2009 System z Expo

October 5 – 9, 2009 – Orlando, FL

Using SOA Web Services with z/VSE

zEO03

Ingo Franzki, IBM

Authorized IBM Training

Trademarks

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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

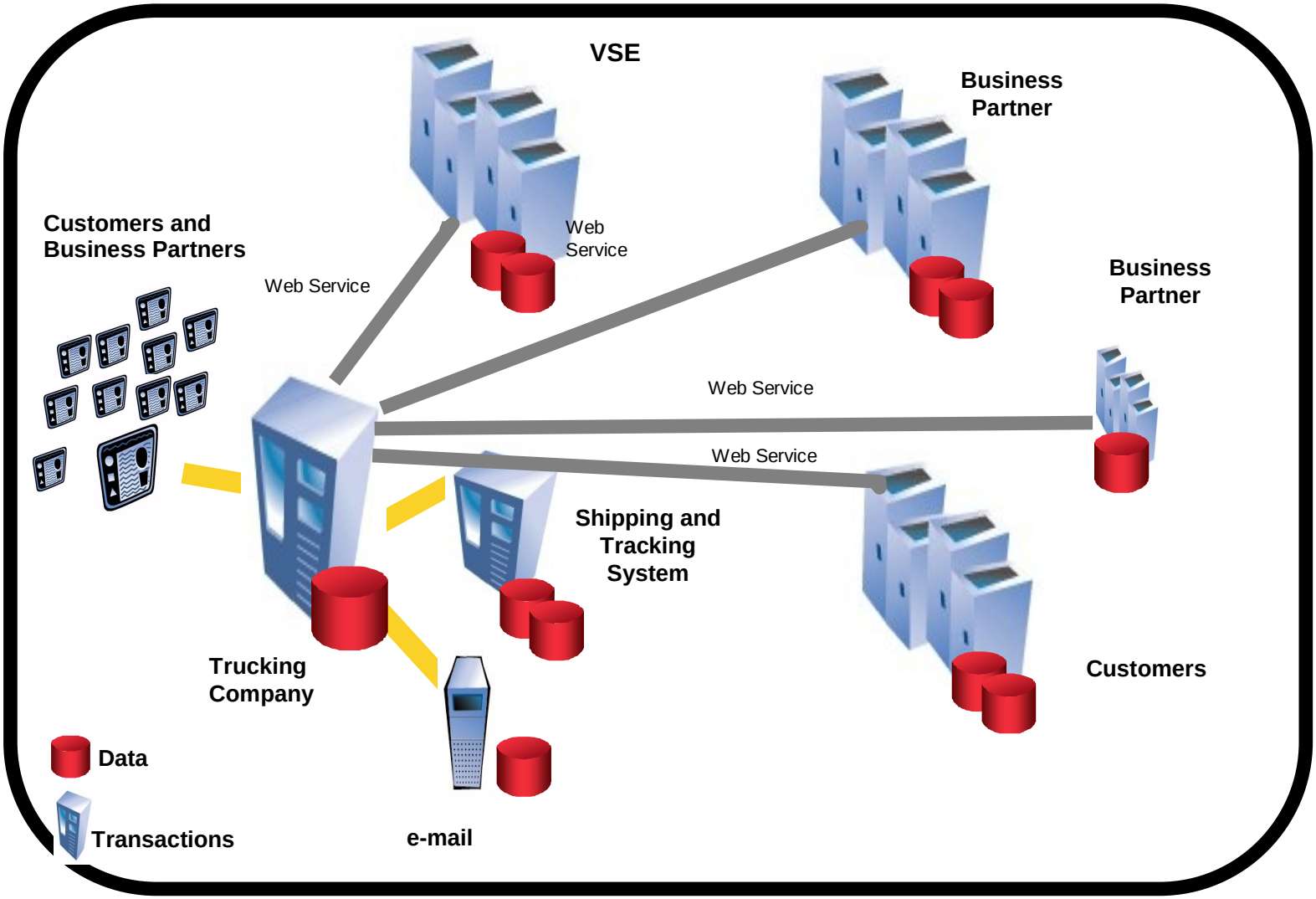
The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

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 technolo
- 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)?

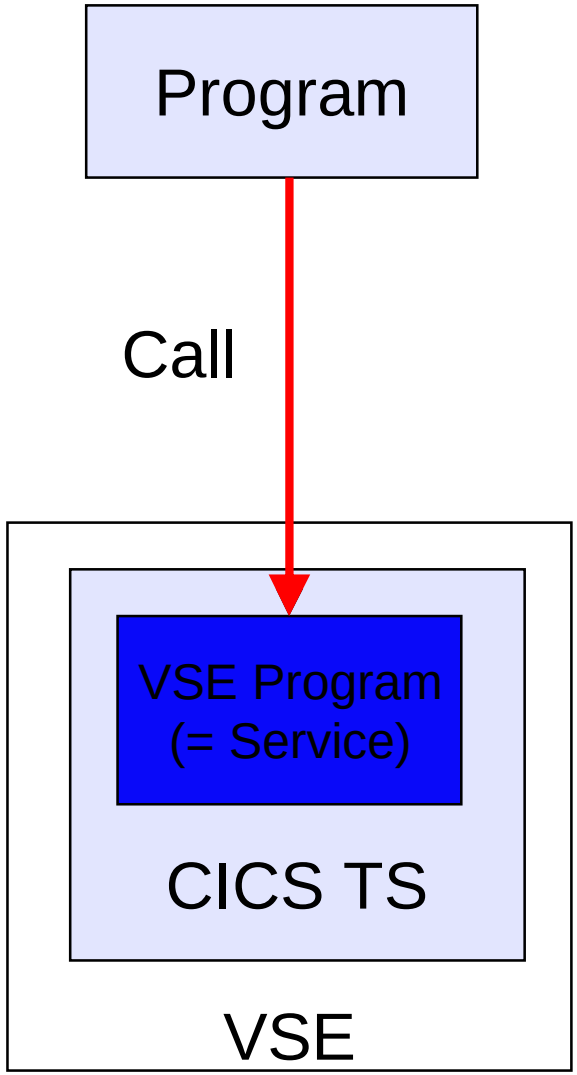
- Web services are the most common technology standards used to implement SOA
  - However, they are not the only technology one can use to develop the parts of an SOA
- Many SOAs -- most, in fact -- involve the integration of legacy data
  - contained in systems that use technology such as MQSeries and 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

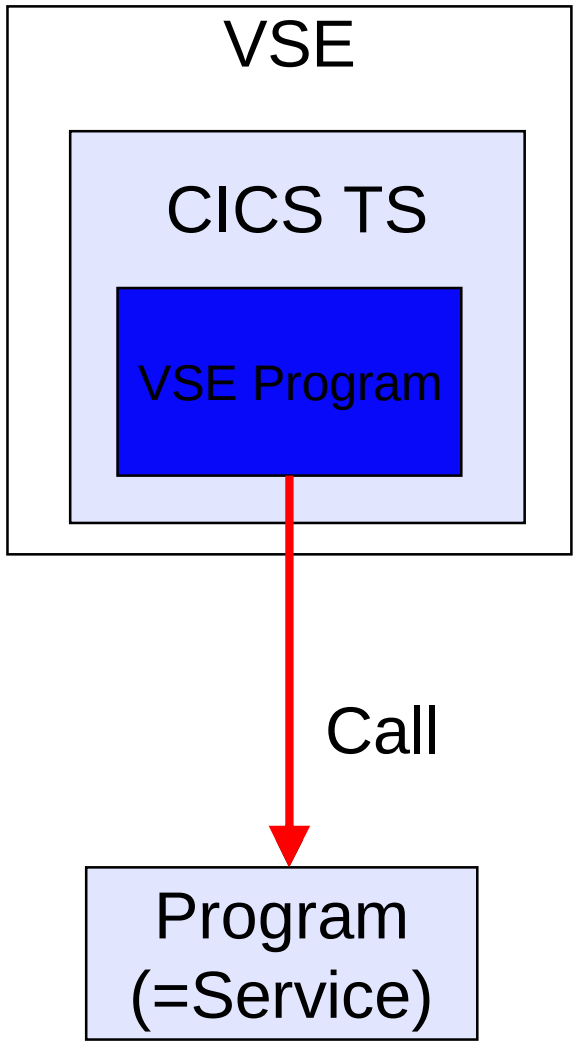
What is a Web Service?

- 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



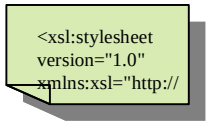
What is a Web Service?

- 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

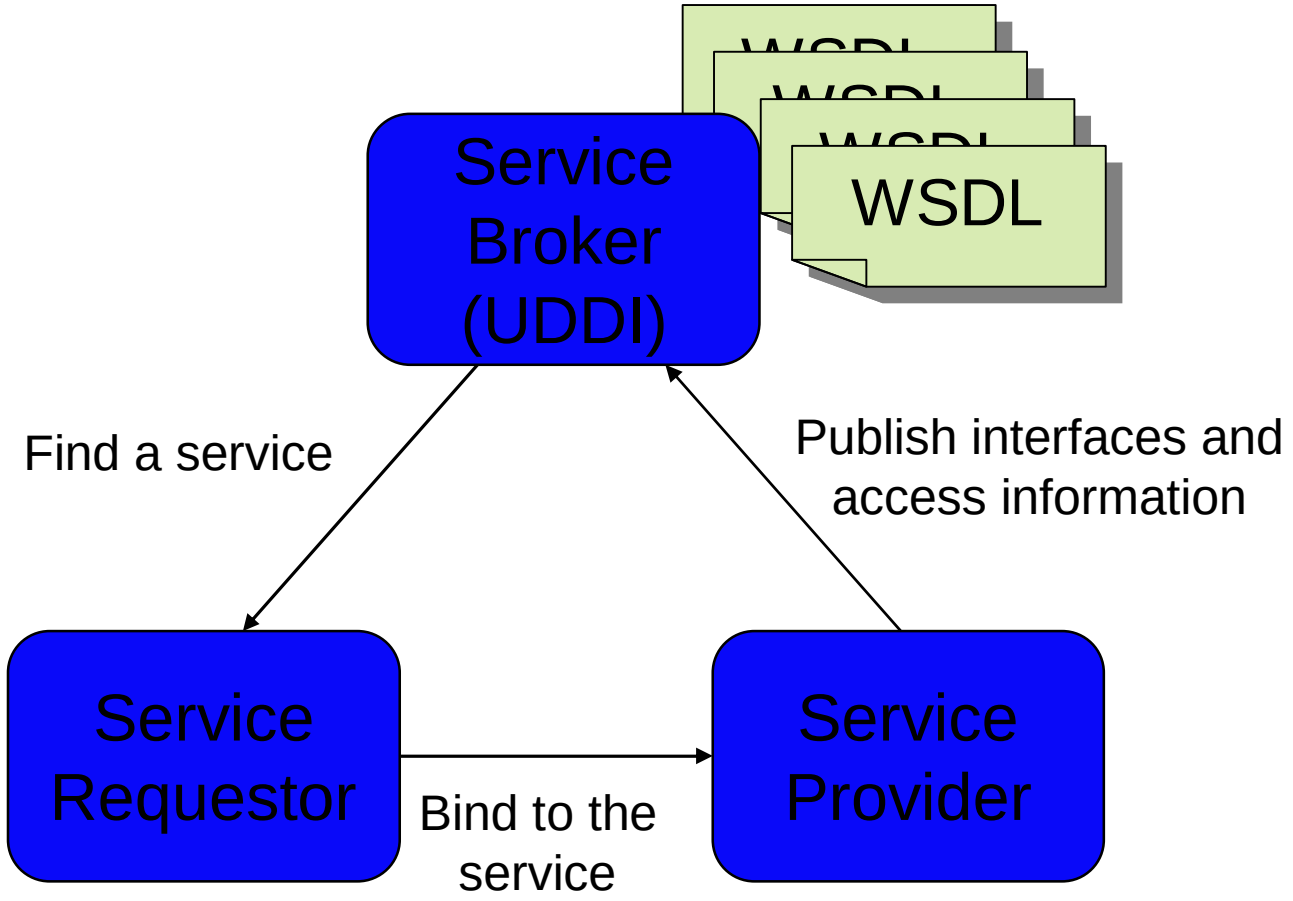
XML



SOAP -

HTTP - Transport Service

TCP/IP - Street

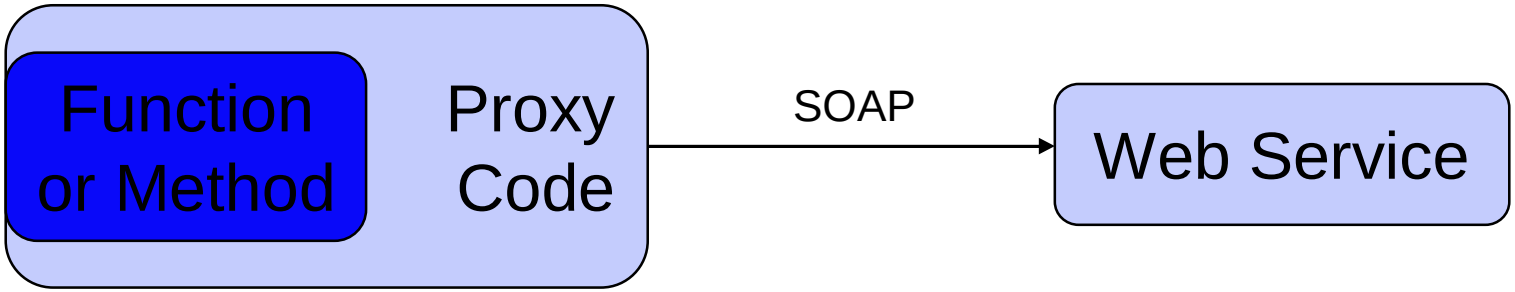


News with z/VSE V4.2

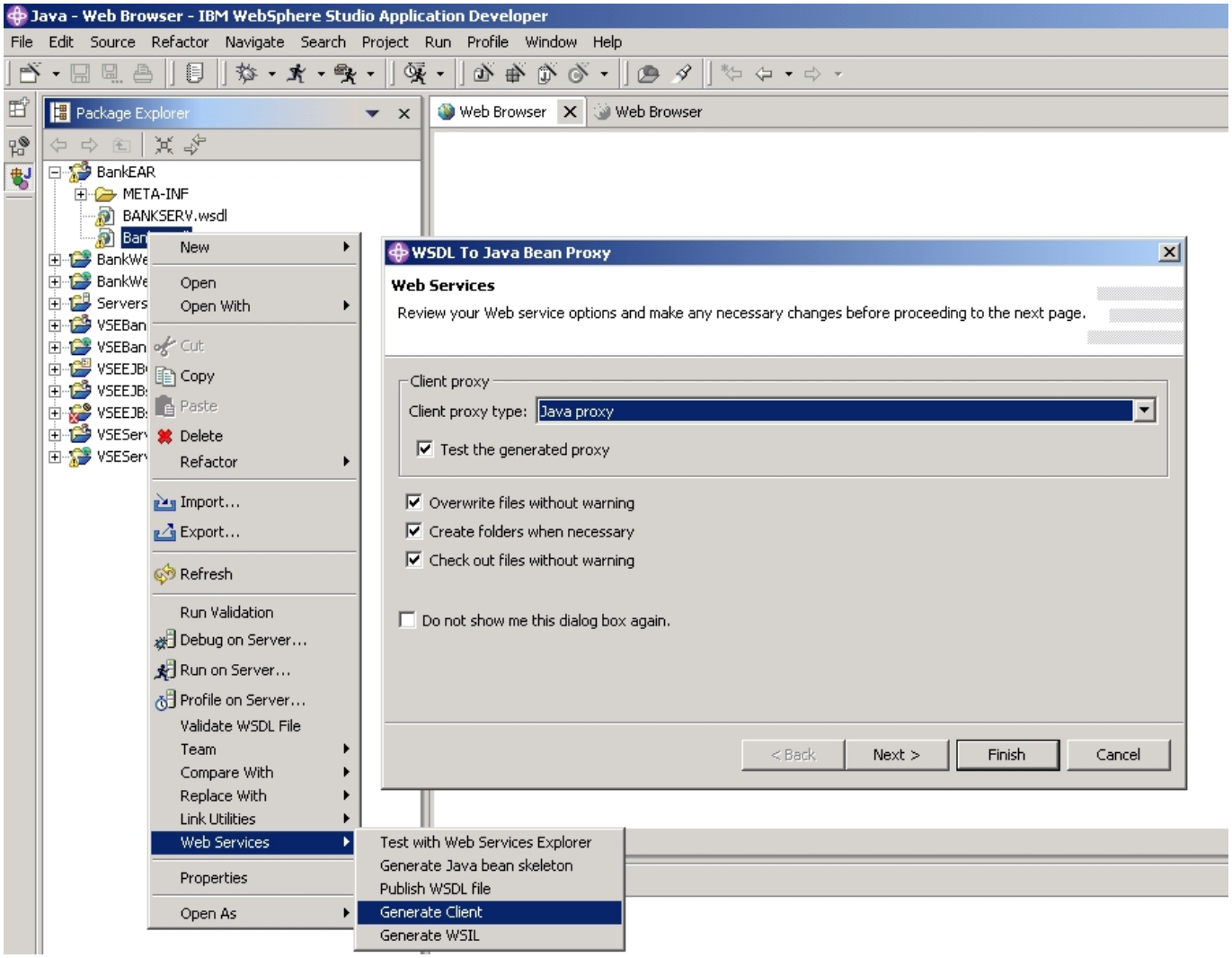
- Web Service Security
  - Message Encryption
    - Transport-layer authentication using: HTTPS/SSL
  - Authentication:
    - Transport-layer authentication using:
      HTTP authentication (Basic and Digest Access Authorization)
      or SSL Client Authentication with HTTPS.
    - Message-layer authentication using:
      A UsernameToken (plaintext password or password digest)
      or an X.509 Certificate (BinarySecurityToken).
- Long-name to short-name mapping
  - To support parameter names longer than 16 characters
  - A mapping table can be used to translate long (external) names to short (internal) names
- New Redbook-style document: “How to use Web Servcies with z/VSE“
  - http://www.ibm.com/servers/eserver/zseries/zvse/documentation/ebusiness.html#soap

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

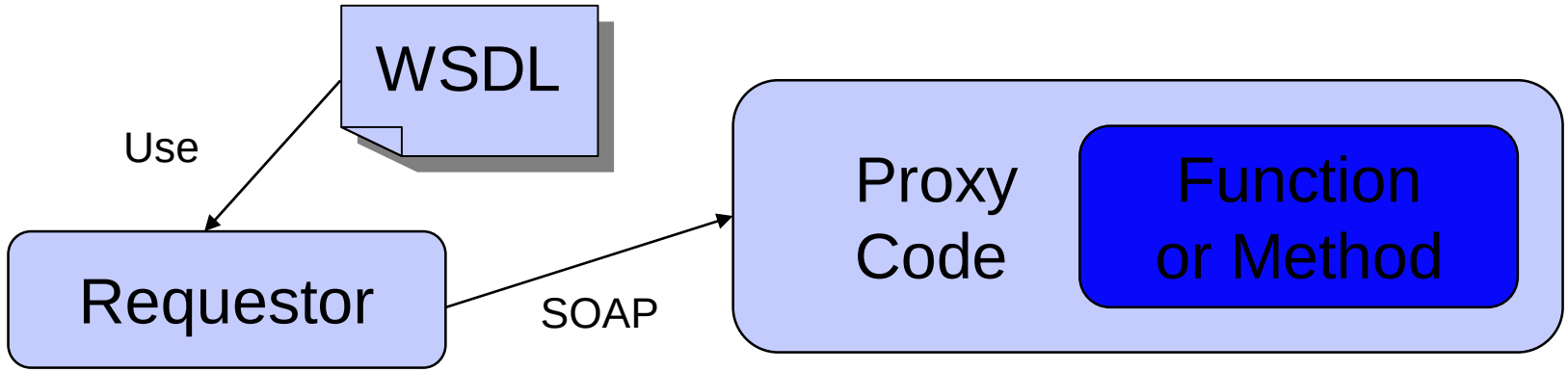


Using Web Services with Java or MS .Net

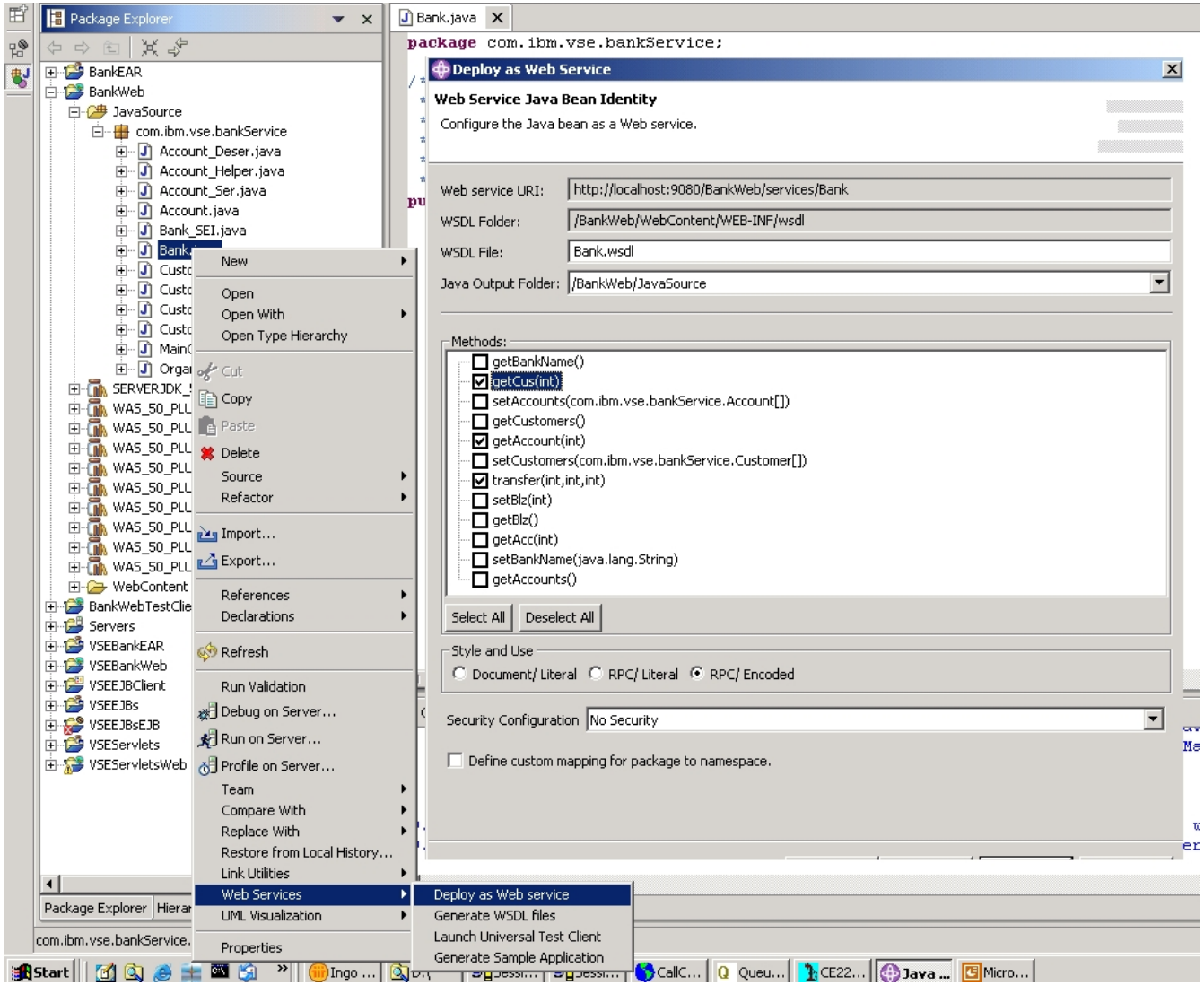


Using Web Services with Java or MS .Net

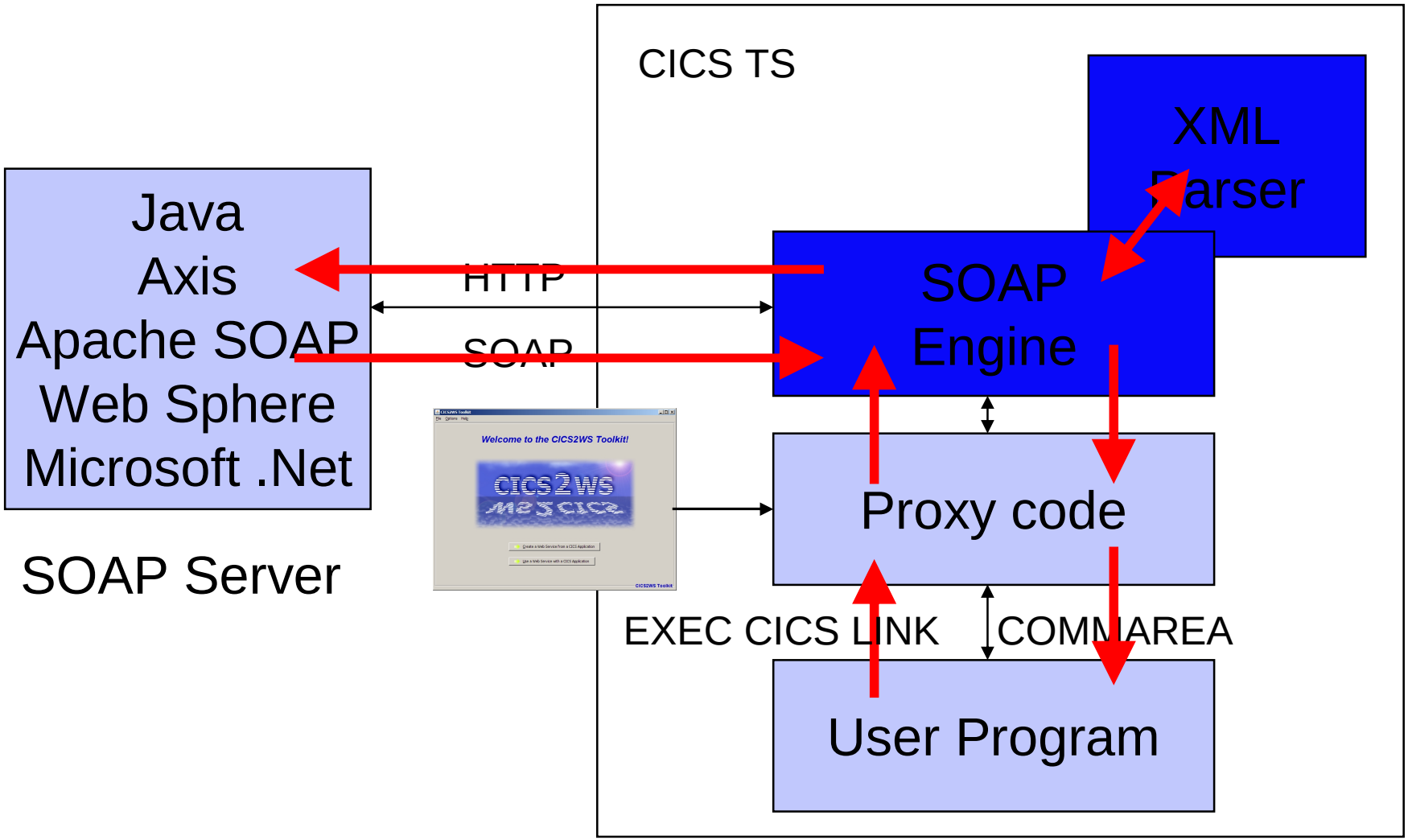
- Create/provide a new Web Service
  - You have a function or method that implements some kind of service that you want to provide
  - Use a tool like Rational Application Developer (RAD/WSAD) or Microsoft Visual Studio to model a Web Service
    - Generate a Web Service Description (WSDL) and publish 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



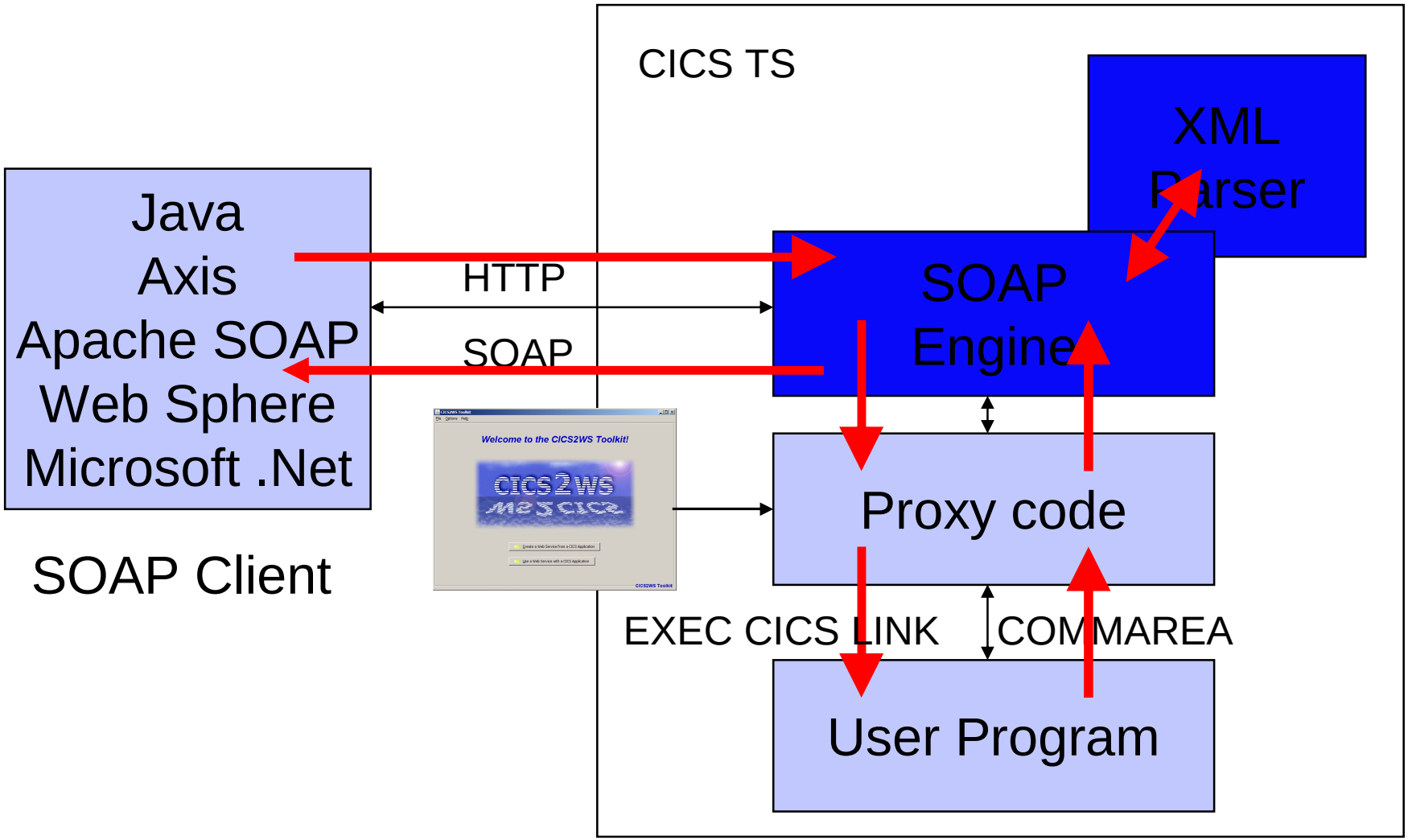
Using Web Services with Java or MS .Net



Using Web Services with VSE – SOAP client



Providing Web Services with VSE – SOAP server



VSE SOAP Engine

- 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?

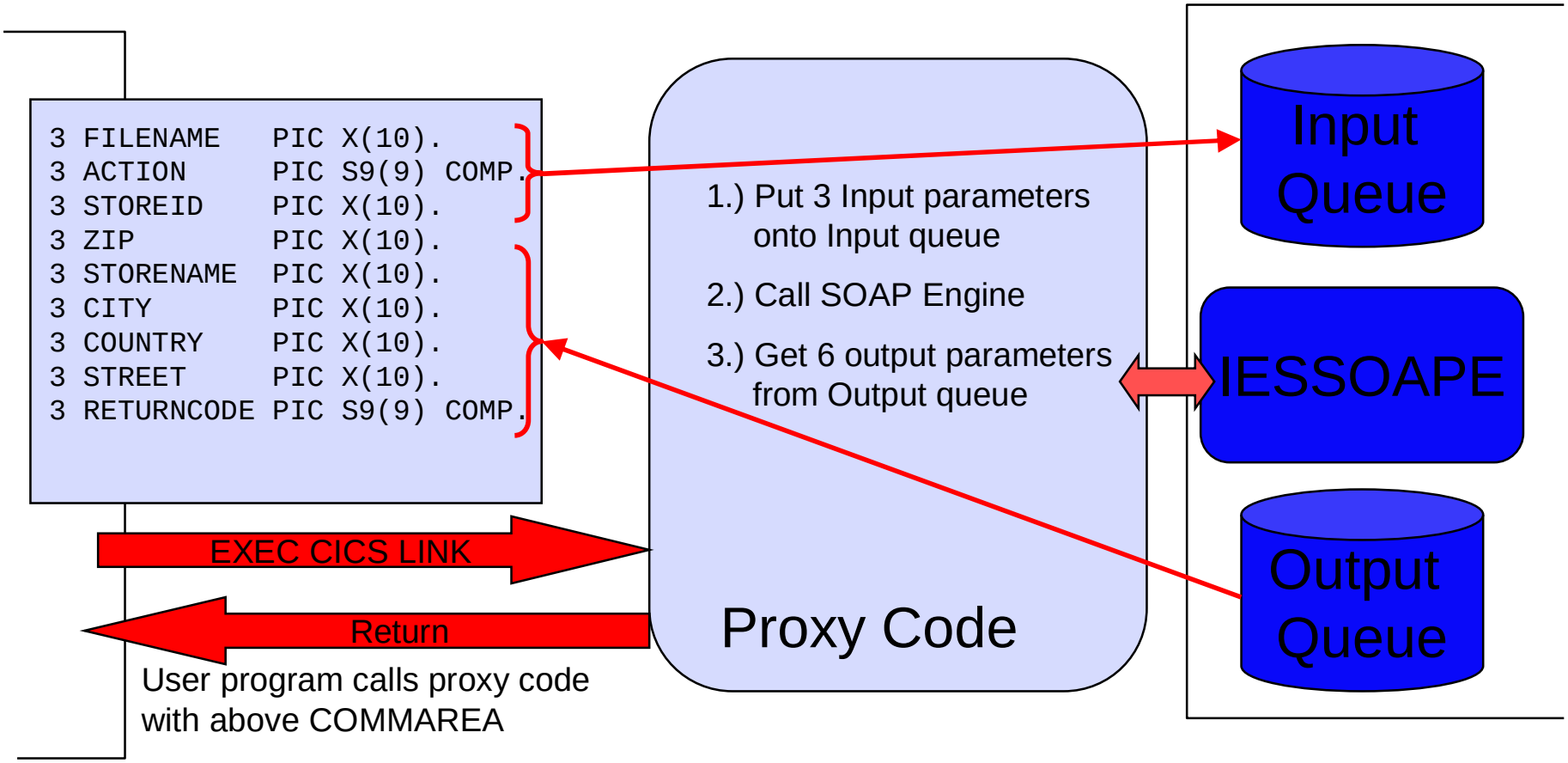
- Although the SOAP Engine interface uses standard CICS methods, special coding is needed to interface with the VSE SOPA Engine.
- Proxy code ‘maps’ between standard COMMAREA interface and SOAP Engine
  - All SOAP specific handling is done in proxy code
  - User applications calls the proxy code or gets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

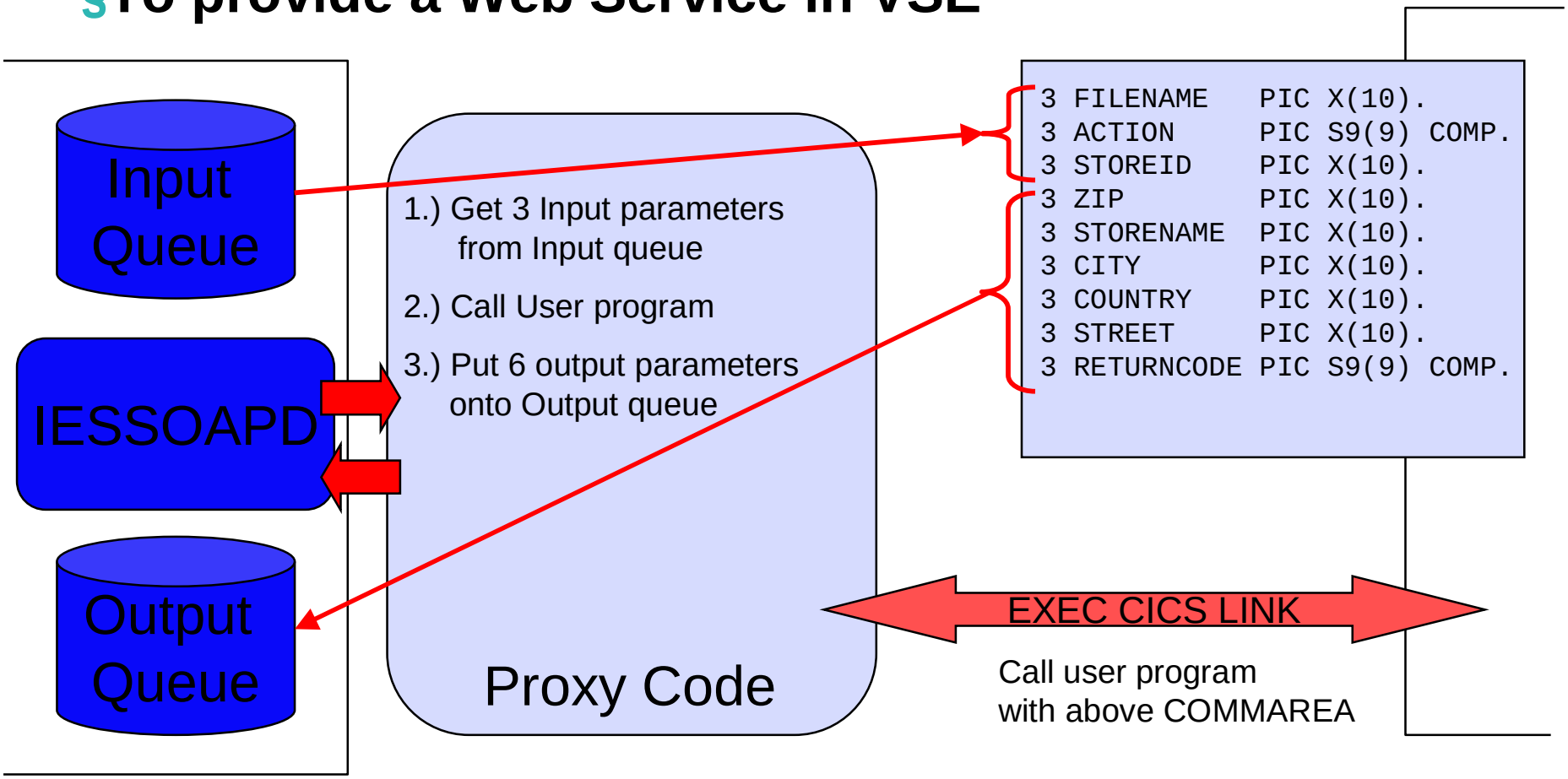


What does the proxy code do?

- 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

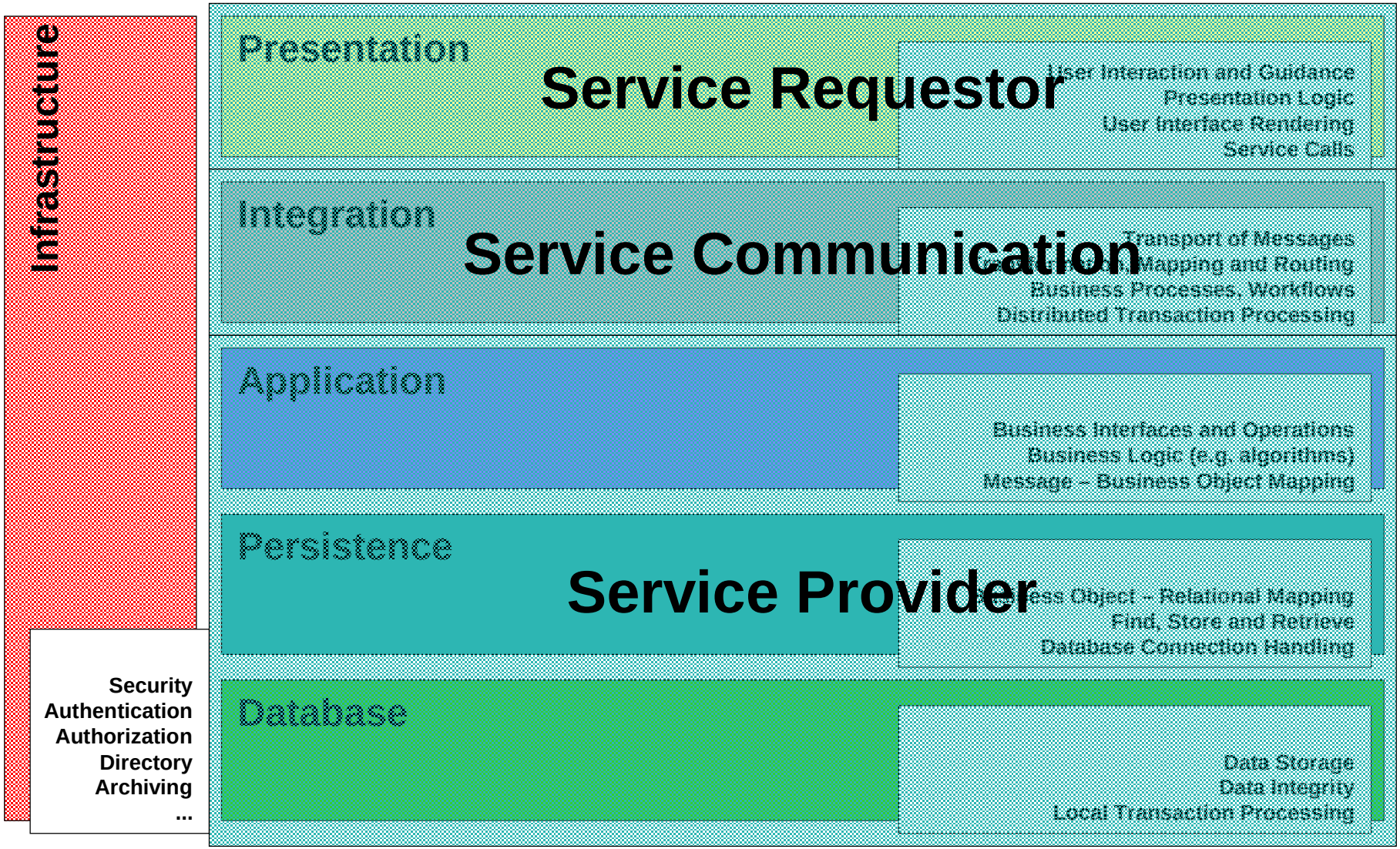
Which programs can be used with Web Services?

- 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

Requirements for web services used with z/VSE

- You need to have an WSDL that describes:
  - where the web service is located (URL)
  - what methods/operations it provides
  - what input and output parameters it expects
- The web service must follow these requirements:
  - It must use the SOAP 1.1 protocol
  - It must use SOAP encoding (use=”encoded”), literal style is not supported by CICS2WS
  - It must use RPC style. Document style is not supported
  - Parameter names must not be larger than 16 characters
    - unless you use long-name to short-name mapping

Layered Software Architecture



How to write the proxy code

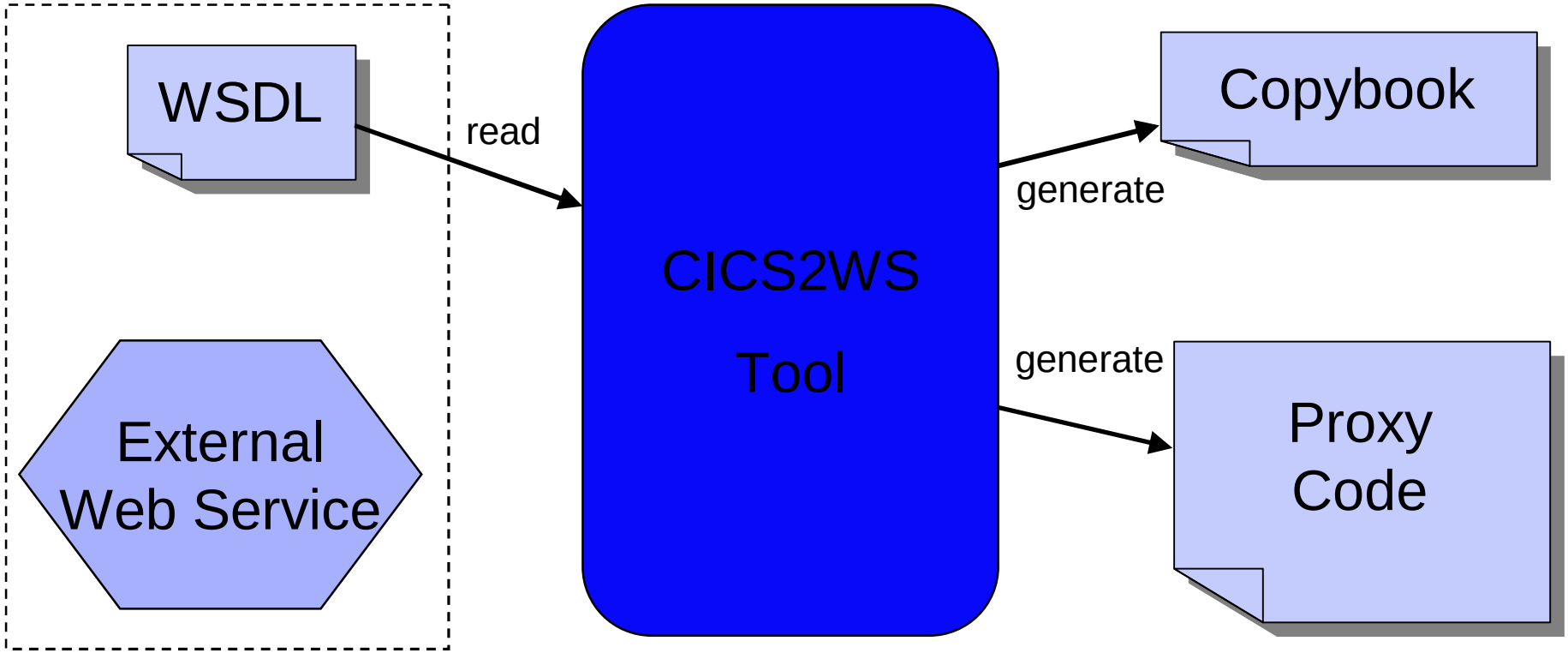
- You can write the proxy code “by hand”
  - Not very difficult, use samples as skeleton
  - COBOL Example (from Rich Smrcina):
    - ftp://ftp.software.ibm.com/eserver/zseries/zos/vse/download/xmps/soap_cobol_rsmrcina.zip
- Use the new CICS2WS tool
  - Generates proxy code and WSDL files
  - Proxy code 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

CICS2WS Tool

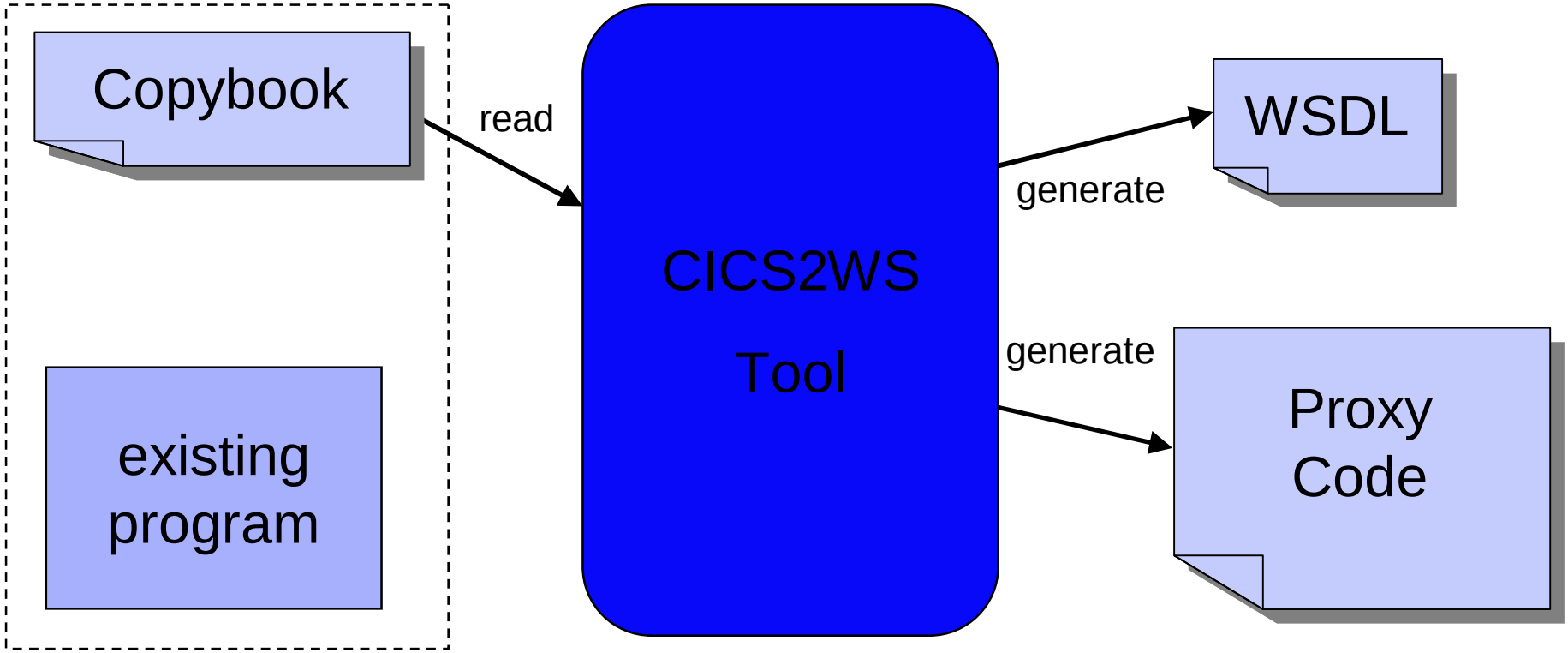
- The tool runs on your PC or workstation
- Implemented in Java
- VSE as a SOAP client (service requestor)
  - Reads the WSDL file
  - Generates the proxy code (Assembler)
  - Generates a COMMAREA mapping (copybook)
    - in COBOL, PL/I or Assembler

- 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)

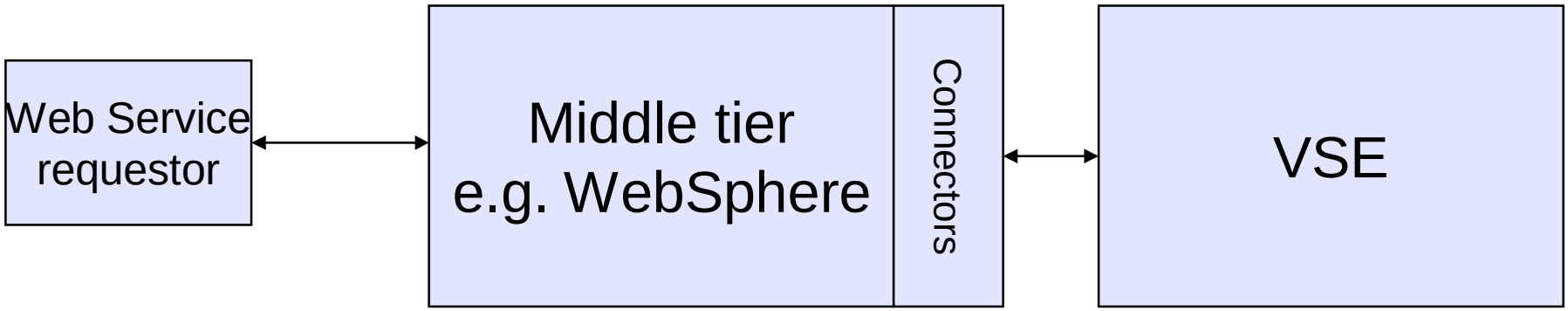


Disadvantages of Web services

- 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

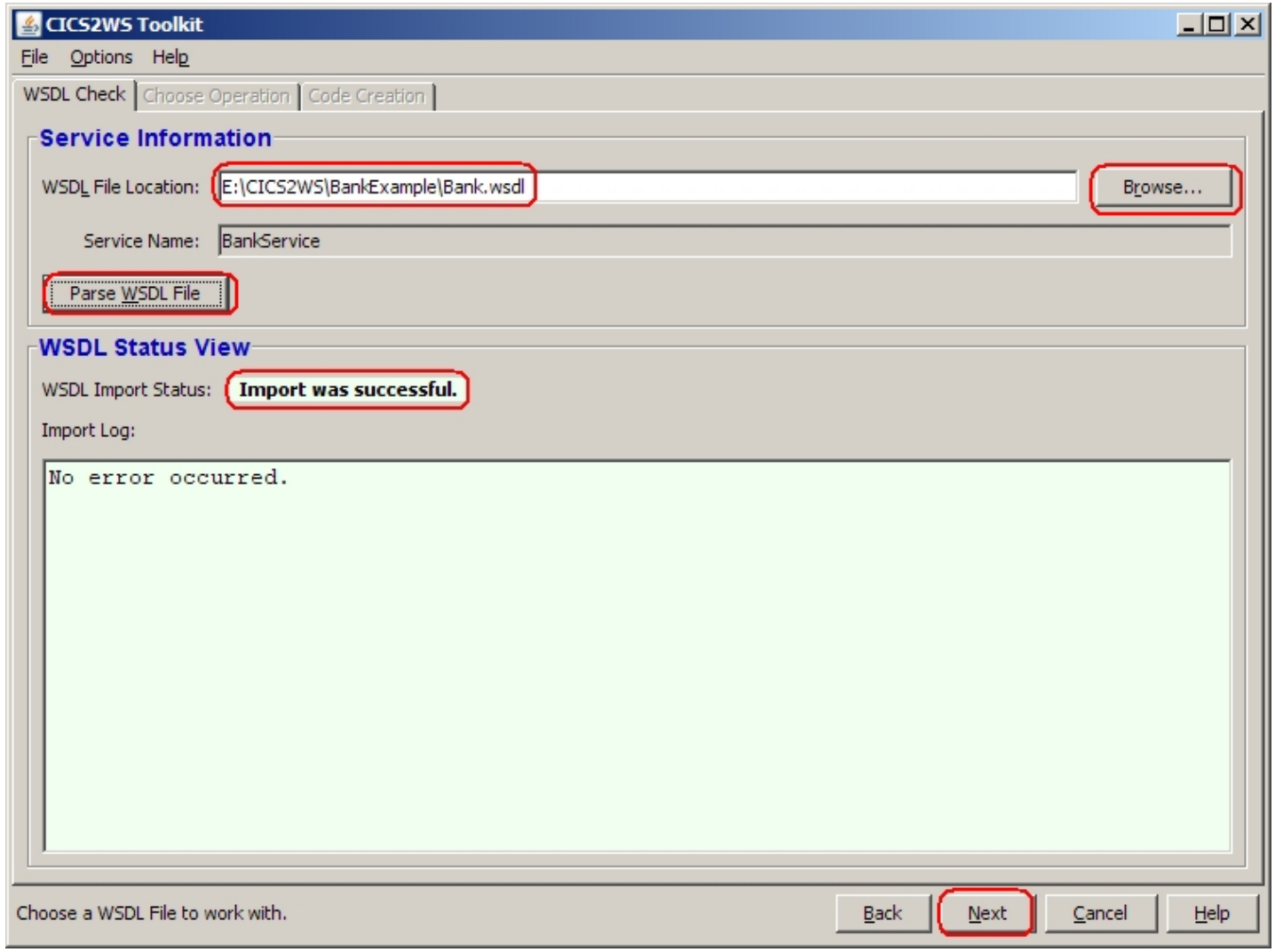


3 tier SOA solutions

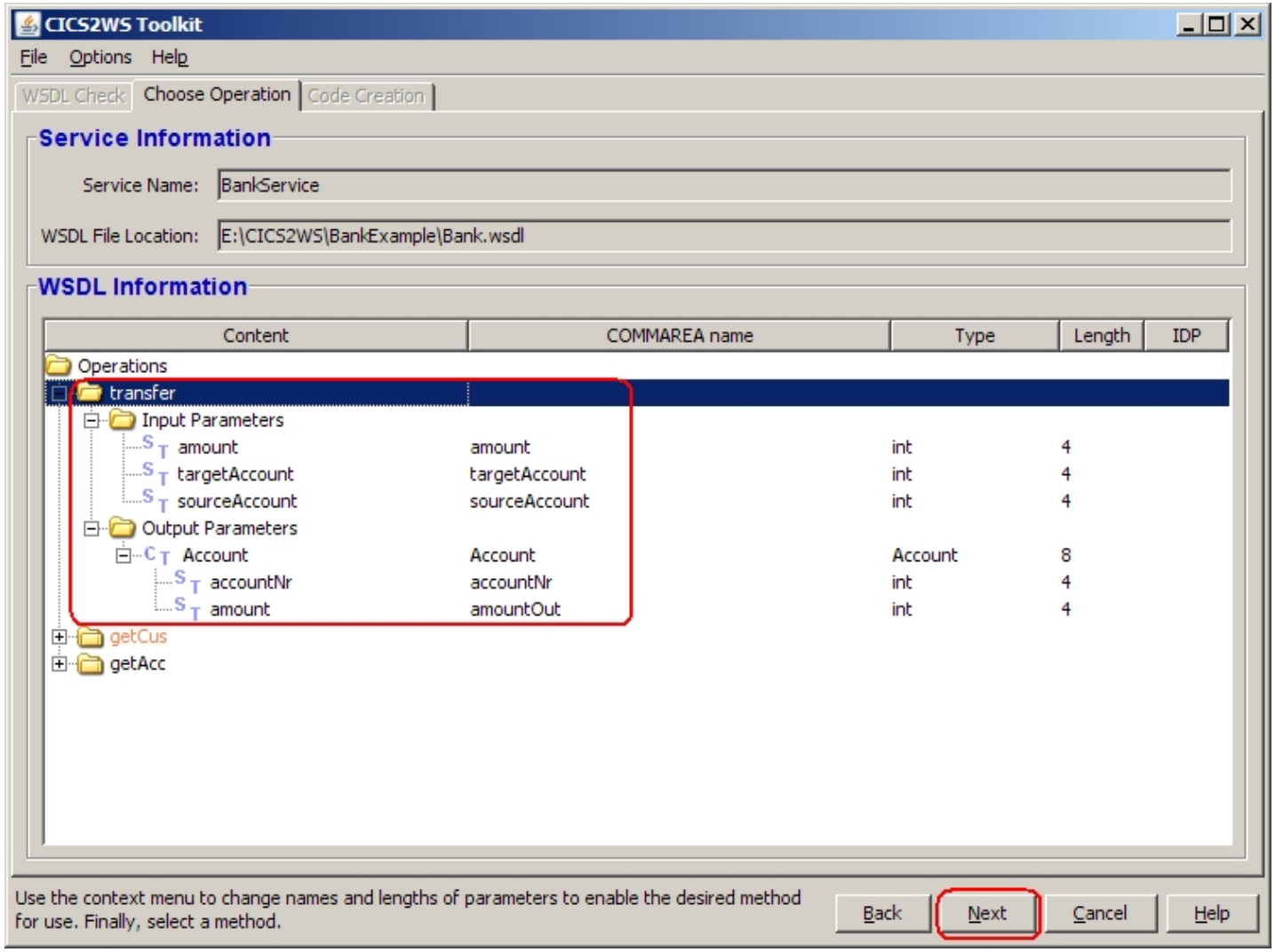
- Access to VSE using connectors
  - CICS Transaction Gateway (CICS programs)
  - DB2 Connect (DB2 data)
  - VSE Java-based Connectors (VSAM, DL/I, Jobs, …)
  - MQ Series
- Middle tier
  - Using modern technology and products
  - E.g.. WebSphere SOA Products (Enterprise Service Bus, WebSphere Process Server)
  - Can also run on Linux on System z

Live Demo – Use a Web Service

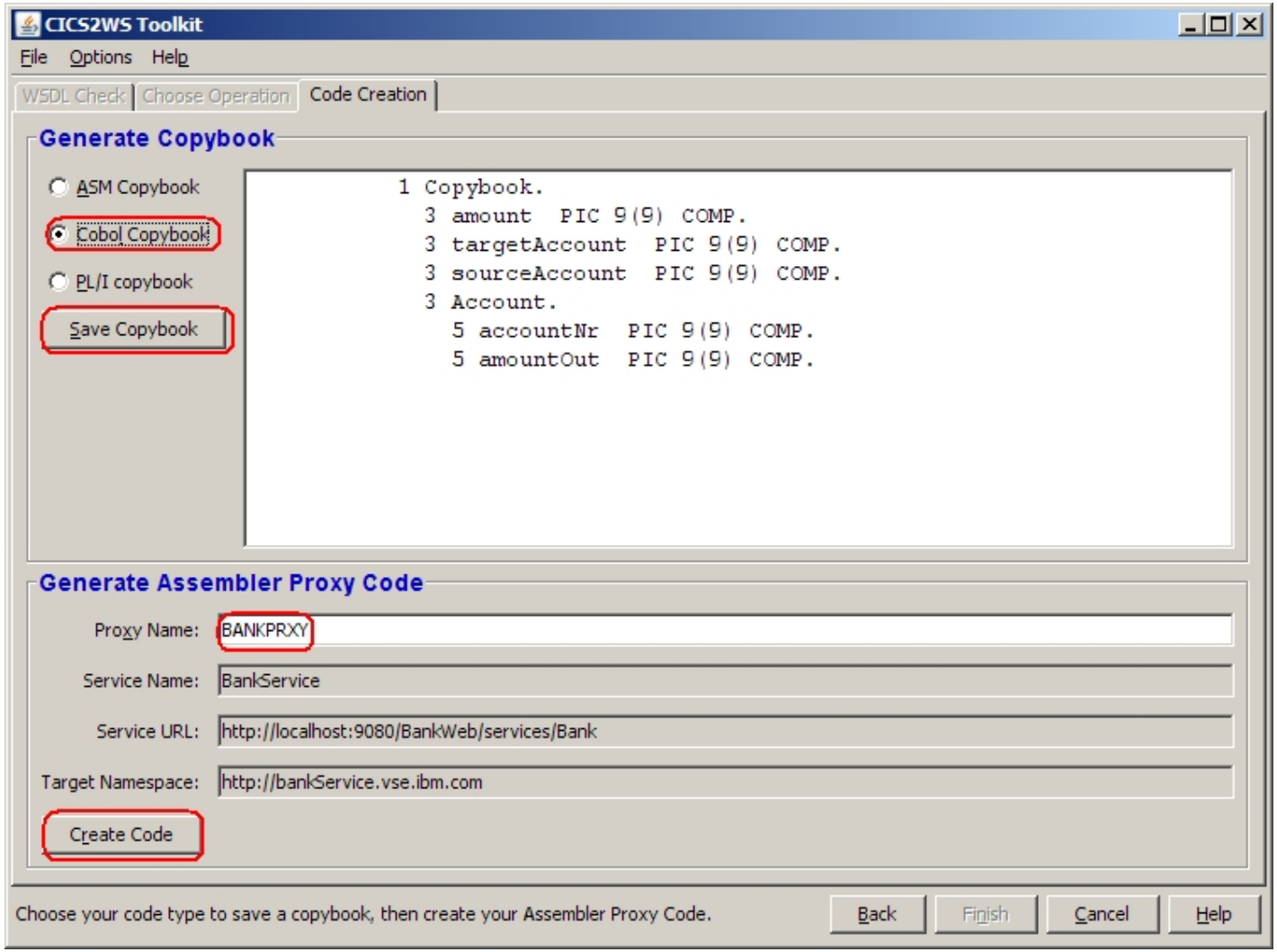
Live Demo – Use a Web Service



Live Demo – Use a Web Service



Live Demo – Use a Web Service



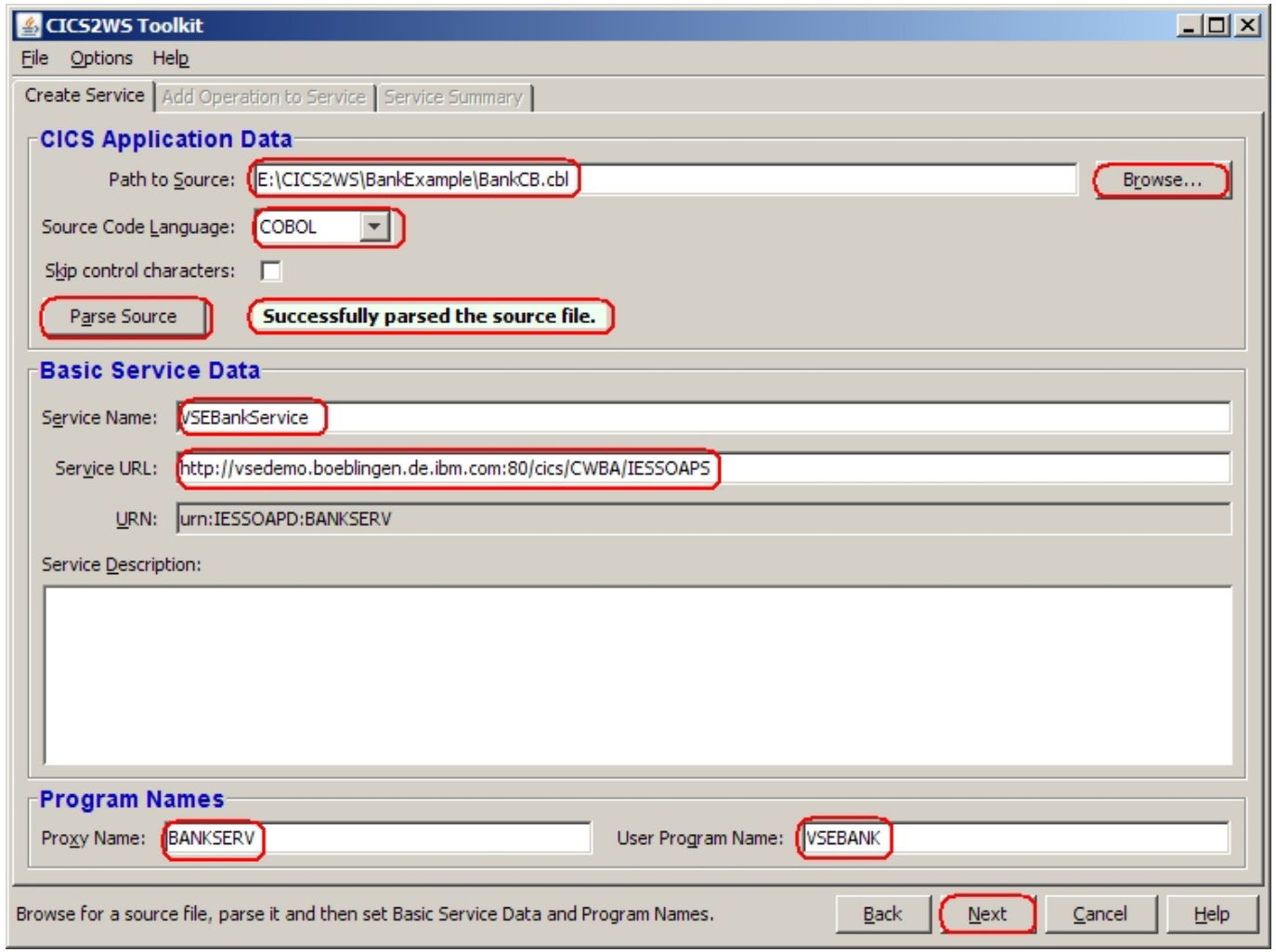
Live Demo – Use a Web Service

```
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.
```

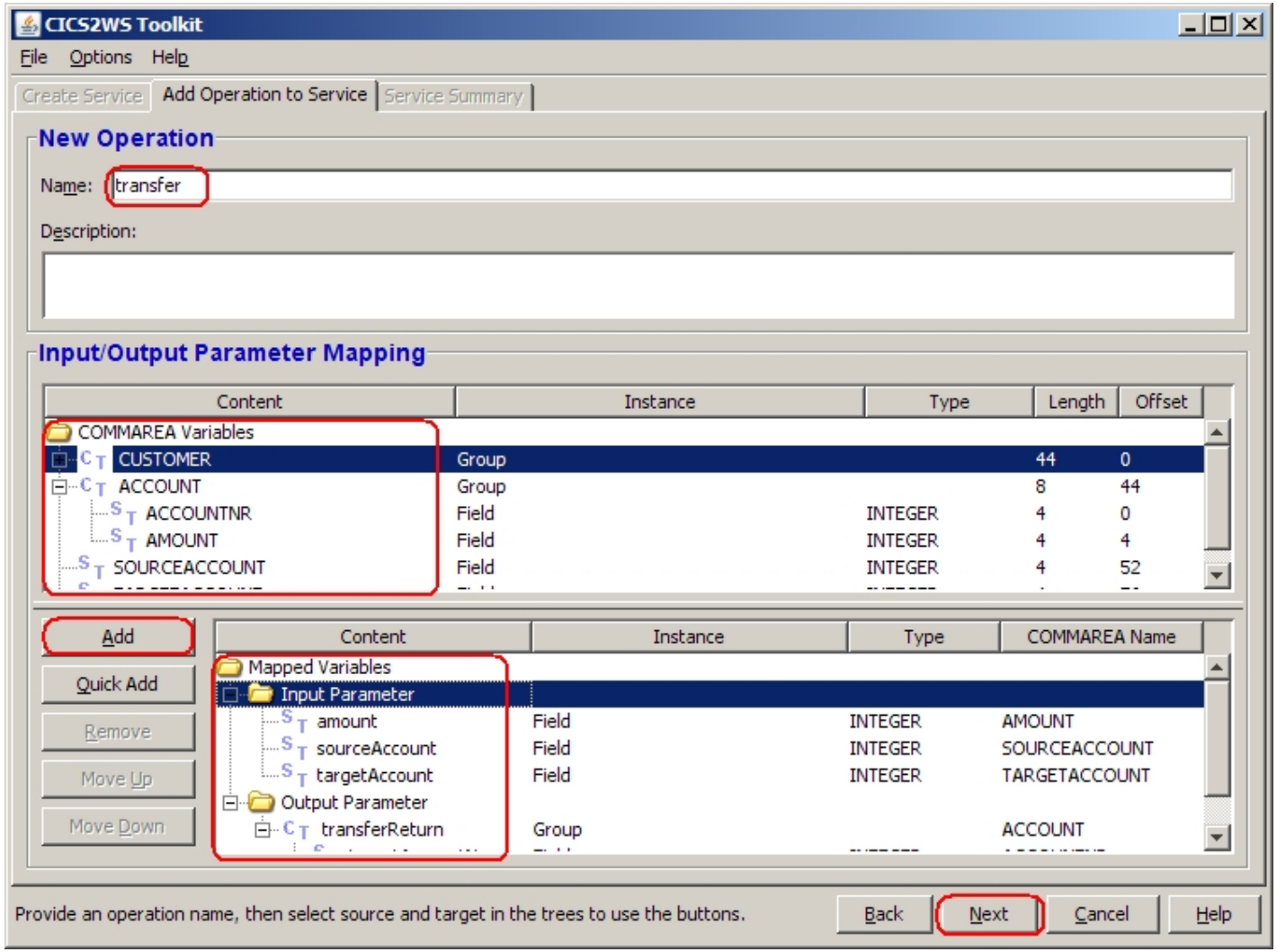
```
*****************************************
 SECTION
*****************************************

OF PARAMETER BLOCK
H OF BLOCK
NSE CODE
NSE CODE 2
) COMMAREA FOR SOAP CALL
UFFER 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

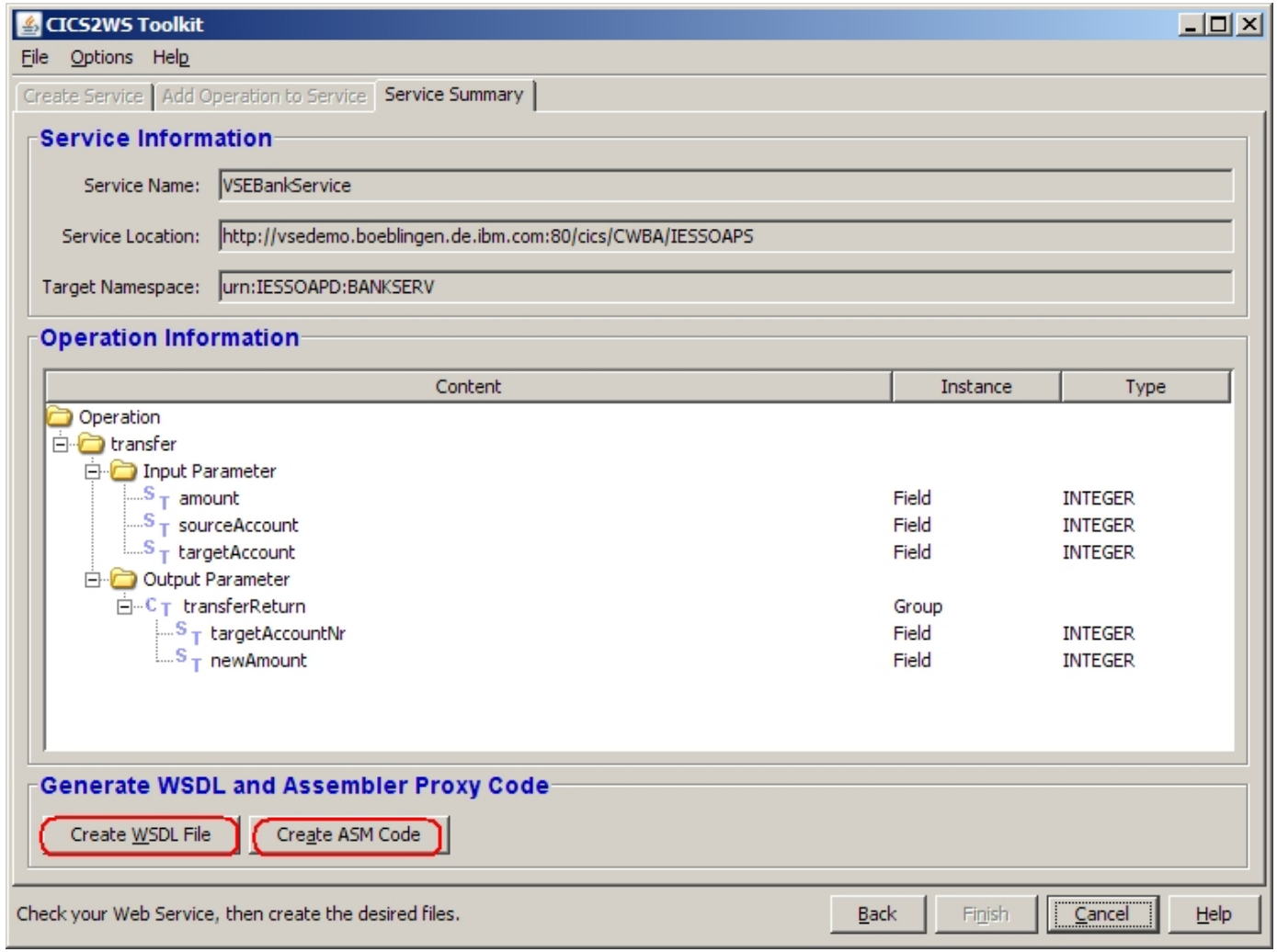
Live Demo – Create a Web Service



Live Demo – Create a Web Service



Live Demo – Create a Web Service



Live Demo – Create a Web Service

```
<?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>
```

```
**********************************************
AGE SECTION
**********************************************

OR OF PARAMETER BLOCK
NGTH OF BLOCK
SPONSE CODE
SPONSE CODE 2
FFER FOR OUTPARAMS


TART 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

- How to use Web Services with z/VSE
  - http://www.ibm.com/servers/eserver/zseries/zvse/documentation/ebusiness.html#soap
- Web Services in z/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 ?