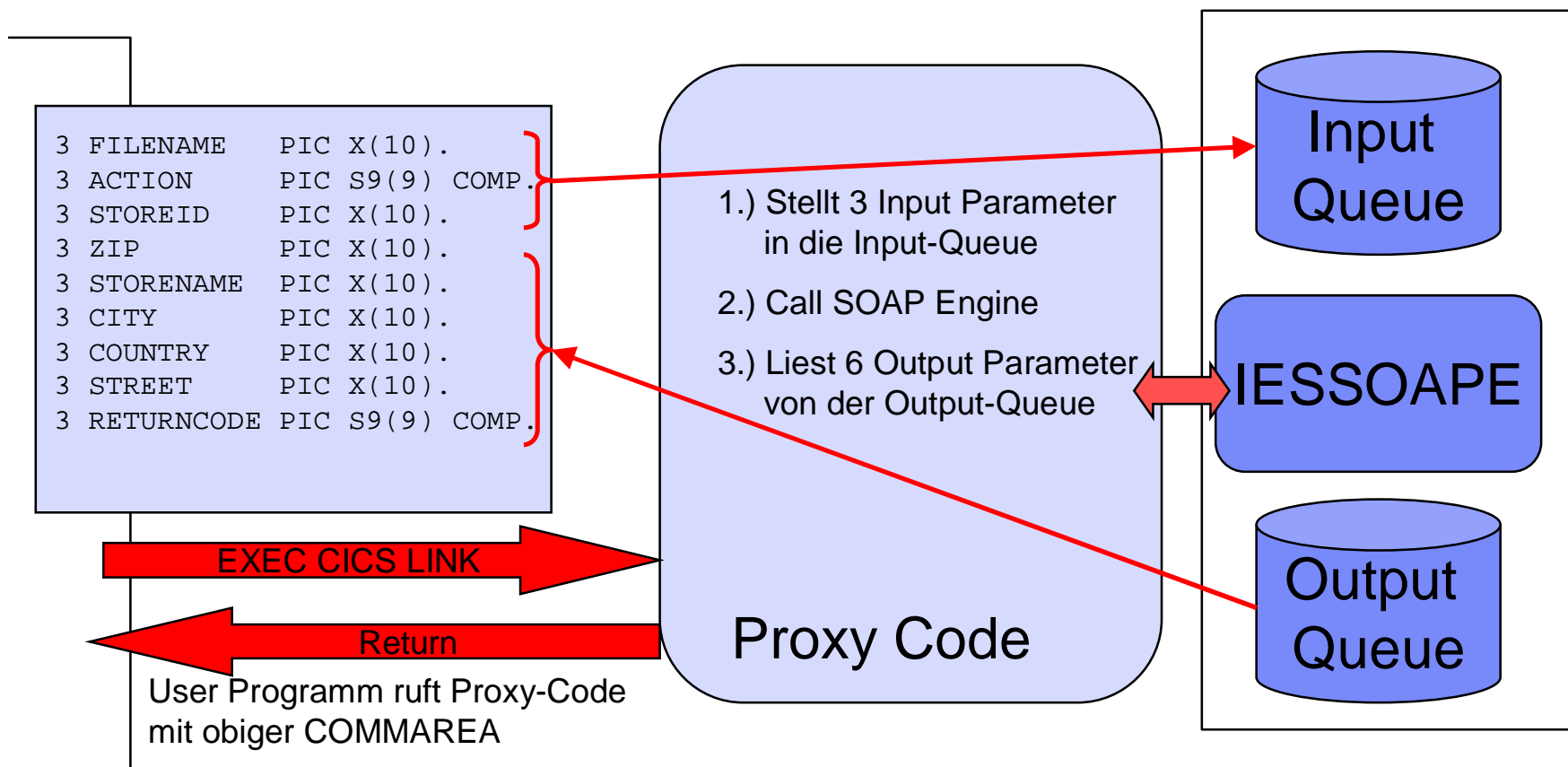
- § **Obwohl die SOAP Engine standard CICS Schnittstellen benützt, werden spezielle Code-Teile gebraucht, um die VSE SOPA Engine aufzurufen**
- § **Der Proxy-Code 'übersetzt' zwischen der COMMAREA Schnittstelle und der SOAP Engine**
  - Alles SOAP spezifische wird im Proxy-Code gemacht
  - Anwendungen rufen den Proxy-Code, oder werden von ihm gerufen, mittels einem lokalen Programm-Call (EXEC CICS LINK) mit einer COMMAREA
  - Das COMMAREA Format kann frei definiert werden
  - Der Proxy-Code kopiert die Felder der COMMAREA in TS-Queue-Entries und umgekehrt

## Warum braucht man den Proxy-Code?

- § **Alle SOAP Implementierungen benützen irgend eine Art von “Proxy-Code”**
  - Java (RAD/WSAD)
  - Microsoft .Net (MS Visual Studio)
  - ...
- § **Der Proxy-Code wird typischerweise aus den Informationen der WSDL generiert**
- § **Der Proxy-Code muss normalerweise nicht vom Benutzer modifiziert oder verstanden werden**
- § **VSE benützt die selbe Technik wie andere SOAP Implementierungen**

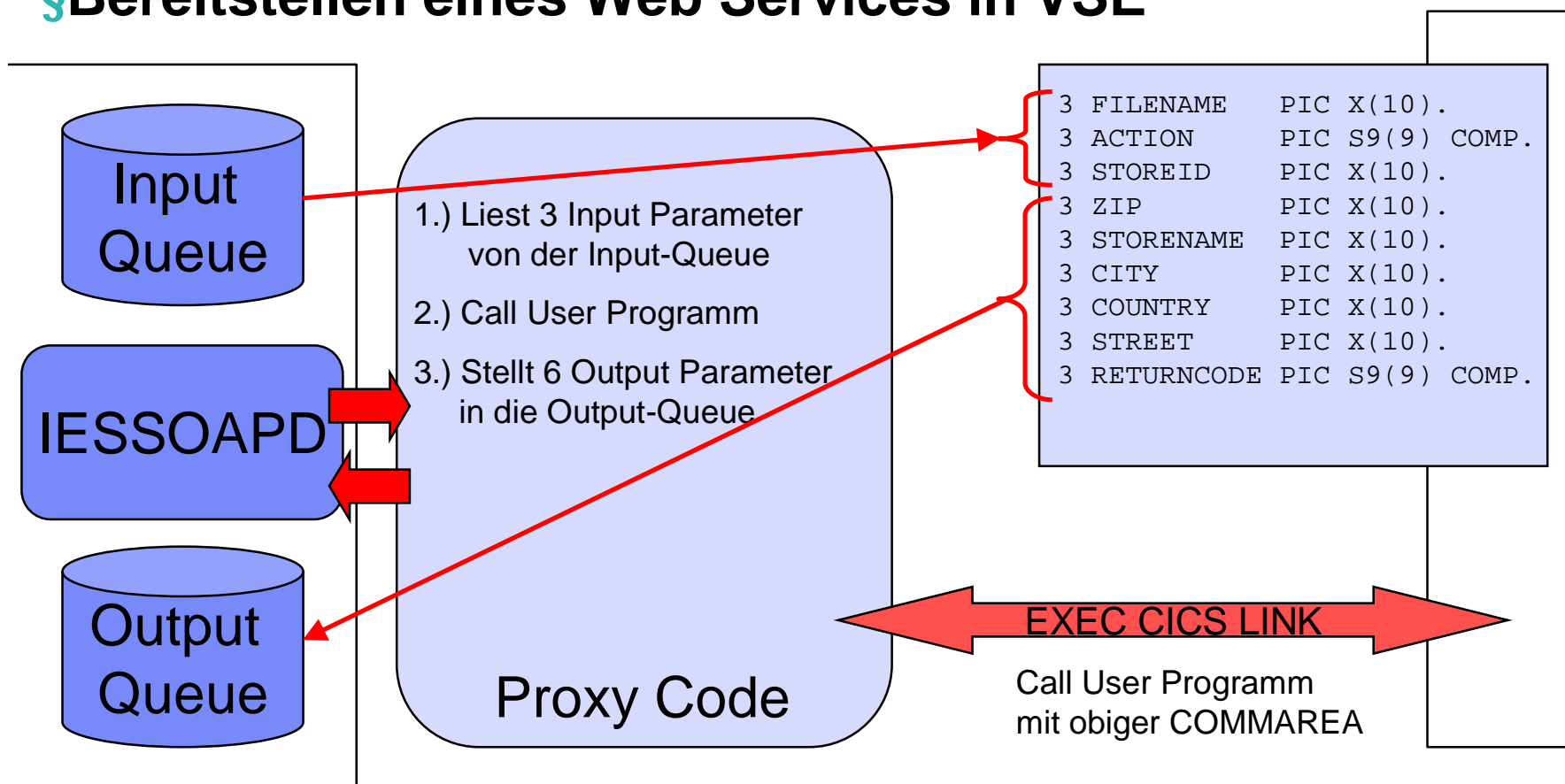
# Was genau macht der Proxy-Code?

## § Aufruf eines externen Web Services



## Was genau macht der Proxy-Code?

### § Bereitstellen eines Web Services in VSE



## Welche Programme können benützt werden?

### § Welche VSE Programme können als Web Service benützt werden?

- Alle CICS TS Programme die den “Service” implementieren den Sie bereitstellen wollen
  - In jeder Programmier-Sprache (COBOL, PL/1, C, Assembler)
- Sie sollten die Business-Logik vom User-Interface trennen
  - 3270 Screens oder BMS Maps können nicht verwendet werden
- Der Proxy-Code ruft Ihr Programm mit **EXEC CICS LINK** und einer **COMMAREA**

### § Welche VSE Programme können einen externen Web Service rufen?

- Alle CICS TS Programme die ein anders Programm mit einer COMMAREA aufrufen können
  - In jeder Programmier-Sprache (COBOL, PL/1, C, Assembler)
- Ihr Programm ruft den Proxy-Code mit **EXEC CICS LINK** und einer **COMMAREA**



# Agenda

- § **Was ist ein Web Service?**
- § **Begriffe und Theorie**
  - SOA, SOAP, WSDL, SOAP RPC, XML
- § **Web Services mit Java/.Net verwenden**
  - Proxy-Code
- § **Web Services mit VSE verwenden**
  - VSE SOAP Engine, Programming Interfaces
- § **CICS2WS Tool**
  - Live Demo
- § **Andere Goodies**

## Wie wird der Proxy-Code erzeugt?

### § Sie können den Proxy-Code “von Hand” schreiben

- Ist nicht sehr schwierig, anhand von Beispielen
- COBOL Beispiel (von Rich Smrcina):
  - [ftp://ftp.software.ibm.com/eserver/zseries/zos/vse/download/xmps/soap\\_cobol\\_rsmrcina.zip](ftp://ftp.software.ibm.com/eserver/zseries/zos/vse/download/xmps/soap_cobol_rsmrcina.zip)

### § CICS2WS Tool

- Generiert den Proxy-Code und WSDL Dateien
- Der Proxy-Code wird in Assembler erzeugt
  - Es wird kein zusätzlicher kostenpflichtiger Compiler benötigt (z.B. COBOL oder PL/I)
  - Der Code ist sehr einfach
  - Normalerweise muss der Code nicht angepasst werden
- <http://www.ibm.com/servers/eserver/zseries/zvse/downloads/>

# CICS2WS Tool

## § Das Tool läuft auf dem PC

## § Implementiert in Java

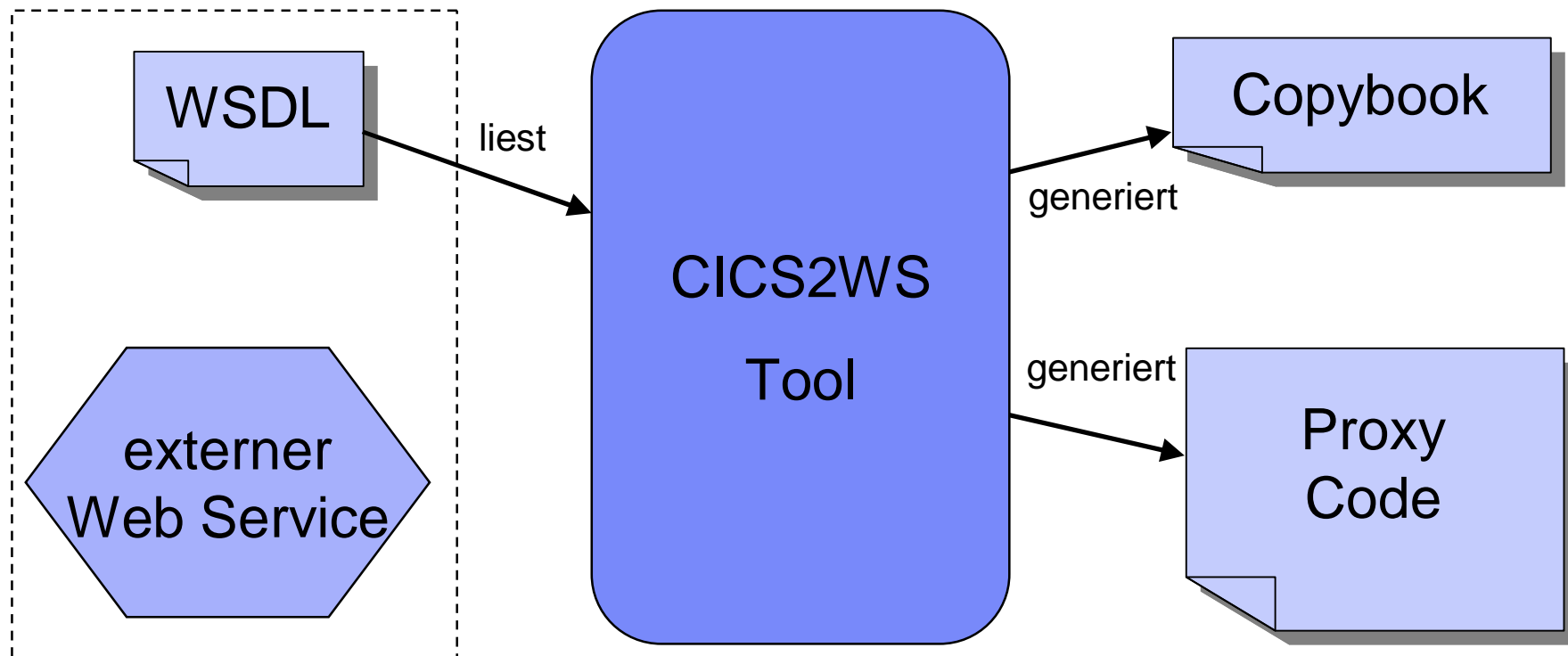
## § VSE als SOAP Client (Service Requestor)

- Liest die WSDL Datei ein
- Generiert den Proxy-Code (Assembler)
- Generiert ein COMMAREA Mapping (Copybook)
  - in COBOL, PL/I oder Assembler

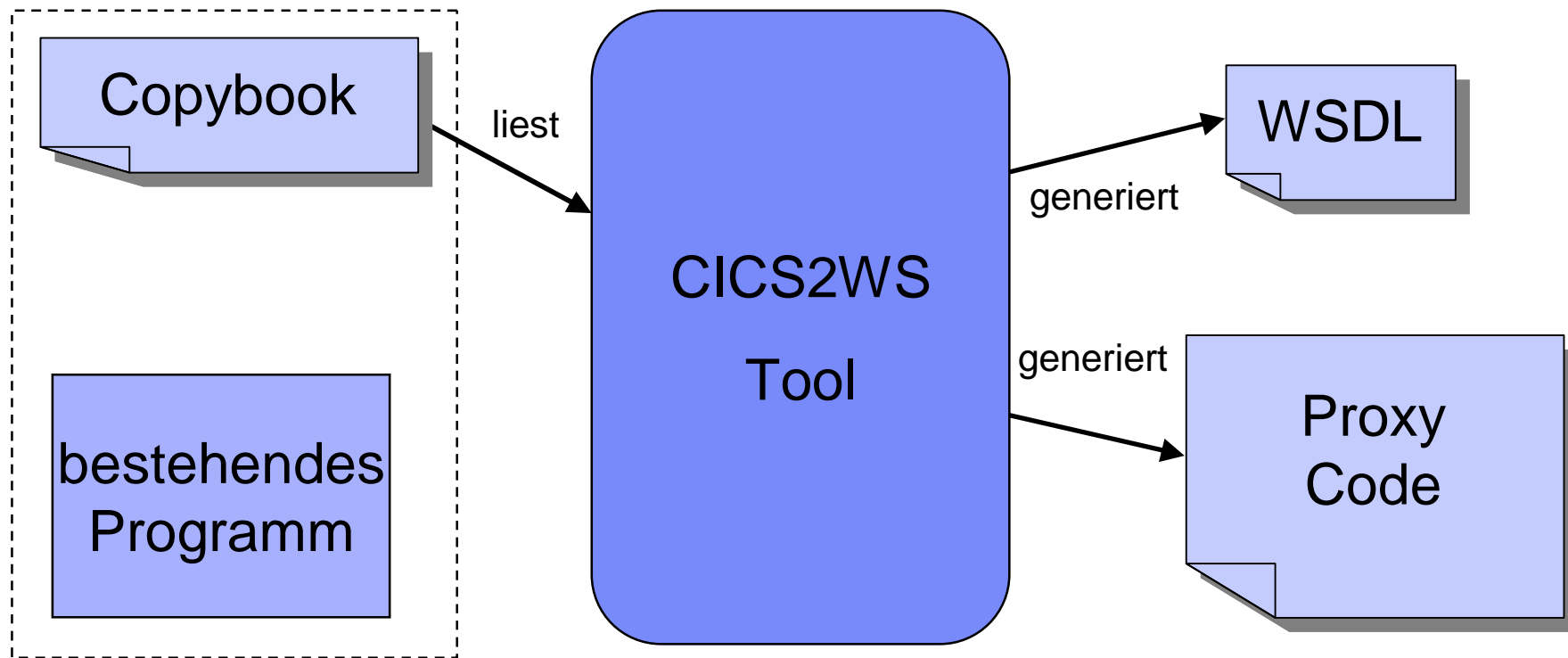
## § VSE als SOAP Server (Service Provider)

- Liest ein vorhandenes COMMAREA Mapping ein (Copybook)
  - in COBOL, PL/I oder Assembler
- Generiert den Proxy-Code (Assembler)
- Generiert die WSDL Datei

## VSE als SOAP Client (Service Requestor)



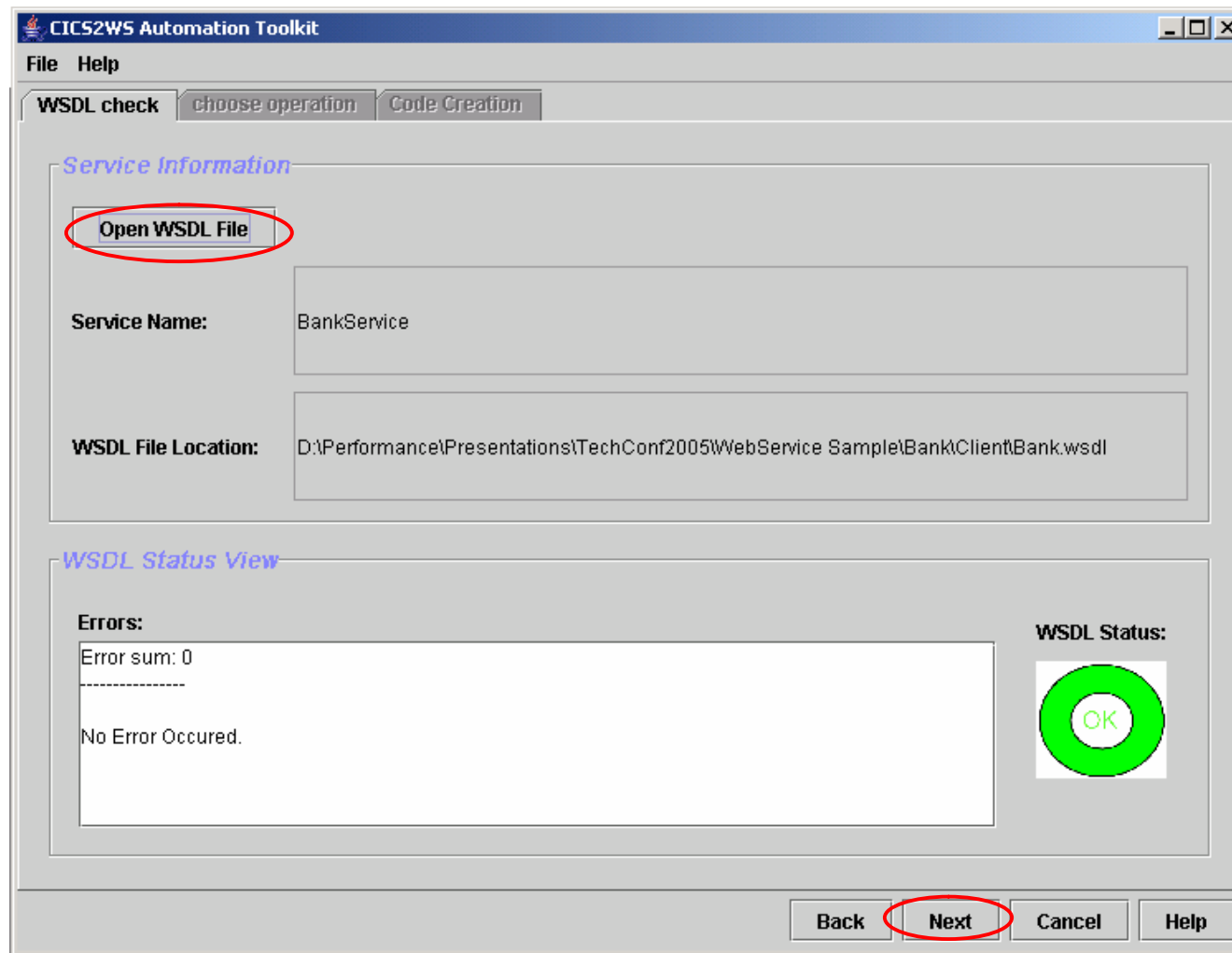
## VSE als SOAP Server (Service Provider)



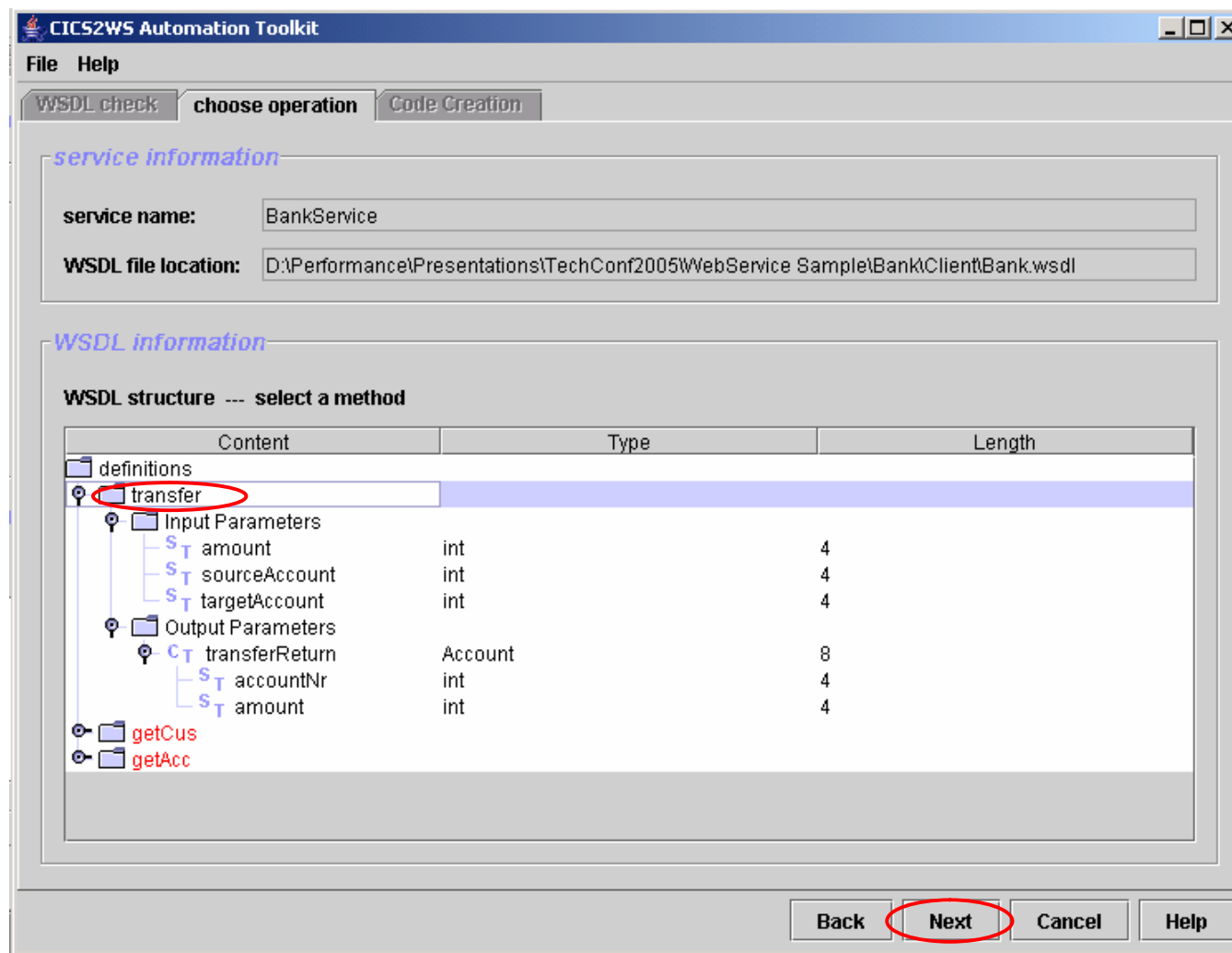
## Live Demo – Bestehenden Web Service benutzen



## Live Demo – Bestehenden Web Service benutzen

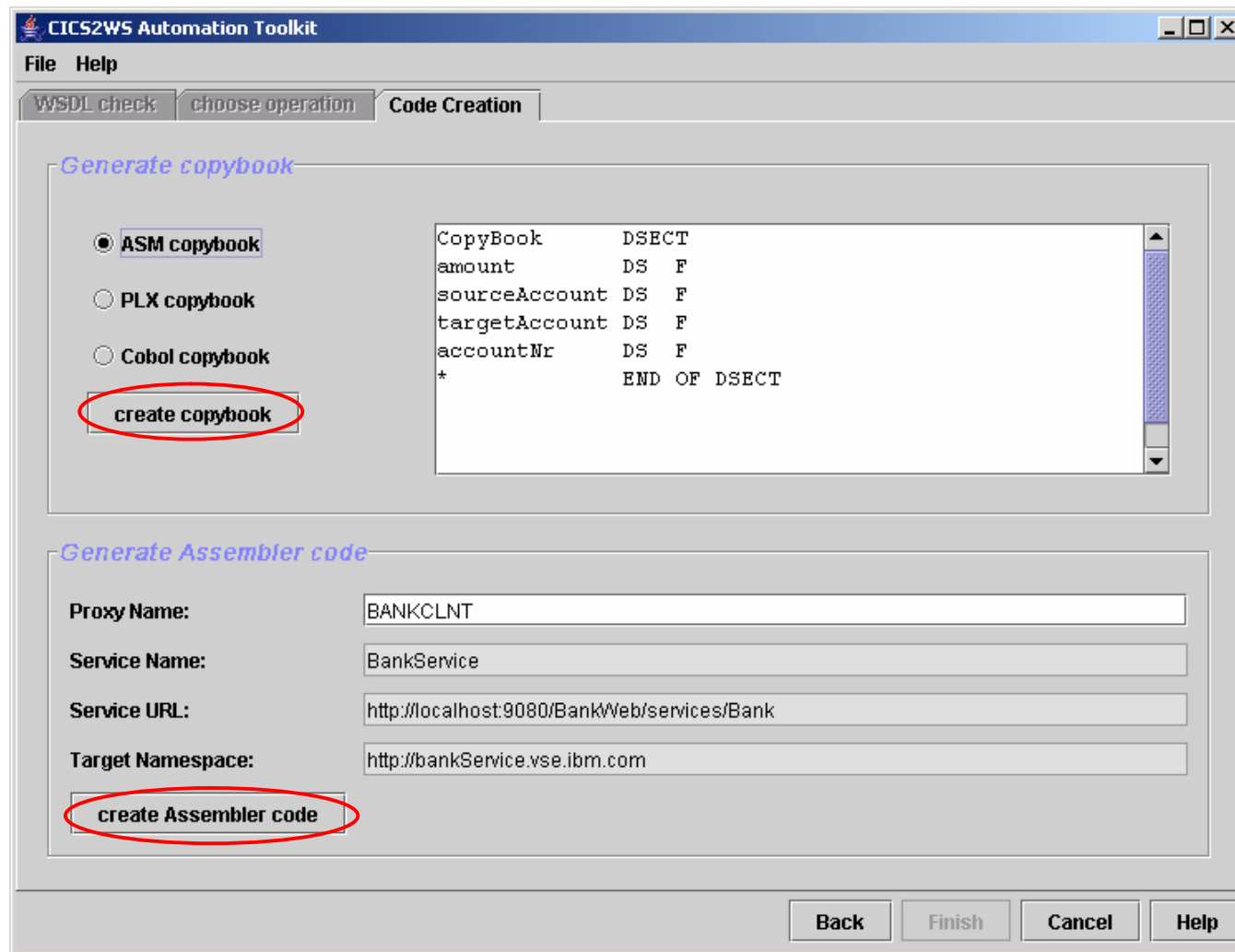


## Live Demo – Bestehenden Web Service benutzen





## Live Demo – Bestehenden Web Service benutzen



# Live Demo – Bestehenden Web Service benutzen

```

CopyBook      DSECT
amount        DS   F
sourceAccount DS   F
targetAccount DS   F
accountNr     DS   F
*             END OF DSECT
    
```

```

*****
MIC STORAGE SECTION
*****
A      ADDR OF PARAMETER BLOCK
H      LENGTH OF BLOCK
F      RESPONSE CODE
F      RESPONSE CODE 2
XL(CLNCALEN) COMMAREA FOR SOAP CALL
XL44 BUFFER FOR OUTPARAMS
      DS   F      LENGTH OF PARAM 19
IP19_PTR    DS   A      PTR OF PARAM 19
*****
* *****
* END OF DYNAMIC STORAGE SECTION
* *****
BANKCLNT AMODE 31
BANKCLNT RMODE ANY
BANKCLNT CSECT
* *****
* START OF PROGRAM SECTION
* *****
      DFHEIENT CODEREG=(R3),      Base registers for program code      X
      DATAREG=(R13),             Base register for data                X
      EIBREG=(R11)               Base register for CICS EIB
*
      USING BANKCLNT+4096,R4
      LA   R4,4095(R3)
...
    
```

## Live Demo – Web Service erzeugen



## Live Demo – Web Service erzeugen

**CICS2WS Automation Toolkit**

File Help

create Service | add operation to service | service summary

*CICS application data*

Path to source: P005WebService SampleBankServerBankCB.ch browse...

source code language: COBOL

*basic service data*

service name: VSEBANKService

service URL: http://sdedemo.boeblingen.de.ibm.com:1080/cics/CWBA/IESSOAPS

URN: urn:IESSOAPD:BANKSERV

Service description:  
Sample VSE Web Service. Uses CICS program VSEBANK as backend.

*program names*

Proxy Name: BANKSERV USER Program Name: VSEBANK

Back Next Cancel Help

# Live Demo – Web Service erzeugen

**CICS2WS Automation Toolkit**

File Help

create Service | **add operation to service** | service summary

*new operation*


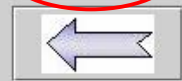
**name:** transfer

**Description:**  
Transfers money from one account to another.

*input/output parameter mapping*

**COMMAREA variables:**

| Content            | Instance | Type    | Length | Offset |
|--------------------|----------|---------|--------|--------|
| commarea variables |          |         |        |        |
| CT CUSTOMER        | Group    |         | 44     | 0      |
| CT ACCOUNT         | Group    |         | 8      | 44     |
| ST ACCOUNTNRField  |          | INTEGER | 4      | 0      |

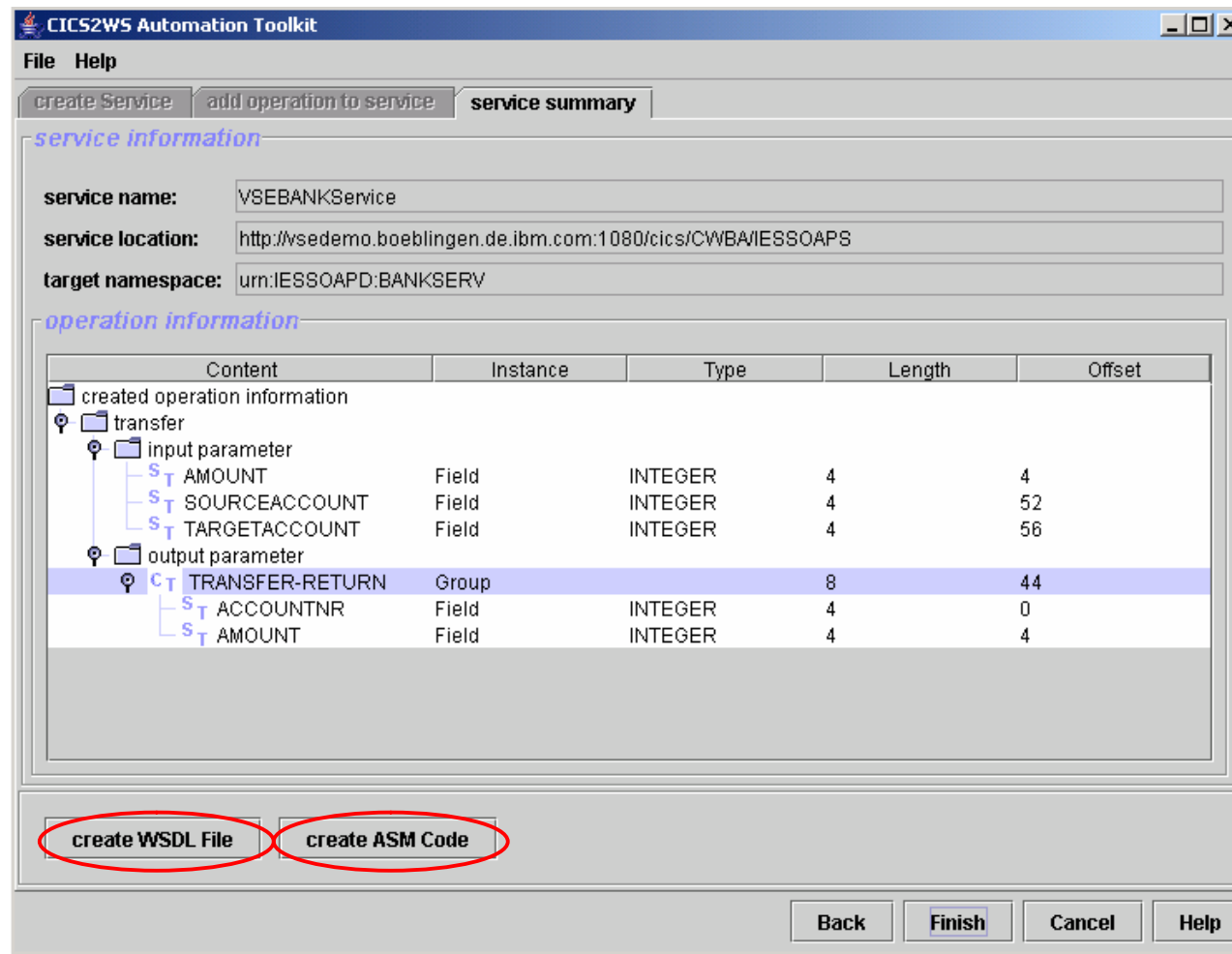



**mapped variables**

| Content            | Instance | Type    | Length | Offset |
|--------------------|----------|---------|--------|--------|
| input parameter    |          |         |        |        |
| ST AMOUNT          | Field    | INTEGER | 4      | 4      |
| ST SOURCEACCOUNT   | Field    | INTEGER | 4      | 52     |
| ST TARGETACCOUNT   | Field    | INTEGER | 4      | 56     |
| output parameter   |          |         |        |        |
| CT TRANSFER-RETURN | Group    |         | 8      | 44     |
| ST ACCOUNTNR       | Field    | INTEGER | 4      | 0      |

Back Next Cancel Help

# Live Demo – Web Service erzeugen



# Live Demo – Web Service erzeugen

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions targetNamespace="urn:IESSOAPD:BANKSERV" xmlns:impl="urn:IESSOAPD:BANKSERV"
  xmlns:intf="urn:IESSOAPD:BANKSERV" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <wsdl:types>
  - <schema elementFormDefault="qualified" targetNamespace="urn:IESSOAPD:BANKSERV"
    xmlns:impl="urn:IESSOAPD:BANKSERV" xmlns:intf="urn:IESSOAPD:BANKSERV"
    xmlns="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/">
    <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
    <element name="AMOUNT" type="xsd:int" />
    <element name="SOURCEACCOUNT" type="xsd:int" />
    <element name="TARGETACCOUNT" type="xsd:int" />
    <element name="TRANSFER-RETURN" type="impl:TRANSFER-RETURN" />
  - <complexType name="TRANSFER-RETURN">
    - <sequence>
      <element name="ACCOUNTNR" type="xsd:int" />
      <element name="AMOUNT" type="xsd:int" />
    </sequence>
  </complexType>
```

```
*****
PAGE SECTION
*****
OR OF PARAMETER BLOCK
NGTH OF BLOCK
SPONSE CODE
SPONSE CODE 2
FFER FOR OUTPARAMS

*****
START OF USER PROGRAM COMMAREA
*BankCB.cbl DSECT
CCUSTOMERNR DS A
CFIRSTNAME DS CL20
CLASTNAME DS CL20
CACCOUNTNR DS A
CAMOUNT DS A
CSOURCEACCOUNT DS A
CTARGETACCOUNT DS A
* END OF DSECT
CARLEN EQU *-COMMAREA LENGTH OF COMMAREA
* *****
* END OF DYNAMIC STORAGE SECTION
* *****
BANKSERV AMODE 31
BANKSERV RMODE ANY
BANKSERV CSECT
...
```

# Agenda

- § **Was ist ein Web Service?**
- § **Begriffe und Theorie**
  - SOA, SOAP, WSDL, SOAP RPC, XML
- § **Web Services mit Java/.Net verwenden**
  - Proxy-Code
- § **Web Services mit VSE verwenden**
  - VSE SOAP Engine, Programming Interfaces
- § **CICS2WS Tool**
  - Live Demo
- § **Andere Goodies**



## Andere Goodies

§ **VSE Health Checker**

§ **Keyman/VSE**

§ **TCP/IP Configuration Dialog**

§ **VSE Navigator: Compile-Helper-Plugin**

§ <http://www.ibm.com/servers/eserver/zseries/zvse/downloads/>

# VSE Health Checker

**VSE Health Checker - Data loaded from: VSECON(vsecon.boeblingen.de.ibm.com)**

File Selected Configuration Help

**Storage Below** | Storage Above | CDSA / SDSA | RDSA / UDSA | ECDSA / ESDSA | ERDSA / EUDSA

| Area                 | Size (KB) | Size (MB) |
|----------------------|-----------|-----------|
| Free GETVIS-24       | 2376      | 2.32      |
| Used GETVIS-24...    | 3768      | 3.67      |
| Free DSA             | 3840      | 3.75      |
| User DSA (UDSA)      | 256       | 0.25      |
| Readonly DSA (...)   | 256       | 0.25      |
| Shared DSA (SD...)   | 256       | 0.25      |
| CICS DSA (CDSA)      | 512       | 0.50      |
| Partition Program... | 4         | 0.00      |

Job Name: CICSICCF | Peak DSA Size: 1280 KB  
 Phase Name: DFHSIP | Peak DSA Used: 784 KB  
 Part. Begin Addr.: 500000 | DSA Limit: 5120 KB  
 Storage Prot.: Active | Max. Ever Used GETVIS-24: 11260 KB

Note: issue a GETVIS nn,RESET after CICS startup to get a correct max\_ever\_used\_24 value.

Legend:  
 ■ Public Key (90)  
 ■ Partition Key (B0)  
 ■ Readonly Key (00)

# Keyman/VSE

The screenshot displays the Keyman/VSE application window with the following data:

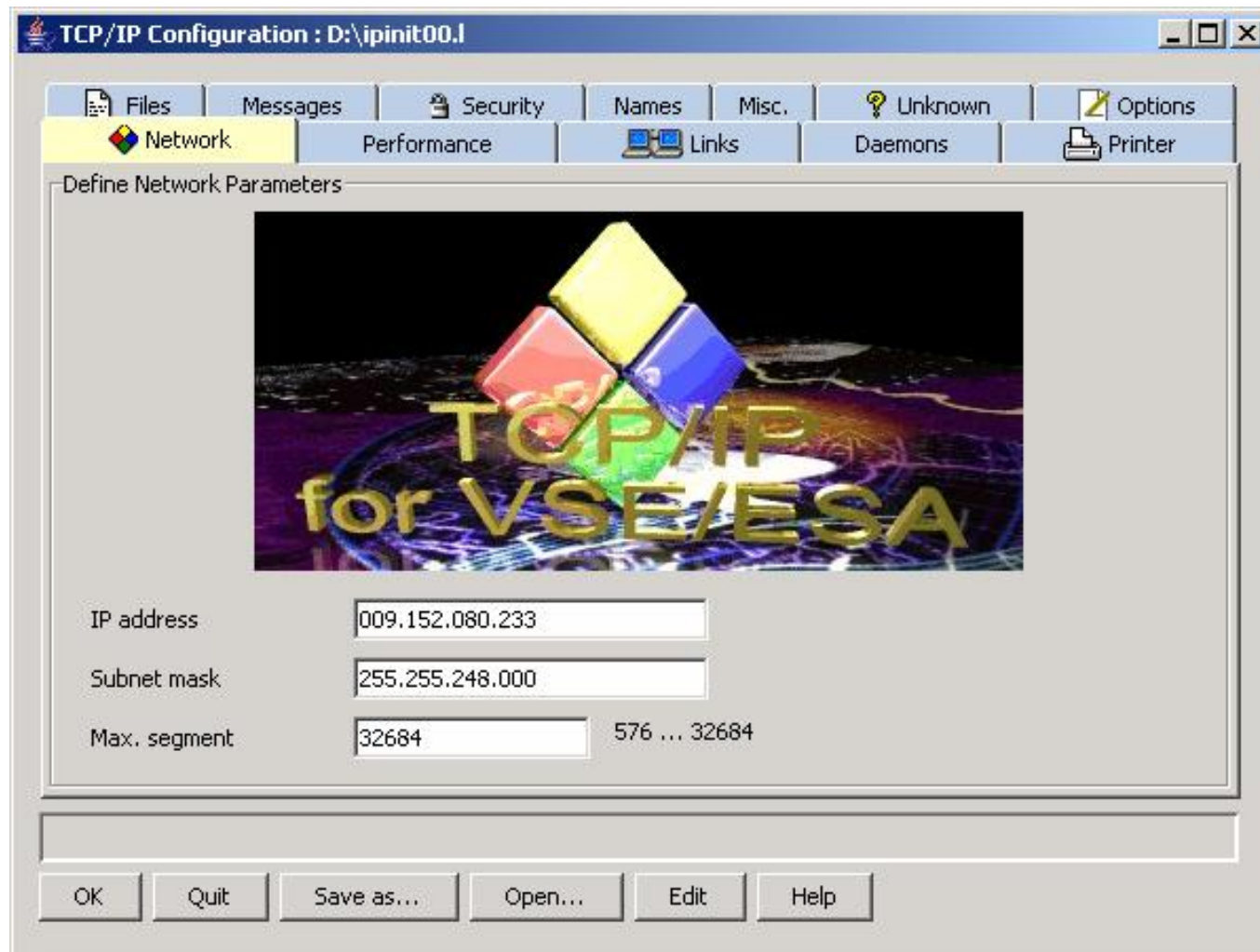
| Certificate Item            | Length | Type | VSE User | Valid |
|-----------------------------|--------|------|----------|-------|
| VSE/ESA Development CA ROOT | 1024   | Root | -        | Yes   |
| VSE Server Certificate      | 1024   | User | -        | Yes   |

The VSEFRAN2 Keyring Library - CRYPTO.KEYRING window shows the following data:

| Member Name | Type | Creation              | Last... |
|-------------|------|-----------------------|---------|
| SSLTEST     | CERT | 03.09.2003 - 09:12:58 | --      |
| SSLTEST     | PRVK | 03.09.2003 - 09:12:49 | --      |
| SSLTEST     | ROOT | 03.09.2003 - 09:12:55 | --      |

# TCP/IP Configuration Dialog



# VSE Navigator: Compile-Helper-Plugin

The screenshot shows the VSE Navigator application window. On the left is a tree view of the system structure, including jobs like DEMOUSA, DV53, and VSEFRAN2. A table in the background lists jobs with columns for Job, Disposition, Class, Priority, Records, and Pages. The 'Compile Helper Plugin' dialog is open, showing a search for 'COMPCHPL.20759' using the keyword scheme 'E:\VSECon\c-compile.scheme'. The search results show two errors:

```

---> COMPCHPL.20759 [0] at line 251
174      |      char  chActClass; /* Job Class of Entry */
175      |          POWER_ENTRY Entry;
=====> .....a.....
*-ERROR=====> a - EDC0350 SYNTAX ERROR.
176      |      char  bFound; /* at least 1 valid Entry found */
177      |          /* Line Buffer for 7 Lines */
178      |      char  szLine1[256];
---> COMPCHPL.20759 [0] at line 358
265      |      if(POWERMainEntryPoint==NULL)
265      31 + if(POWERMainEntryPoint==(void *)0)
=====> .....a.....
*-ERROR=====> a - EDC0041 IDENTIFIER POWERMainEntryPoint MUST BE DECLARED BEFORE IT
    
```

# Dokumentation

## § Web Services in VSE (from Rich Smrcina)

- <http://www.zjournal.com/Article.asp?ArticleId=1059>
- Includes COBOL sample code

## § Web Services

- <http://www.ibm.com/servers/eserver/zseries/zvse/documentation/ebusiness.html#soap>

## § What is SOA?

- <http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>

## § z/VSE e-business Connectors, User's Guide (SC33-8231)

## § Downloads:

- <http://www.ibm.com/servers/eserver/zseries/zvse/downloads/>

# Fragen ?

