



Simpler Calls to Web Services From IMS Apps

John Bachiochi
Product Specialist | SysperTec
john.bachiochi@syspertec.us

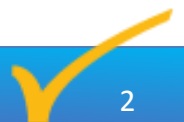
June 10, 2014

Patrick Fournier
Solution Architect | SysperTec
patrick.fournier@syspertec.us



Agenda - Calling Remote Services from IMS apps

- **Challenge:**
 - ✓ Insert unfamiliar web techno into legacy IMS application code
- **Solution:**
 - ✓ Virtel handles HTTP and XML/SOAP behind the scene
- **Implementation:**
 - ✓ Virtel installation & tailoring
 - ✓ Solution components generation (Virtel Studio IDE)
 - ✓ IMS program changes
- **Conclusion + Q/A**



The Challenge

Calling remote web services from IMS application programs is cumbersome: IMS COBOL programmers must deal with XML/SOAP data format and HTTP communication protocol that they are not familiar with.



Mainframe Applications No Longer Run in Isolation



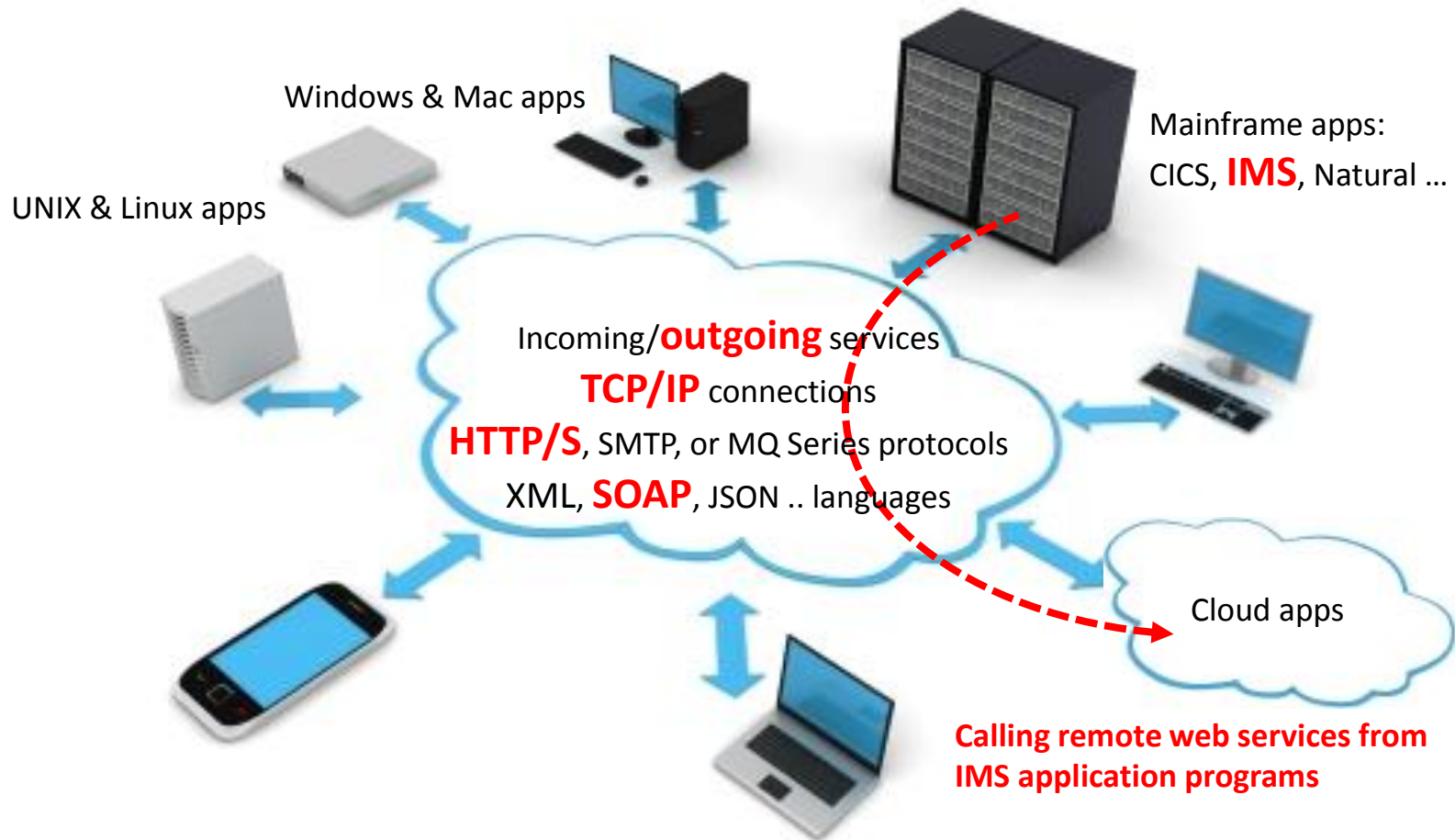
Modern Data Processing = Heterogeneous

Modern data processing combines heterogeneous application components that communicate through web connections



Modern Data Processing = Heterogeneous

Modern data processing combines heterogeneous application components that communicate through web connections



Calling a Web Service from an IMS Program

- **Application code must be changed to:**
 - ✓ Convert data between COBOL and XML/SOAP formats
 - ✓ Handle the HTTP/S protocol
- **Cumbersome + unfamiliar to IMS programmers**
 - ✓ Foreign (non traditional IMS COBOL) techno
 - ✓ Don't have skills/understanding
 - ✓ Reluctant to insert web techno into legacy code



The Solution

By handling XML/SOAP data format conversion and HTTP communication protocol behind the scene, Virtel IMS Extender provides a simple IMS COBOL solution to call remote web services from IMS application programs.



Virtel Web Suite

ONE WEB-ENABLEMENT TECHNO → MULTIPLE APPLICATIONS



Virtel Web Access – Browser-Based 3270 TE

- ✓ Serve 3270 screens to thin-client web browsers



Virtel Web Modernization – Web User Interface (WUI/GUI)

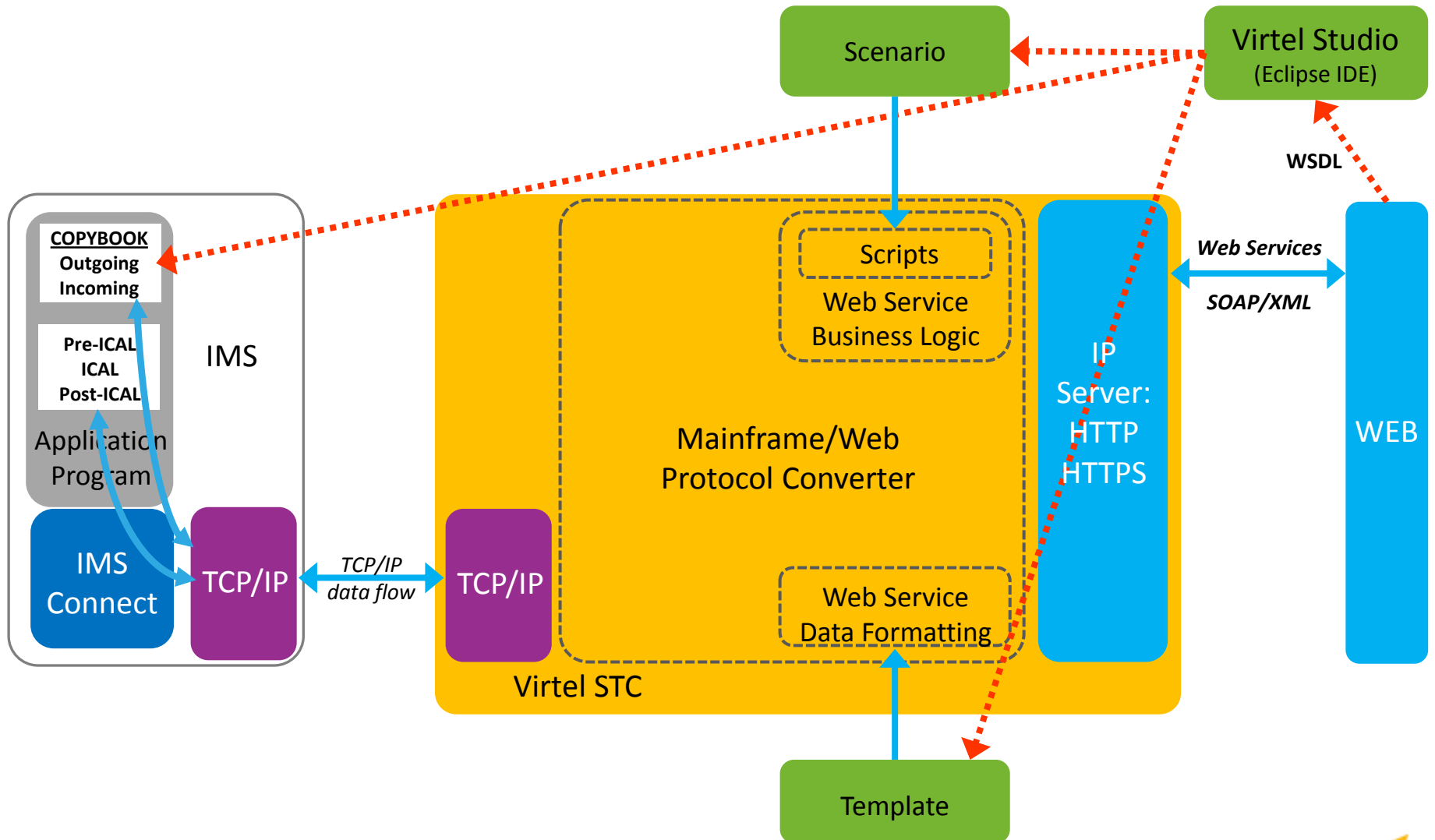
- ✓ Create rich user-friendly GUI for 3270 applications



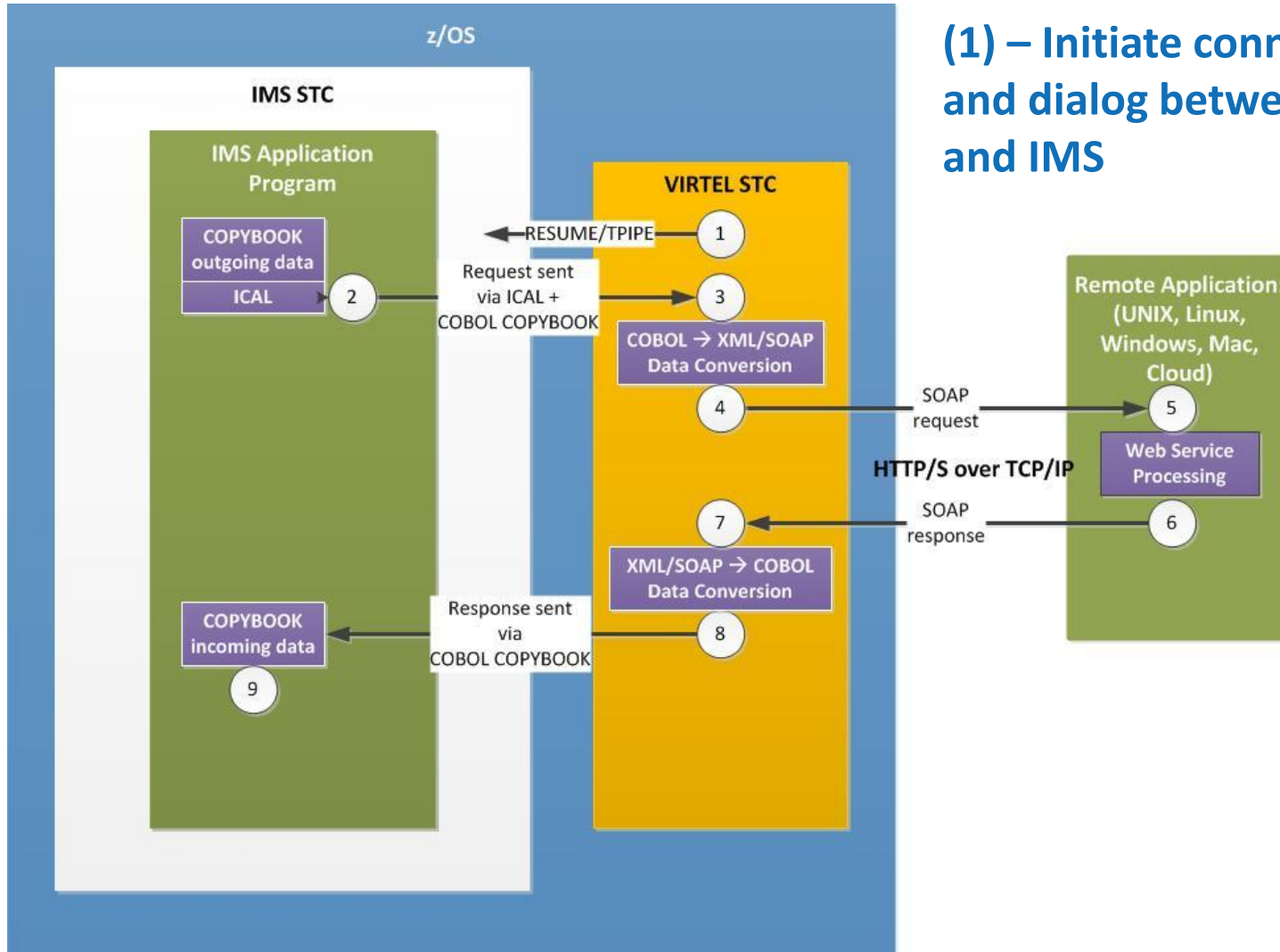
Virtel Web Integration – App-to-App Connector

- ✓ Create interactive bidirectional app-to-app connections (web services) between heterogeneous platforms over the Internet
→ e.g. calling remote web services from IMS applications

Virtel IMS Extender - Architecture

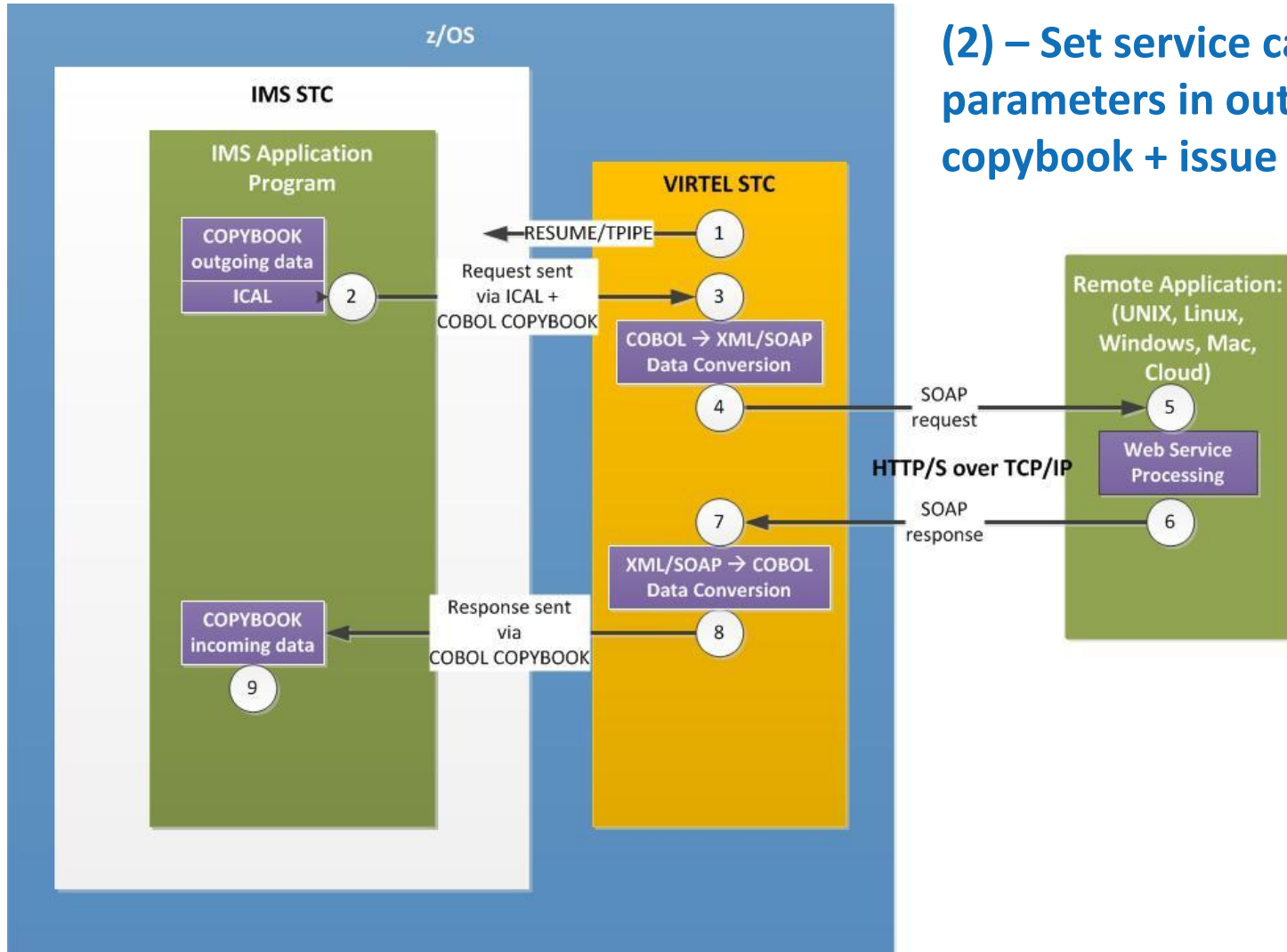


Virtel IMS Extender – Process - Step #1



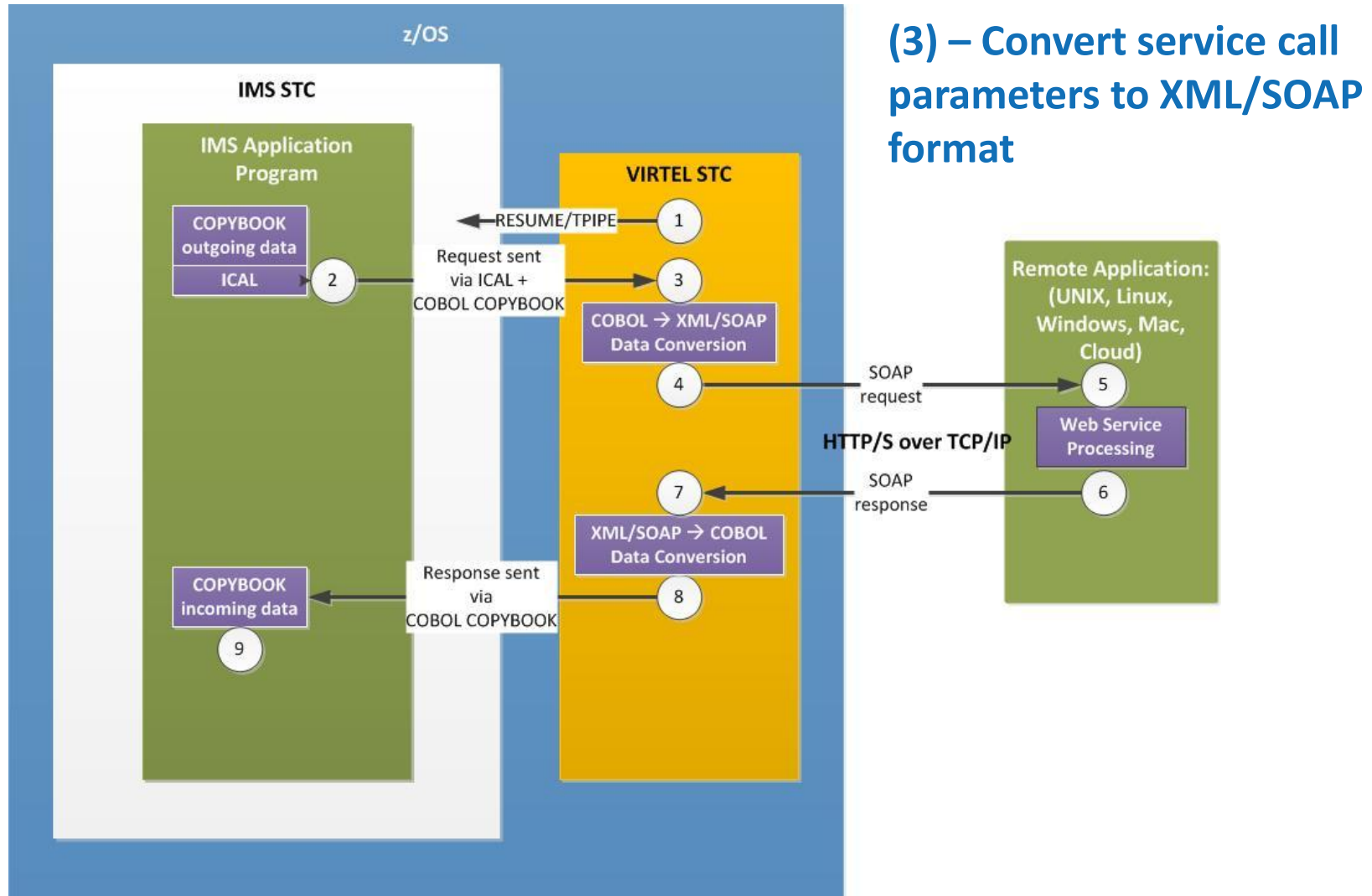
(1) – Initiate connection and dialog between Virtel and IMS

Virtel IMS Extender – Process - Step #2



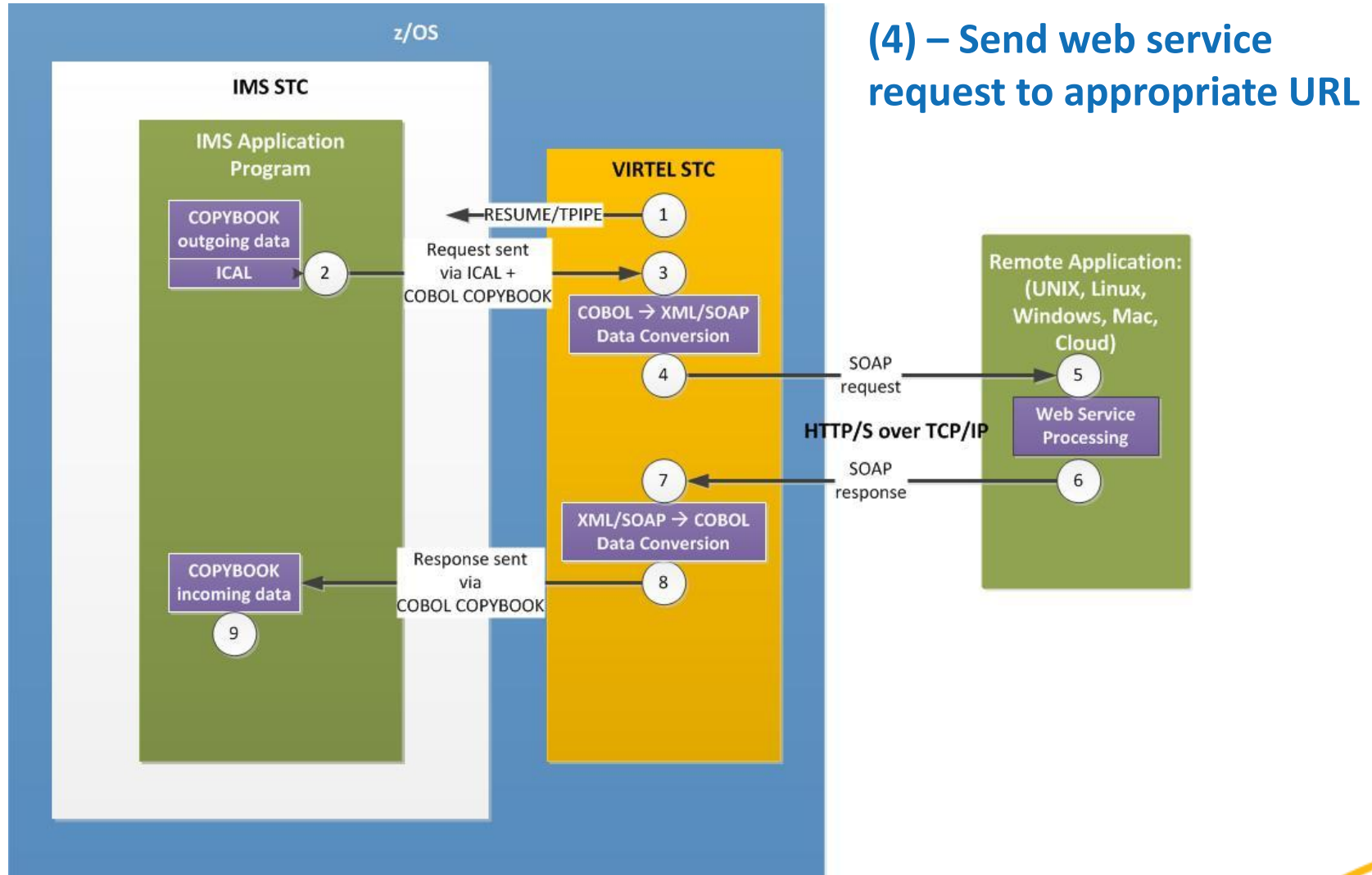
(2) – Set service call parameters in outgoing copybook + issue ICAL

Virtel IMS Extender – Process - Step #3

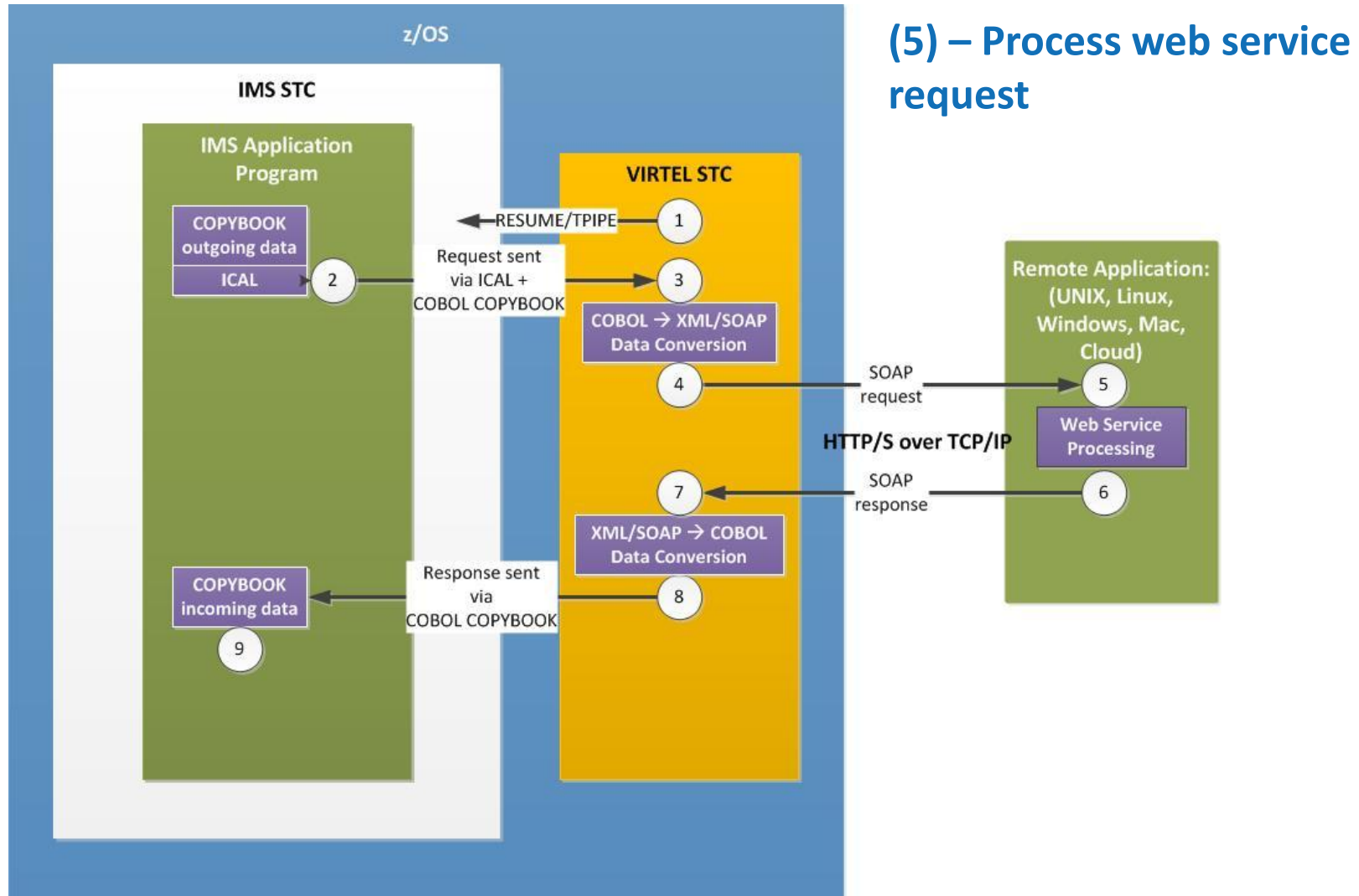


(3) – Convert service call parameters to XML/SOAP format

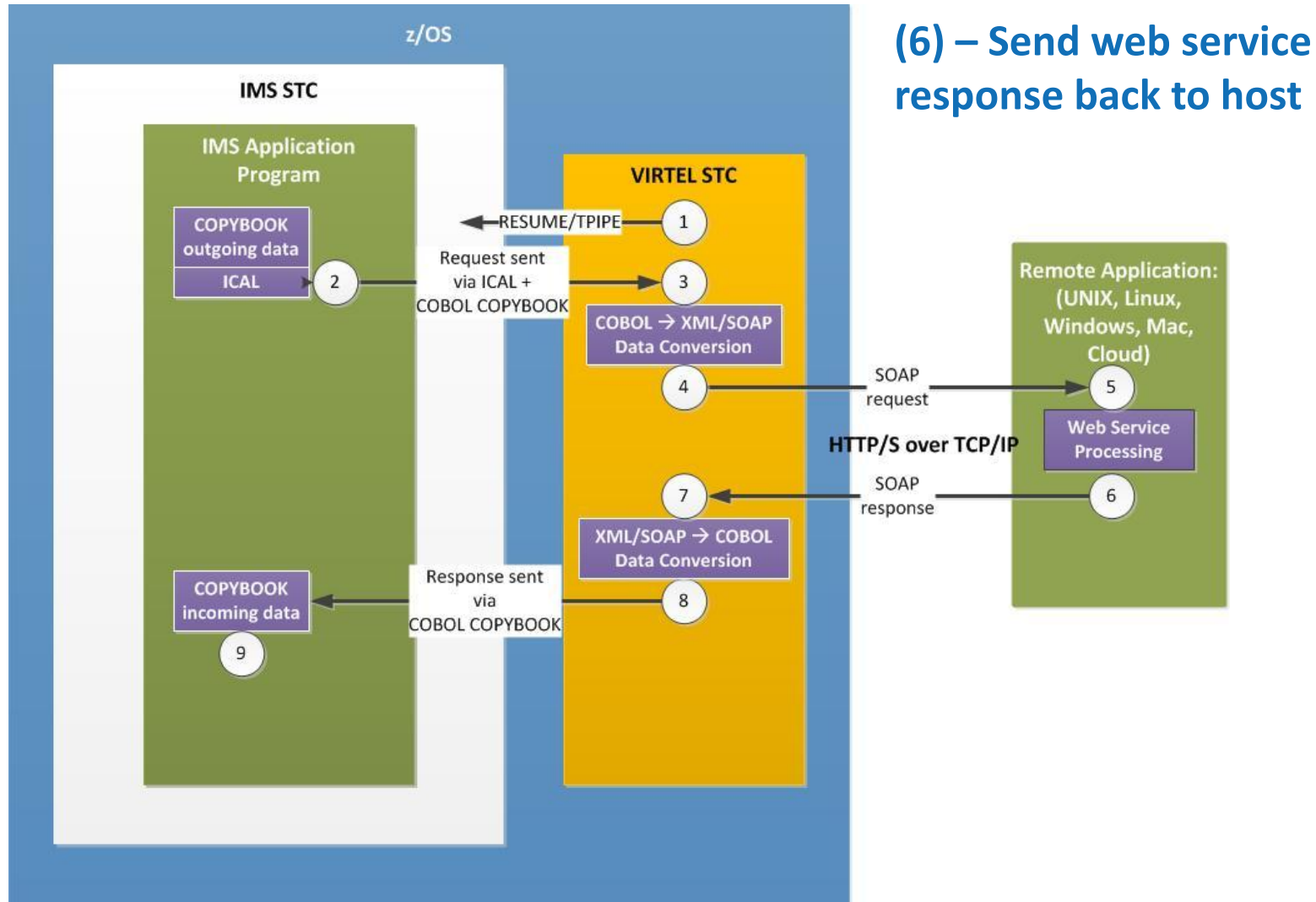
Virtel IMS Extender – Process - Step #4



Virtel IMS Extender – Process - Step #5

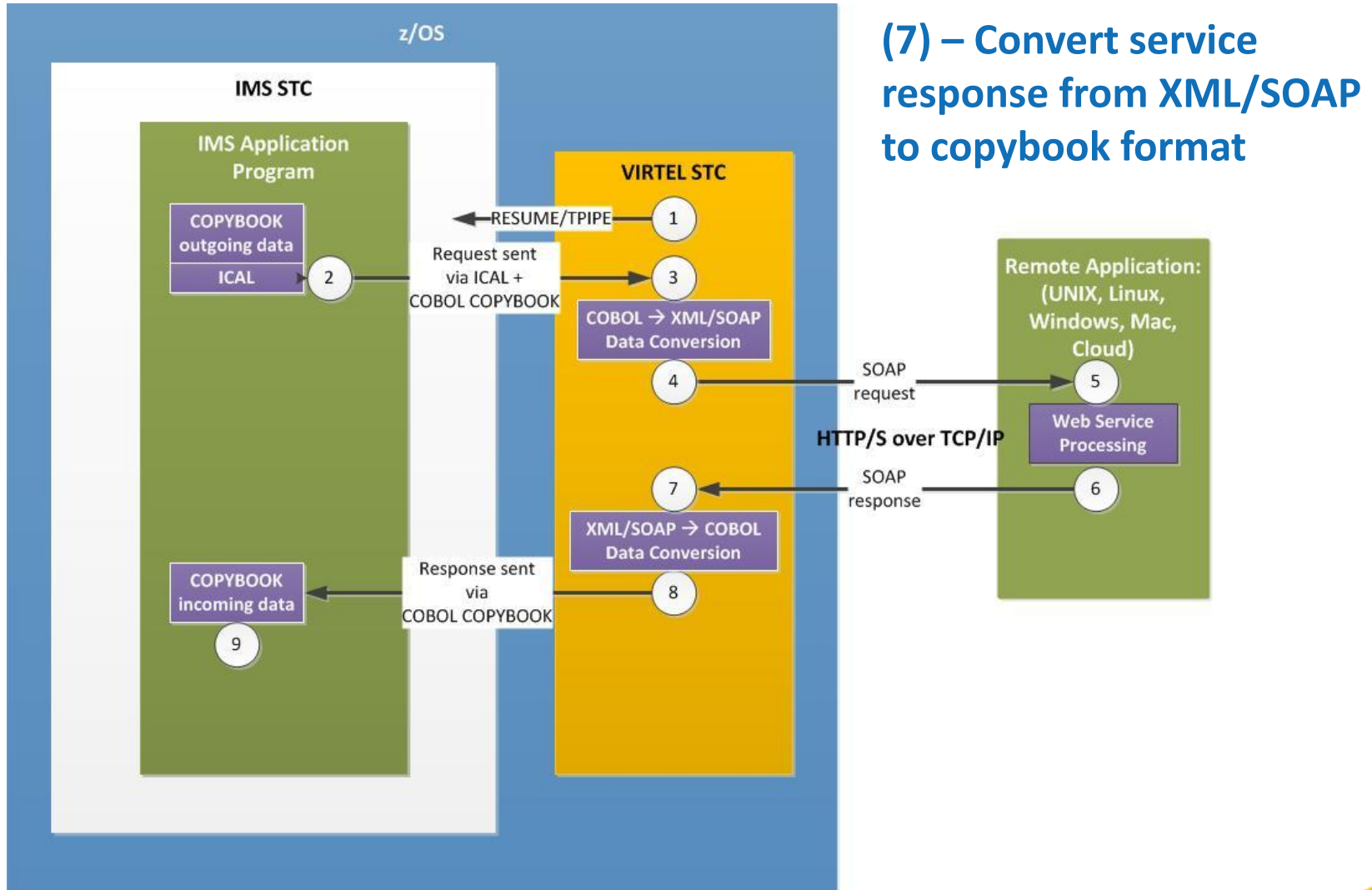


Virtel IMS Extender – Process - Step #6



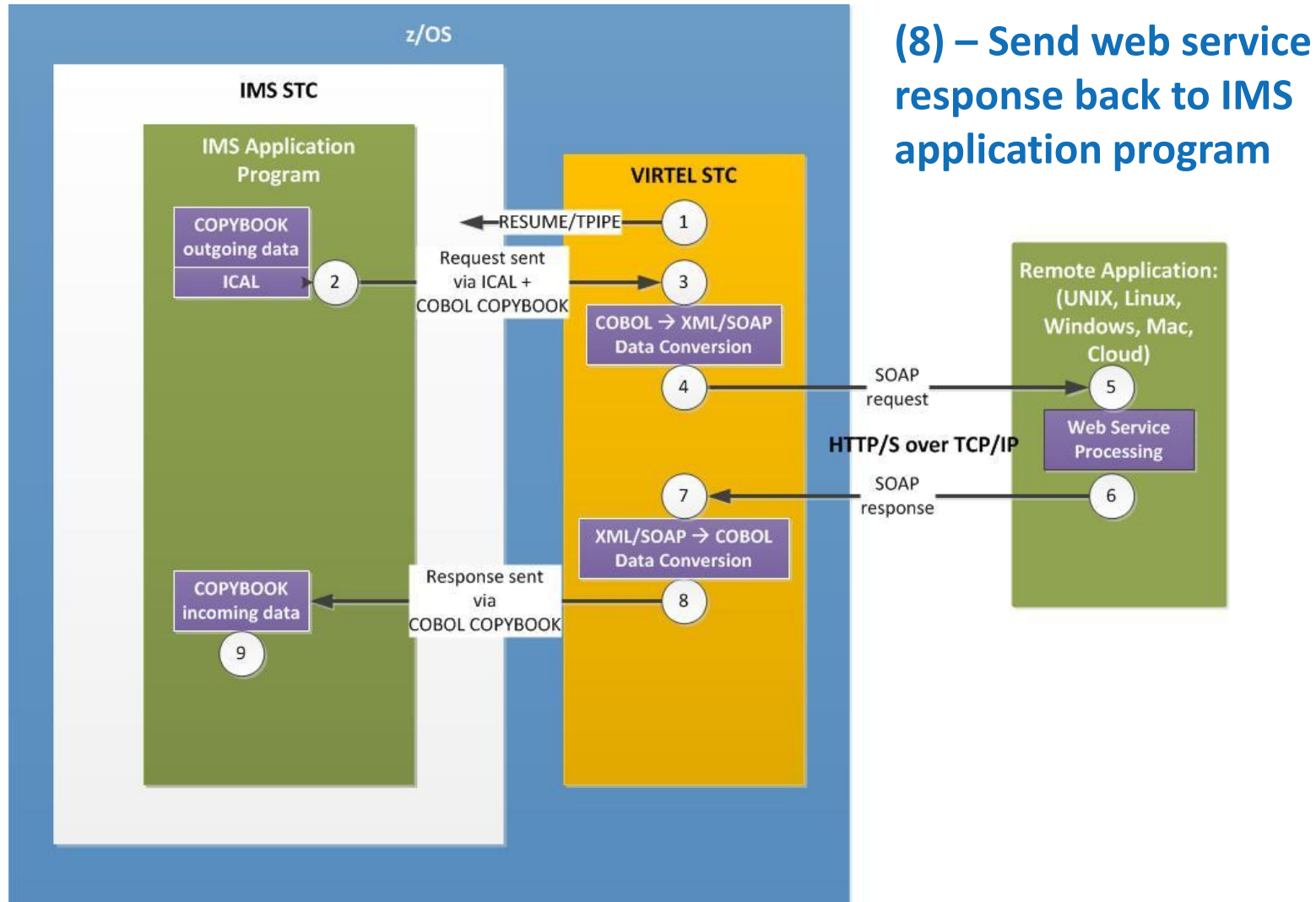
(6) – Send web service response back to host

Virtel IMS Extender – Process - Step #7



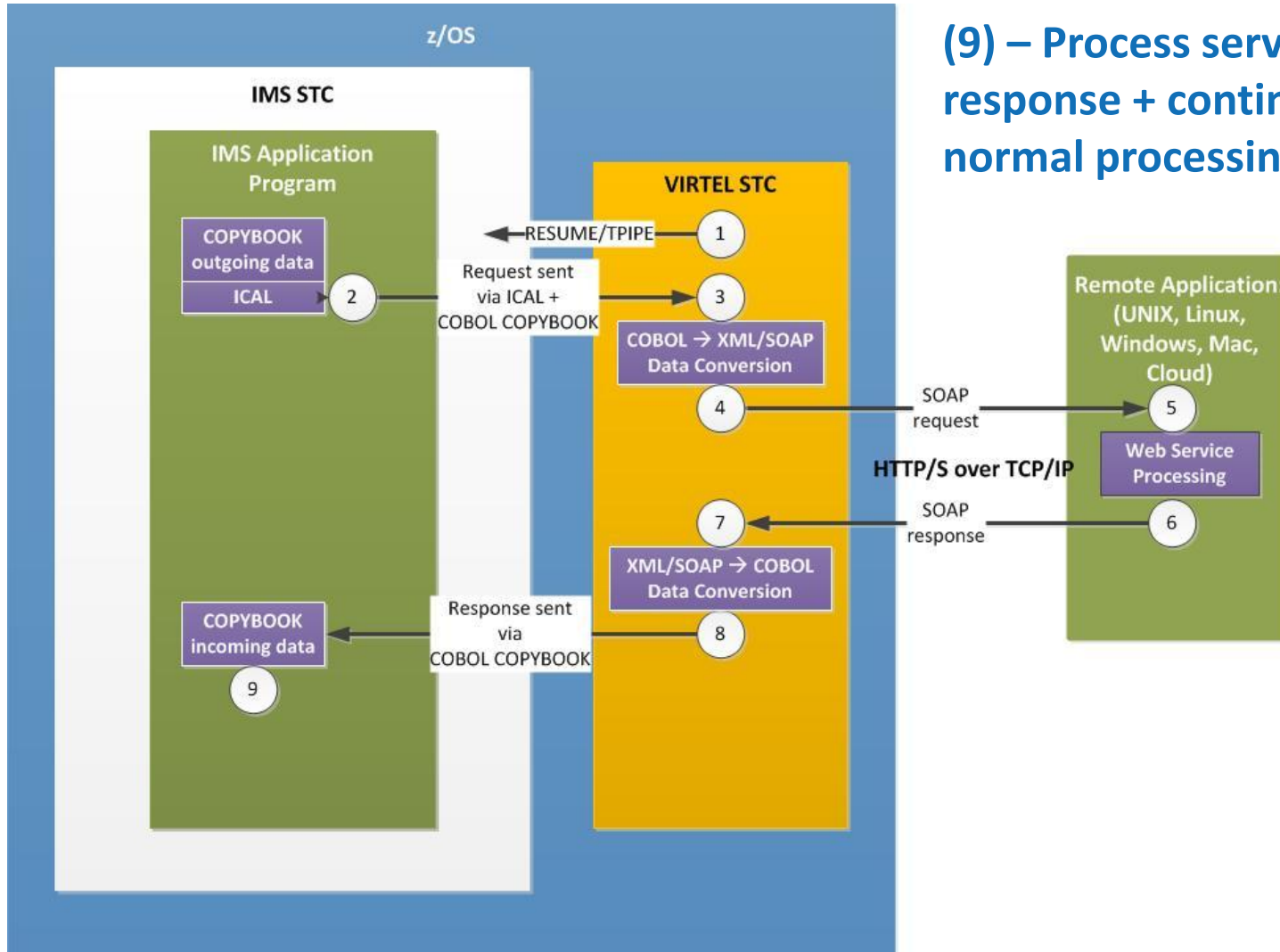
(7) – Convert service response from XML/SOAP to copybook format

Virtel IMS Extender – Process - Step #8



(8) – Send web service response back to IMS application program

Virtel IMS Extender – Process - Step #9



(9) – Process service response + continue with normal processing

Implementation

Virtel Installation & Tailoring

Virtel Studio (Eclipse IDE)

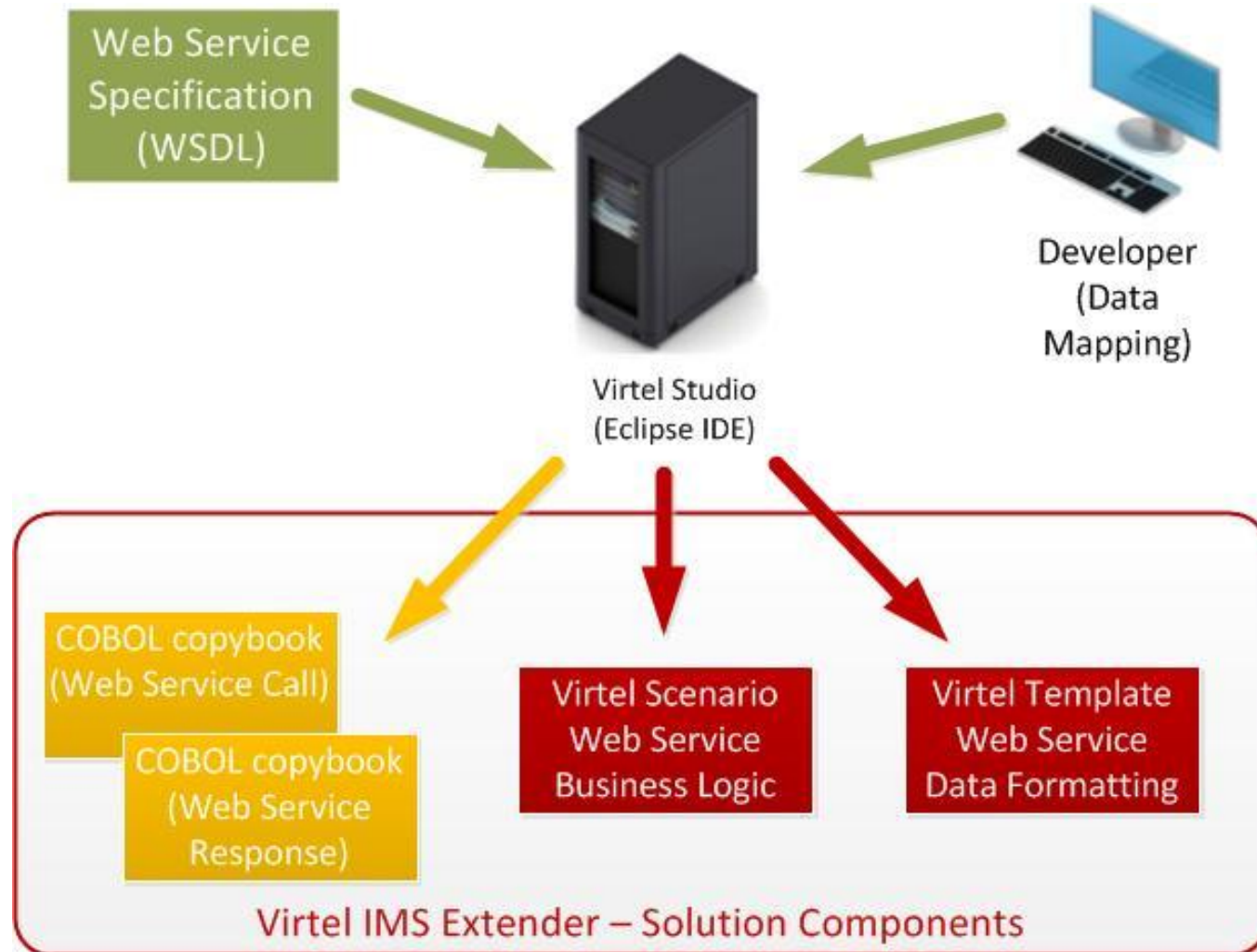
IMS Program Modifications



Virtel - Installation and Tailoring

- **Installation:**
 - ✓ Upload Virtel libraries to z/OS system
 - ✓ Simple installation process takes ~ 2 hours
- **Tailoring (web service path definition):**
 - ✓ Define lines between:
 - ▶ IMS and Virtel
 - ▶ Virtel and Web
 - ✓ Define URL to call remote web service
 - ✓ Define name correspondence between:
 - ▶ Web service ↔ scenario ↔ IMS transaction

Studio – Generate Solution Components



WSDL - Web Service Specifications

- Provided by web server support → input to Studio:

```
<?xml version='1.0' encoding='UTF-8'?><wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://wscompany.demos.syspertec.com/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:ns1="http://schemas.xmlsoap.org/soap/http"
name="CompanyService" targetNamespace="http://wscompany.demos.syspertec.com/">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://wscompany.demos.syspertec.com/" elementFormDefault="unqualified"
targetNamespace="http://wscompany.demos.syspertec.com/" version="1.0">
      <xs:element name="search" type="tns:search"/>
      <xs:element name="searchResponse" type="tns:searchResponse"/>
      <xs:complexType name="search">
        <xs:sequence>
          <xs:element minOccurs="0" name="prefix" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="searchResponse">
        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="return" type="tns:company"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="company">
        <xs:sequence>
          <xs:element minOccurs="0" name="city" type="xs:string"/>
          <xs:element minOccurs="0" name="country" type="xs:string"/>
          <xs:element minOccurs="0" name="name" type="xs:string"/>
          <xs:element minOccurs="0" name="price" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="search">
    <wsdl:part element="tns:search" name="parameters">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="searchResponse">
    <wsdl:part element="tns:searchResponse" name="parameters">
    </wsdl:part>
  </wsdl:message>
  ... etc ...
```

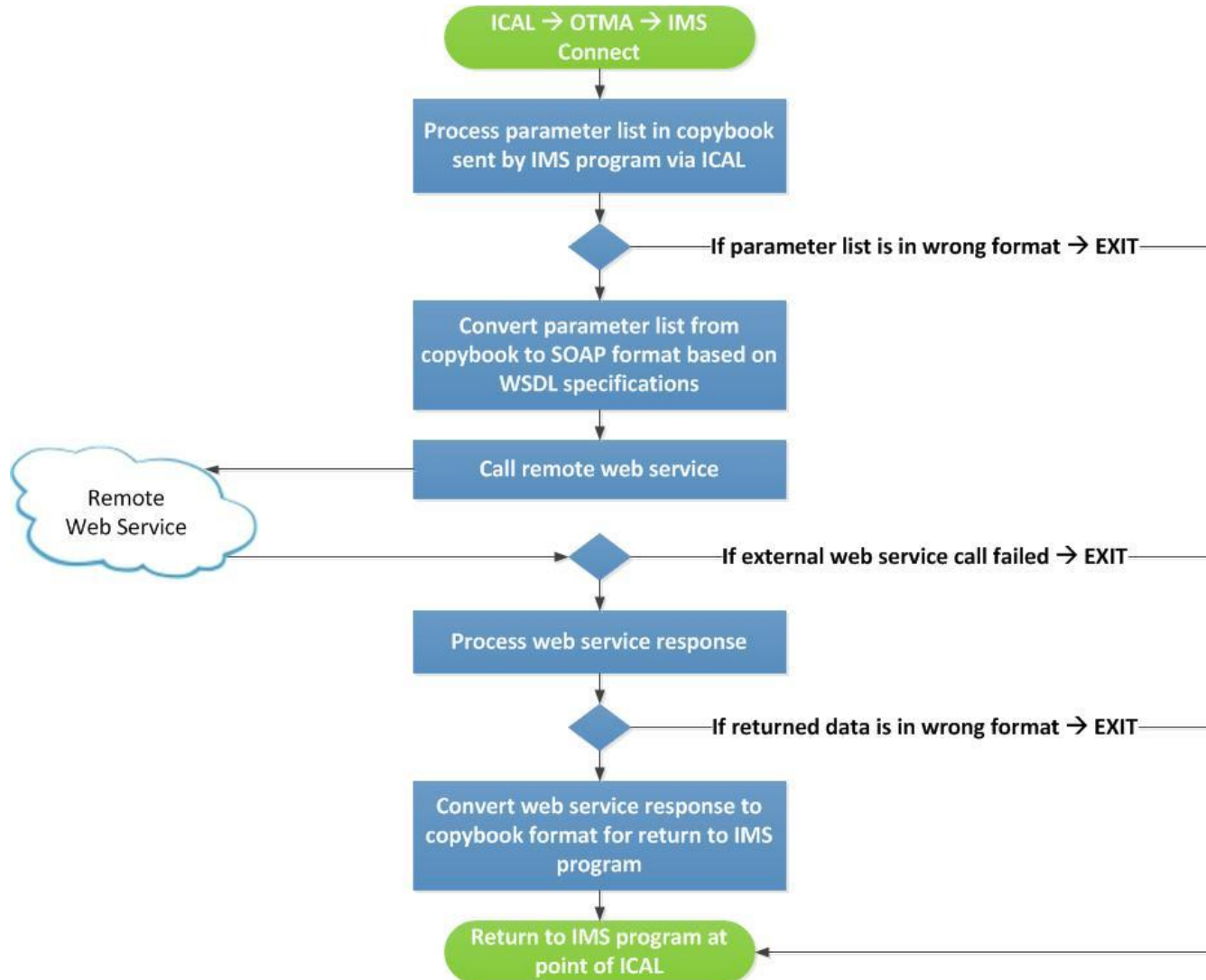

Virtel Scenario – Web Service “Business Logic”

● Generated by Studio → input to Virtel at runtime

```
COMPSEAR SCREENS APPL=COMPSEAR,EXEC=NO
*
  SCENARIO INITIAL
*
* Input structure definition
*
INPUT  MAP$ BEGIN,WITH='CsearchINPUT'
      MAP$ AREA,WITH='VIRTEL-VERSION',LENGTH=4,TYPE=X
      MAP$ AREA,WITH='VIRTEL-TRANSACTION',LENGTH=8,TYPE=X
      MAP$ AREA,WITH='VIRTEL-USERDATA',LENGTH=32,TYPE=X
      MAP$ AREA,WITH='VIRTEL-CHECKSUM',LENGTH=16,TYPE=X
      MAP$ AREA,WITH='PREFIX',TYPE=X,LENGTH=32
INPUT  MAP$ END
*
* Output structure definition
*
OUTPUT MAP$ BEGIN
OVERS  MAP$ AREA,FROM-CONSTANT,'/V1/',LENGTH=4
OCODE  MAP$ AREA,FROM-VARIABLE,VAR='STATUS',LENGTH=8
OMSG   MAP$ AREA,FROM-VARIABLE,VAR='ERRMSG',LENGTH=128
OVIRT  MAP$ AREA,FROM-VARIABLE,VAR='APPLID',LENGTH=8,TYPE=X
OSCEN  MAP$ AREA,FROM-VARIABLE,VAR='SCENARIO',LENGTH=8,TYPE=X
OSITE  MAP$ AREA,FROM-VARIABLE,VAR='SITE',LENGTH=32,TYPE=X
OSNURI MAP$ AREA,FROM-VARIABLE,VAR='SERVICE-NAME-URI',LENGTH=64, *
      TYPE=X
OSNLOC MAP$ AREA,FROM-VARIABLE,VAR='SERVICE-NAME-LOCAL',LENGTH=64, *
      TYPE=X
OOPER  MAP$ AREA,FROM-VARIABLE,VAR='OPERATION',LENGTH=64,TYPE=X
      MAP$ EVENTUAL-AREA,FROM-CONSTANT,'0',WHEN=(ELEMENT, *
... etc ...
```

```
... etc ...
      'ns1:searchResponse'),EVENT='Response',LENGTH=1
      MAP$ ELSETHEN-AREA,FROM-CONSTANT,'1',EVENT='Fault',LENGTH=1
      MAP$ AREA,FROM-VARIABLE,VAR='VIRTEL-USERDATA',LENGTH=32,TYPE=X
OENV   MAP$ BEGIN,WITH='soap:Envelope'
OHEADER MAP$ BEGIN,WITH='soap:Header'
OHEADER MAP$ END
OBODY  MAP$ BEGIN,WITH='soap:Body'
OFAULT MAP$ BEGIN,EVENT='Fault'
      MAP$ AREA,WITH='faultcode',TYPE=X,LENGTH=32
      MAP$ AREA,WITH='faultstring',TYPE=X,LENGTH=32
      MAP$ AREA,WITH='faultactor',TYPE=X,LENGTH=32
      MAP$ AREA,FROM-CONSTANT,'',LENGTH=32
OFAULT MAP$ END
O0000001 MAP$ BEGIN,EVENT='Response'
O0000002 MAP$ BEGIN,WITH='return',OCCURS=10
      MAP$ AREA,WITH='city',TYPE=X,LENGTH=32
      MAP$ AREA,WITH='country',TYPE=X,LENGTH=32
      MAP$ AREA,WITH='name',TYPE=X,LENGTH=32
      MAP$ AREA,WITH='price',TYPE=X,LENGTH=32
O0000002 MAP$ END
O0000001 MAP$ END
OBODY  MAP$ END
OENV   MAP$ END
OUTPUT MAP$ END
*
START  SET$ ENCODING,UTF-8,'IBM1147'
INPUT  TOVAR$ FROM-VARIABLE,VAR='$INFILES$'
      CASE$ 'VIRTEL-CHECKSUM',(EQ, *
      X'2EC8587FCEC5F0CC3E94F8D49B245441',PROSITE)
... etc ...
```

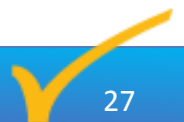
Virtel Scenario – Web Service “Business Logic”



Virtel Template – Web Service Data Formatting

- Generated by Studio → input to Virtel at runtime

```
<?xml version="1.0" encoding="UTF-8"?>
<!--VIRTEL start="{{" end="}" -->{{ SET-CONTENT-TYPE "text/xml"}}{{SET-OUTPUT-ENCODING-UTF-8 ""}}{{SET-LOCAL-OPTIONS (XML-ESCAPES)}}
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://wscompany.demos.syspertec.com/" >
  <soapenv:Header/>
  <soapenv:Body>
    <ns1:search>
      <prefix>{{TRIMMED-VALUE-OF "PREFIX"}}</prefix>
    </ns1:search>
  </soapenv:Body>
</soapenv:Envelope>
```



COBOL Copybooks – Data Structure Definitions

* Structure for operation

* Generated on Wed May 28 08:45:37 PDT 2014

```
01 CsearchINPUT.  
02 VIRT-VERSION PIC X(4) VALUE "/V1/".  
02 VIRT-TRANSACTION PIC X(8).  
02 VIRT-USERDATA PIC X(32).  
02 VIRT-CHECKSUM PIC X(16)  
VALUE X"2EC8587FCEC5F0CC3E94F8D49B245441".  
02 PREFIX PIC X(32).
```

* Structure for operation

* Generated on Wed May 28 08:45:37 PDT 2014

```
01 CsearchOUTPUT.  
02 VIRT-VERSION PIC X(4) VALUE "/V1/".  
02 VIRT-CODE PIC 9(8).  
02 VIRT-ERR-MSG PIC X(128).  
02 VIRT-APPLID PIC X(8).  
02 VIRT-SCENARIO PIC X(8).  
02 VIRT-SITE PIC X(32).  
02 VIRT-SERVICE-NAME-URI PIC X(64).  
02 VIRT-SERVICE-NAME-LOCAL PIC X(64).  
02 VIRT-OPERATION PIC X(64).  
02 VIRT-FAULT PIC 9.  
02 VIRT-USERDATA PIC X(32).  
* if VIRT-FAULT=1  
02 VIRT-FAULT-DATA.  
03 VIRT-FAULT-CODE PIC X(32).  
03 VIRT-FAULT-STRING PIC X(32).  
03 VIRT-FAULT-ACTOR PIC X(32).  
03 VIRT-FAULT-TYPE PIC X(32).  
03 VIRT-FAULT-DETAIL PIC X.  
* if VIRT-FAULT=0  
02 RESPONSE REDEFINES VIRT-FAULT-DATA.  
03 RETURN-RENAMED OCCURS 10.  
04 RETURN-CITY PIC X(32).  
04 RETURN-COUNTRY PIC X(32).  
04 RETURN-NAME PIC X(32).  
04 RETURN-PRICE PIC X(32).
```



Call Web Service from IMS COBOL Program

... etc ...

```
*****  
* Call external application via ICAL  
*****
```

```
3000-SYNC-ICAL.
```

```
IF RE-ISSUE = 1
```

```
MOVE 0 TO RE-ISSUE
```

```
GO TO 3000-SYNC-ICAL-EXIT
```

```
END-IF
```

```
DISPLAY 'SYCALOUT:SYNC-ICAL'
```

```
* Send ICAL
```

```
MOVE 'COMPSEAR' TO VIRT-TRANSACTION
```

```
MOVE MID-REQUEST TO PREFIX
```

```
COMPUTE AIBOALEN = LENGTH OF CsearchINPUT
```

```
COMPUTE AIBRLEN = LENGTH OF AIB
```

```
COMPUTE AIBOAUSE = LENGTH OF CsearchOUTPUT
```

```
DISPLAY 'SYCALOUT:CALLOUT REQUEST (' CsearchINPUT ')'
```

```
CALL 'AIBTDLI' USING ICAL AIB
```

```
      CsearchINPUT CsearchOUTPUT
```

```
IF AIBRETRN = ZEROS
```

```
DISPLAY 'SYCALOUT:CALLOUT RESPONSE ('  
      CsearchOUTPUT ')'
```

```
ELSE IF AIBRETRN = 256 AND AIBREASN = 260
```

```
DISPLAY 'SYCALOUT:ICAL FAILED - REQUEST TIMEOUT'
```

```
ELSE IF AIBRETRN = 256 AND AIBREASN = 12
```

```
DISPLAY 'SYCALOUT:PARTIAL RESPONSE DATA RETURNED'
```

```
DISPLAY 'SYCALOUT:CALLOUT RESPONSE(' CsearchOUTPUT ')'
```

```
END-IF
```

```
MOVE AIBRETRN TO AIBRETRN-DISP
```

```
MOVE AIBREASN TO AIBREASN-DISP
```

```
MOVE AIBERRXT TO AIBERRXT-DISP
```

```
MOVE AIBRETRN-DISP TO MOD-AIBRETRN
```

```
MOVE AIBREASN-DISP TO MOD-AIBREASN
```

```
MOVE AIBERRXT-DISP TO MOD-AIBERRXT
```

```
.
```

```
3000-SYNC-ICAL-EXIT.
```

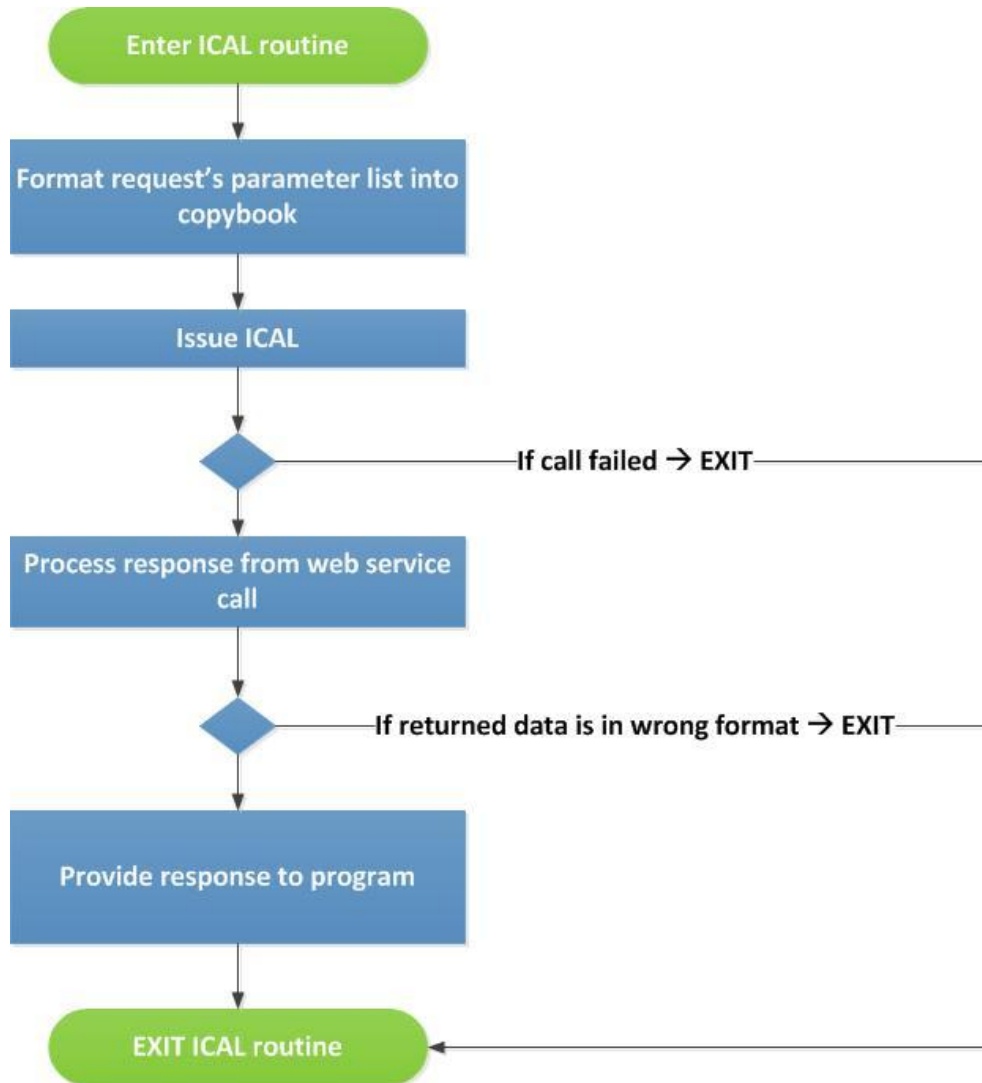
```
EXIT.
```

```
4000-ICAL-MOD.
```

... etc ...



Call Web Service from IMS COBOL Program



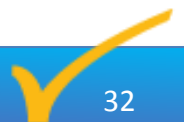
Conclusion

Virtel IMS Extender – Solution Highlights



Virtel IMS Extender – Solution Highlights

- Using WSDL + data mapping → Studio generates:
 - ✓ COBOL copybooks for insertion into IMS program
 - ✓ Virtel scenario → web service logic
 - ✓ Virtel template → data formatting
- Simple IMS program modification:
 - ✓ Insert COBOL copybooks
 - ✓ Insert ICAL + pre and post ICAL processing
 - ✓ No reference to XML/SOAP or HTTP/S protocol
- Virtel handles behind the scene:
 - ✓ Data format conversion between COBOL and XML/SOAP
 - ✓ HTTP/S over TCP/IP protocol



Thank You

Patrick Fournier | patrick.fournier@sypertec.us

John Bachiochi | john.bachiochi@sypertec.us

+1 (925) 954-9649

