



| IBM System z – WAVV 2008

Using WebServices with z/VSE

Ingo Franzki – ifranzki@de.ibm.com



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

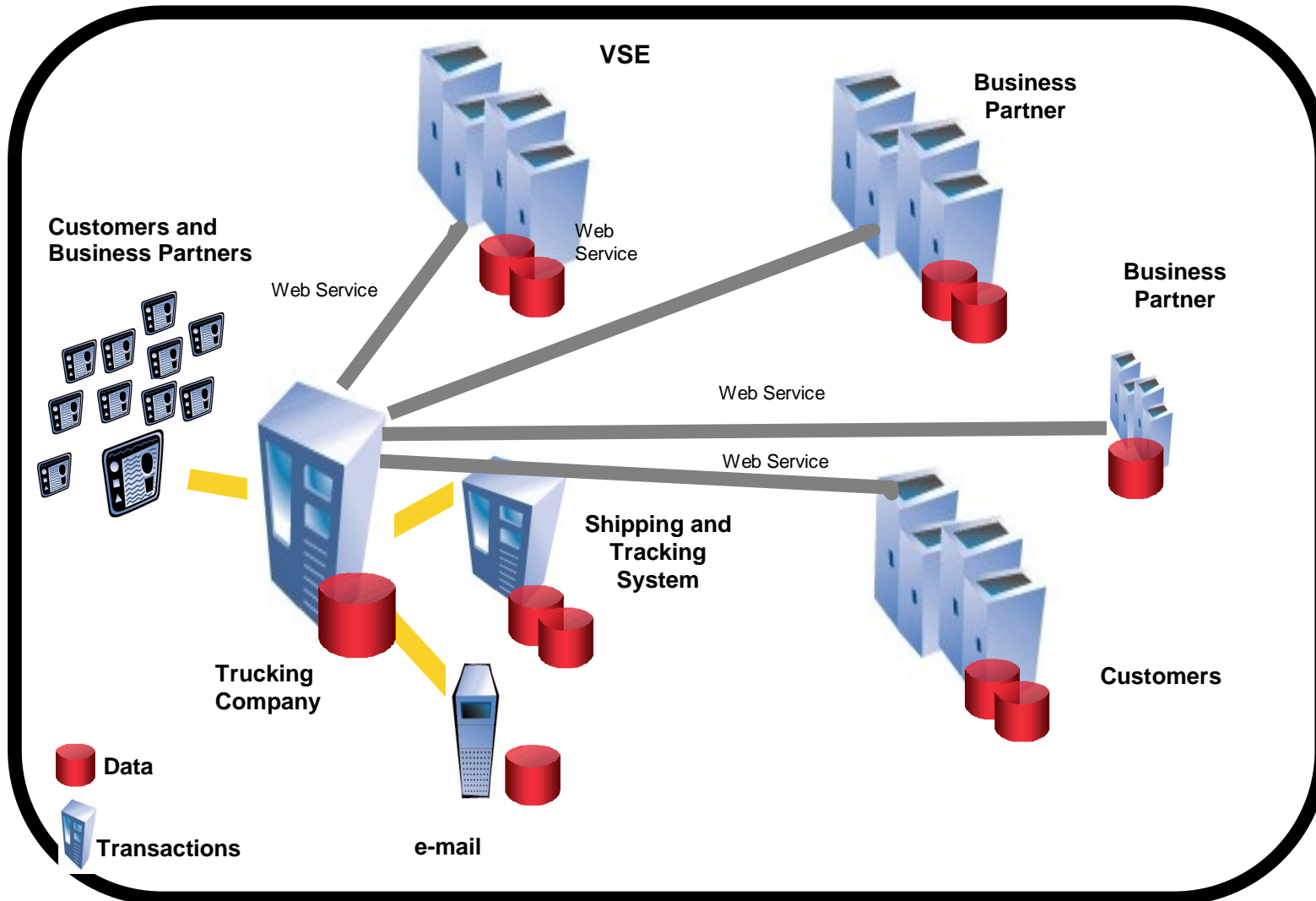
UNIX is a registered trademark of The Open Group in the United States and other countries.

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.

Roadmap for dynamic e-business - SOA



What is Service-Oriented Architecture (SOA)?

§ SOA is an IT architectural style

- supports **integrating your business** as linked services that can be accessed when needed over a network, enabling your business to adapt to changing conditions and requirements
- These services are **self-contained** and have **well-defined interfaces** to let the users of those services -- called clients or consumers -- know how to interact with them

§ SOA results in "**loosely coupled**" application components

- The code is not necessarily tied to a particular database, or even a particular infrastructure.

§ It is this loose coupling that enables the combination of services into diverse applications.

- It also enables much greater **code reuse**, cutting your workload at the same time that it increases your capabilities.

§ Because a service and the client accessing that service are not tied to each other

- a service used to process an order could be completely replaced, and the client-services placing orders would never know.

What is Service-Oriented Architecture (SOA)?

- § **From a business standpoint, a Service-Oriented Architecture is focused on**
 - developing technology that helps you **accomplish your business tasks**
 - rather than allowing technological constraints to dictate your activities.
- § **For example, the process of selling, manufacturing, shipping, and getting paid for an item may involve dozens of steps and several different databases and computer systems.**
- § **But at the heart of things, the process encompasses a handful of human activities, for example:**
 - Salesmen finds a likely customer
 - Customer orders product
 - Production department produces product
 - Production department ships product
 - Billing department bills for product
 - Customer pays for product

What is Service-Oriented Architecture (SOA)?

§ Implementing SOA can bring you a great number of benefits, including the following:

- Greater alignment of business and IT
- **Component-based** systems
- **Loosely coupled** components and systems
- A network-based infrastructure, enabling geographically and technologically diverse resources to work together
- On-demand, built-on-the-fly-applications
- Greater **code reuse**
- Better **process standardization** throughout the enterprise
- Easier centralization of corporate control

What is Service-Oriented Architecture (SOA)?

- § **Web services** are the most common technology standards used to implement SOA
 - However, they are not the only technology one can use to develop the parts of an SOA
- § **Many SOAs -- most, in fact -- involve the integration of legacy data**
 - contained in systems that use technology such as MQSeries and Common Object Request Broker Architecture (CORBA) or even CICS.
- § **Many of these technologies have been adapted for the SOA world, and they can be used with or without a Web services wrapper.**
- § **But, Web services is rapidly becoming the de facto standard used to support SOA.**

Why would a VSE customer do SOA ?

§ SOA is modern (hype) and strategic

- The management says: We also have to do SOA

§ Easy integration of existing VSE programs into the modern world

- Reducing the interface complexity
- Reuse of existing applications as services
- Use of standard protocols (XML, SOAP, HTTP)

§ Encapsulation of VSE programs

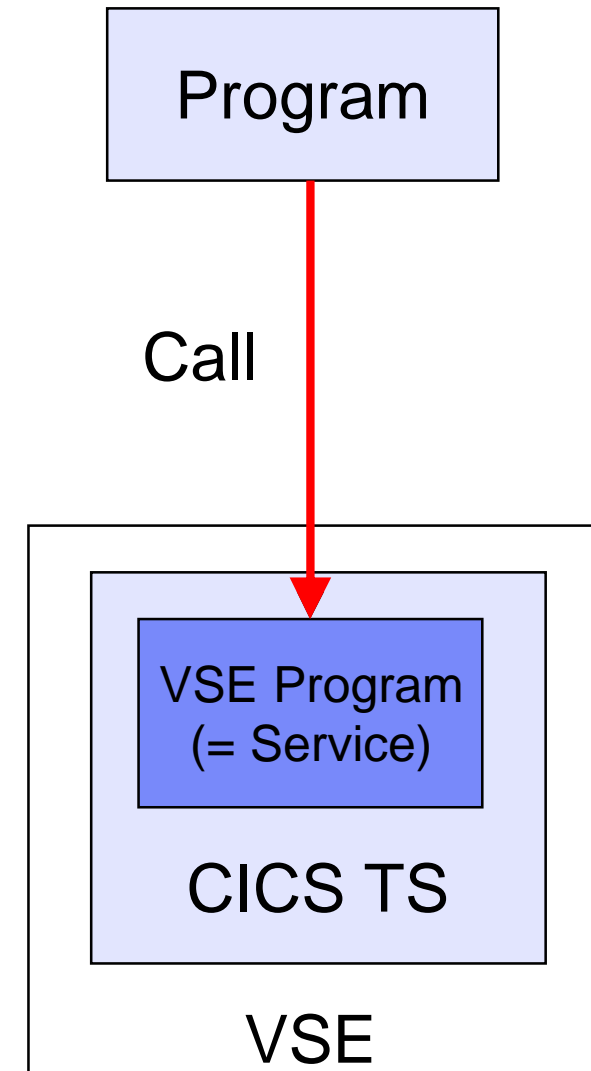
- Disconnecting business and display logic

§ Integration of VSE into a Microsoft .Net environment

- You do not want to use Java
- You already have a Microsoft environment

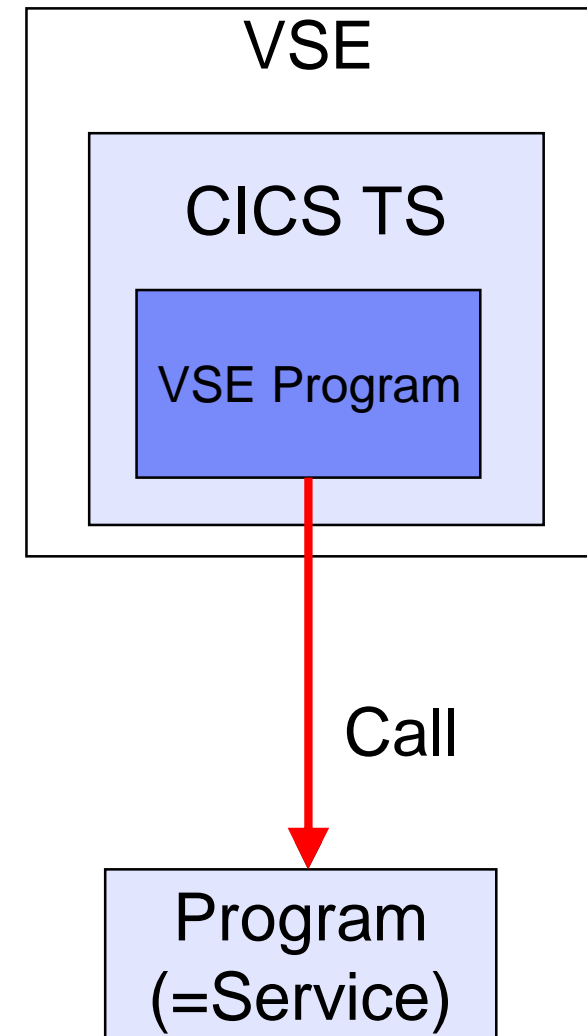
What is a Web Service?

- § 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

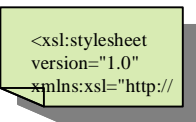


What is a Web Service?

- § 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



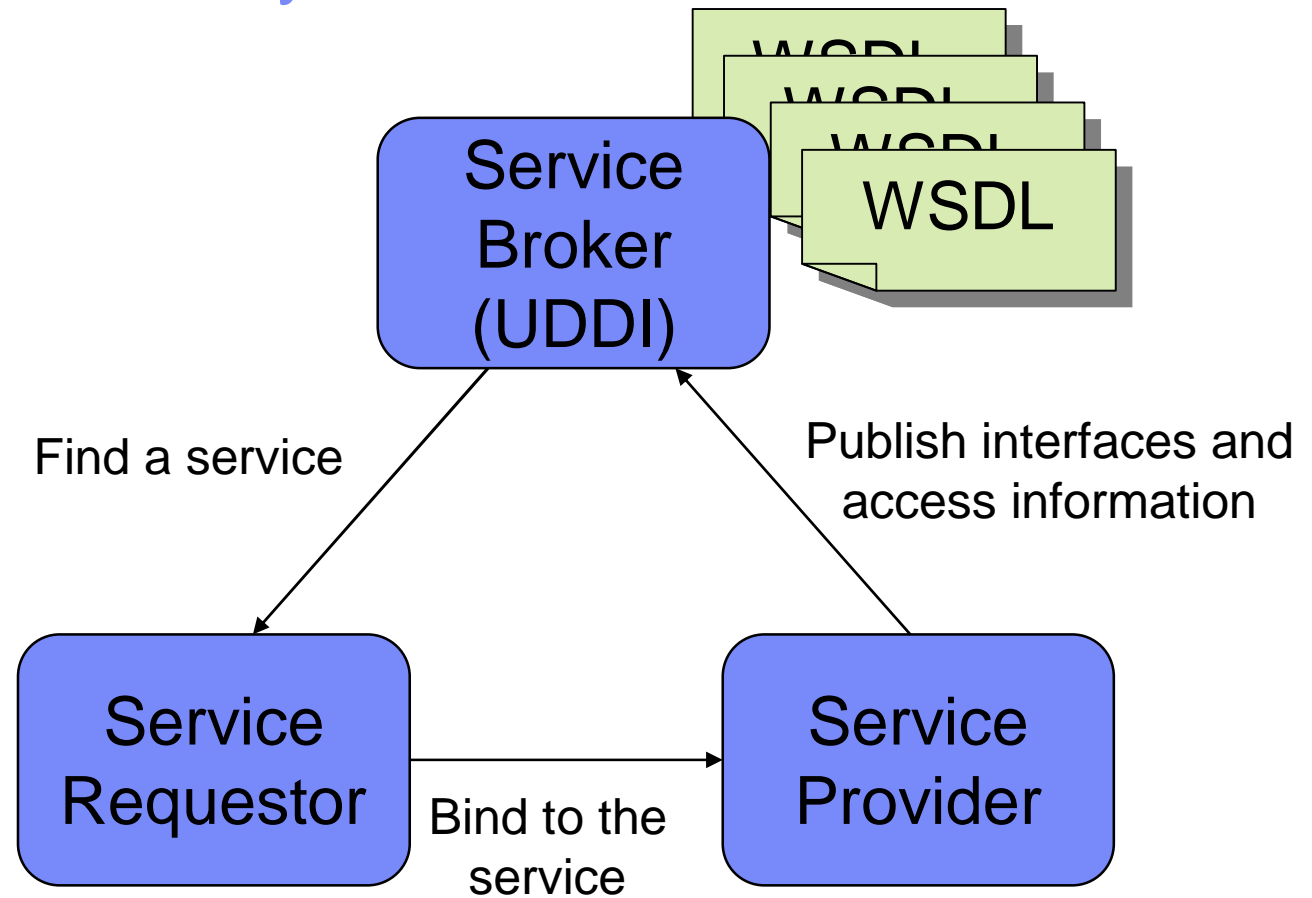
Web Services - Summary

XML 

SOAP 

HTTP - Transport Service

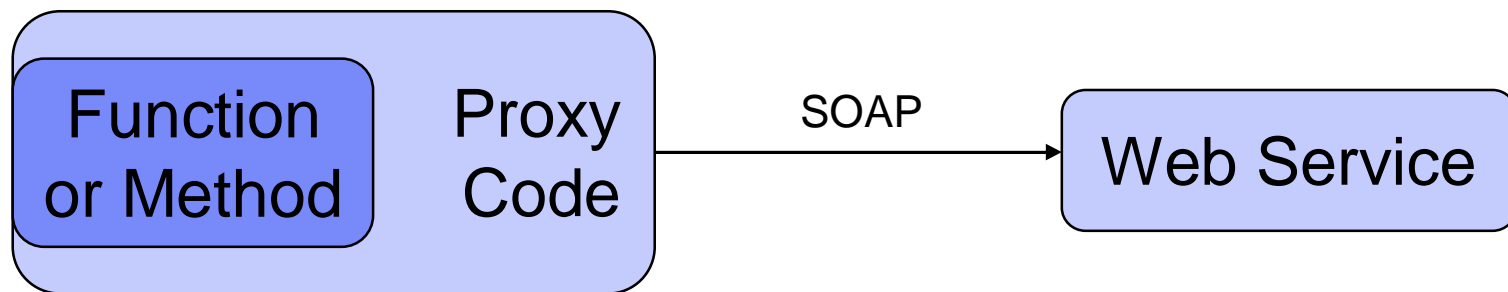
TCP/IP - Street



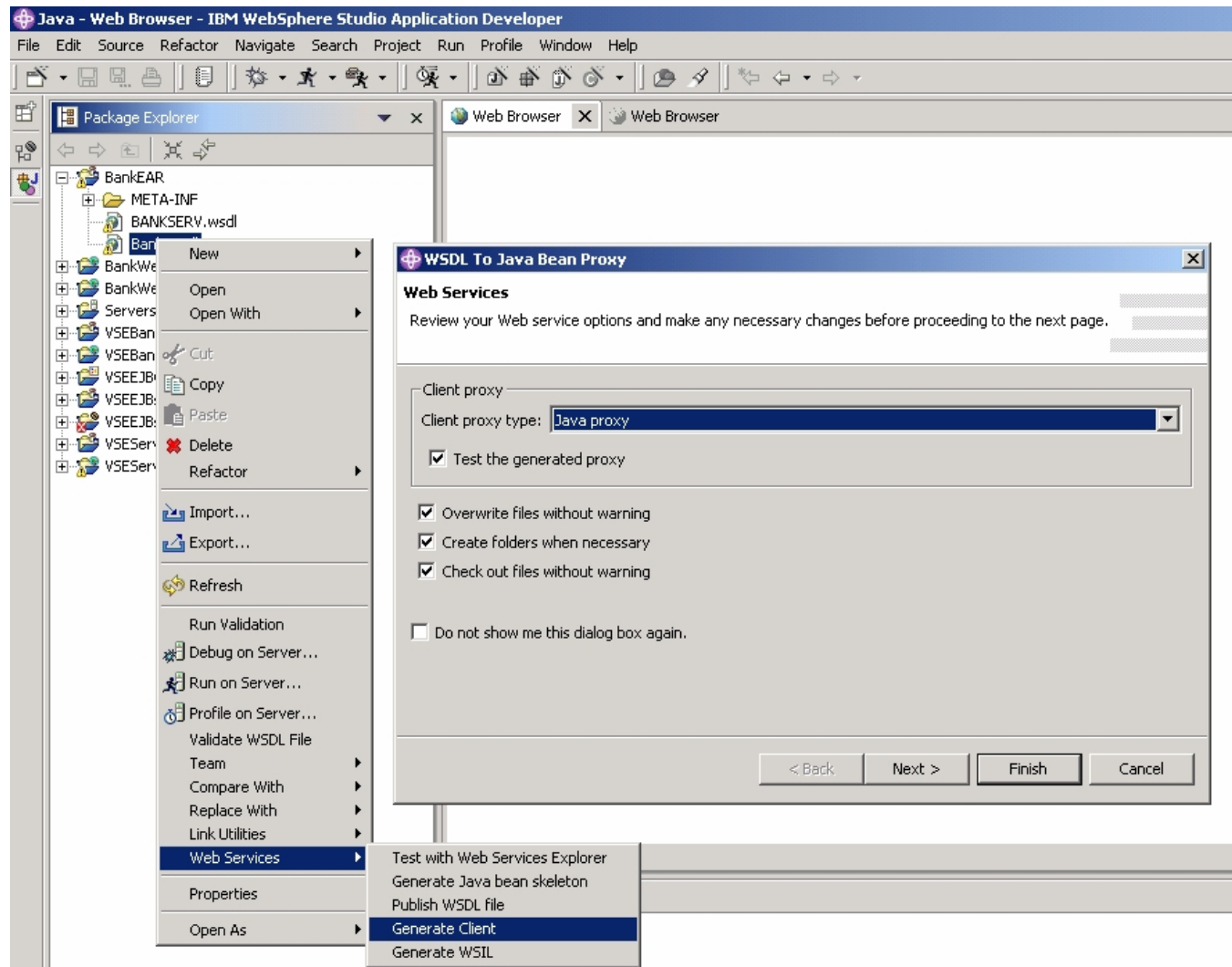
Using Web Services with Java or MS .Net

§ 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 Visual Studio 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



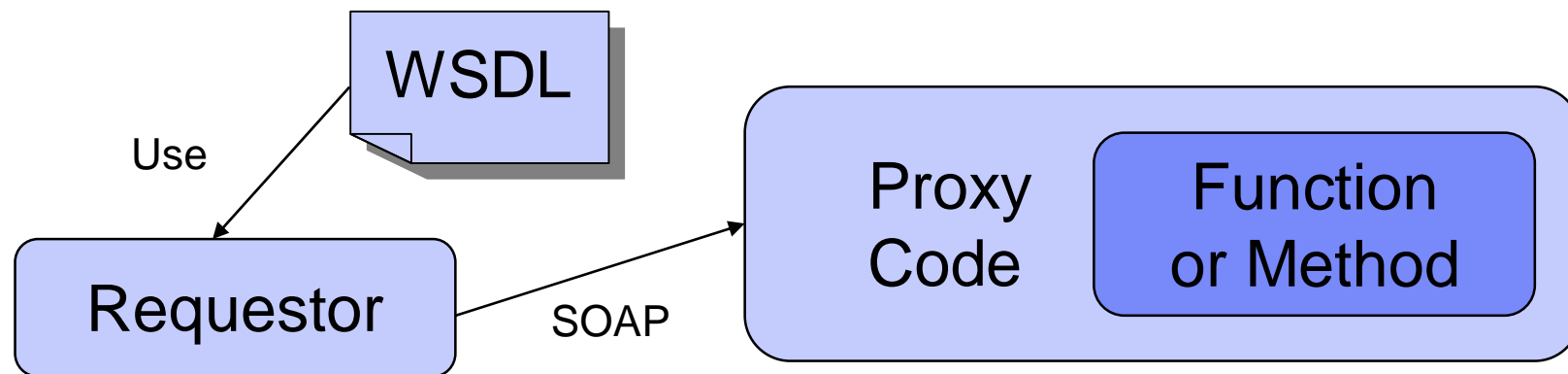
Using Web Services with Java or MS .Net



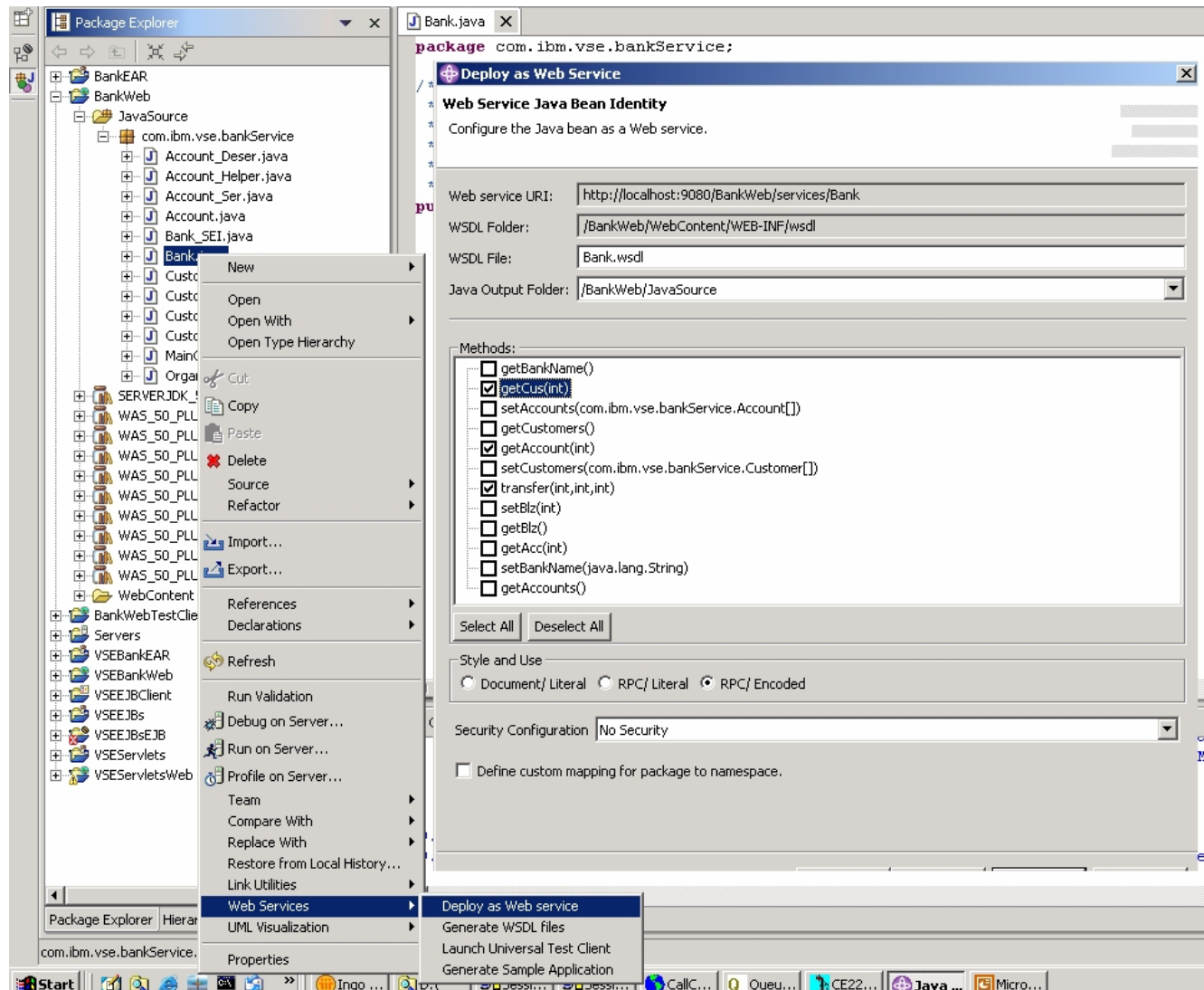
Using Web Services with Java or MS .Net

§ Create/provide a new Web Service

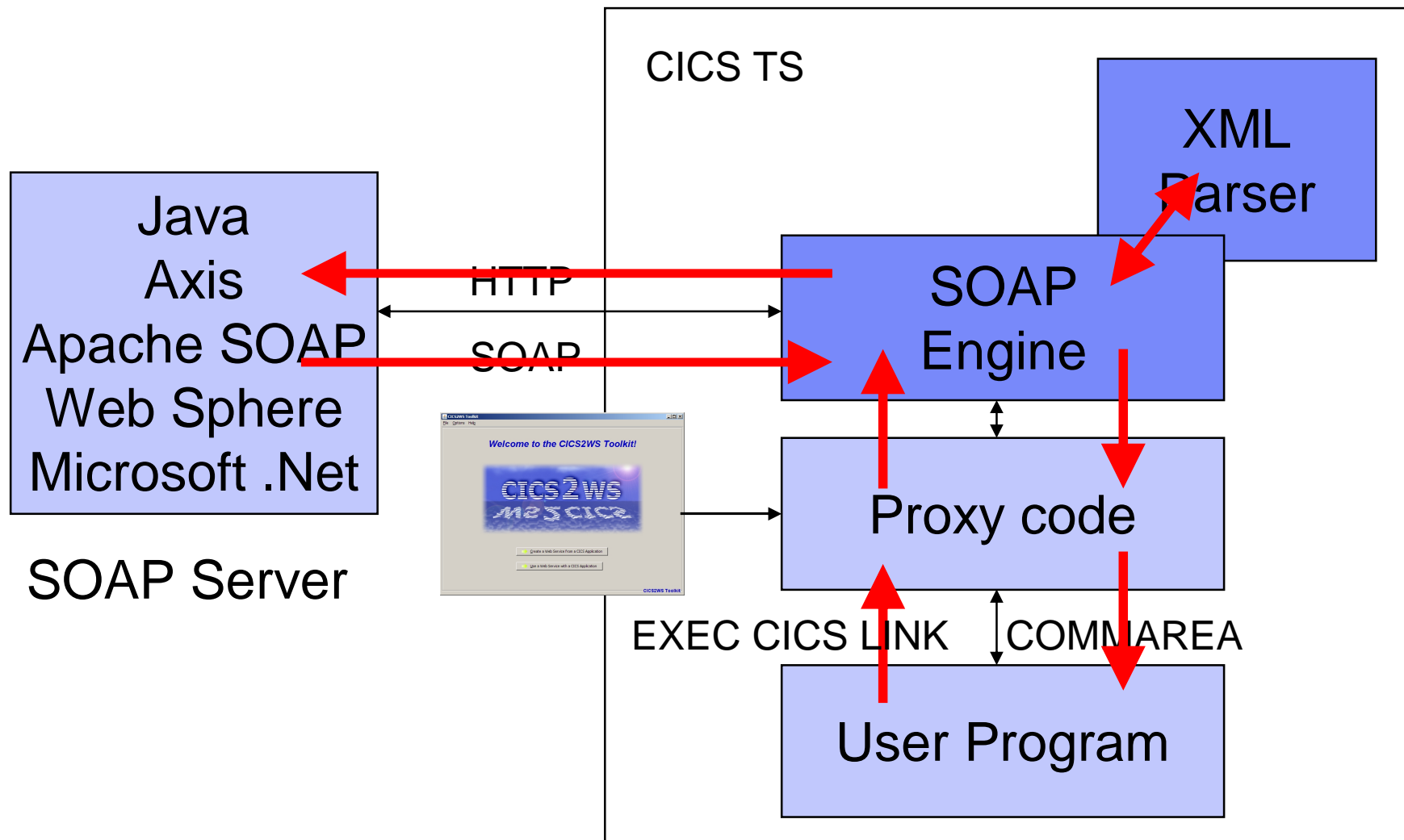
- You have a function or method that implements some kind of service that you want to provide
- Use a tool like Rational Application Developer (RAD/WSAD) or Microsoft Visual Studio to model a Web Service
 - Generate a Web Service Description (WSDL) and publish it
 - Generate “proxy code” that makes the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



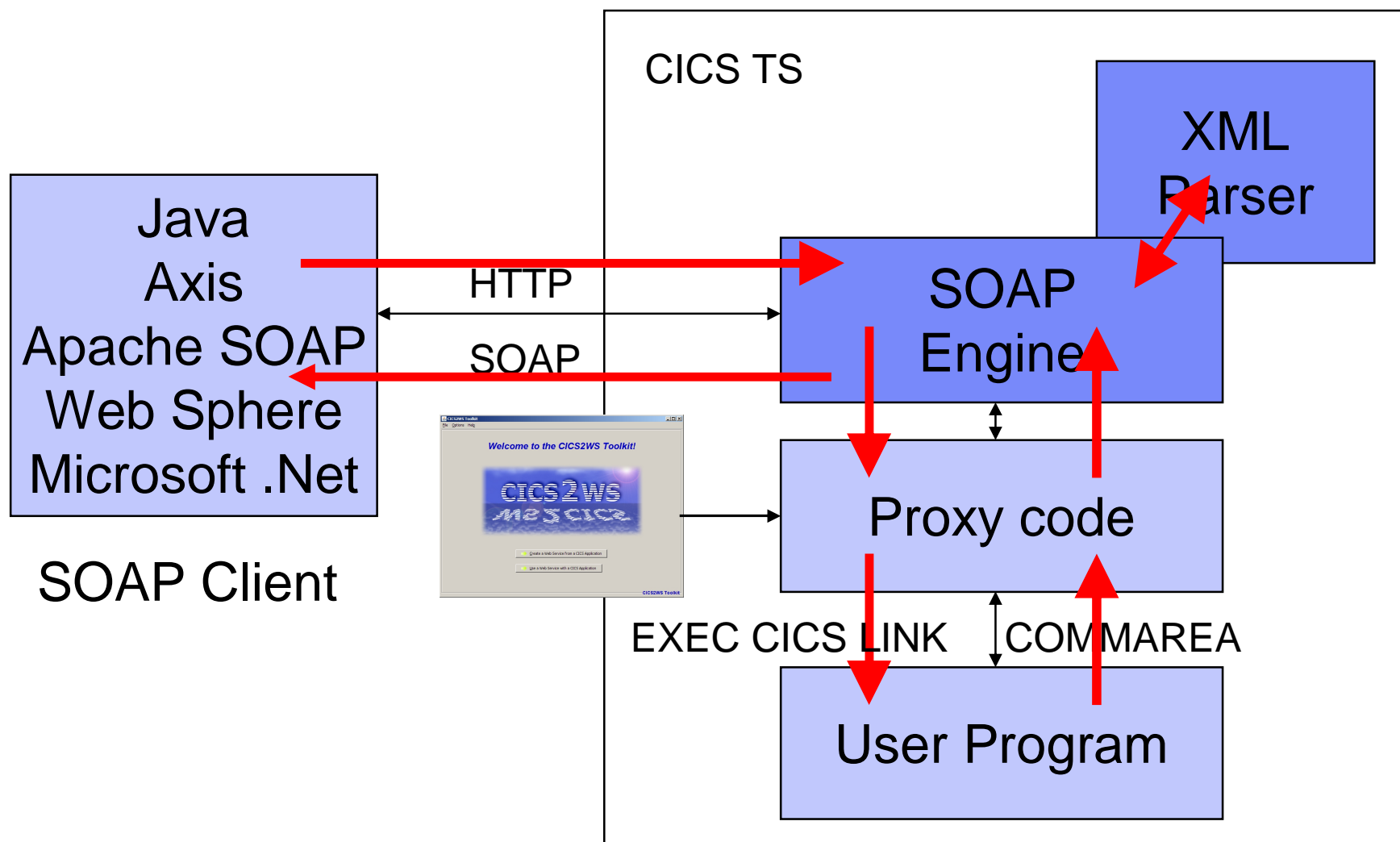
Using Web Services with Java or MS .Net



Using Web Services with VSE – SOAP client



Providing Web Services with VSE – SOAP server



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

§ System requirements

- VSE/ESA 2.6, 2.7 or z/VSE 3.1
- CICS Transaction Server
 - CICS/VSE 2.3 can not be used
- TCP/IP

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?

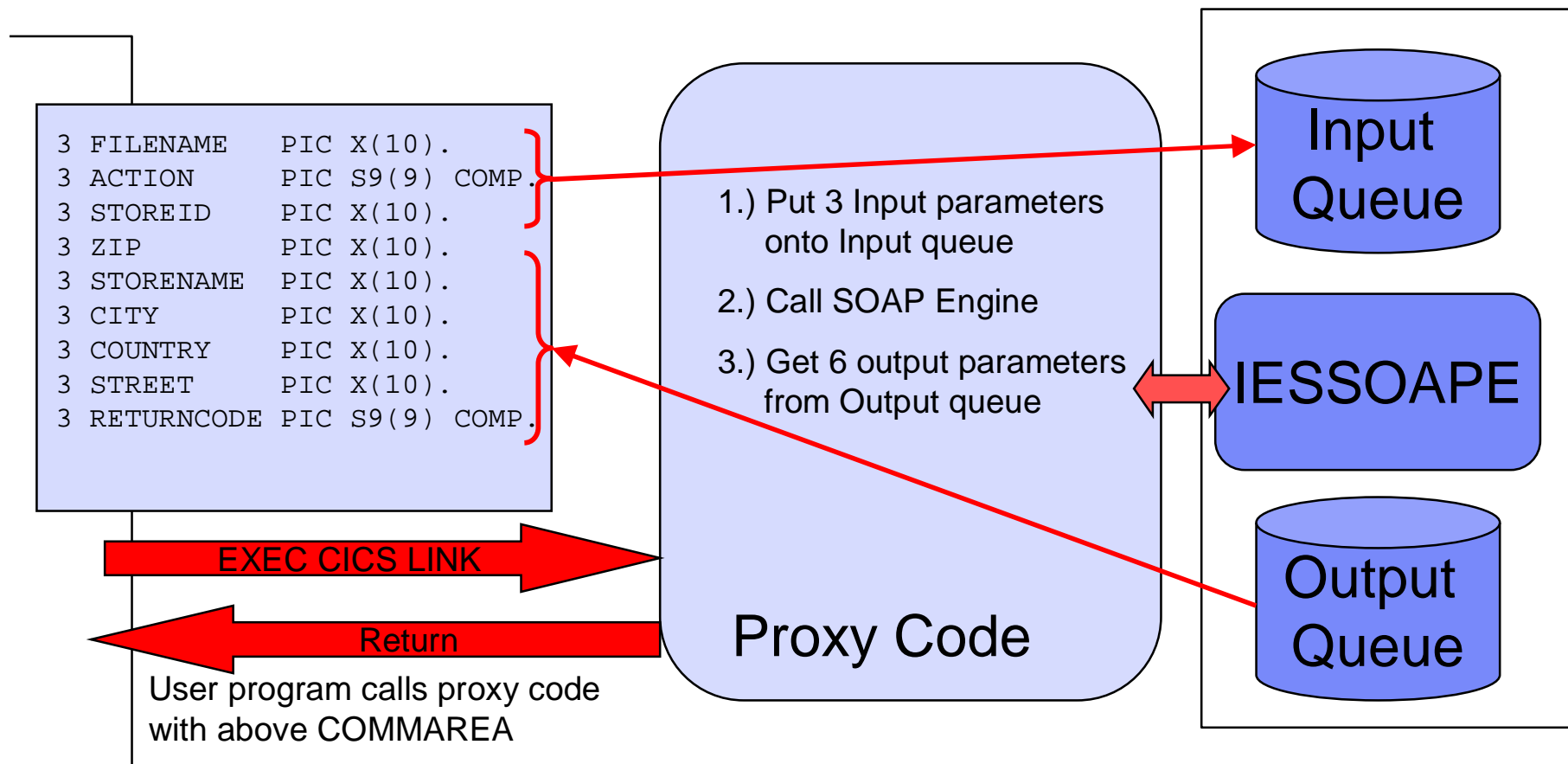
- § **Although the SOAP Engine interface uses standard CICS methods, special coding is needed to interface with the VSE SOPA Engine.**
- § **Proxy code 'maps' between standard COMMAREA interface and SOAP Engine**
 - All SOAP specific handling is done in proxy code
 - User applications calls the proxy 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



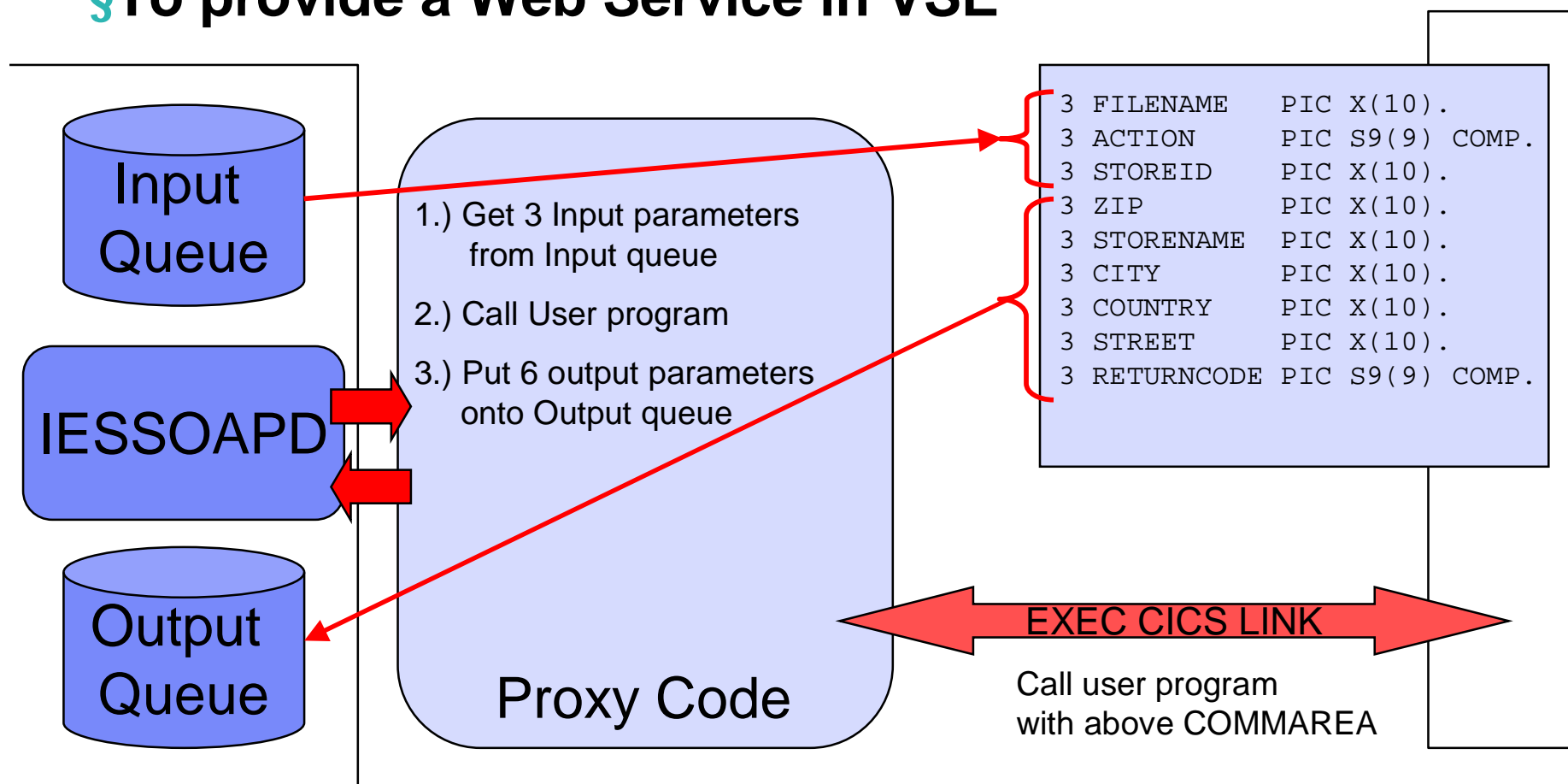
What does the proxy code do?

§ 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**

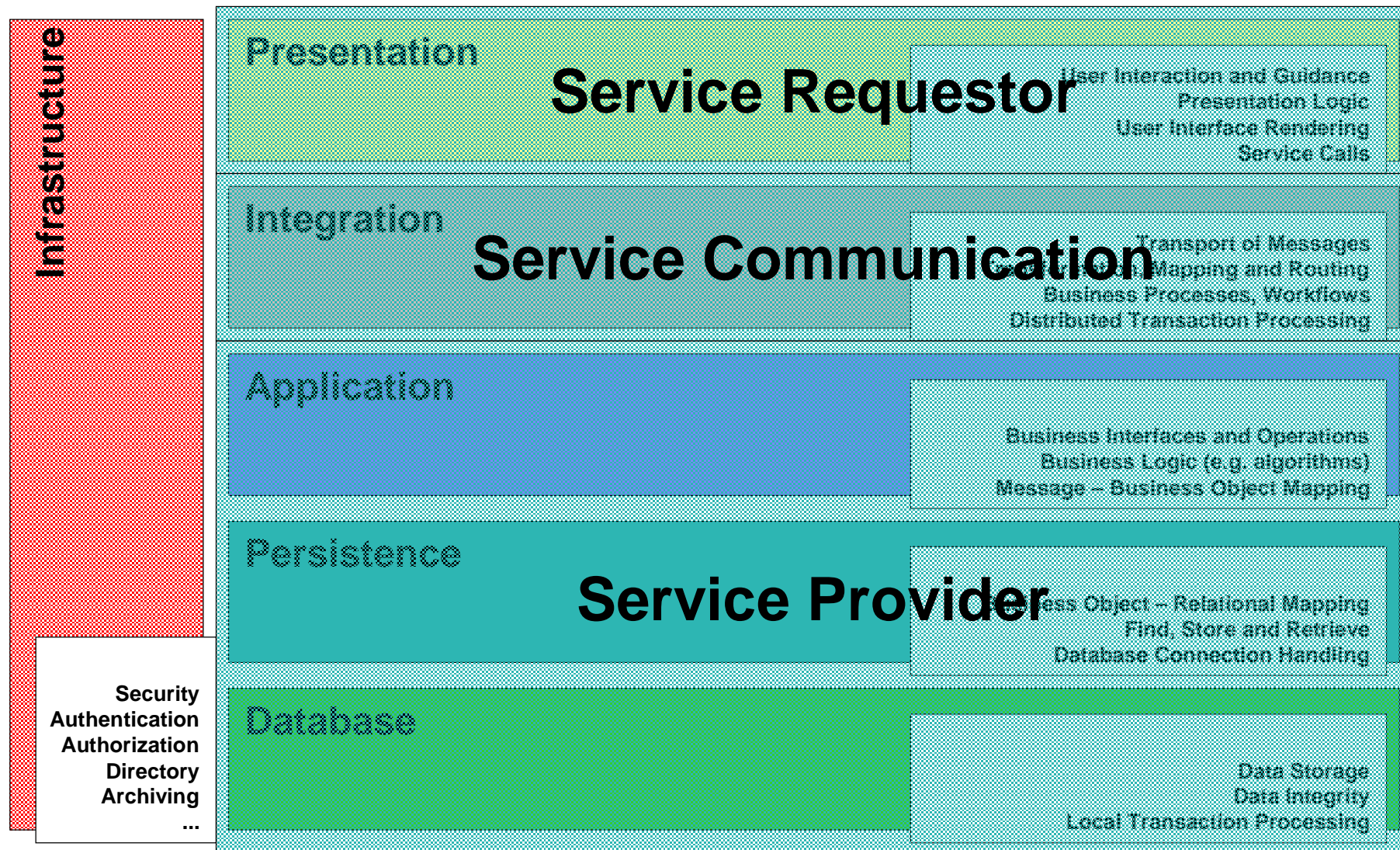
§ Which VSE programs can call an external Web Service?

- 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**

§ VSE SOAP Engine requires CICS TS

- But you can use MRO or remote program definitions to use programs running in CICS/VSE 2.3

Layered Software Architecture



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

§ Use the new CICS2WS tool

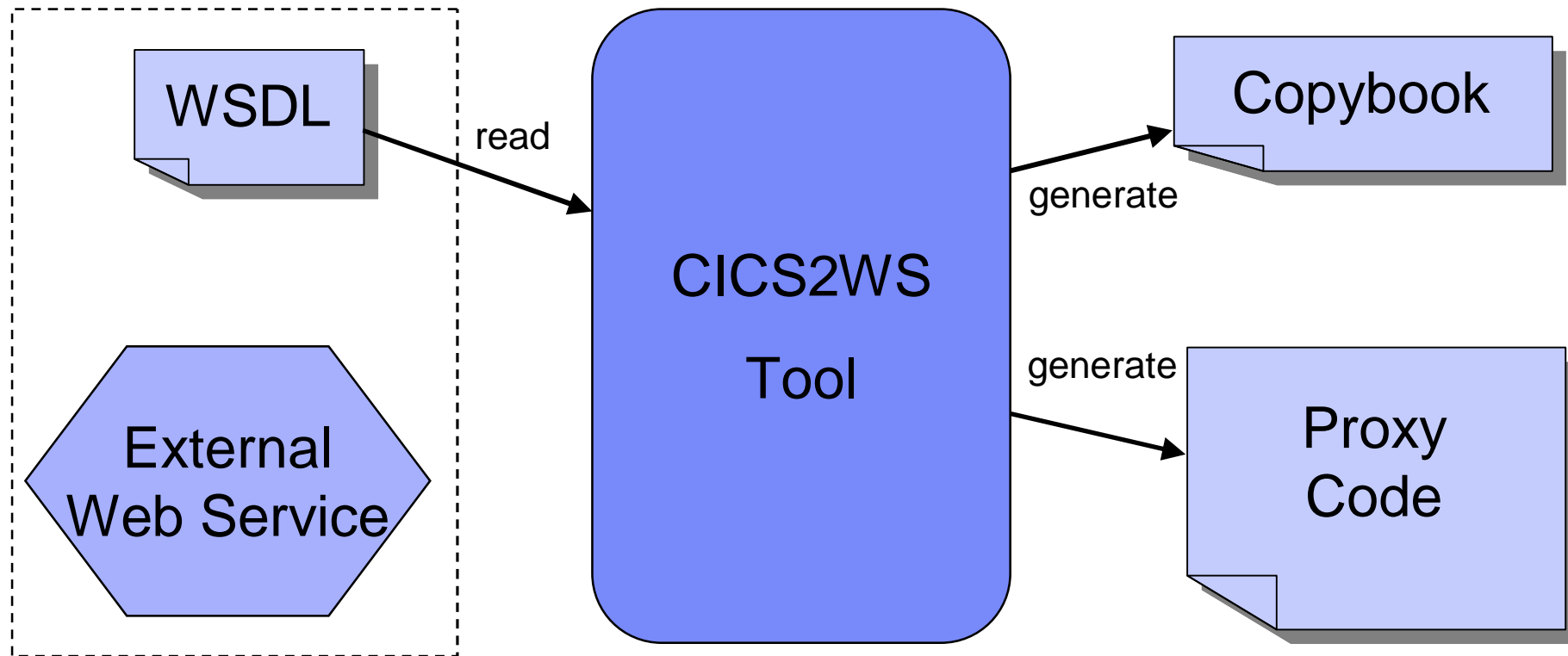
- Generates proxy code and WSDL files
- Proxy code is in assembler language
 - No extra charged compiler needed (e.g. COBOL or PL/I)
 - Code is very simple, straight forward
 - Usually no manual changes needed in proxy code

New CICS2WS Tool

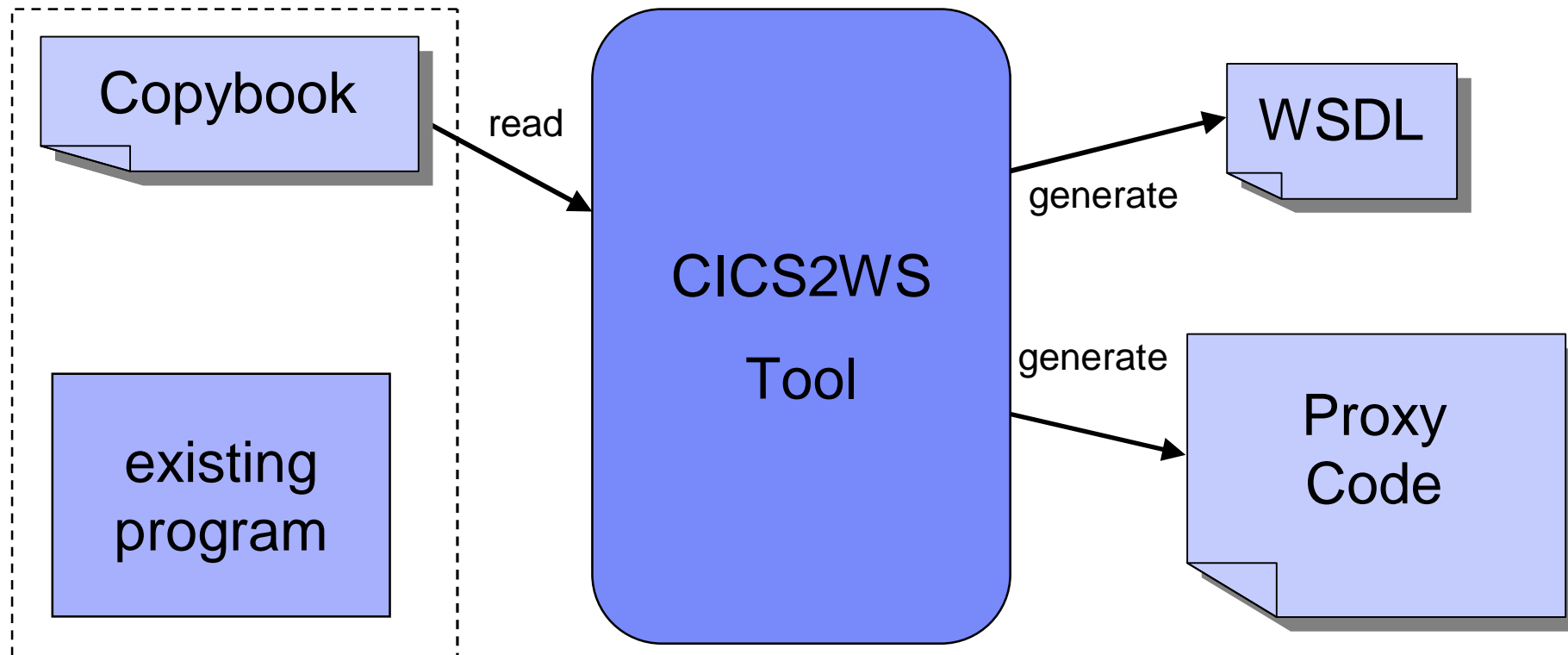
- § **The tool runs on your PC or workstation**
- § **Implemented in Java**
- § **VSE as a SOAP client (service requestor)**
 - Reads the WSDL file
 - Generates the proxy code (Assembler)
 - Generates a COMMAREA mapping (copybook)
 - in COBOL, PL/I or Assembler
- § **VSE as a SOAP server (service provider)**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



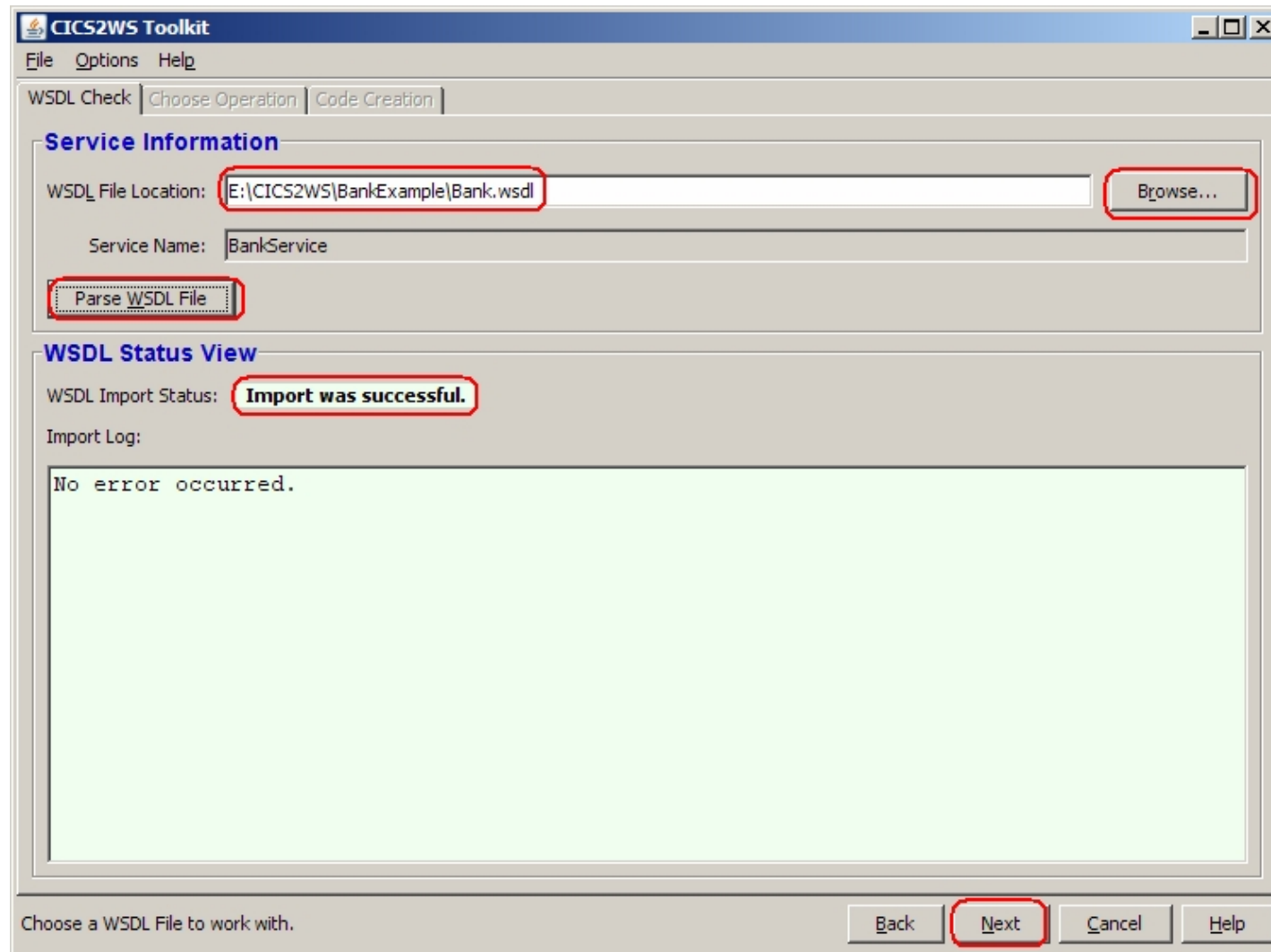
VSE as a SOAP server (service provider)



Live Demo – Use a Web Service



Live Demo – Use a Web Service



Live Demo – Use a Web Service

CICS2WS Toolkit

File Options Help

WSDL Check Choose Operation Code Creation

Service Information

Service Name: BankService

WSDL File Location: E:\CICS2WS\BankExample\Bank.wsdl

WSDL Information

Content	COMMAREA name	Type	Length	IDP
Operations				
transfer				
Input Parameters				
amount	amount	int	4	
targetAccount	targetAccount	int	4	
sourceAccount	sourceAccount	int	4	
Output Parameters				
Account	Account	Account	8	
accountNr	accountNr	int	4	
amount	amountOut	int	4	
getCus				
getAcc				

Use the context menu to change names and lengths of parameters to enable the desired method for use. Finally, select a method.

Back Next Cancel Help

Live Demo – Use a Web Service

The screenshot shows the CICS2WS Toolkit interface. The window title is "CICS2WS Toolkit" and it has a menu bar with "File", "Options", and "Help". Below the menu bar are three tabs: "WSDL Check", "Choose Operation", and "Code Creation".

The "Generate Copybook" section is active. It contains three radio buttons: "ASM Copybook", "Cobol Copybook" (which is selected and circled in red), and "PL/I copybook". Below these is a "Save Copybook" button, also circled in red. To the right of these controls is a text area containing the following COBOL code:

```
1 Copybook.  
3 amount PIC 9(9) COMP.  
3 targetAccount PIC 9(9) COMP.  
3 sourceAccount PIC 9(9) COMP.  
3 Account.  
5 accountNr PIC 9(9) COMP.  
5 amountOut PIC 9(9) COMP.
```

The "Generate Assembler Proxy Code" section is below. It contains four text input fields: "Proxy Name" (with "BANKPRXY" entered and circled in red), "Service Name" (with "BankService" entered), "Service URL" (with "http://localhost:9080/BankWeb/services/Bank" entered), and "Target Namespace" (with "http://bankService.vse.ibm.com" entered). Below these fields is a "Create Code" button, also circled in red.

At the bottom of the window, there is a status bar with the text "Choose your code type to save a copybook, then create your Assembler Proxy Code." and four buttons: "Back", "Finish", "Cancel", and "Help".

Live Demo – Use a Web Service

1 Copybook.

```

3 amount    PIC 9(9) COMP.
3 targetAccount  PIC 9(9) COMP.
3 sourceAccount  PIC 9(9) COMP.
3 Account.
    5 accountNr  PIC 9(9) COMP.
    5 amountOut  PIC 9(9) COMP.
    
```

```

*****
SECTION
*****
OF PARAMETER BLOCK
H OF BLOCK
NSE CODE
NSE CODE 2
) COMMAREA FOR SOAP CALL
BUFFER FOR OUTPARAMS
IP19_LEN  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

CICS2WS Toolkit

File Options Help

Create Service | Add Operation to Service | Service Summary

CICS Application Data

Path to Source: E:\CICS2WS\BankExample\BankCB.cbl Browse...

Source Code Language: COBOL

Skip control characters:

Parse Source Successfully parsed the source file.

Basic Service Data

Service Name: VSEBankService

Service URL: http://vsedemo.boeblingen.de.ibm.com:80/cics/CWBA/IESSOAPS

URN: urn:IESSOAPD:BANKSERV

Service Description:

Program Names

Proxy Name: BANKSERV User Program Name: VSEBANK

Browse for a source file, parse it and then set Basic Service Data and Program Names. Back Next Cancel Help

Live Demo – Create a Web Service

CICS2WS Toolkit

File Options Help

Create Service Add Operation to Service Service Summary

New Operation

Name:

Description:

Input/Output Parameter Mapping

Content	Instance	Type	Length	Offset
COMMAREA Variables				
CT CUSTOMER	Group		44	0
CT ACCOUNT	Group		8	44
ST ACCOUNTNR	Field	INTEGER	4	0
ST AMOUNT	Field	INTEGER	4	4
ST SOURCEACCOUNT	Field	INTEGER	4	52

Buttons: Add, Quick Add, Remove, Move Up, Move Down

Content	Instance	Type	COMMAREA Name
Mapped Variables			
Input Parameter			
ST amount	Field	INTEGER	AMOUNT
ST sourceAccount	Field	INTEGER	SOURCEACCOUNT
ST targetAccount	Field	INTEGER	TARGETACCOUNT
Output Parameter			
CT transferReturn	Group		ACCOUNT

Provide an operation name, then select source and target in the trees to use the buttons.

Buttons: Back, Next, Cancel, Help

Live Demo – Create a Web Service

The screenshot shows the CICS2WS Toolkit interface with the following configuration:

- Service Information:**
 - Service Name: VSEBankService
 - Service Location: http://vsedemo.boeblingen.de.ibm.com:80/cics/CWBA/IESSOAPS
 - Target Namespace: urn:IESSOAPD:BANKSERV
- Operation Information:**

Content	Instance	Type
Operation		
transfer		
Input Parameter		
S_T amount	Field	INTEGER
S_T sourceAccount	Field	INTEGER
S_T targetAccount	Field	INTEGER
Output Parameter		
C_T transferReturn	Group	
S_T targetAccountNr	Field	INTEGER
S_T newAmount	Field	INTEGER
- Generate WSDL and Assembler Proxy Code:**
 - Buttons: Create WSDL File, Create ASM Code

At the bottom, there are navigation buttons: Back, Finish, Cancel, and Help. A note at the bottom left says: "Check your Web Service, then create the desired files."

Disadvantages of Web services

§ When should you not use Web Services?

- When you have very high performance requirements
 - Communication using SOAP/XML is very time consuming
- When you transport large amounts of data
 - XML data can get really huge
- If you require transaction security
 - No 2 phase commit
- When you want to access the data directly
 - SOAP is program to program communication

§ Similar functions provide

- CICS Transaction Gateway
- MQ Series

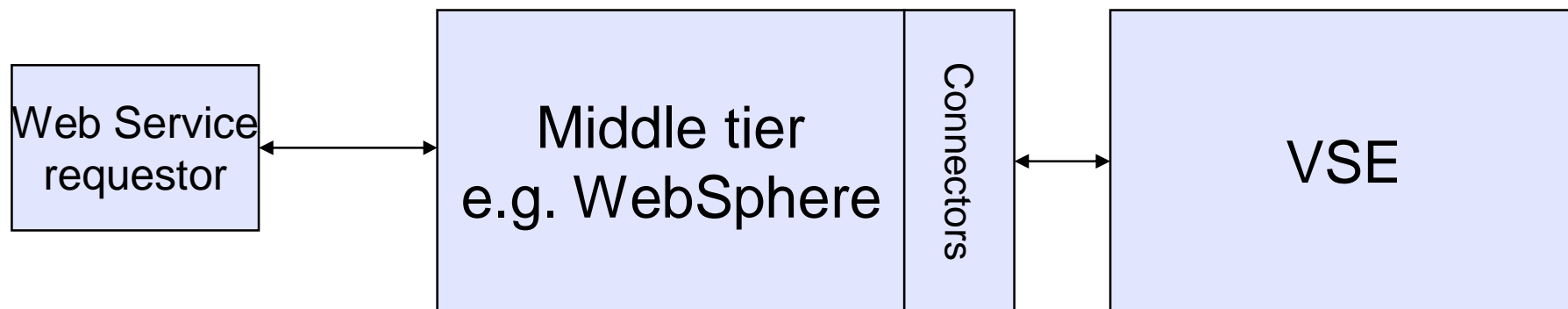
Other possibilities to participate into SOA solutions

§ 2 Tier Solutions

- The Web Service requestor or provider runs on VSE itself

§ 3 Tier Solutions

- The Web Service is implemented on a middle tier system, but accesses VSE data or programs



3 tier SOA solutions

§ Access to VSE using connectors

- CICS Transaction Gateway (CICS programs)
- DB2 Connect (DB2 data)
- VSE Java-based Connectors (VSAM, DL/I, Jobs, ...)
- MQ Series

§ Middle tier

- Using modern technology and products
- E.g.. WebSphere SOA Products (Enterprise Service Bus, WebSphere Process Server)
- Can also run on Linux on System z

Documentation

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

- <http://www.zjournal.com/index.cfm?section=article&aid=281>
- <http://www.zjournal.com/index.cfm?section=article&aid=320>
- Includes COBOL sample code

§ **Web Services**

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

§ **What is SOA?**

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

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

Questions ?

