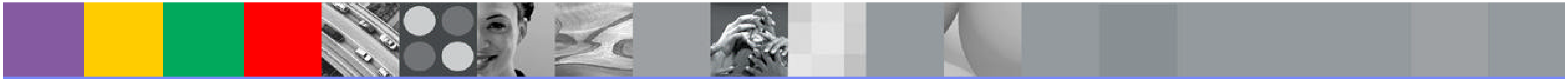
IBM Software Group

SOAP for CICS Feature

CICS Transaction Server for z/OS V2



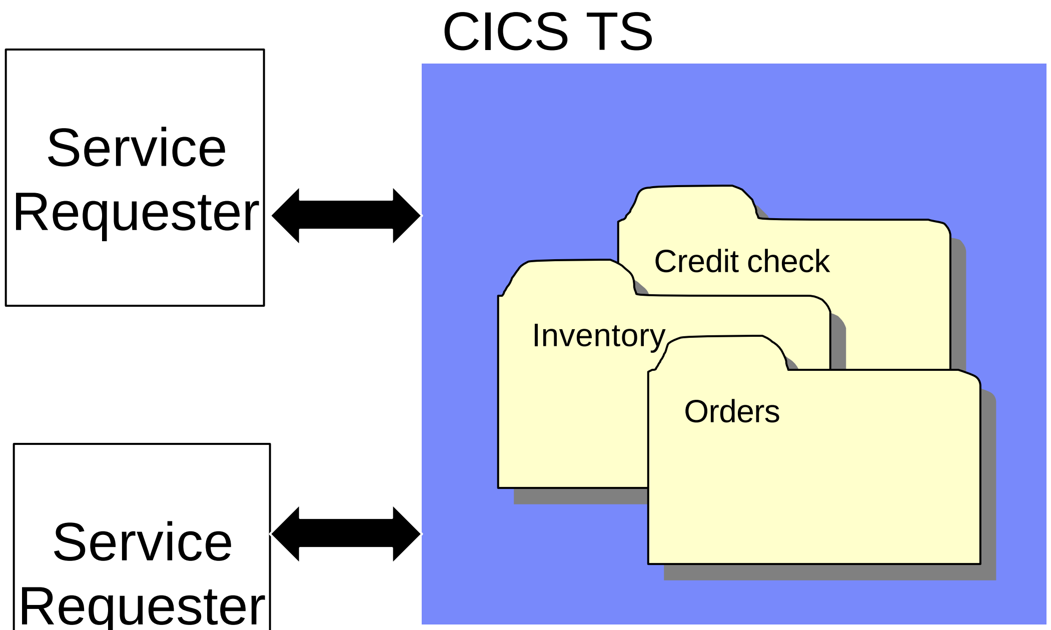
February 2004

Agenda

- **Business needs**
- **Components**
- **Development**
- **Setup**
- **Conclusion**

CICS and Enterprise e-business

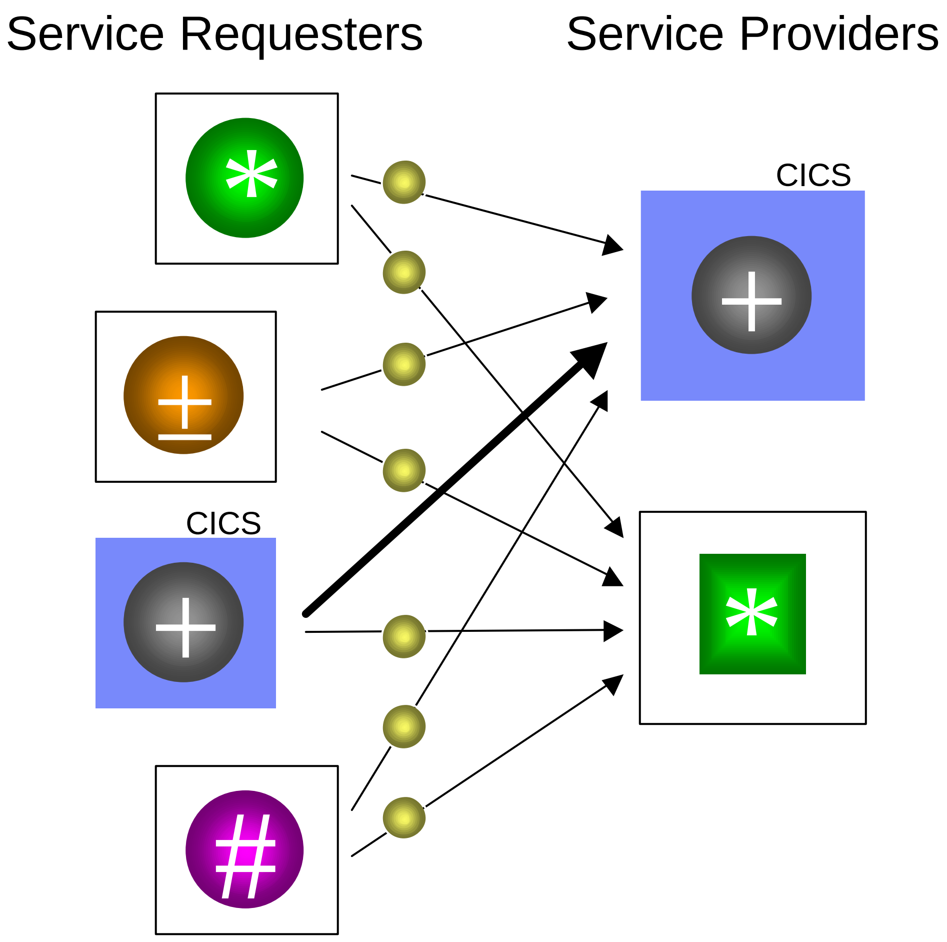
- Enterprise level e-business involves one computer system requesting a service from another
- This service is either a request for information, such as an account balance or it is a request to process information, such as a financial transaction
- Many leading enterprises have chosen IBM Mainframes running CICS software to provide these services
- The proven strengths of CICS running on IBM mainframes has, over time, led to a huge investment in CICS application code and skills
- To be agile in today's fast-moving marketplace companies must extend existing application that embrace new technologies



✓Performance
✓Scalability
✓Reliability
✓Systems management
✓Security
✓Proven reputation

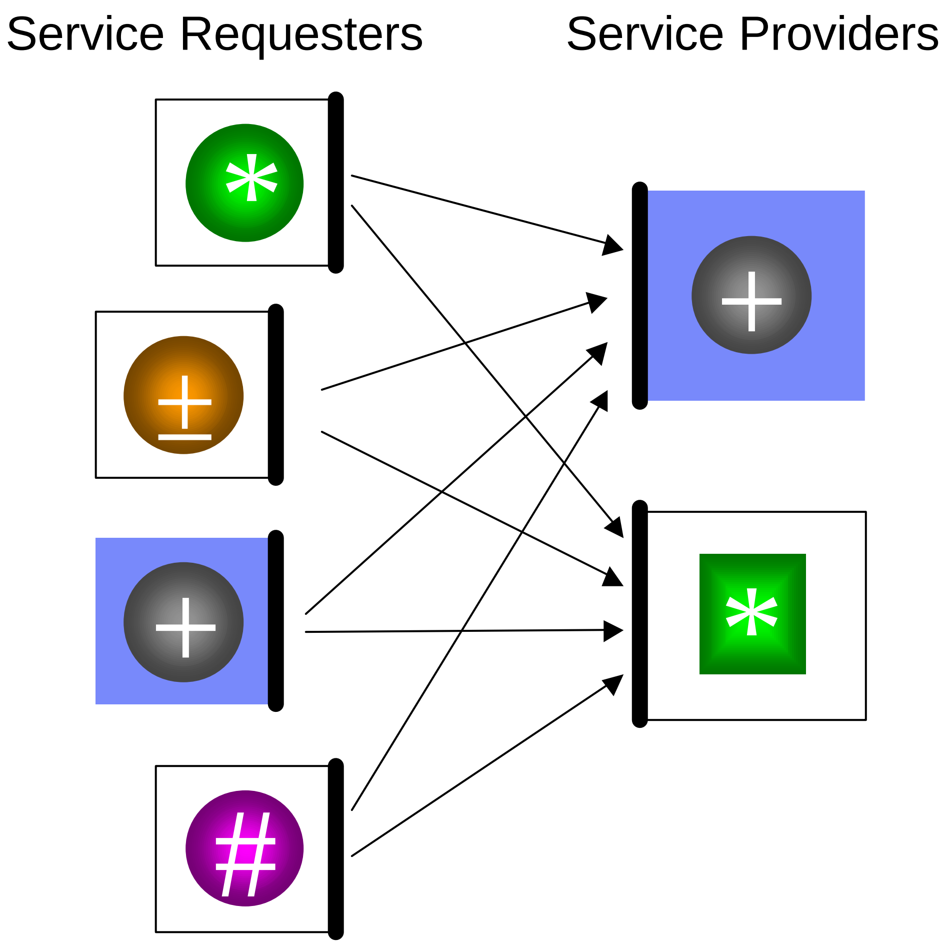
The problems with complex environments

- **Companies often need to implement business services**
  - On different hardware platforms
  - In different operating environments
  - In different programming languages
  - A multiple of interfaces, data types, programming models
- **Services can not all naturally communicate**
- **Proprietary and inflexible methods of integration can solve these communication problems in the short term**
  - Layers of different connectors, adapters, brokers and integration software can soon become difficult to implement and expensive to service and maintain
  - Complexity inhibits responsiveness to new market opportunities
- **There is a need for a simpler solution**



A simple SOAP solution

- **SOAP (Simple Object Access Protocol) is an industry standard designed to solve this problem**
- **SOAP for CICS is now available as a fully supported, no charge, orderable feature of CICS TS V2**
- **SOAP for CICS enables new and existing mission critical CICS applications to interact securely with Web services independently of platform, environment, or programming language**
- **It allows CICS to be a service requester as well as a service provider**



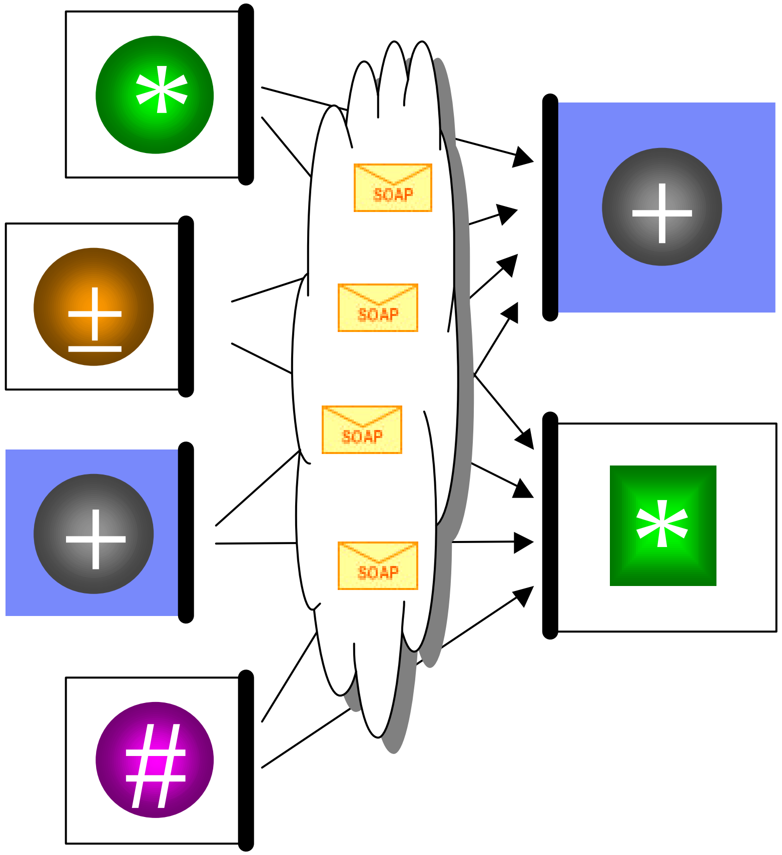
A simple SOAP solution

- Web service developers can now rapidly build open standards based two or three tier applications, independently of the CICS services that they will interact with

- SOAP for CICS enables enterprises to react quickly to new opportunities, whilst reducing the cost and resource associated with business integration

Service Requesters

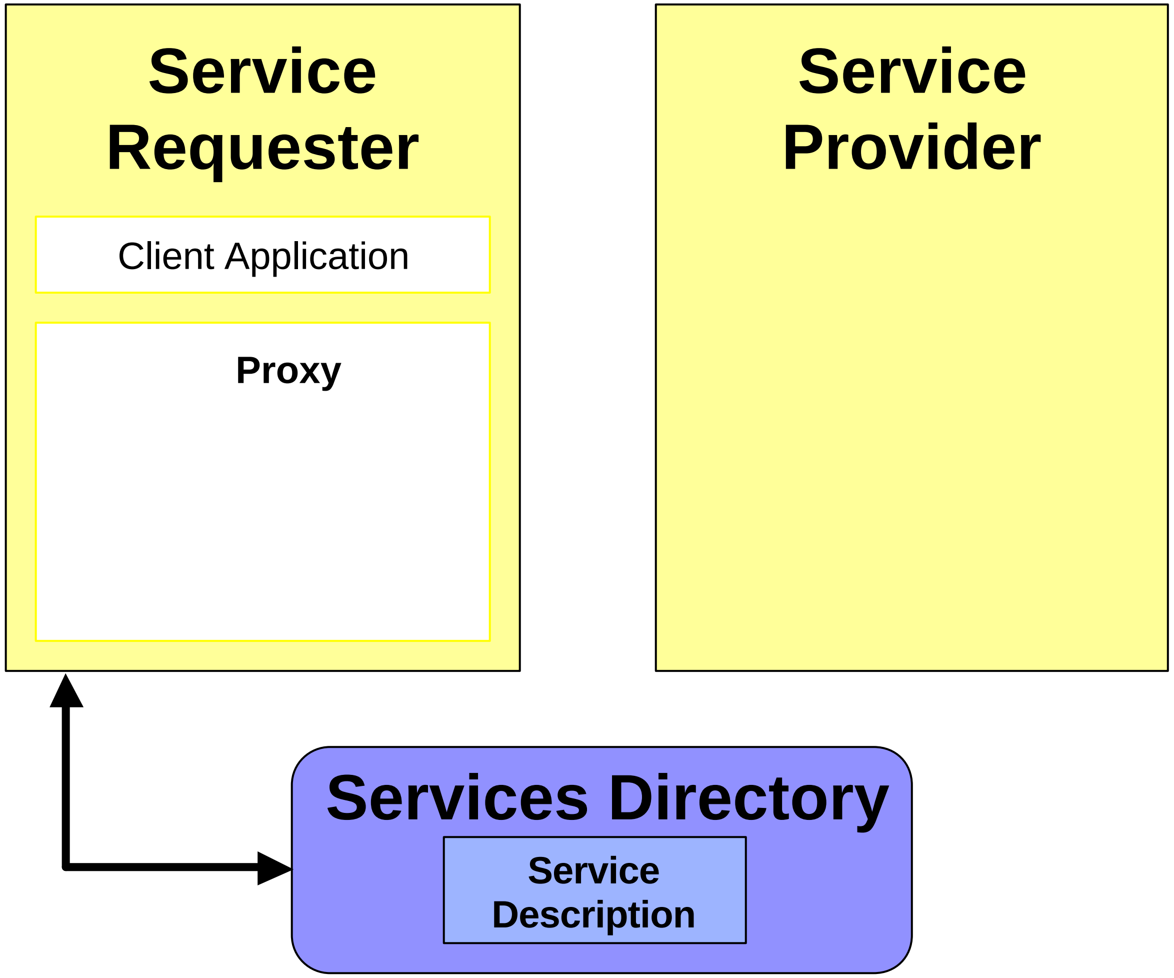
Service Providers



Components - principles of SOAP

- **The service requester uses the service description, in the services directory, to find the location and the description of the service provider**

- **The client application in the service requester should not handle the SOAP messages directly, but instead use a proxy**

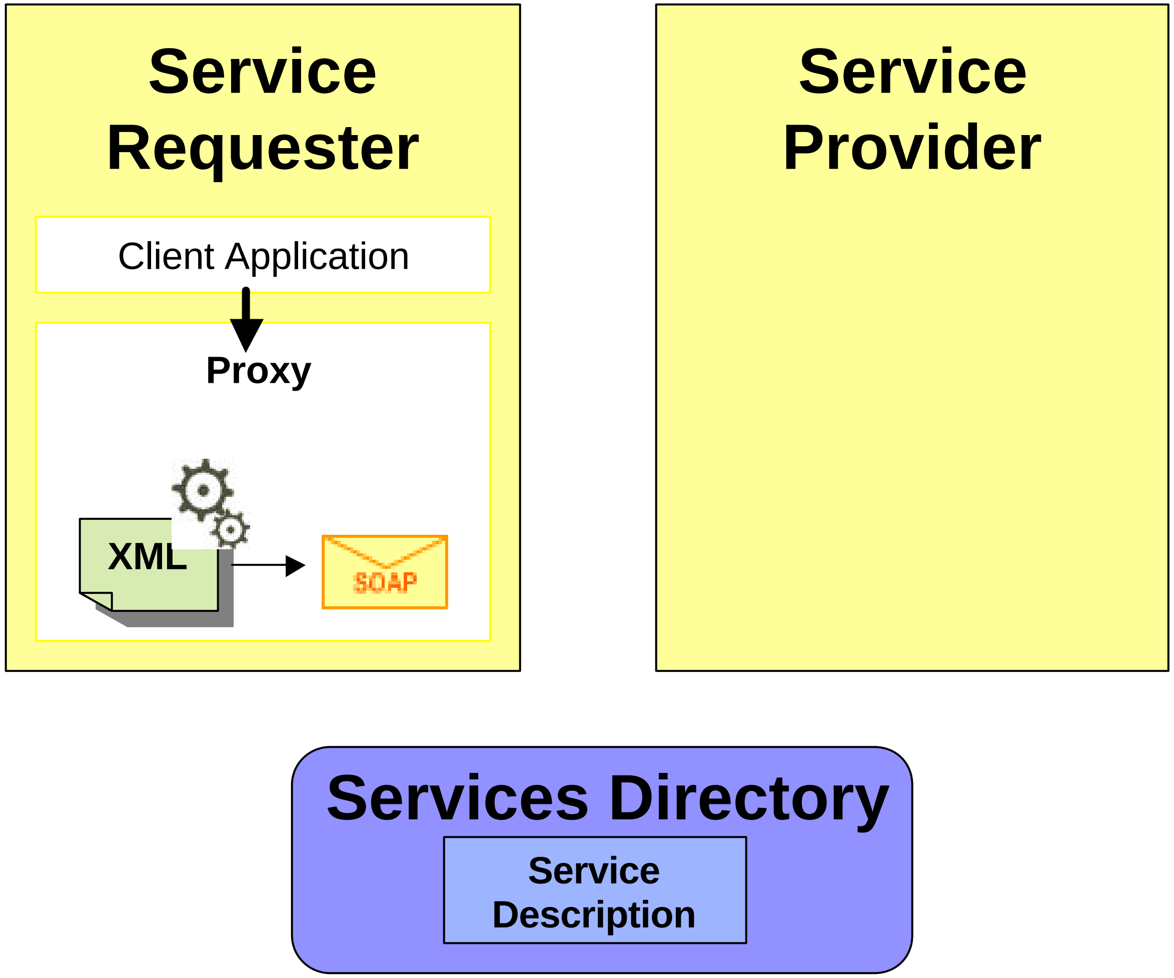


Components - principles of SOAP

1. **The client application invokes the proxy to request the service**

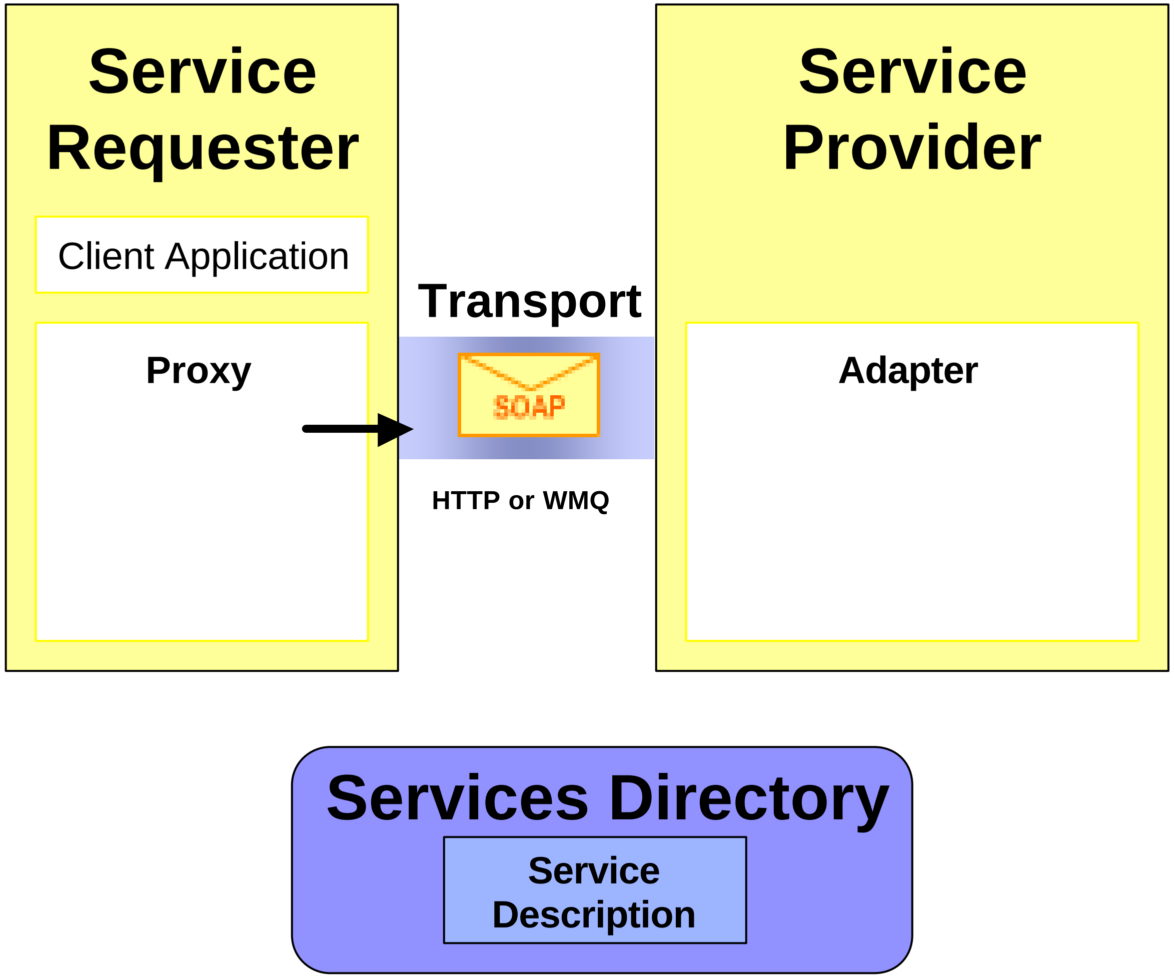
- **The proxy creates a SOAP request as an XML document and includes information from the Client application**

- **The SOAP message includes headers which contain additional information used by the middleware**



Components - principles of SOAP

2. **The proxy sends the SOAP request to the service provider using the transport specified in the Service Description**

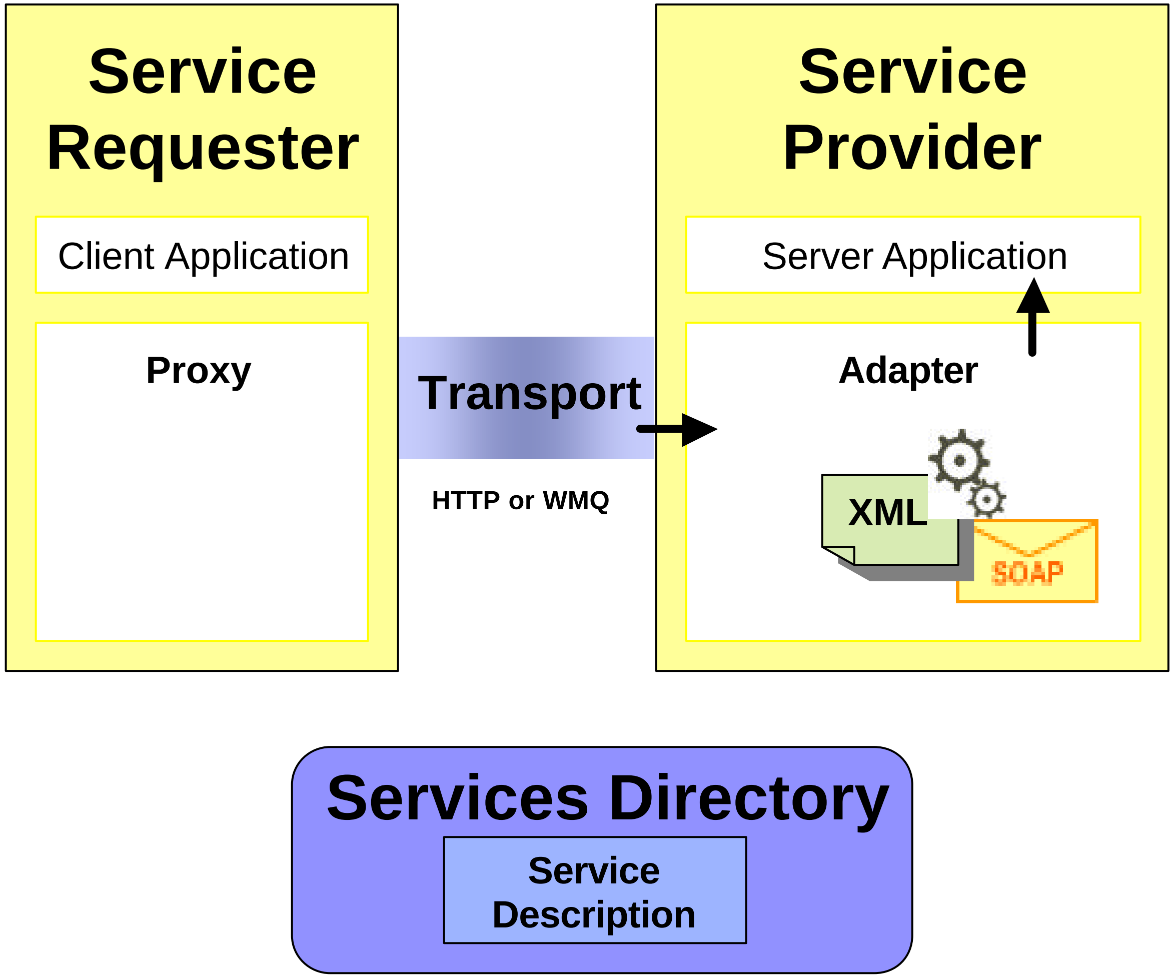


Components - principles of SOAP

3. **An existing CICS server application is enabled for Web services by using the SOAP for CICS Feature and an adapter component**

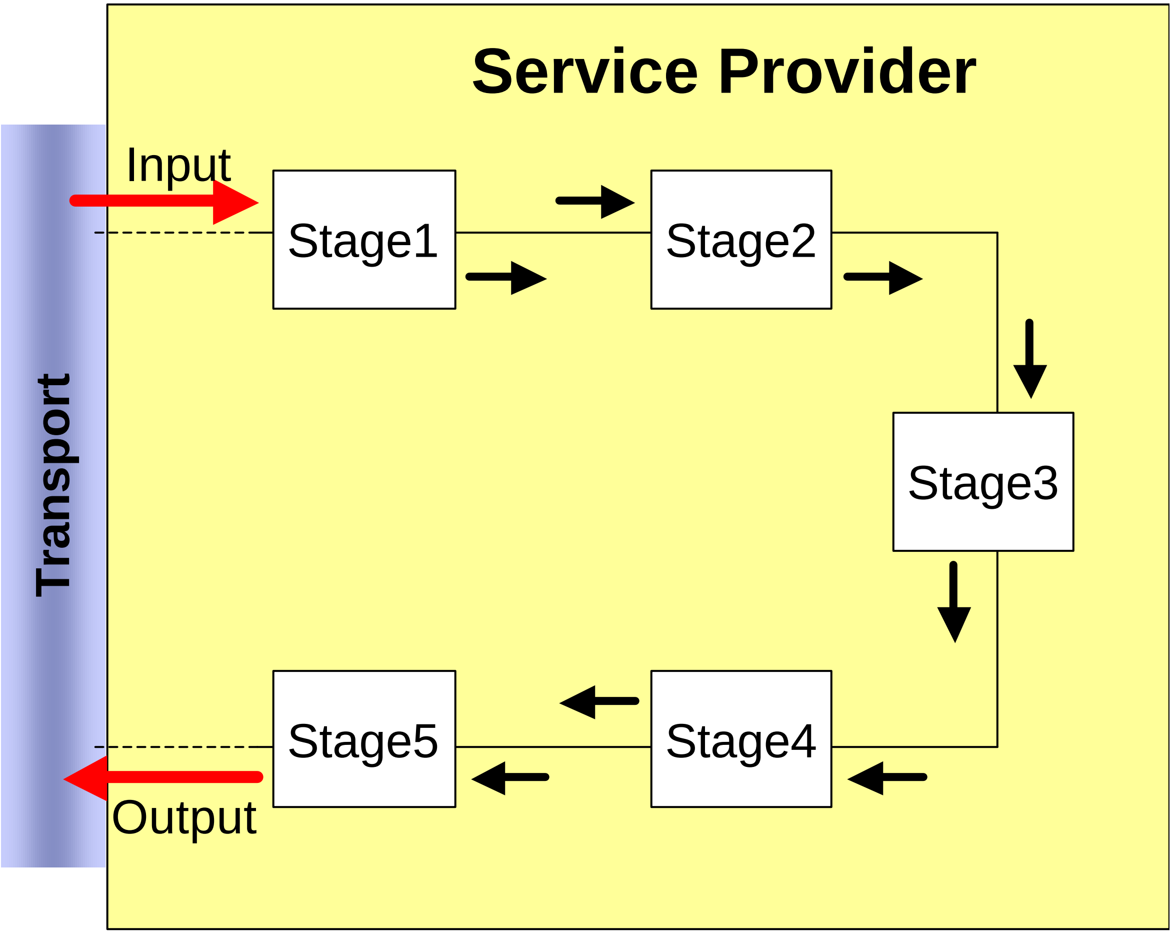
- **The adapter processes the SOAP request. It transforms the SOAP body to the form request by the server application, for example a CICS COMMAREA.**

- **The adapter component performs a similar role to the proxy. It transforms the SOAP messages so they can be understood by the server application**



Components – definition of a pipeline

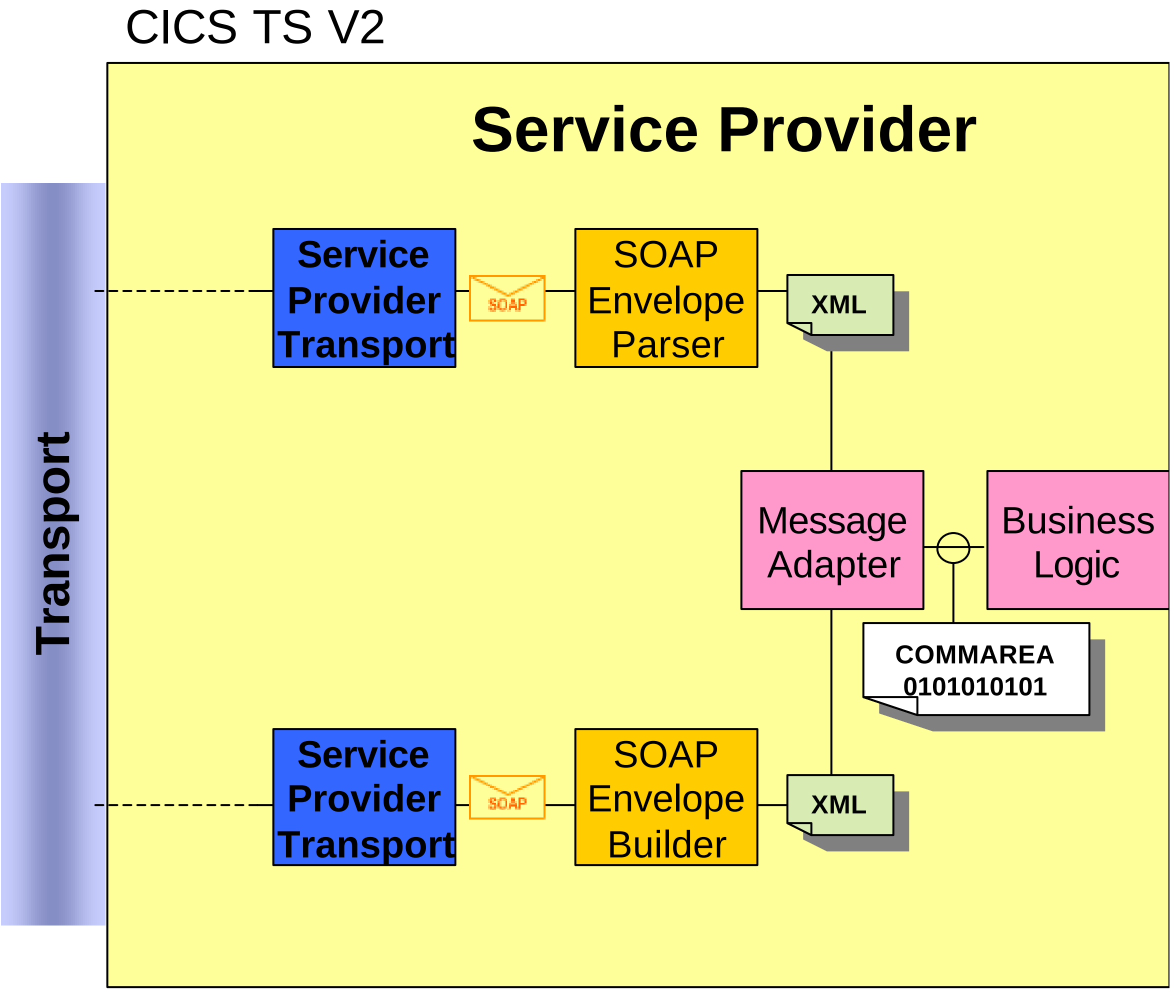
- **CICS uses a pipeline to process SOAP requests**
- **A pipeline is a sequence of processes arranged so that the output from one process is used as input to the next process**
- **Data is passed from one stage to the next in one or more containers. For optimum performance these containers exist in memory. CICS does not have a limit on the amount of data held in a container**



Components – use of a pipeline for CICS

In the SOAP for CICS pipeline:

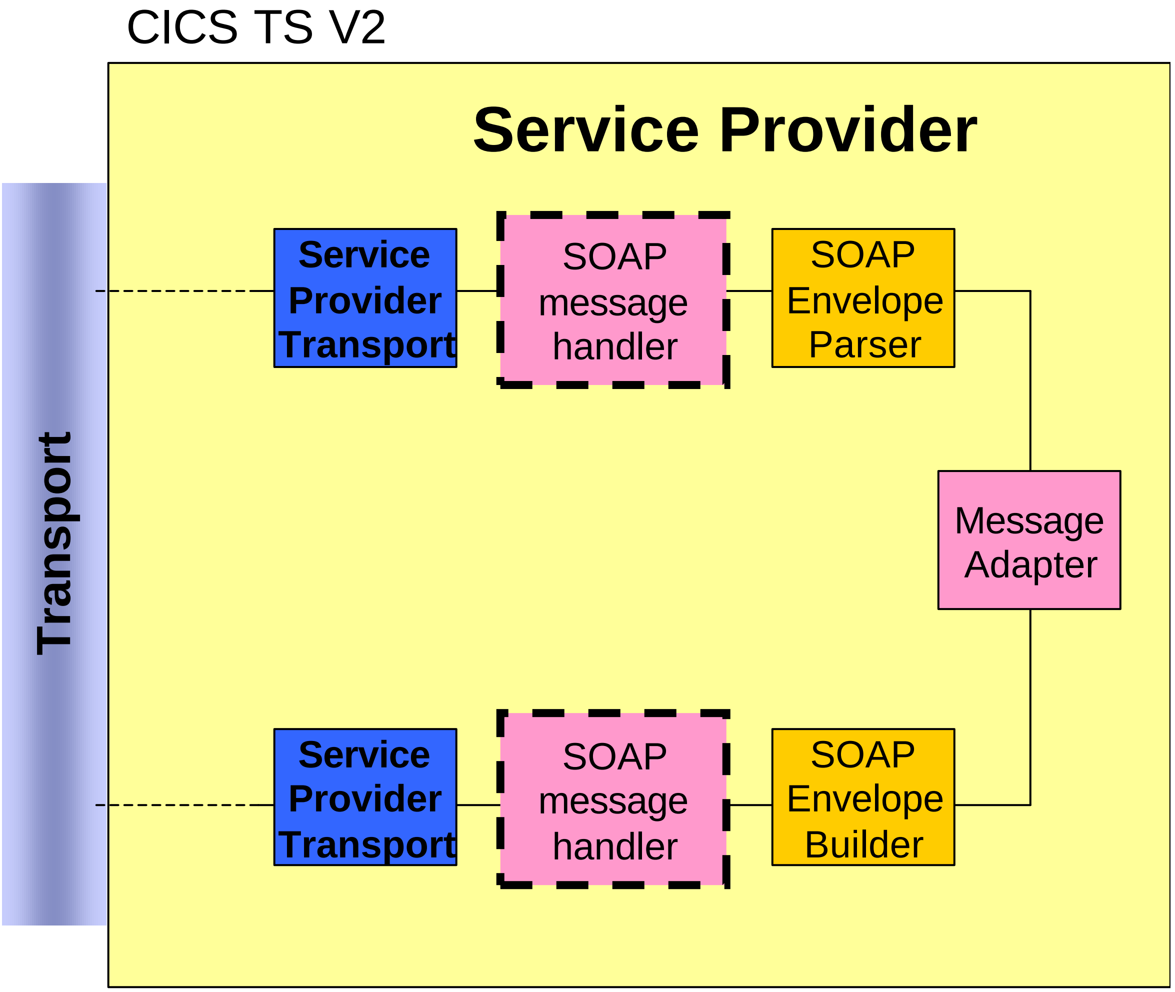
- **The service provider transport supports HTTP and WebSphere MQ®**
- **The service provider transport stage strips off the headers and passes the SOAP message onto the next stage in the pipeline**
- **The SOAP envelope parser strips off the SOAP envelope and passes the XML body onto the message adapter**
- **The message adapter parses the XML body and calls the business logic**
- **The business logic returns a response to the message adapter**
- **The message adapter generates the XML body and passes it to the SOAP envelope builder then on to the service provider transport stage**



Components – use of a pipeline for CICS

- **You can customize the process using a SOAP message handler.**

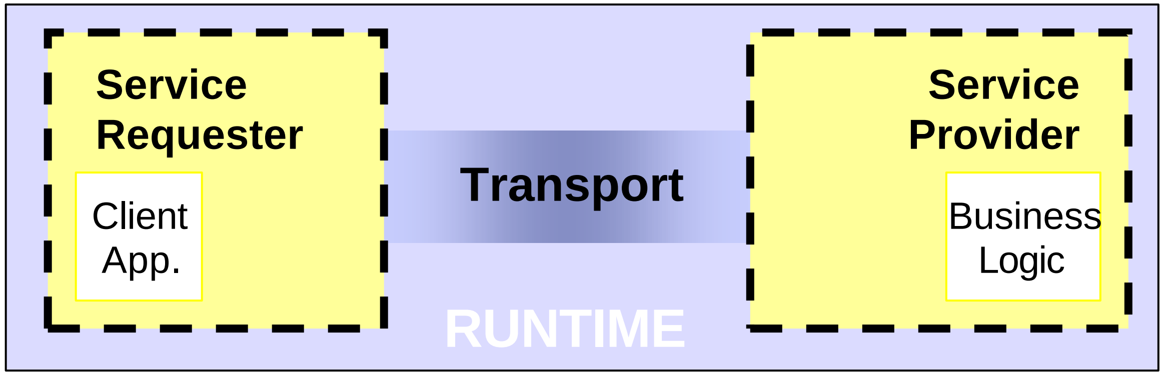
- **For example:**
  - You can extract information from the headers and perform logging
  - You can customise security processing, reject requests, and handle errors
  - You can record state information for use in outbound processing



Development – getting started

- **This section tells you how to enable a CICS program as a Web service**

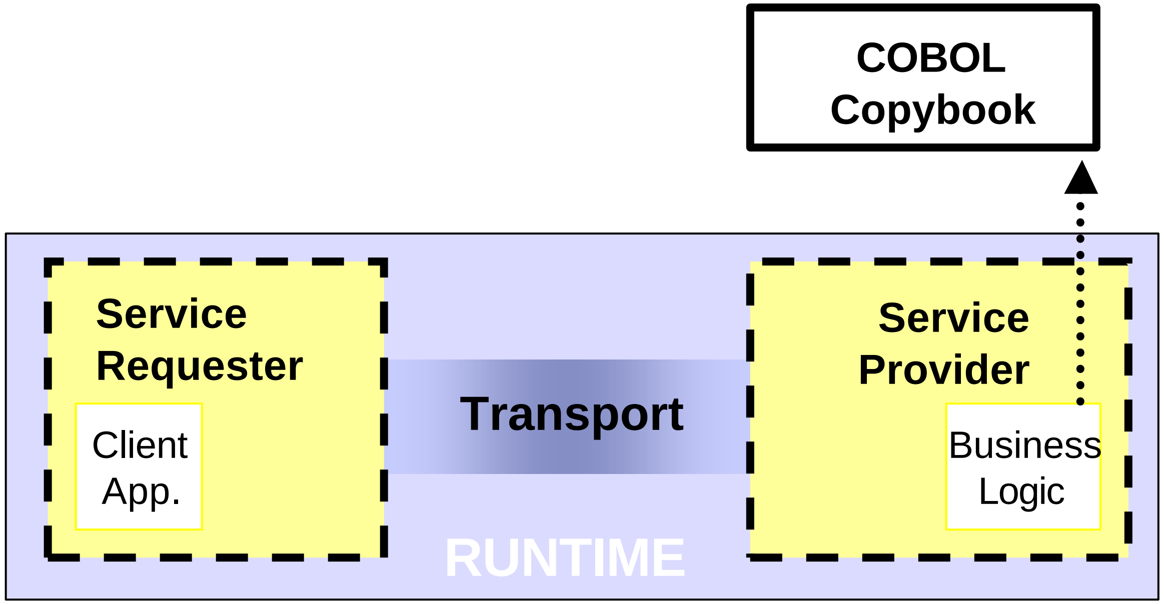
- **We explain the components that you must produce and how you can use tools (such as WebSphere® Studio Enterprise Developer) to help you do this**



Development – the service provider interface

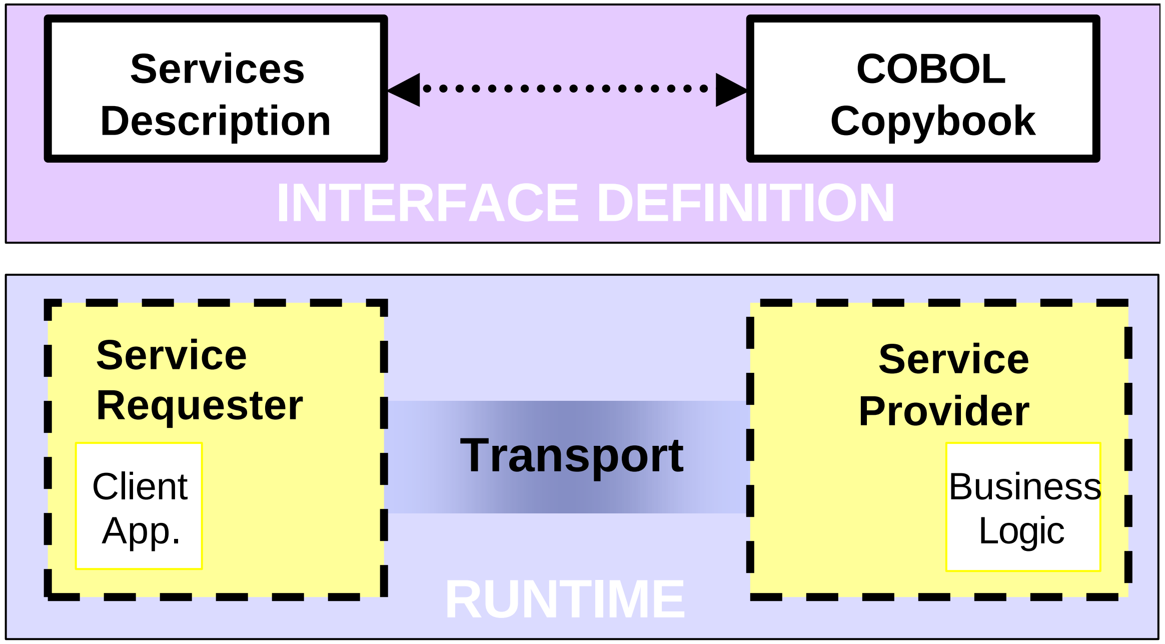
- **Typically your CICS program will have a COMMAREA interface, defined in a COBOL copybook**

- **COBOL Copybook**
  - When CICS programs are invoked, they receive information about the task to be performed in a data structure called a COMMAREA
  - The COMMAREA also has space for the program to return information to its caller
  - The definition of this COMMAREA must be known to both the caller of the program and the target program itself
  - This shared information can be defined once in a separate file known as a copybook
  - This copybook is used when compiling the target program and its callers



Development – the interface definition

- **Service Description**
  - Each Web service that you create is defined using Web Services Description Language (WSDL). This is a form of eXtensible Markup Language (XML). The information in the service description specifies the service provided and the details of their interfaces
- **You will need to produce a service description that matches the COMMAREA interface**
  - There are tools to create WSDL directly from COBOL copybooks, and create COBOL copybooks from WSDL



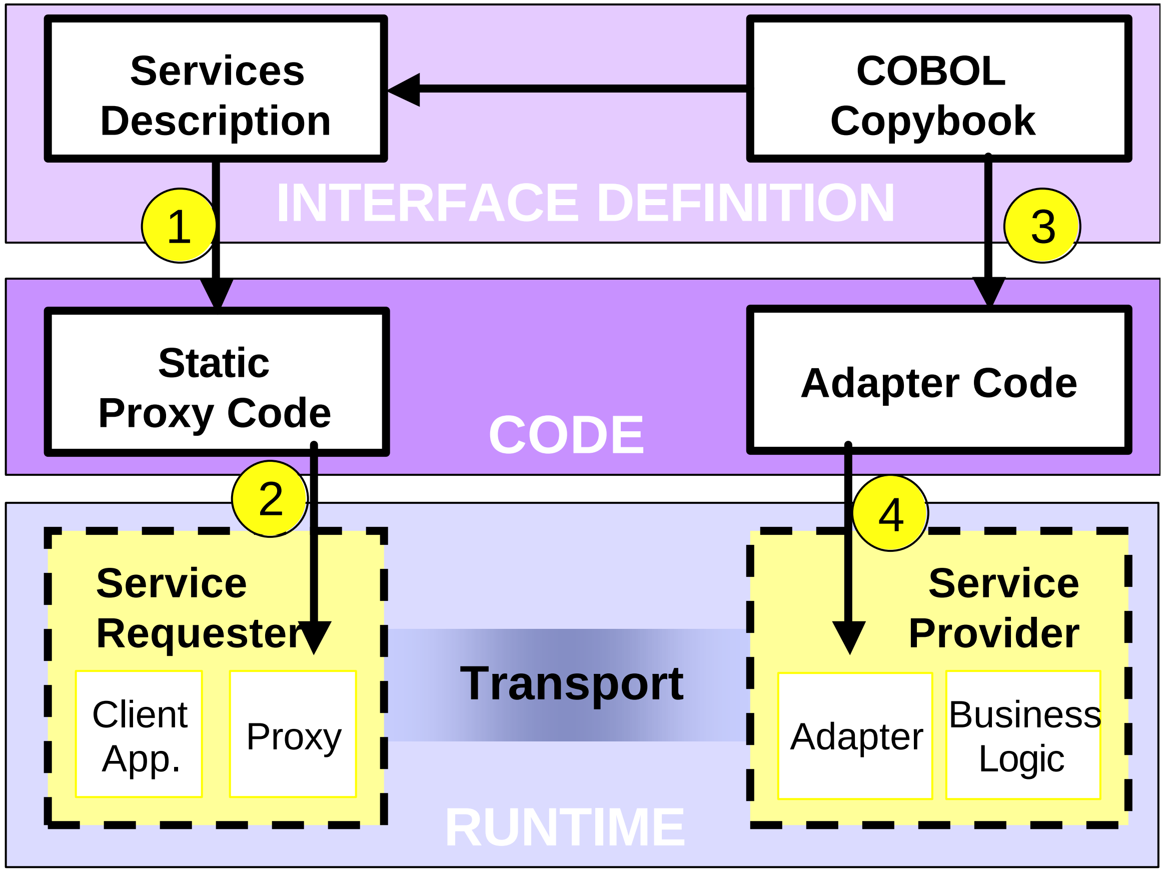
Development – client proxies and CICS adapters

1. **The service requester needs to know the location of the service and the type of messages used to invoke it. It gets this information from the service description**

2. **You need to develop a proxy which contains all the functions required to invoke the service. The proxy must be deployed along with the client application**

3. **The server application needs to know how to map the messages sent from the service requester into its COMMAREA interface. The adapter uses information from the service description and the copybook**

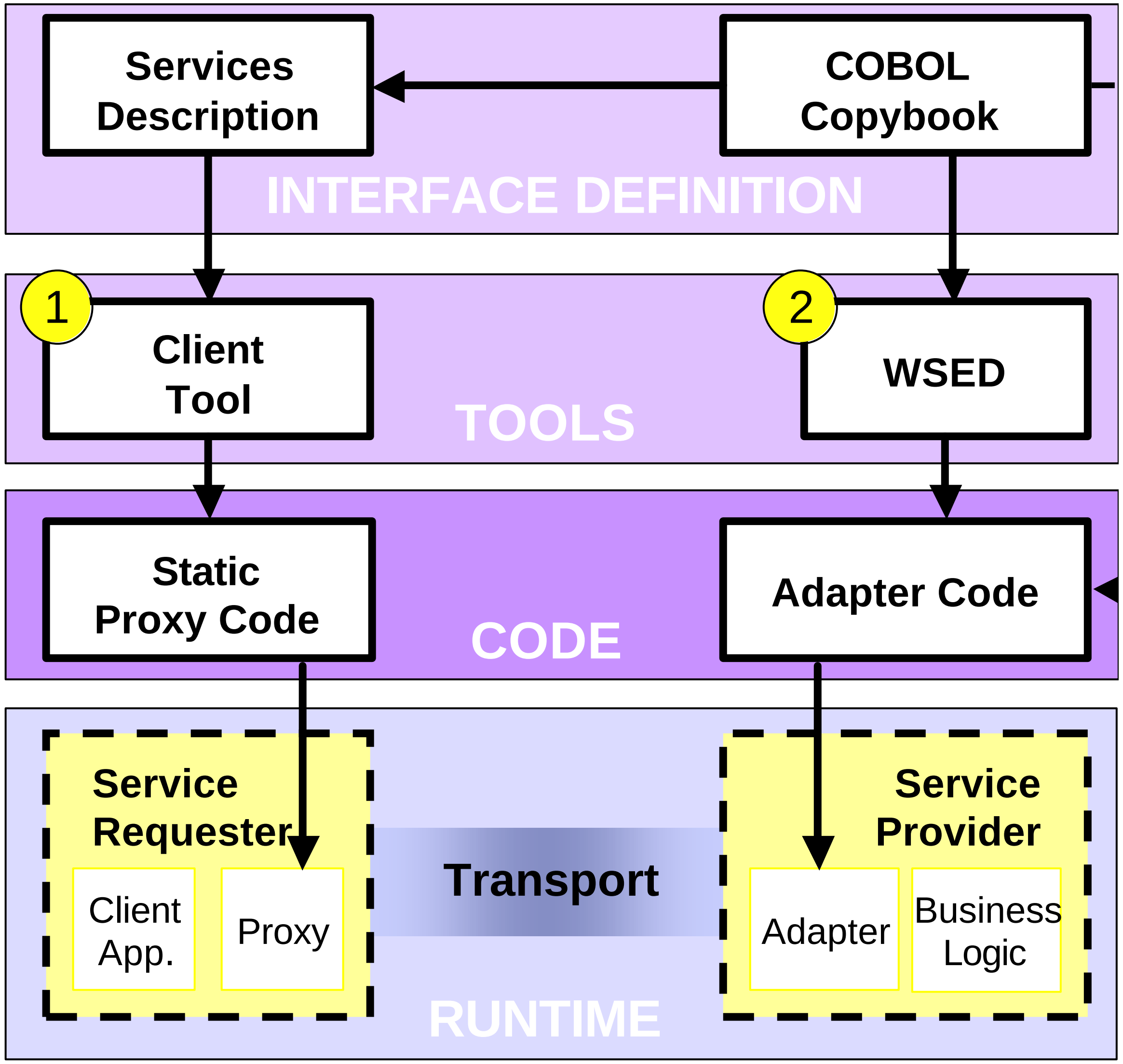
4. **You need to develop the adapter component which contains all the functions to handle the SOAP messages. The adapter component must be deployed in CICS**



Development – generating proxies, WSDL, and adapters

1. **For service requesters we advise you to use a development environment (such as WebSphere Studio or Visual Studio® .Net which produces both the client application and the proxy from the service description**

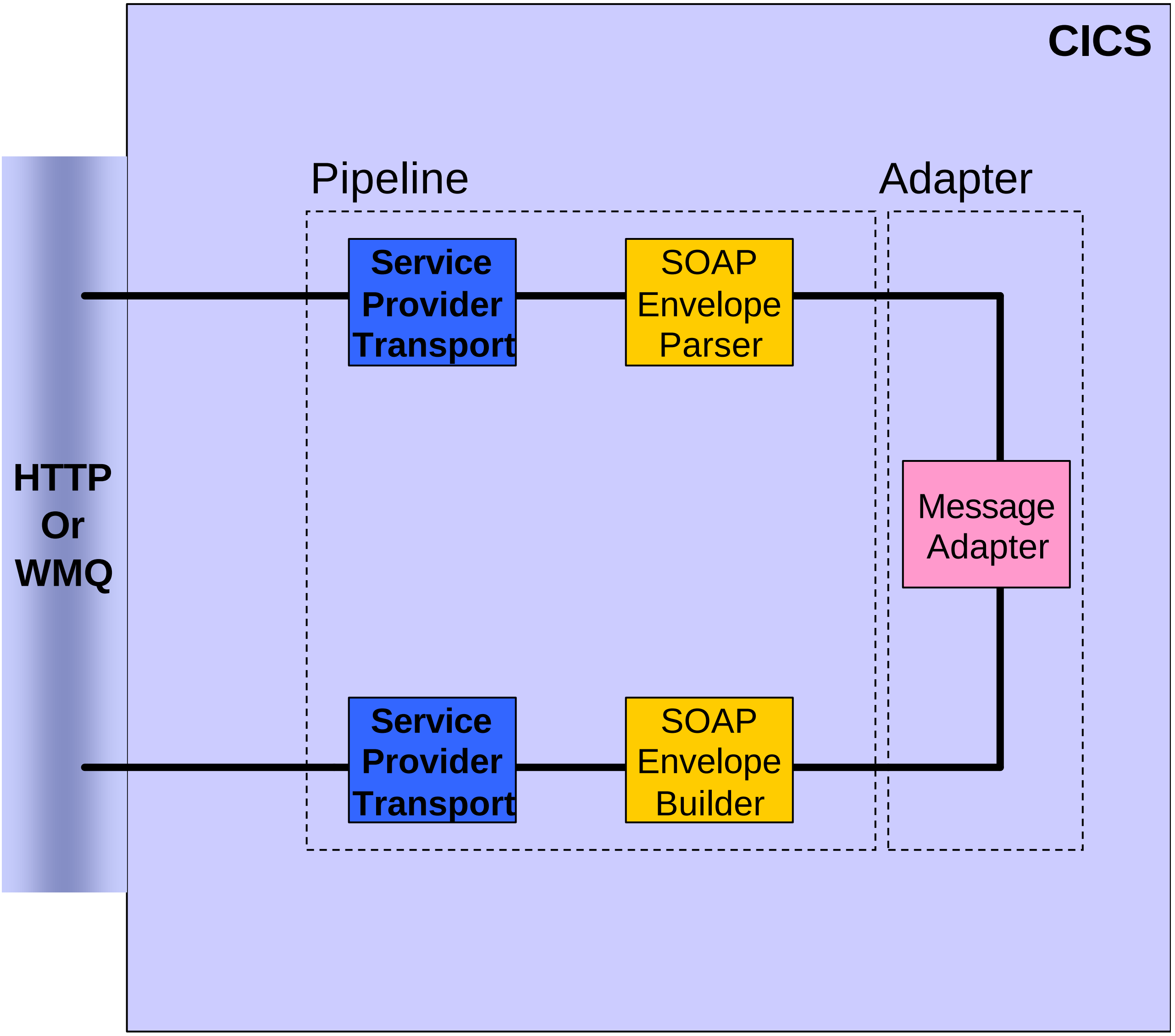
2. **We recommend that you use the WebSphere Studio Enterprise Developer (WSED) wizard which uses the copybook to produce both the adapter code for deployment into CICS and the service description**



Setup - introduction

- **This section tells you how to set up a CICS region to process SOAP requests**

- **There are two steps**
  - Setting up the HTTP or WMQ transport
  - Installing the pipeline and the adapters



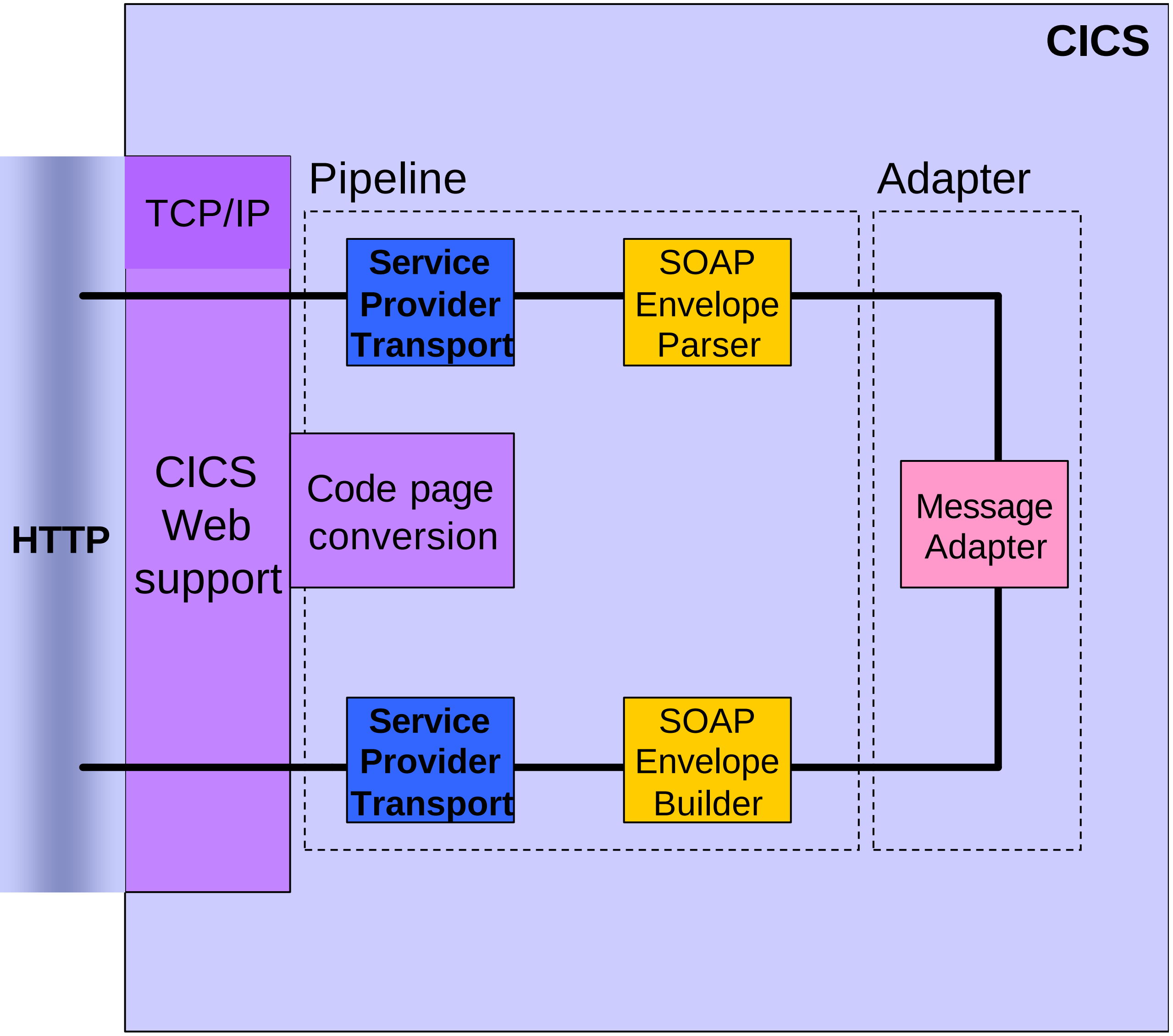
Setup - setting up HTTP

- **The HTTP transport for the SOAP for CICS feature uses the CICS Web support**

- **If you are not currently using the CICS Web support, then setting this up involves the following:**
  - Setting up TCP/IP for the CICS Web support
  - Generating a CICS table for the relevant code page conversion

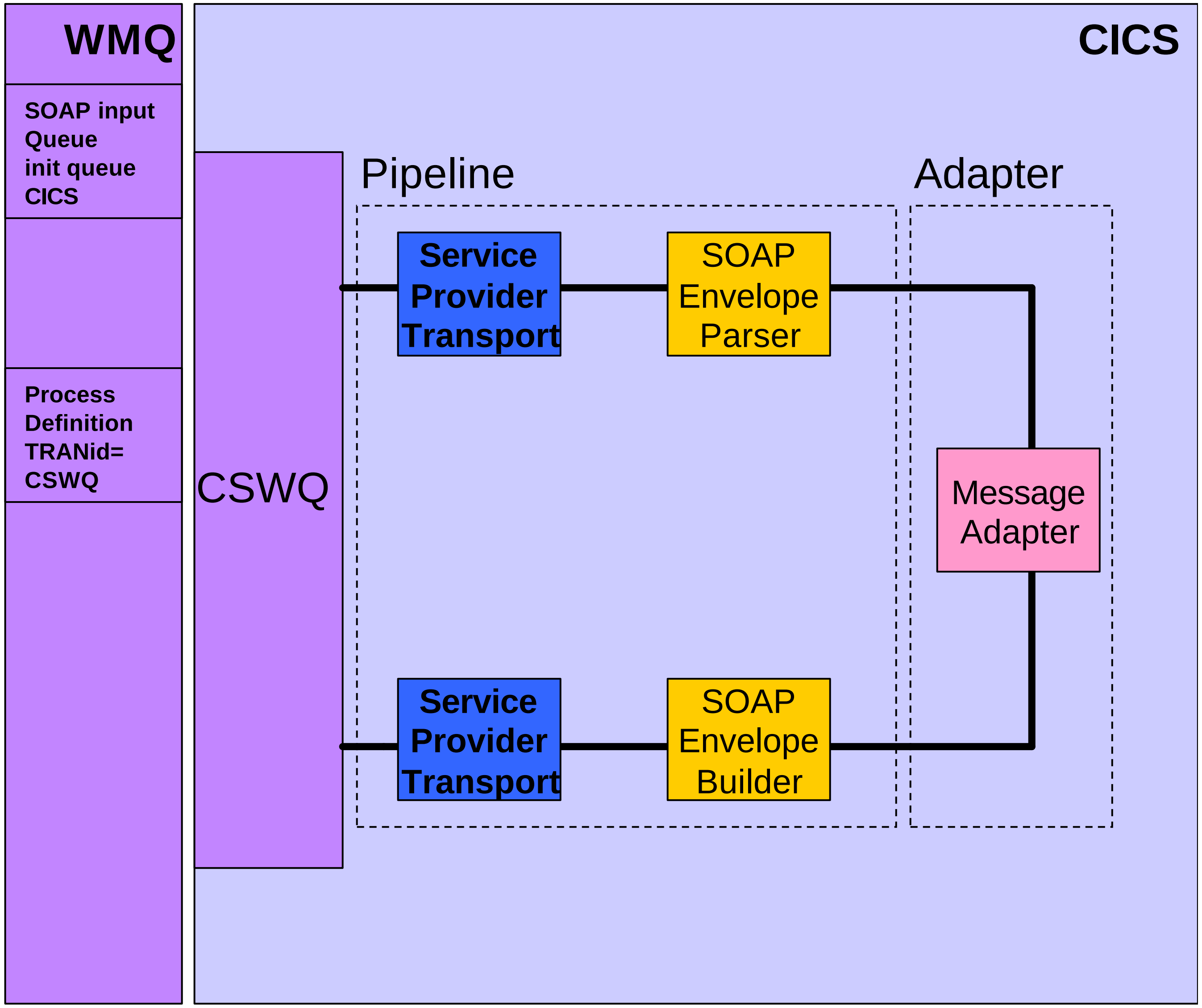
- **Use the supplied sample program to verify that the set up has been successful**

- **You are now ready to receive SOAP message using HTTP**



Setup - setting up WebSphere MQ

- **If CICS and WebSphere MQ have been set up to talk to each other, there is no additional set up needed in the CICS region**
- **However, some set up is required in WebSphere MQ**
  - Define local input queues for every service into CICS
- **These queues will store the SOAP messages until they are processed**
  - Define a process object in WebSphere MQ. This specifies the CICS transaction CSMQ
- **CICS uses CSWQ to get SOAP messages from the queues and process them**

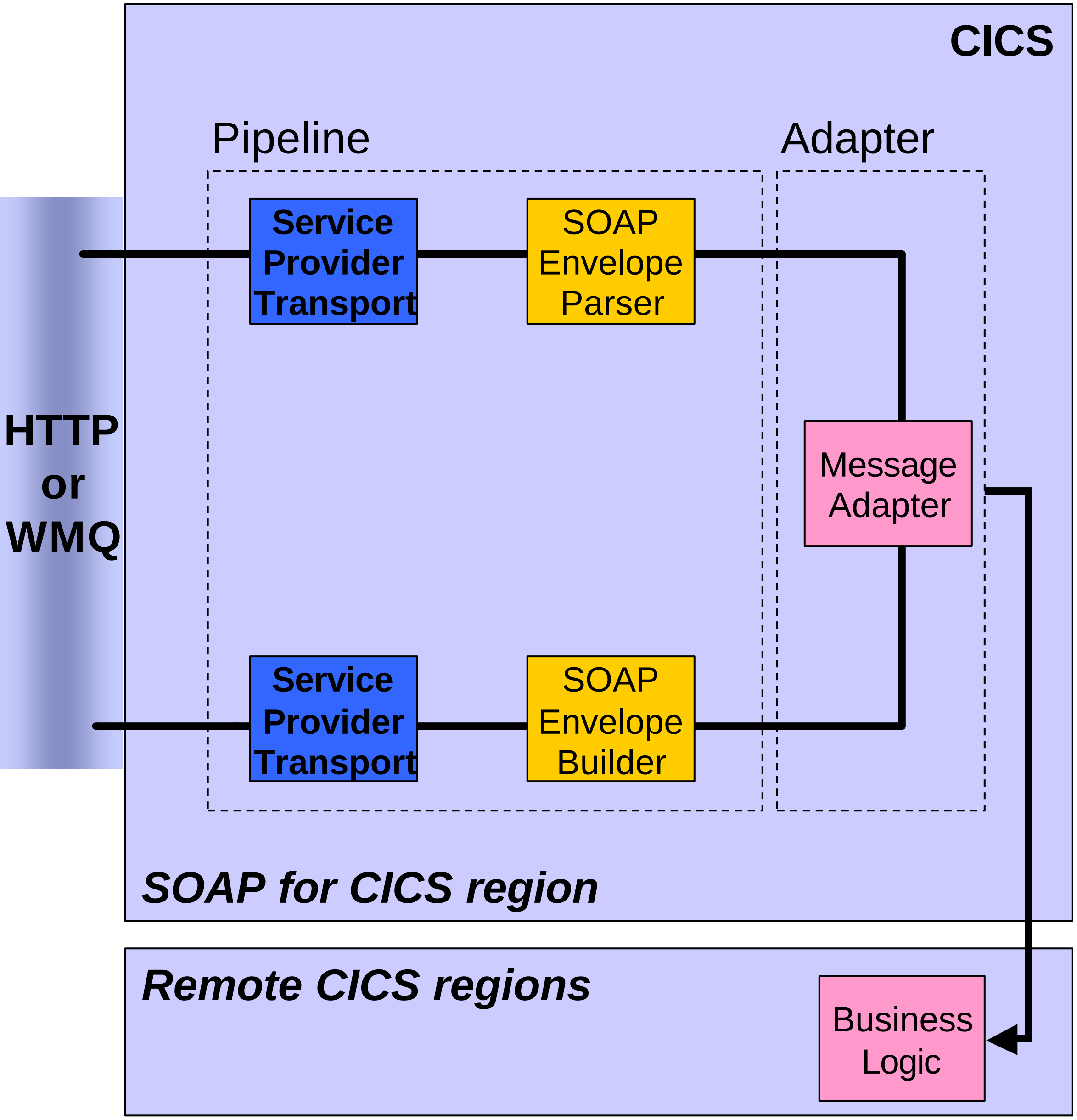


Setup – installing the pipeline and the adapters

- The pipeline and the adapters are standard CICS programs. These are installed in the normal way

- The pipeline and adapters must be installed in the same CICS region

- The service provider program (or business logic) can also reside in this region, or workload balanced into a remote CICS region using CPSM or other CICS techniques



Conclusion

- SOAP for CICS is now available as a fully supported, no charge, orderable feature of CICS TS V2
- SOAP for CICS enables new and existing mission critical CICS applications to interact securely with Web services independently of platform, environment, or programming language, either as a service requester or service provider
- Web service developers can now rapidly build open standards based two or three tier applications, independently of the CICS services that they will interact with
- Development tools play an important part in Web services, able to generate proxies and clients from single WSDL service description
- SOAP for CICS enables enterprises to react quickly to new opportunities, whilst reducing the cost and resource associated with business integration

Further information

- **SOAP for CICS Feature**
  - ibm.com/cics/soap/
- **Web Services Architecture - W3C**
  - w3.org/TR/ws-arch/
- **Web Services Zone - IBM developerWorks**
  - ibm.com/developerworks/webservices/
- **XML on z/OS and OS/390: Introduction to a Service Oriented Architecture**
  - Redbook: SG24-6826
- **Websphere V5 Web Services Handbook**
  - Redbook: SG24-6891
- **Self-Study Guide: WebSphere Studio Application Developer and Web Services**
  - Redbook: SG24-6407

Trademarks

- The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:
  - AIX
  - CICS
  - CICS Transaction Gateway
  - DB2
  - DFSMS
  - IBM
  - IMS
  - Language Environment
  - OS/390
  - RISC System/6000
  - RACF
  - RMF
  - S/390
  - VisualAge
  - WebSphere
  - z/OS
  - zSeries
- Java and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- SUN is a registered trademark of Sun Microsystems, Inc. in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a registered trademark owned by Linus Torvalds.
- Microsoft, Windows, Windows NT, Windows 2000 and the Windows logo are trademarks of Microsoft Corporation in the United States, other coutries, or both.
- Other company, product or service names may be trademarks or service marks of others.