



# E19

## Implementing Web Services in VSE

Ingo Franzki (ifranzki@de.ibm.com)

**zSeries® EXPO**

**FEATURING Z/OS, Z/VM, Z/VSE  
AND LINUX ON ZSERIES**

September 19 - 23, 2005

San Francisco, CA

# Trademarks

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

CICS*	IBM*	Virtual Image
DB2*	IBM logo*	Facility
DB2 Connect	IMS	VM/ESA*
DB2 Universal	Intelligent	VSE/ESA
Database	Miner	VisualAge*
e-business logo*	Multiprise*	VTAM*
Enterprise Storage	MQSeries*	WebSphere*
Server	OS/390*	xSeries
HiperSockets	S/390*	z/Architecture
	SNAP/SHOT	z/VM
	*	z/VSE
		zSeries

\* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

LINUX is a registered trademark of Linus Torvalds

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 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.

# Agenda

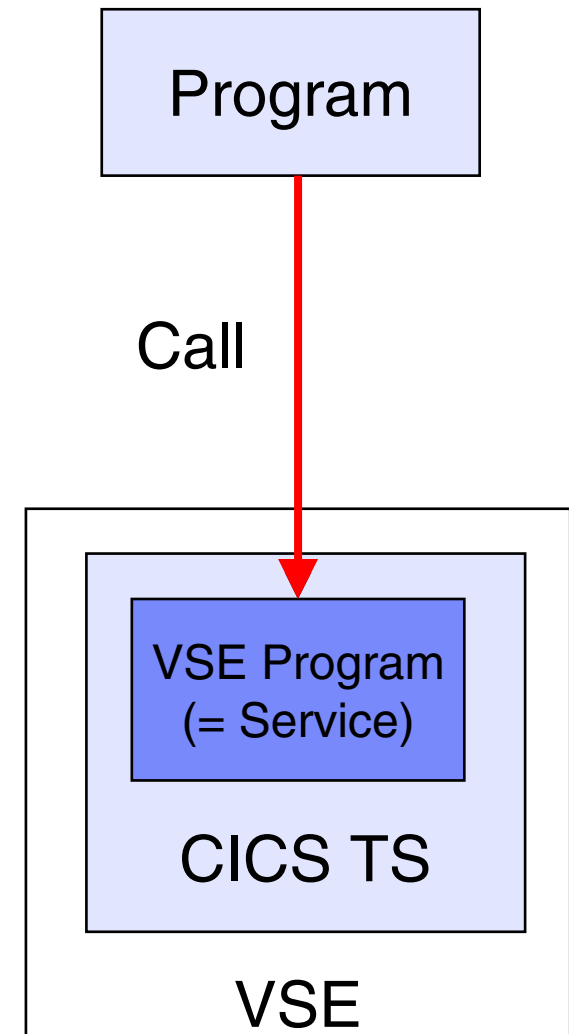
## What is a Web Service?

### Terms and theory

- SOA, SOAP, WSDL, SOAP RPC, XML
- **Using Web Services with Java/.Net**
  - Proxy code
- **Using Web Services with VSE**
  - VSE SOAP Engine, programming Interfaces
- **CICS2WS Tool**
  - Live demo

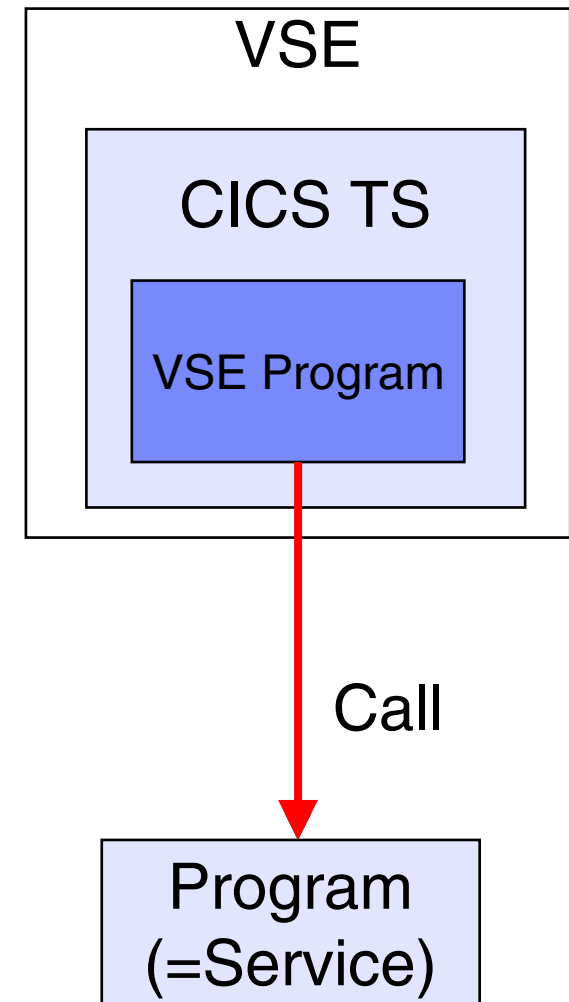
## What is a Web Service?

- **Assume you have a VSE program that implements some kind of important business logic**
- **Someone else (outside VSE) wants to use this program**
  - 1. Possibility: Rewrite the same logic
    - May need access to VSE data
    - Changes/Fixes in VSE code needs to be re-done in new code also
  - 2. Possibility: Call the VSE program from remote
    - VSE program can be treated as a **Web Service**
    - VSE is the Web Service provider



## What is a Web Service?

- Assume **someone has a program** that implements some kind of important business logic
- You want to use this program inside a VSE application
  - 1. Possibility: Rewrite the same logic
    - May need access to the remote data
    - Changes/Fixes in code needs to be re-done in VSE code also
  - 2. Possibility: Call the external program from VSE
    - External program can be treated as a **Web Service**
    - VSE is the Web Service Requestor



# Agenda

## What is a Web Service?

### Terms and theory

- SOA, SOAP, WSDL, SOAP RPC, XML
- **Using Web Services with Java/.Net**
  - Proxy code
- **Using Web Services with VSE**
  - VSE SOAP Engine, programming Interfaces
- **CICS2WS Tool**
  - Live demo

## What is Service-Oriented Architecture (SOA)?

- **"Things should be made as simple as possible, but no simpler." - Albert Einstein**
- **Einstein made that famous statement many decades ago, and it's still relevant today for building superior software systems.**
- **Unfortunately, as anyone who has been in the IT industry for long can point out, far too many software systems have failed Einstein's test.**
  - Some are made too simple to carry out the duties they are supposed to perform.
  - Others are made too complex, and the costs of building and maintaining them have rocketed, not to mention the nearly impossible tasks of integrating different systems together.
- **It seems that reaching the right level of simplicity is more like a dream than reality.**
- **Where have we gone wrong?**



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

## What is Service-Oriented Architecture (SOA)?

- **As we build more and more software systems, we see similar situations and patterns appearing:**
- **Naturally, we want to reuse the functionality of existing systems rather than building them from scratch.**
- **A **real dependency** is a state of affairs in which one system depends on the functionality provided by another.**
- **If the world only contained real dependencies, Einstein's test would have been satisfied long time ago.**
- **The problem is that we also create **artificial dependencies** along with real dependencies.**

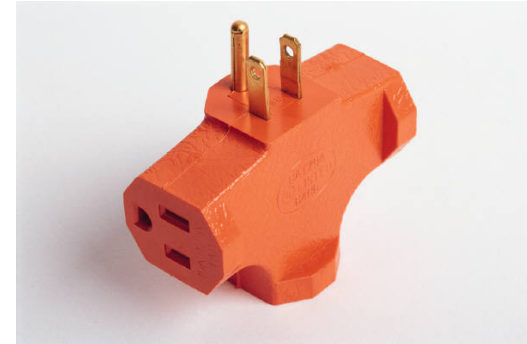
Taken from: <http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>



## What is Service-Oriented Architecture (SOA)?

### ■ An example:

- If you travel overseas on business, you know that you must bring power adapters along with you or your life will be miserable.
- The **real dependency** is that you need power.
- The **artificial dependency** is that your plug must fit into the local outlet.
- Looking at all the varying sizes and shapes of those plugs from different countries, you would notice that some of them are small and compact while many others are big and bulky.



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

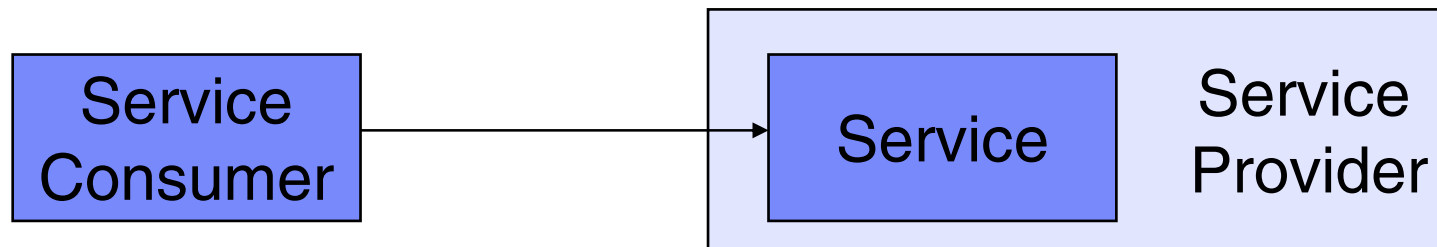
## What is Service-Oriented Architecture (SOA)?

- The lesson here is that we **cannot remove artificial dependencies**, but we can reduce them.
- If the artificial dependencies among systems have been reduced, ideally, to their minimum, we have achieved **loose coupling**.
- In that sense, Einstein was just talking about was loose coupling.
- We might rework his famous principle thus:  
**"Artificial dependencies should be reduced to the minimum but real dependencies should not be altered."**

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

## What is Service-Oriented Architecture (SOA)?

- **Now we are able to define a Service Oriented Architecture (SOA):**
  - SOA is an architectural style whose goal is to achieve **loose coupling** among interacting software agents.
  - A **service** is a **unit of work** done by a **service provider** to achieve desired end results for a service consumer.
  - Both **provider** and **consumer** are roles played by software agents on behalf of their owners.



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

## What is Service-Oriented Architecture (SOA)?

- **How does SOA achieve **loose coupling** among interacting software agents?**
- **It does so by employing two architectural constraints:**
  - A small set of **simple** and ubiquitous **interfaces** to all participating software agents.
    - Only generic semantics are encoded at the interfaces.
    - The interfaces should be universally available for all providers and consumers.
  - **Descriptive messages** constrained by an extensible schema delivered through the interfaces.
    - No, or only minimal, system behavior is prescribed by messages.
    - A schema limits the vocabulary and structure of messages.
    - An extensible schema allows new versions of services to be introduced without breaking existing services.

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

## What is a Web Service?

- **The definition of Web Service has always been under hot debate within the **W3C Web Services Architecture Working Group** (<http://www.w3.org/2002/ws/arch/>).**
- **A Web Service is a SOA with at least the following additional constraints:**
  - Interfaces must be based on Internet protocols such as **HTTP, FTP, and SMTP**.
  - Except for binary data attachment, messages must be in **XML**.
- **There are two main styles of Web Services:**
  - **SOAP Web Services** (Simple Object Access Protocol)
  - **REST Web Services** (Representational State Transfer)

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

## What is a Web Service?

- **A SOAP Web Service introduces the following constraints:**
  - Except for binary data attachment, messages must be carried by **SOAP** (Simple Object Access Protocol).
  - The description of a service must be in **WSDL** (Web Service Description Language).
- **A SOAP Web Service is the most common and marketed form of web service in the industry.**
  - Some people simply collapse "Web Service" into SOAP and WSDL services.
  - SOAP provides "a message construct that can be exchanged over a variety of underlying protocols".
  - In other words, SOAP acts like an **envelope that carries its contents**.
- **One advantage of SOAP is that it allows rich message exchange patterns ranging from traditional request-and-response to broadcasting and sophisticated message correlations.**

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

## What is a Web Service?

- **SOAP RPC Web Service**

- Encodes RPC (remote procedure calls) in SOAP messages.
- In other words, SOAP RPC "tunnels" new application-specific RPC interfaces through an underlying generic interface.

- **Document-centric SOAP Web Service**

- Carries any kind of (XML) document
- Works if the receiver 'knows' what to do with it

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

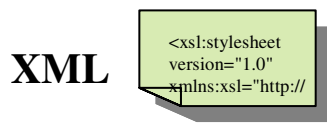
## What is WSDL?

- **Web Services Description Language (WSDL) provides a **model and an XML format** for **describing Web Services**.**
- **WSDL enables one to **separate the description of the abstract functionality** offered by a service from **concrete details** of a service description such as “**how**” and “**where**” that functionality is offered.**

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

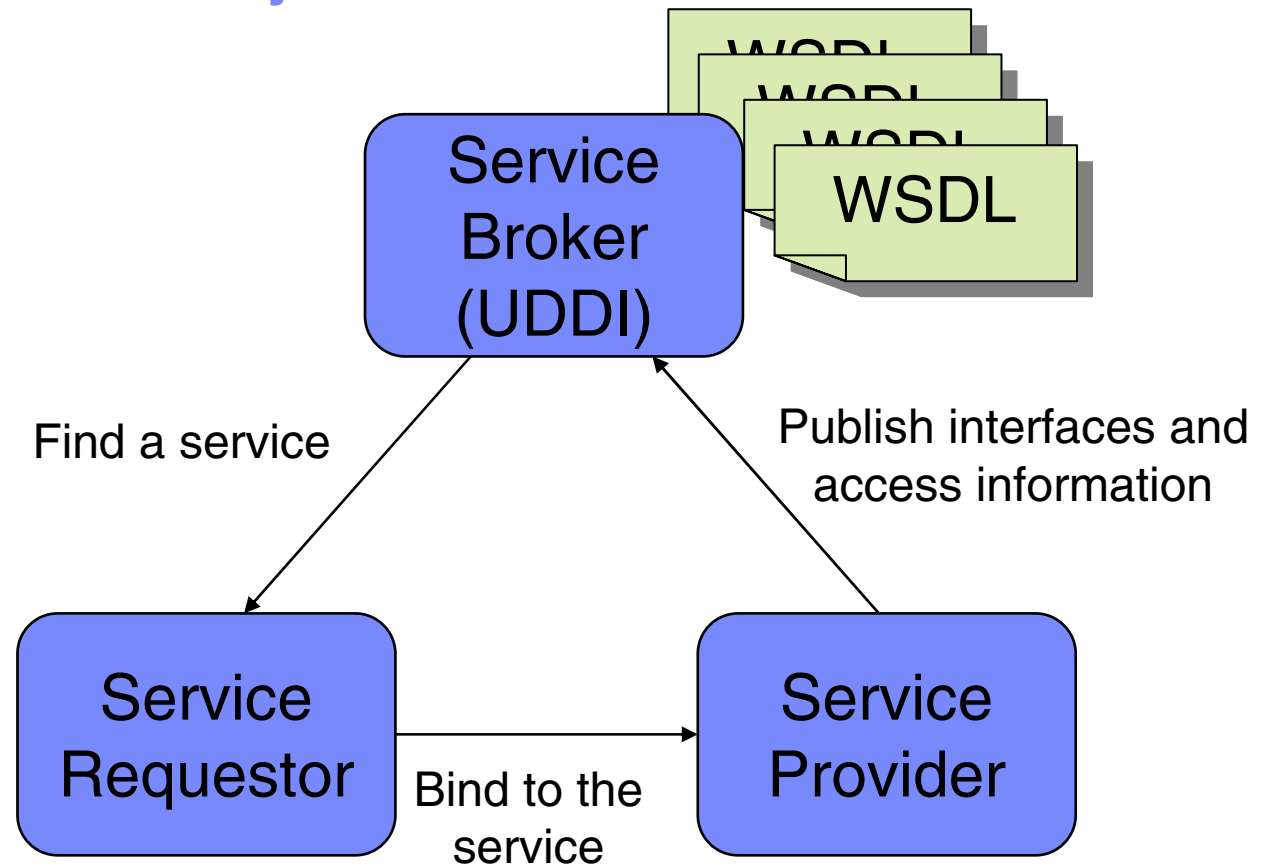


# Web Services - Summary



**HTTP - Transport Service**

**TCP/IP - Street**



## Web Services - Summary

### 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

## What is a Web Service?

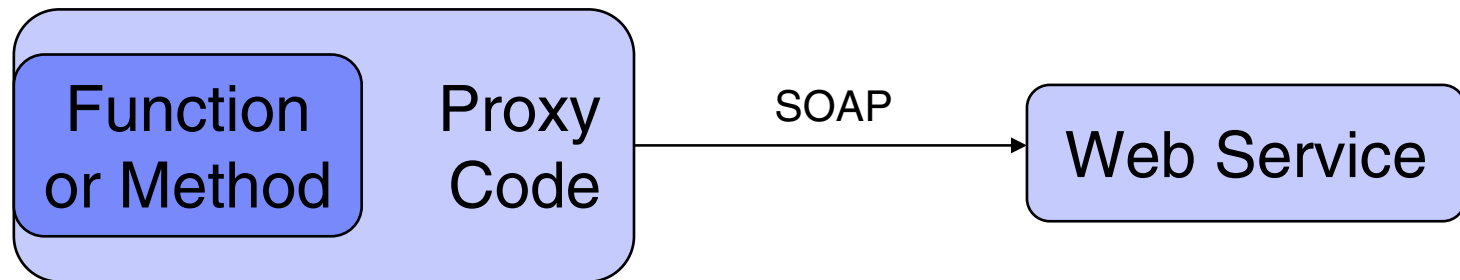
### Terms and theory

- SOA, SOAP, WSDL, SOAP RPC, XML
- **Using Web Services with Java/.Net**
  - Proxy code
- **Using Web Services with VSE**
  - VSE SOAP Engine, programming Interfaces
- **CICS2WS Tool**
  - Live demo

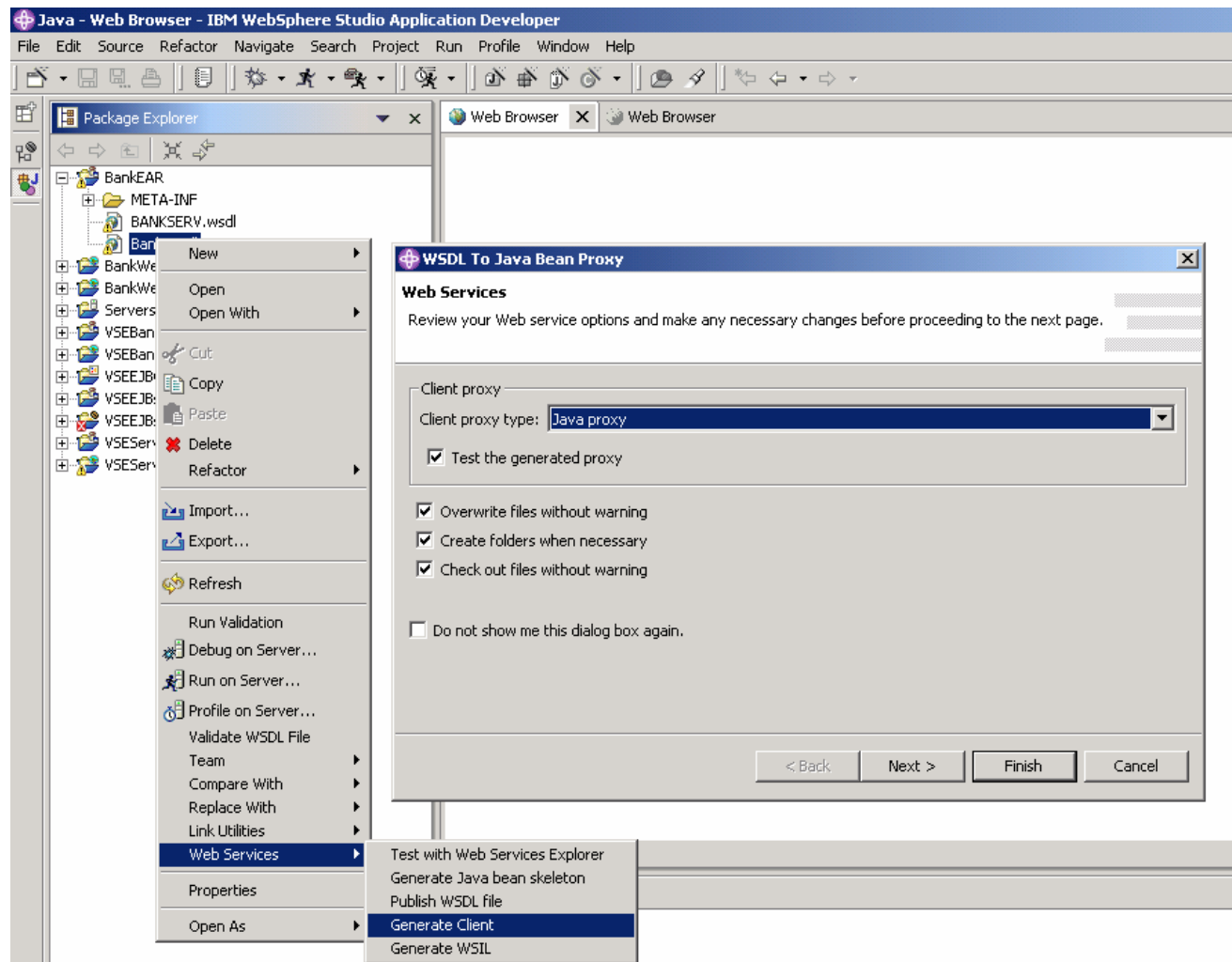
## Using Web Services with Java or MS .Net

### ■ **Use/Call an existing Web Service**

- You know that a specific Web Service exists
- Locate the Web Service Description (WSDL)
- Use a tool like Rational Application Developer (RAD/WSAD) or Microsoft .Net and import the WSDL
  - Generate “proxy code” that implements all things needed to invoke the Web Service
  - Applications will call a function or method of the proxy code as it would implement the service locally



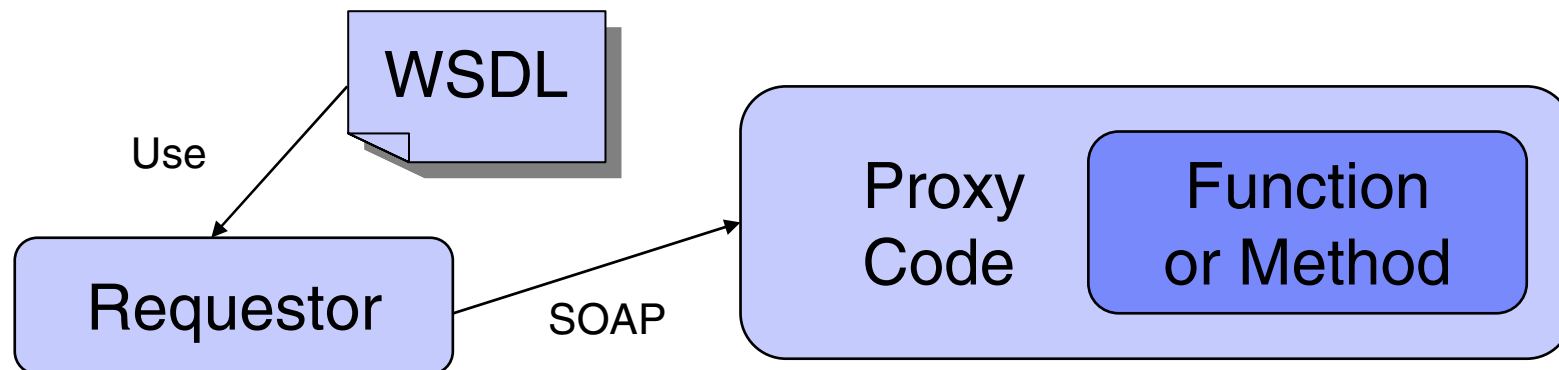
# Using Web Services with Java or MS .Net



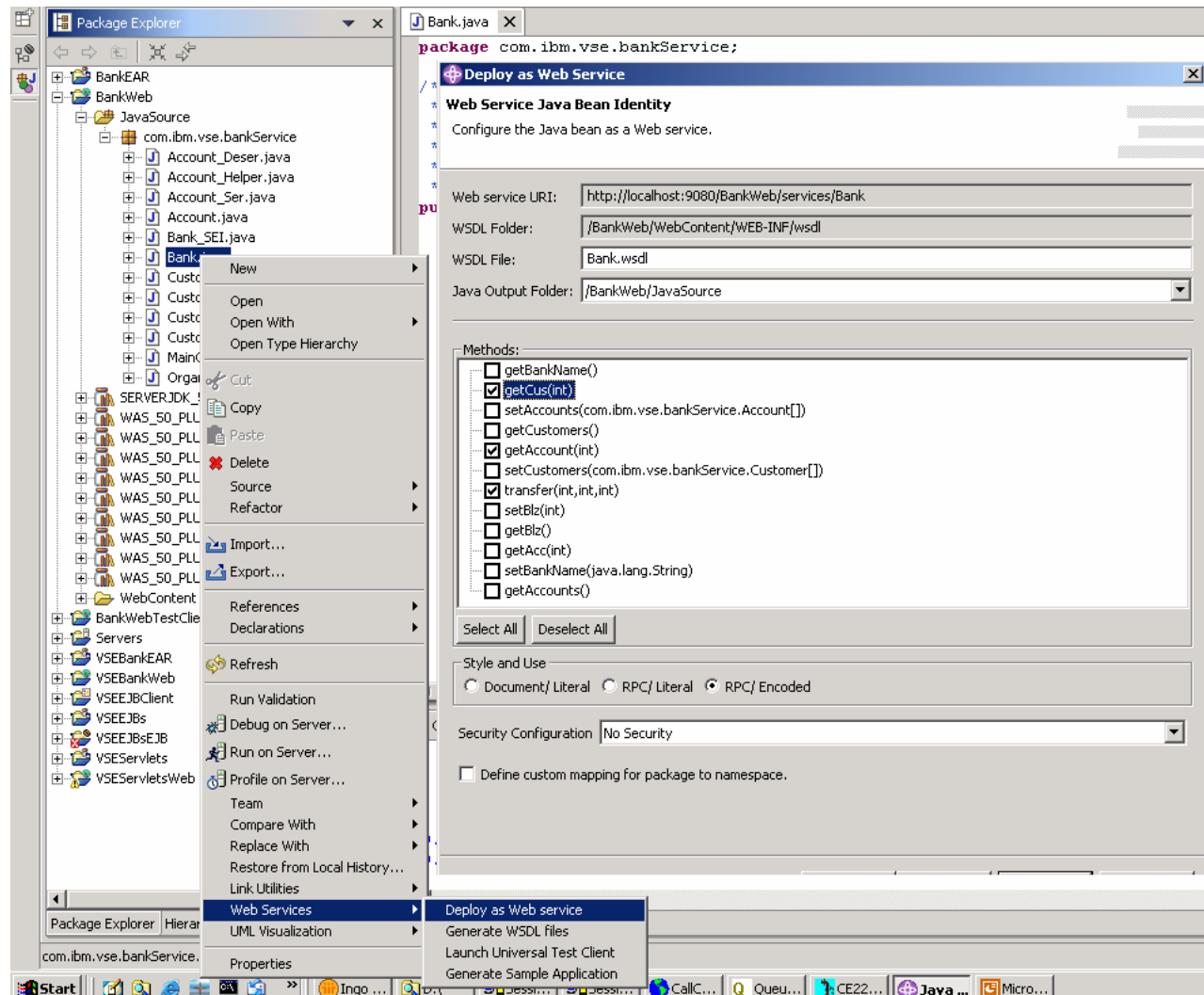
## Using Web Services with Java or MS .Net

- **Create/provide a new Web Service**

- You have a function or method that implements some kind of service that you want to provide
- Use a tool like Rational Application Developer (RAD/WSAD) or Microsoft .Net to model a Web Service
  - Generate a Web Service Description (WSDL) and publish it
  - Generate “proxy code” that makes the function or method callable from outside as a Web Service via SOAP
  - Deploy it in an application server



# Using Web Services with Java or MS .Net



# Agenda

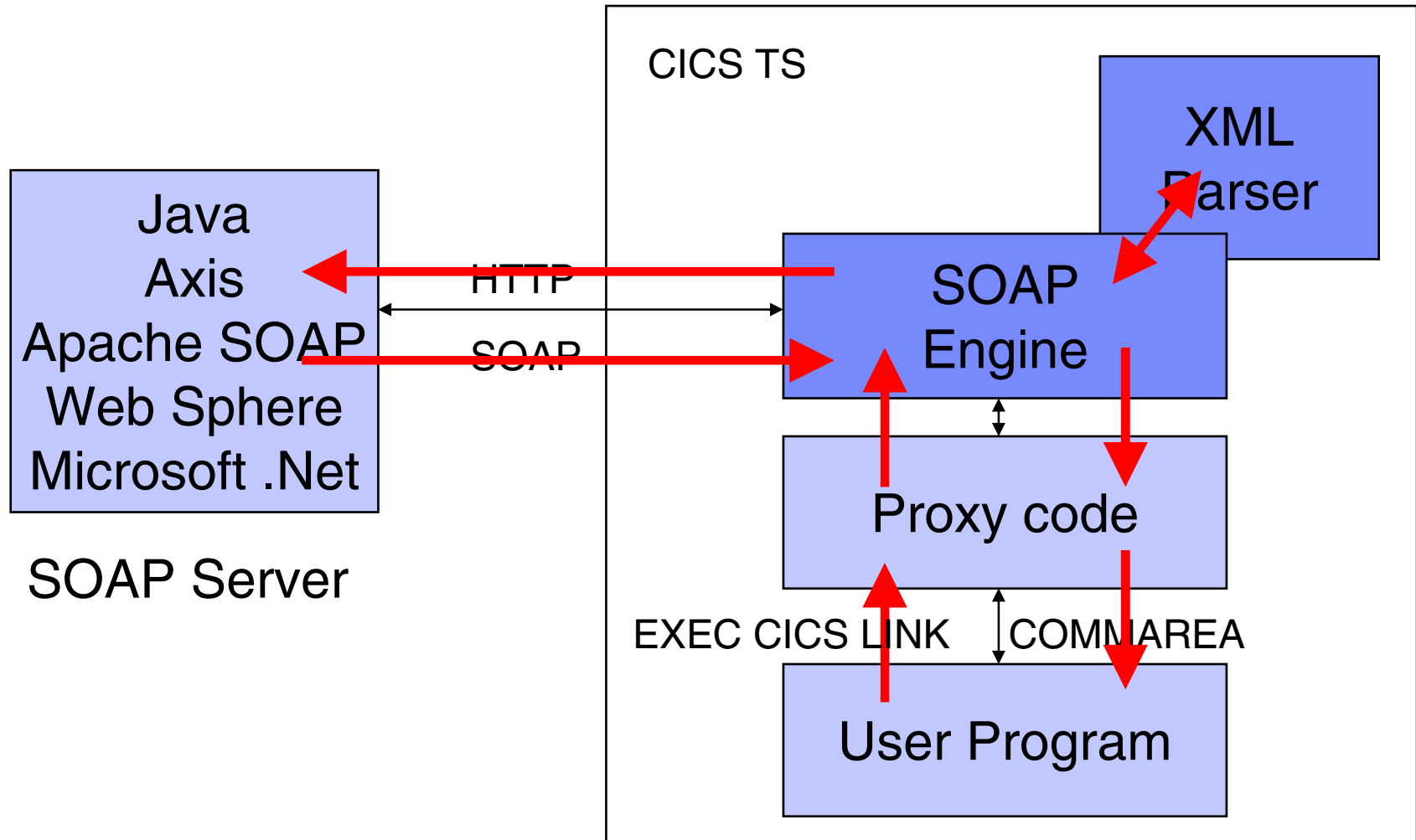
## What is a Web Service?

### Terms and theory

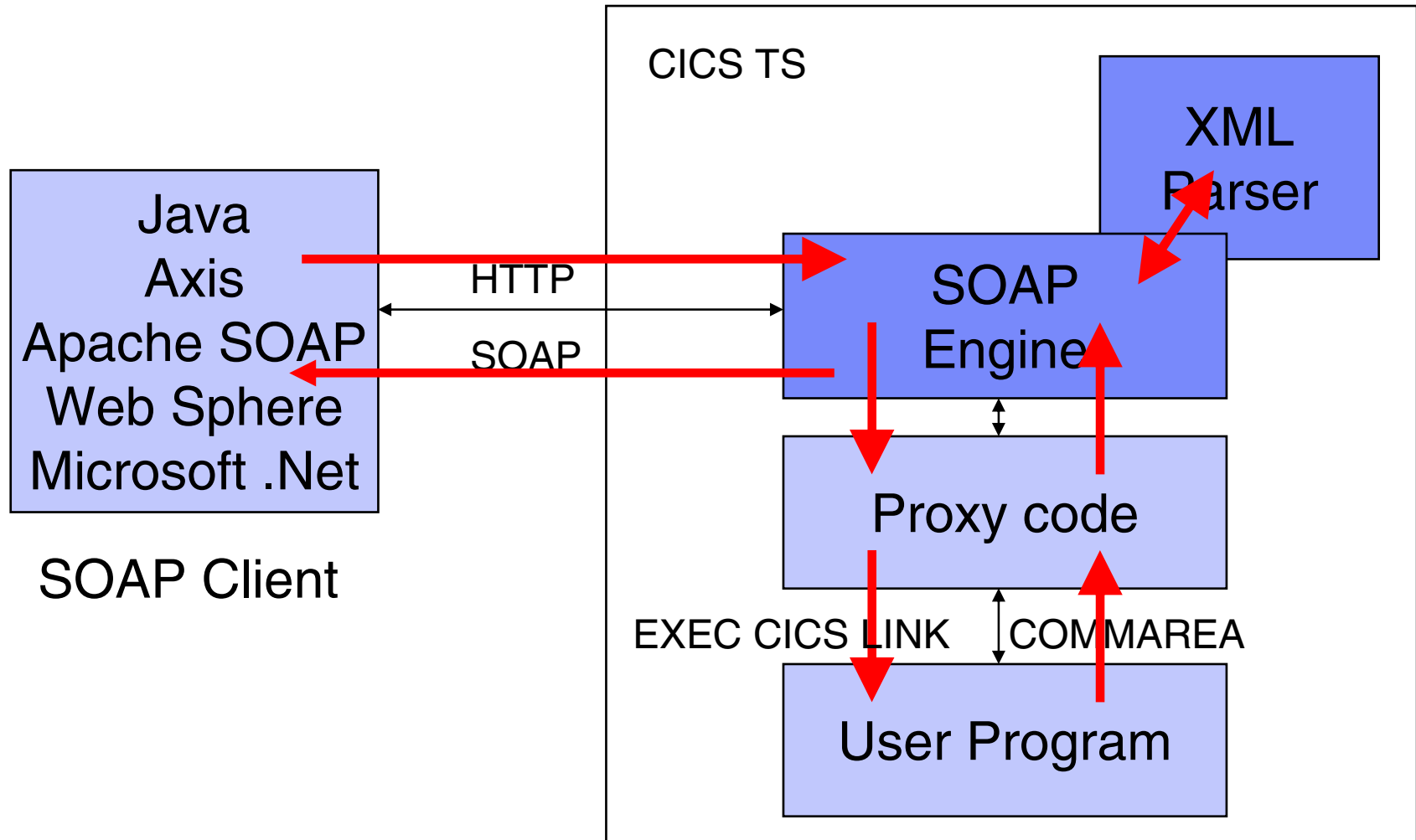
- SOA, SOAP, WSDL, SOAP RPC, XML
- **Using Web Services with Java/.Net**
  - Proxy code
- **Using Web Services with VSE**
  - VSE SOAP Engine, programming Interfaces
- **CICS2WS Tool**
  - Live demo



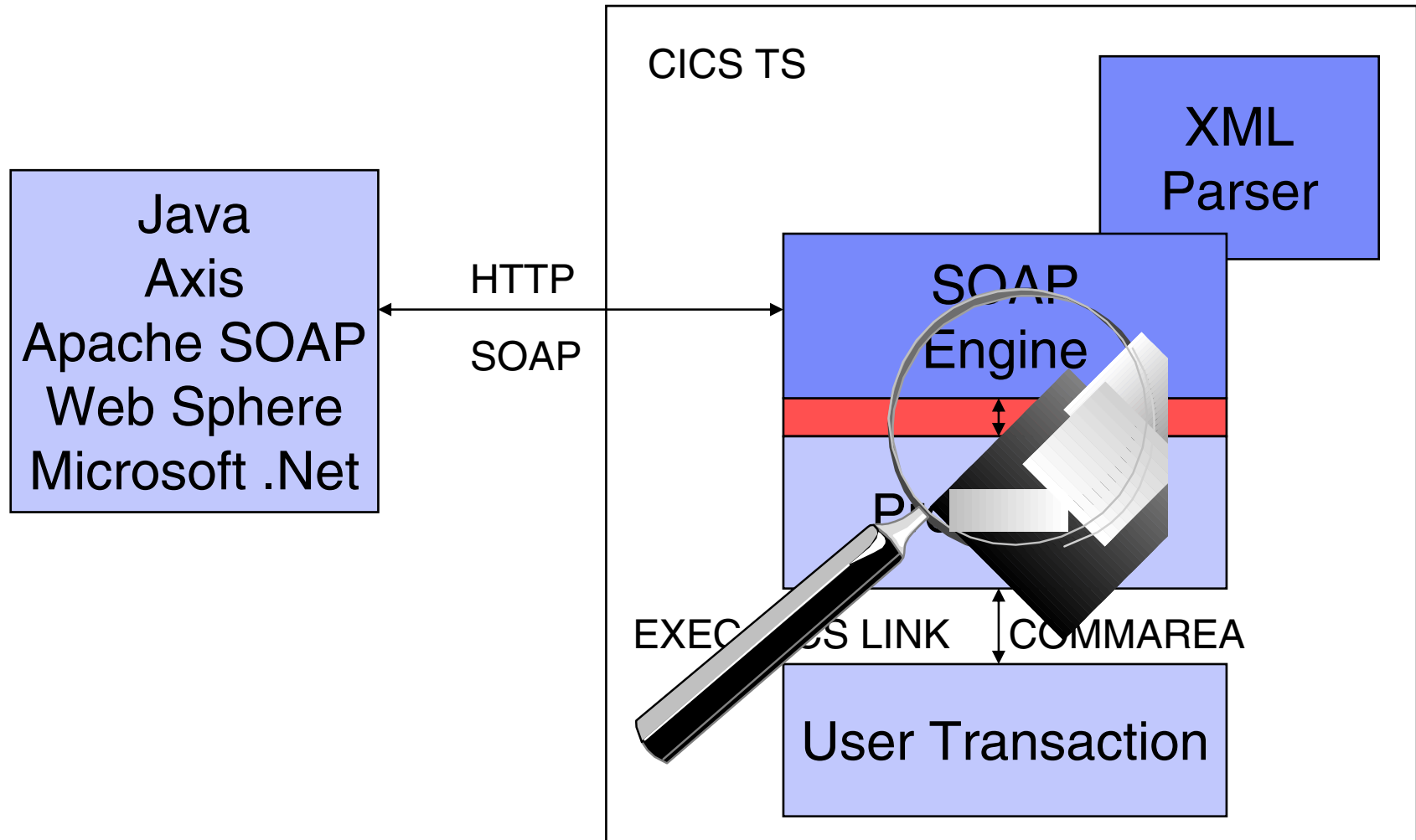
# Using Web Services with VSE – SOAP client



# Providing Web Services with VSE – SOAP server



# Using Web Services with VSE



## VSE SOAP Engine

- **Part of VSE Connectors**
- **Translates SOAP messages into/from a native format**
  - Receives/sends the SOAP message via HTTP
  - Parses/Generates XML
  - Interprets/creates the SOAP message
  - Serializes/de-serializes the input and output values
    - Translates it from textual format into a native format and vice versa
  - Calls service provider program
  - Gets called when invoking external Web Services

## VSE SOAP Engine

### ▪ Input/Output parameters

- Each parameter is represented by a TS-Queue entry
  - Parameter name (e.g. “StockPrice”)
  - Parameter type (e.g. “String”)
  - Parameter value (e.g. “34.5”)
  - Length of the parameter data
- Input parameters are put onto the Input-Queue
- Output parameters are read from the Output Queue

```
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 .
```

## Why use a proxy program?

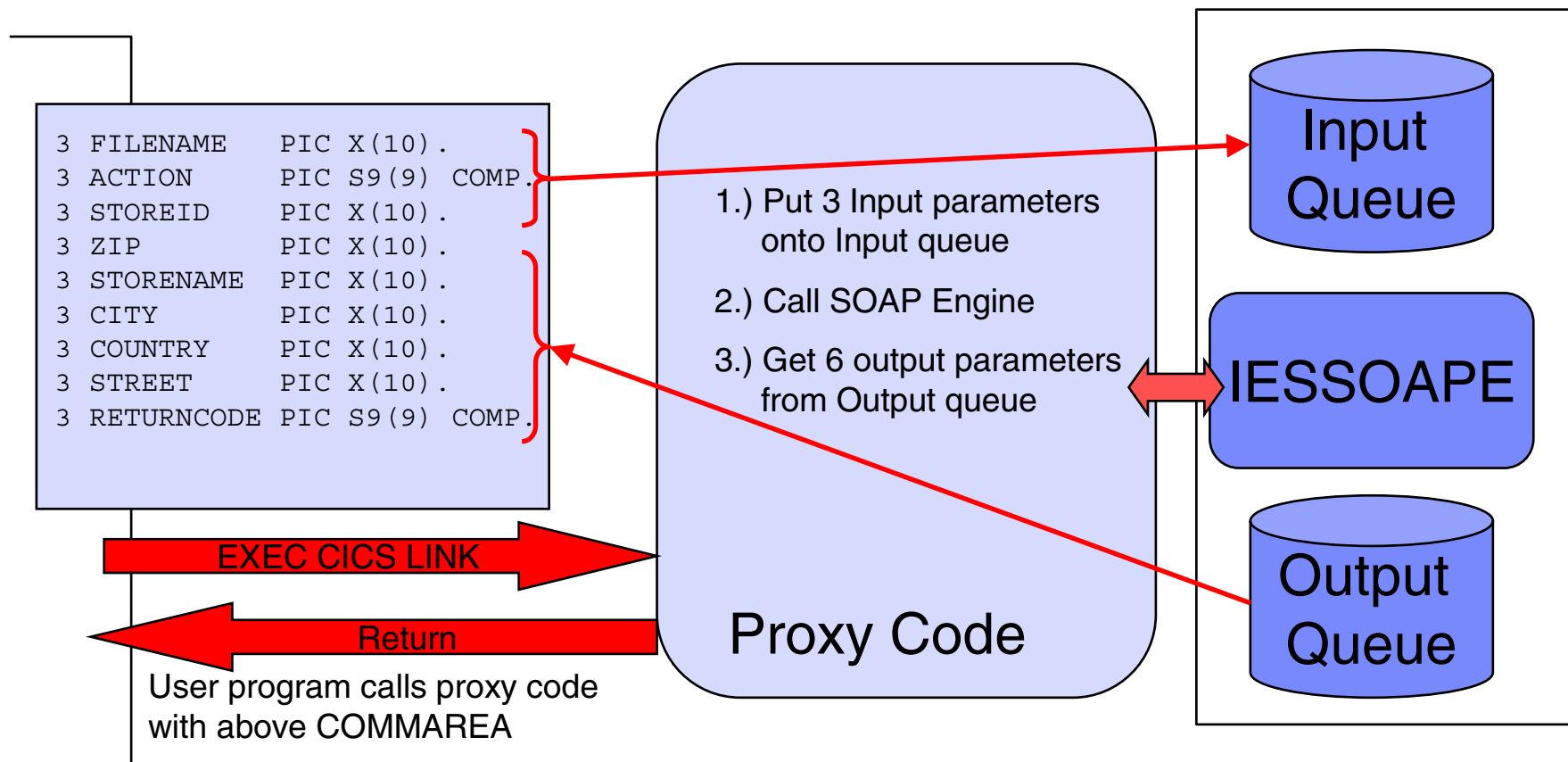
- **Although the SOAP Engine interface uses standard CICS methods, special coding is needed to interface with the VSE SOPA Engine.**
- **Proxy code ‘maps’ between standard COMMAREA interface and SOAP Engine**
  - All SOAP specific handling is done in proxy code
  - User applications calls the proxy code or gets called by the proxy code like a local program call (EXEC CICS LINK) using a COMMAREA
  - User COMMAREA format can be freely defined by user
  - Proxy code copies fields from COMMAREA into TS queue entries and vice versa

## Why use a proxy program?

- **All SOAP implementations use some kind of “proxy code”**
  - Java (RAD/WSAD)
  - Microsoft .Net
  - ...
- **The proxy code maps the implementation specifics of the SOAP engine to a common interface**
- **The proxy code is generated using the information from the WSDL**
- **The proxy code is usually not modified directly by user**
- **VSE uses the same technique as other SOAP implementations**

## What does the proxy code do?

### ■ To call an external Web Service



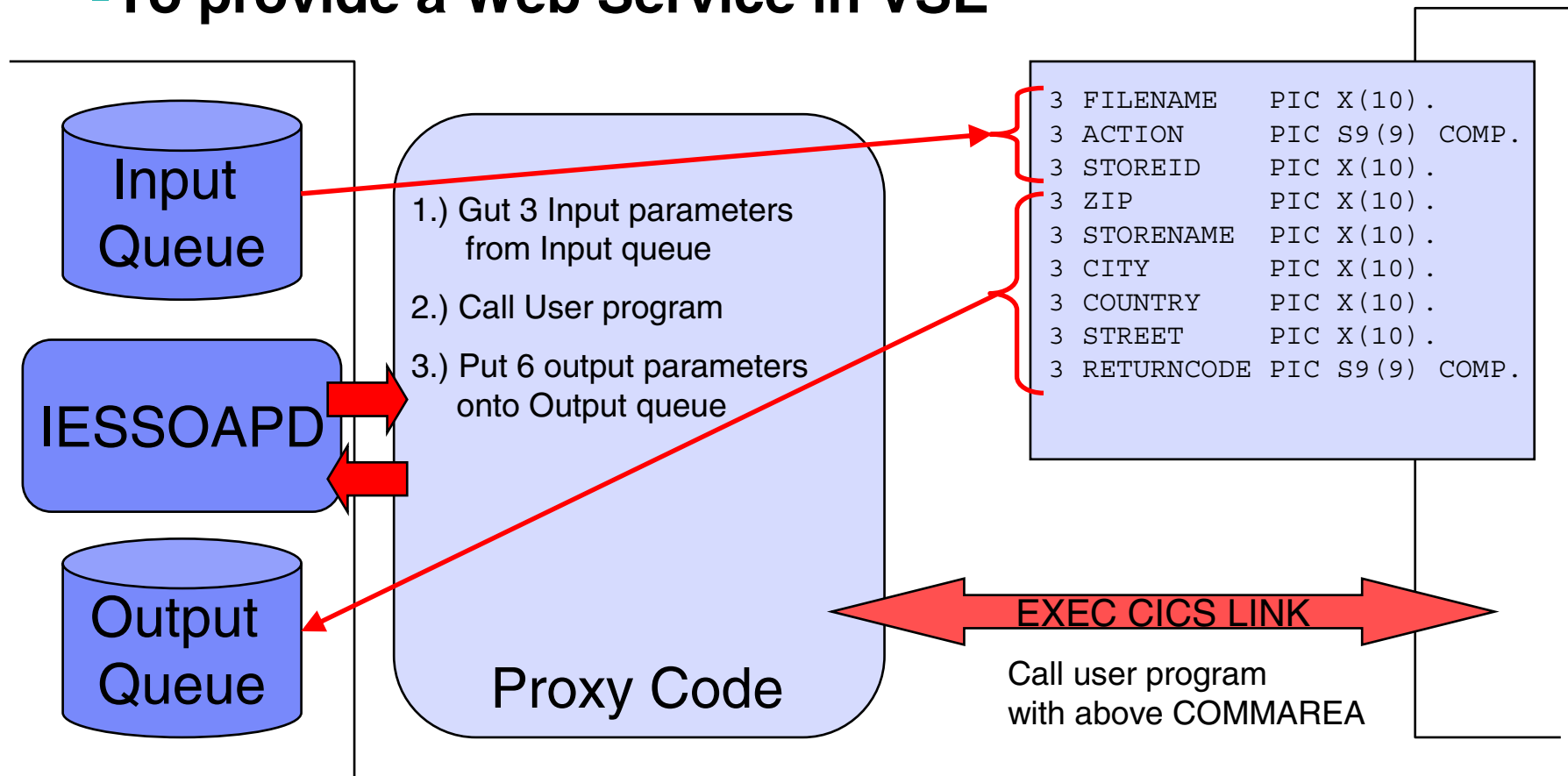


## What does the proxy code do?

- **To call an external Web Service**
  - Proxy code gets invoked via EXEC CICS LINK by user program
  - Put input parameters onto input queue
  - Setup parameter area for Web Service call
    - Endpoint URL
    - Name of method to call
    - Names of Input and Output queues
  - Call SOAP Engine
    - EXEC CICS LINK to IESSOAPE
  - On return
    - Check for errors
    - Get output parameters from output queue
    - Return to user program

# What does the proxy code do?

## ■ To provide a Web Service in VSE



## What does the proxy code do?

- **To provide a Web Service in VSE**
  - Proxy code gets called by SOAP Engine
  - Get input parameters from input queue
  - Prepare user COMMAREA
  - Call user program
    - EXEC CICS LINK service provider program (user program)
  - On return
    - Check for errors
    - Put output parameters onto output queue
    - Return to SOAP engine

## Which programs can be used with Web Services?

### ■ Which VSE programs can be used as an Web Service?

- All CICS TS programs that implement the “service” you want to provide
  - In any programming language (COBOL, PL/1, C, Assembler)
- You should separate business logic from user interface
  - 3270 screens or BMS maps can not be used
- The proxy code calls your program with **EXEC CICS LINK** and an **user defined COMMAREA**

### ■ Which VSE programs can call an external Web Service?

- All CICS TS programs that can call another program with a COMMAREA
  - In any programming language (COBOL, PL/1, C, Assembler)
- Your program calls the proxy code with **EXEC CICS LINK** and an **user defined COMMAREA**

# Agenda

## What is a Web Service?

### Terms and theory

- SOA, SOAP, WSDL, SOAP RPC, XML
- **Using Web Services with Java/.Net**
  - Proxy code
- **Using Web Services with VSE**
  - VSE SOAP Engine, programming Interfaces
- **CICS2WS Tool**
  - Live demo

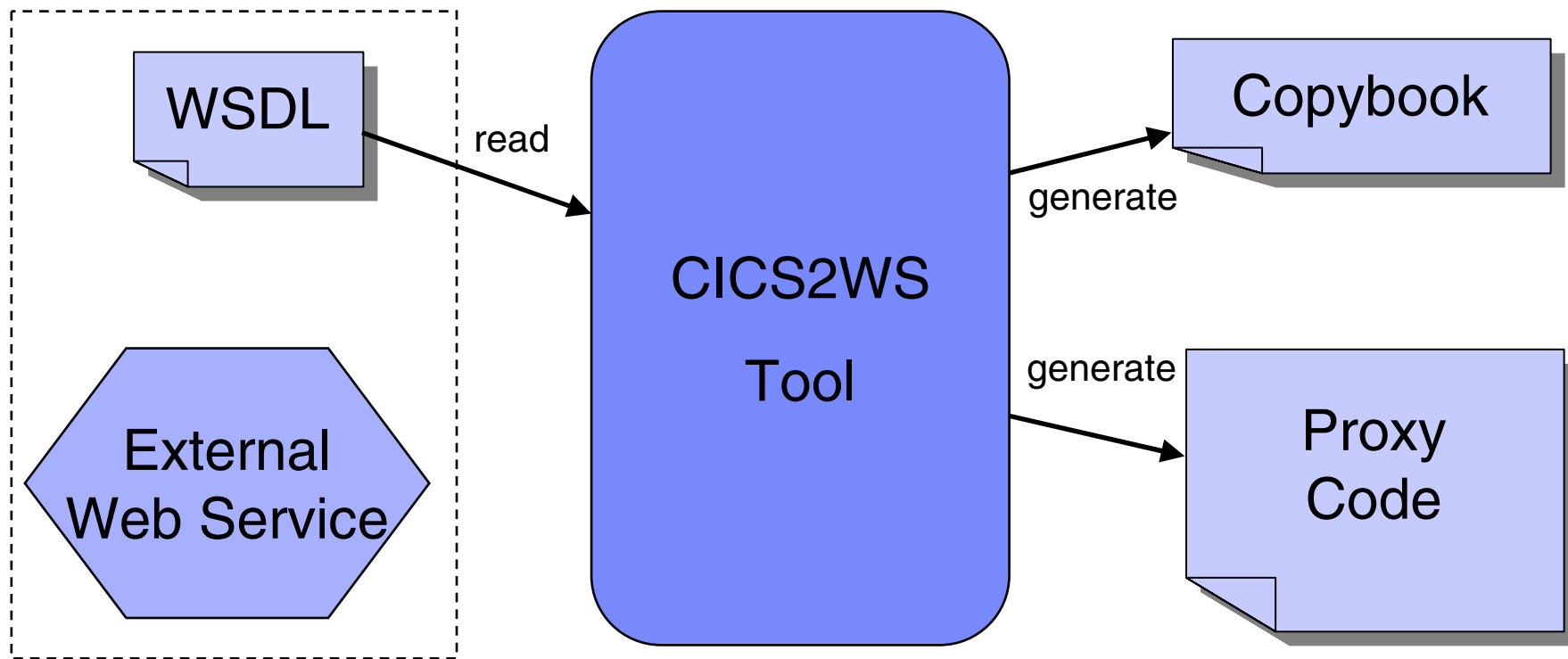
## How to write the proxy code

- **You can write the proxy code “by hand”**
  - Not very difficult, use samples as skeleton
  - COBOL Example (from 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)
- **Use the new CICS2WS tool**
  - Generates proxy code and WSDL files
  - Proxy code is in assembler language
    - No extra charged compiler needed (e.g. COBOL or PL/I)
    - Code is very simple, straight forward
    - Usually no manual changes needed in proxy code

## New CICS2WS Tool

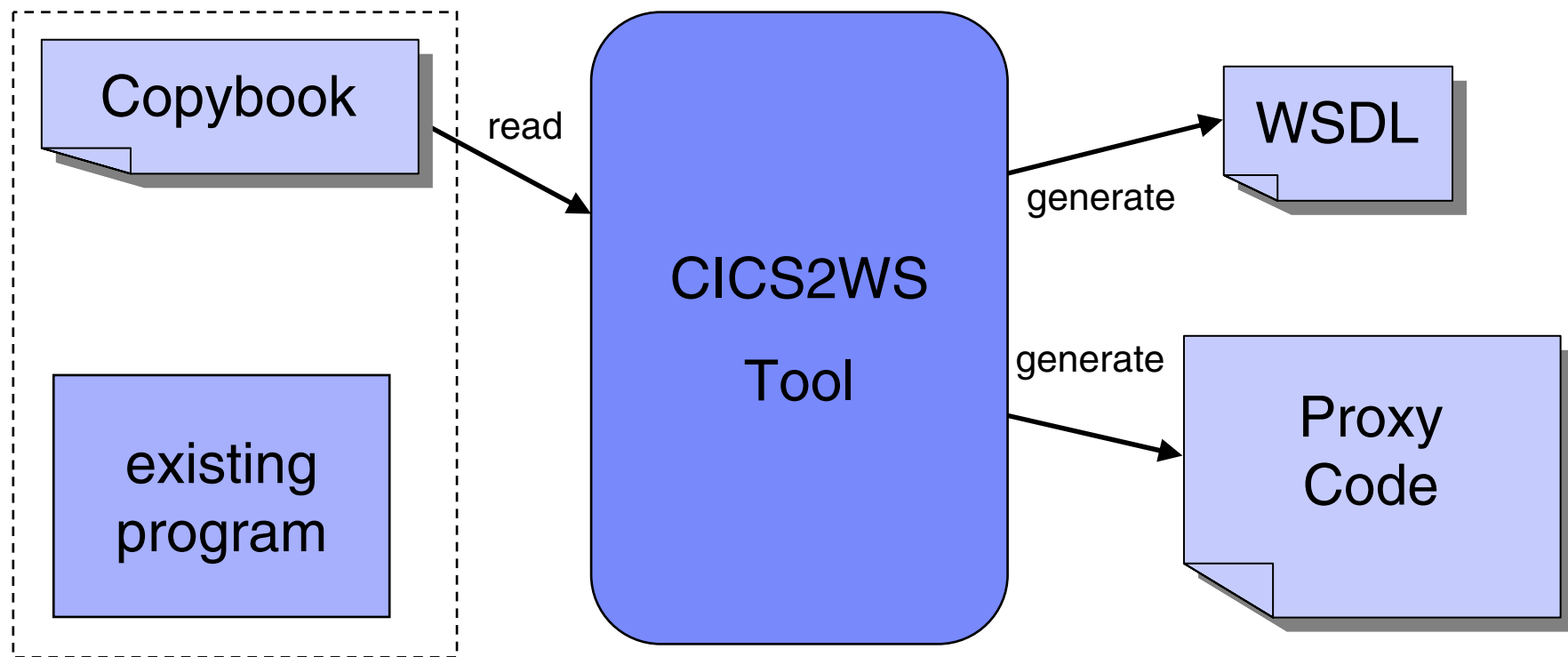
- **The tool runs on your PC or workstation**
- **Implemented in Java**
- **VSE as a SOAP client (service requestor)**
  - Reads the WSDL file
  - Generates the proxy code (Assembler)
  - Generates a COMMAREA mapping (copybook)
    - in COBOL, PL/I or Assembler
- **VSE as a SOAP server (service provider)**
  - Reads a given COMMAREA mapping (copybook)
    - in COBOL, PL/I or Assembler
  - Generates the proxy code (Assembler)
  - Generates the WSDL file

## VSE as a SOAP client (service requestor)

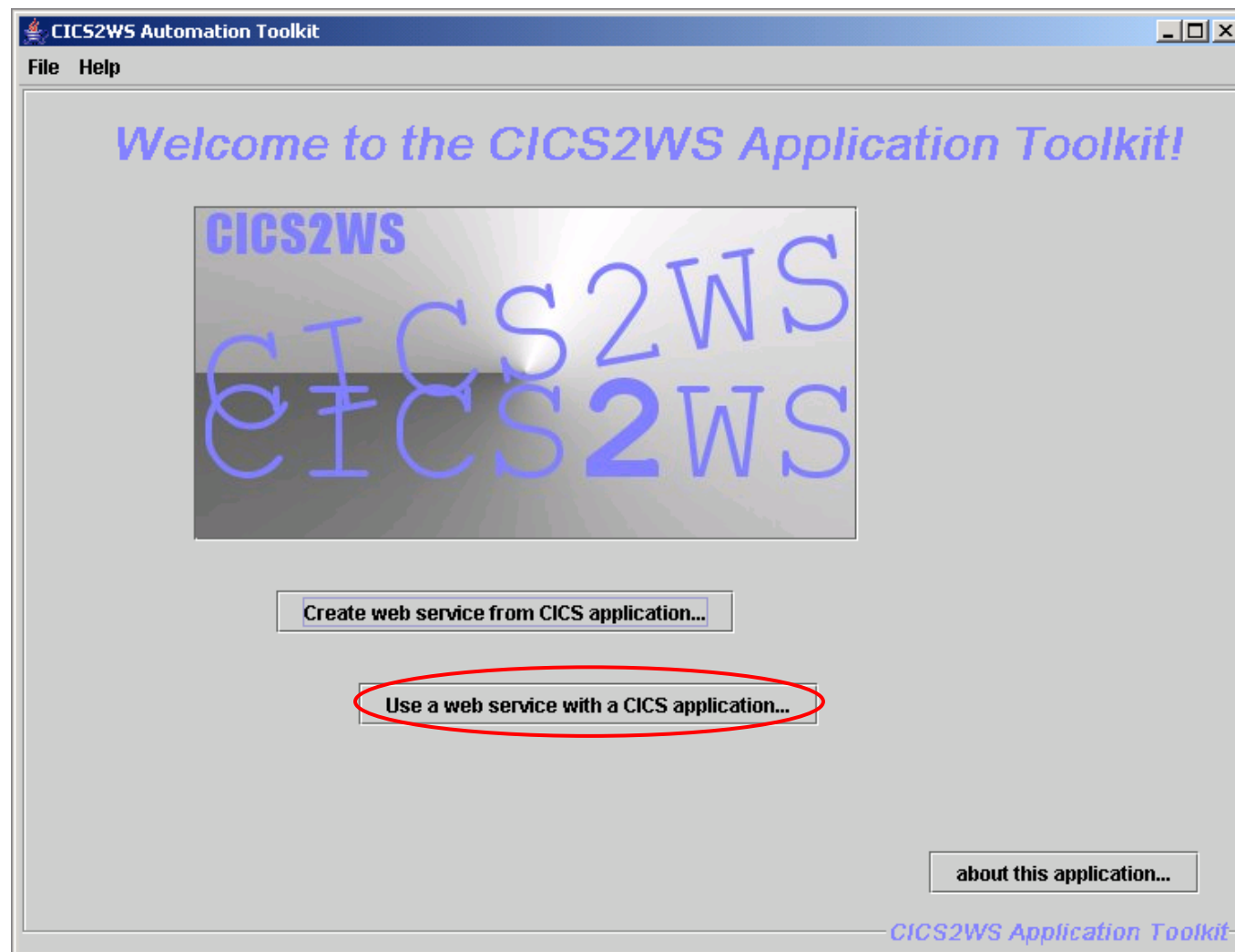




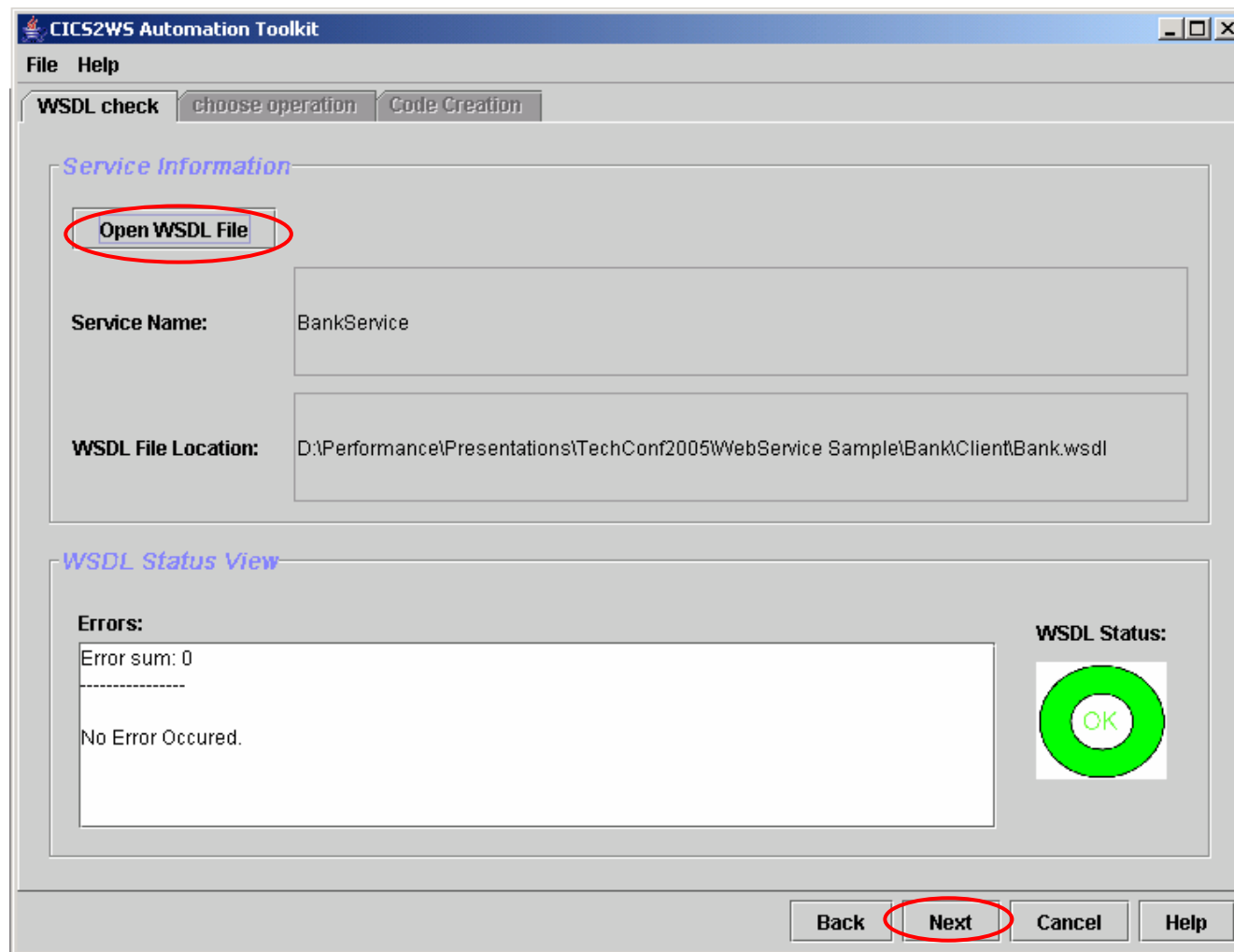
## VSE as a SOAP server (service provider)



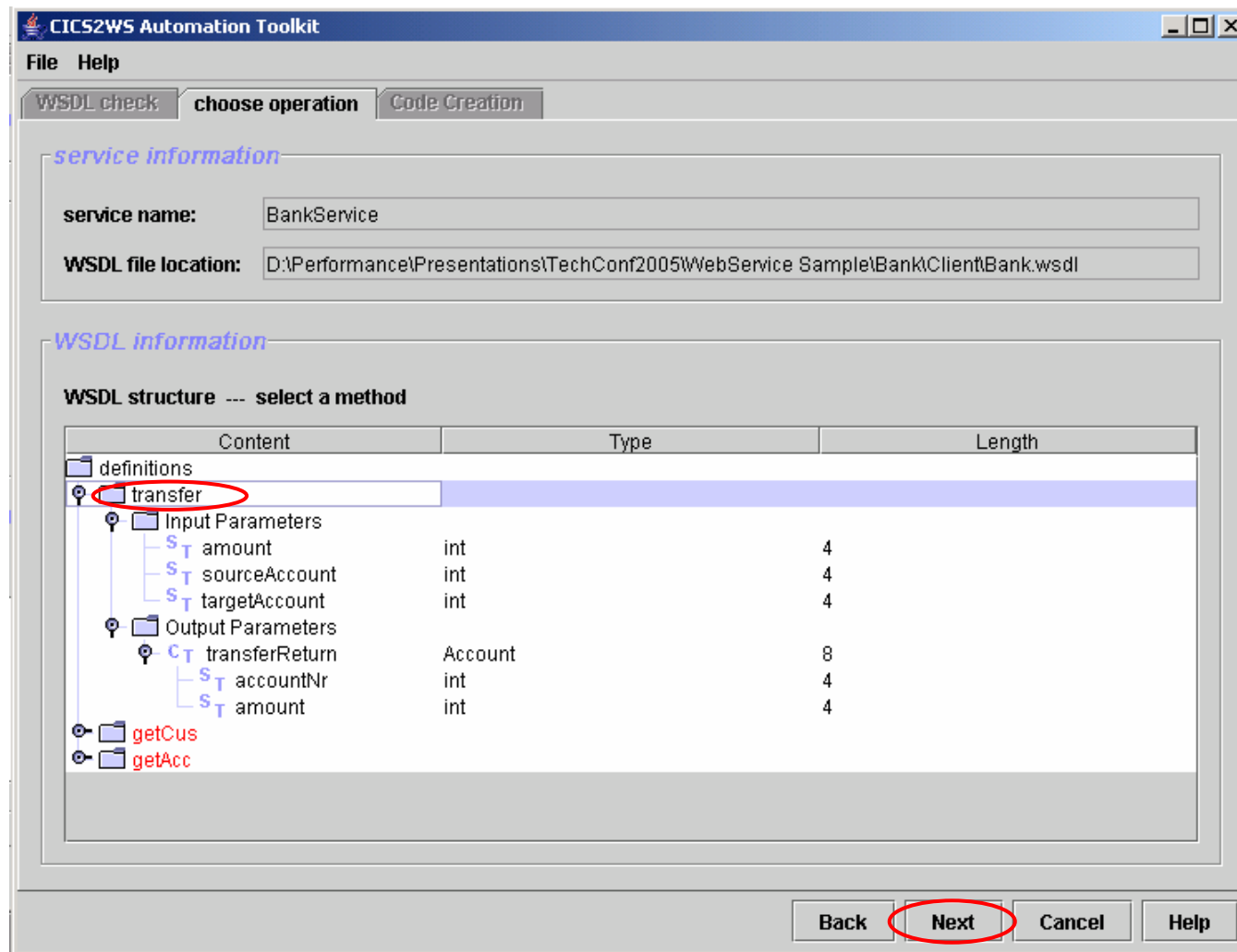
## Live Demo – Use a Web Service



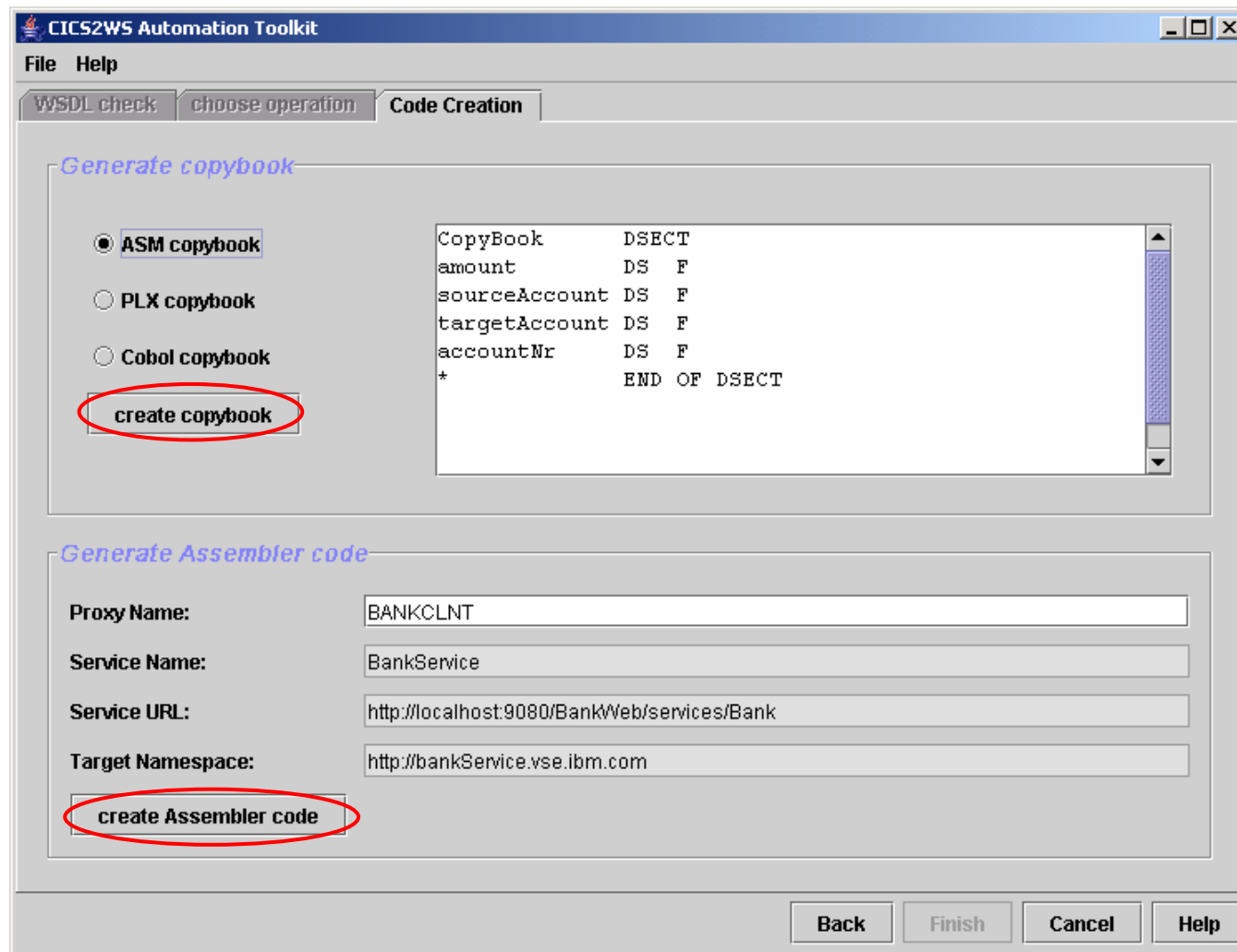
## Live Demo – Use a Web Service



# Live Demo – Use a Web Service



## Live Demo – Use a Web Service



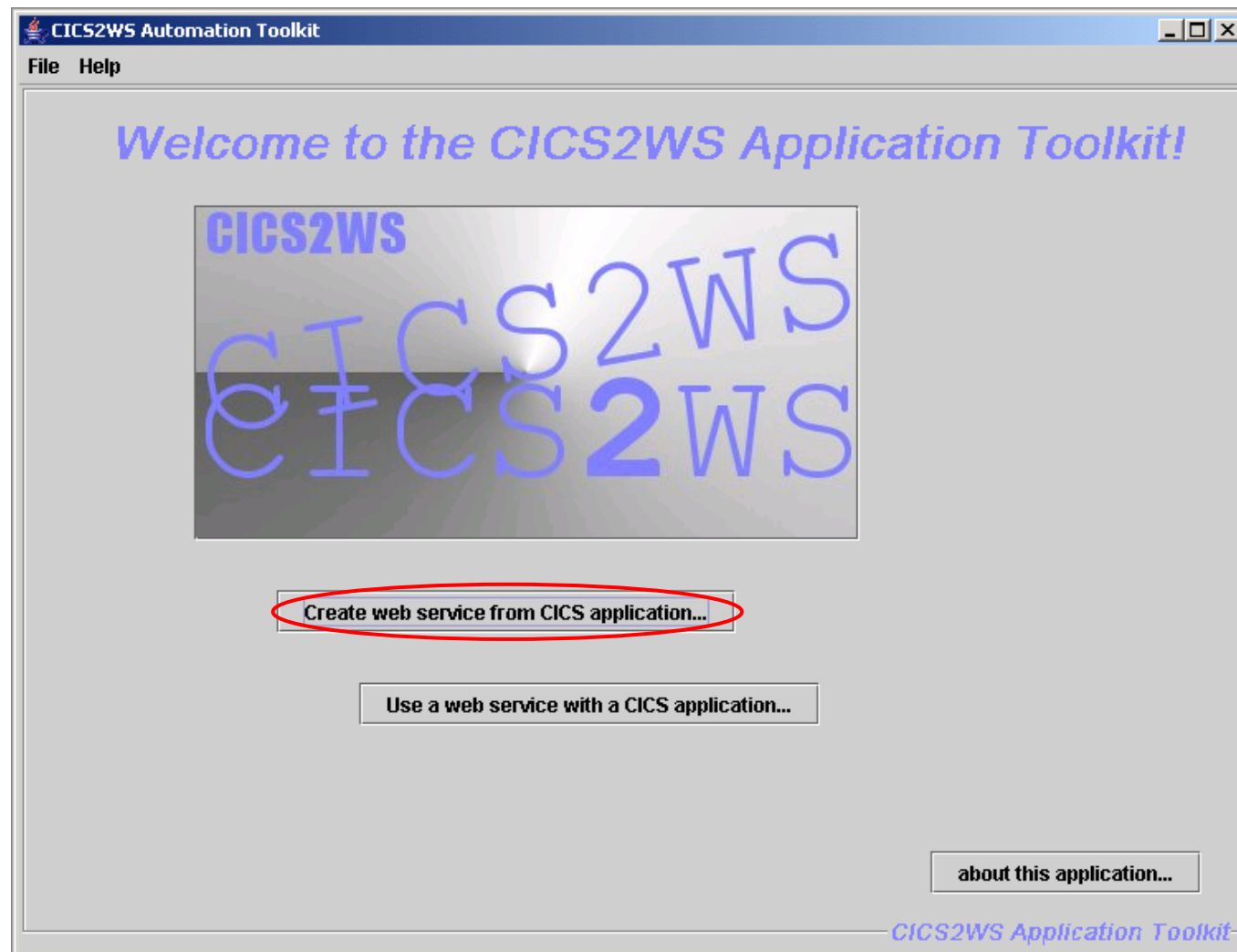
## Live Demo – Use a Web Service

```
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 – Create a Web Service



# Live Demo – Create a Web Service

**CIC52WS Automation Toolkit**

File Help

create Service | add operation to service | service summary

*CICS application data*

Path to source: 005WebService SampleBankServerBankCB.chl browse...

source code language: COBOL

*basic service data*

service name: VSEBANKService

service URL: http://vsedemo.boeblingen.de.ibm.com:1080/cics/CWBVAIESSOAPS

URN: urn:IESOAPD: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 – Create a Web Service

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

→

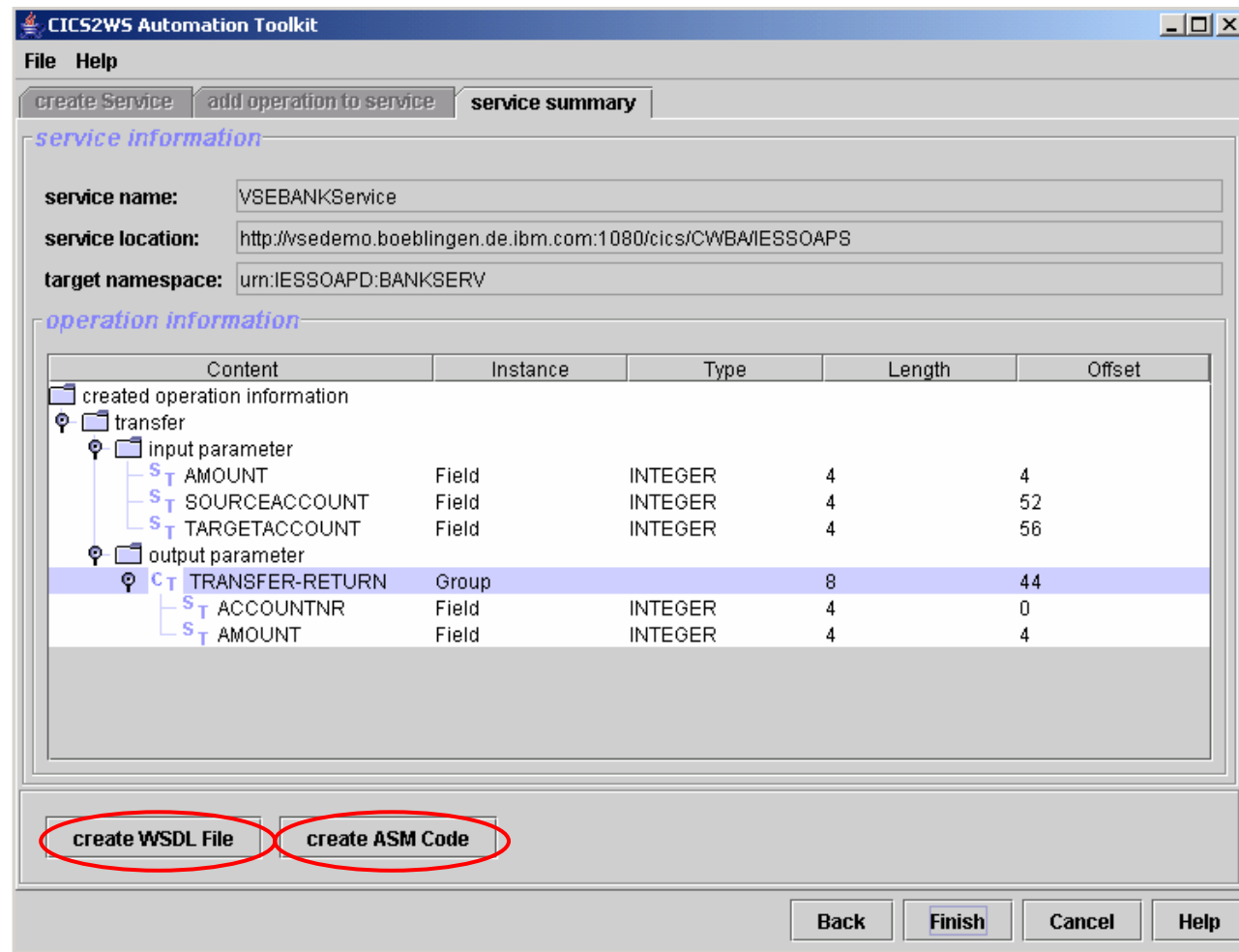
←

**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 – Create a Web Service



# Live Demo – Use a Web Service

```

<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions targetNamespace="urn:IESOAPD:BANKSERV" xmlns:impl="urn:IESOAPD:BANKSERV"
  xmlns:intf="urn:IESOAPD: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:IESOAPD:BANKSERV"
  xmlns:impl="urn:IESOAPD:BANKSERV" xmlns:intf="urn:IESOAPD: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
...
    
```



## Documentation

- **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)***

# Questions ?

