Learning Objectives

As a result of this lecture, you will be able to:

- Comprehend the main ideas of SAP’s Web Service definition
- Understand the architecture of the Web Service Framework
- Understand what Virtual Interfaces, Web Service Definitions, Web service Configurations and other concepts of the Web Service Framework are
Agenda

Web Service Overview
- What are Web services?
- Why do we need them?

SAP NetWeaver
Web Service Technology

SAP Web Application Server 6.40
- Providing Web Services
- Consuming Web Services

Summary

Web Services Definition

Web Services are application functionalities supporting direct interaction by responding to service requests based on open Internet Standards
The nature of Web Services

**Web Services**

- act like a black-box that may require input and deliver a result
- are modular, self-contained and self-describing
- work on top of any communication technology stack
- can be published, discovered and invoked based on open technology standards
- work in synchronous and asynchronous scenarios
- facilitate integration within an enterprise as well as cross enterprises
### Web Services: Examples

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligent product catalog search</td>
<td>Publish and analyze financial reports (XBRL based)</td>
</tr>
<tr>
<td>Product availability check</td>
<td>Electronic bill presentment and payment</td>
</tr>
<tr>
<td>Pricing inquiry</td>
<td>Matching vacancies and job applicants profiles</td>
</tr>
<tr>
<td>Customer credit check</td>
<td>Postal service address check</td>
</tr>
<tr>
<td>Order status check</td>
<td>UDDI registration and discovery services</td>
</tr>
<tr>
<td>Vendor managed inventory</td>
<td>Automated web searches (Google)</td>
</tr>
<tr>
<td>Dynamic auctioning and bidding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Demand forecasts, stock replenishment …</td>
</tr>
</tbody>
</table>

- Dynamic auctioning and bidding
- Publish and analyze financial reports (XBRL based)
- Electronic bill presentment and payment
- Matching vacancies and job applicants profiles
- Postal service address check
- UDDI registration and discovery services
- Automated web searches (Google)

---

**Demo**

**Web Service Example**
Example: Web Service Scenario Credit Limit Check

Customer
- Order
- CRM
- Process Order

Company
- Create Order
- Credit Management System
- Credit Limit Check
- Credit Limit OK?
- OK
- rejected

Financial
- History

Service Provider
- Database

Credit Limit Check Example: Architecture

SAP Web AS 6.30 (Java)
- Cell Phone Shop
- Web Application (JSP)
- Web Service Proxy

Microsoft Excel XP / 2003 Beta
- Credit Limit Check
- Company Worksheet
- Web Service Proxy

SAP Web AS 6.30 (Java)
- Virtual Interface
- Session Enterprise Java Bean
- SAP JCo Proxy
- SAP JCo Proxy

R/3 Enterprise 4.7
- RFC Function Module (External Id -> Internal ID)
- RFC Function Module (Credit Limit Check)
Fundamental Elements of the complete Web Service Solution

- Open Technology Standards for Web Services
  - XML, WSDL, SOAP, UDDI, WSI
- Referent to business semantics
  - RosettaNet, Spec2000, HR-XML, XBRL, IFX, papiNet, ...
- Web Service Technology
- SAP NetWeaver

Agenda

- Web Service Overview
  - What are Web services?
  - Why do we need them?
- SAP NetWeaver
  - Web Service Technology
- SAP Web Application Server 6.40
  - Providing Web Services
  - Consuming Web Services
- Summary
Providing Web services based on Open Standards

- UDDI
- WSDL
- SOAP Processing
- Web Service Configuration
- UDDI Publishing
- WSDL Generation

Consuming Web services based on Open Standards

- UDDI
- WSDL
- SOAP Processing
- Web Service Configuration
- SOAP Interfaces
- Extensible Runtime
- Pluggable Features
- Security
- Transactions
- Protocols

Web Service Consumer

Web Service Provider

Development Environment
- UDDI based Web Service discovery
- Web Service Proxy Generation
- Proxy Configuration

Business Application

SAP Web Application Server

SAP Web Application

Web Service Provider

Extensible runtime
Pluggable features
Security
Transactions
Protocols
Exchange Infrastructure & Web Services

Exchange Infrastructure

- Business Process Engine
- Routing
- Mapping
- Integration Repository & Directory
- SAP Web Application Server
- Web Services Infrastructure
- SOAP
- Native Protocol
- mySAP Component
- NON SAP Component
- Adapter
- RFC
- RMI

Enterprise Services: Runtime View

- Standard Web Service Execution (P2P)
- Web Service Infrastructure
- Integration Server
- Integration Repository & Directory
- Mediated Execution via Web Services
- SAP Web AS as client
- Outbound Proxy
- SAP Web AS as server
- Inbound Proxy
- Application

© 2003 SAP AG
Agenda

Web Service Overview
- What are Web services?
- Why do we need them?

SAP NetWeaver
Web Service Technology

SAP Web Application Server 6.40
- Providing Web Services
- Consuming Web Services

Summary

Web Service within SAP Web Application Server

The Web Service capability within SAP Web AS is
- the foundation for all mySAP.com solutions
- a scalable, reliable Web application infrastructure that delivers high-performance Web Services
- offering Web Service functionality for the ABAP and J2EE personality with a common architecture
- allowing to expose existing functionality (BAPIs, RFMs, IDOCs, EJBs, Java Classes, XI Server Proxies) as Web Service
- easily allowing to consume Web Service from any source
- requiring only configuration (no coding)
Benefits

- Web Service interfaces provide a 'virtual' abstraction from the implementation layer
- Fully integrated into development environment
- Full fledged UDDI capabilities
  - UDDI server implementation
  - UDDI client functionality
- Standard compliant WSDL generation
- Support of client proxy generation for ABAP and J2EE
- Extensible SOAP Runtime
- Pluggable Features
  - Security
  - Protocols
  - ....

SAP Web Services Infrastructure
- Role-based Approach -

Provider-side

- Web Service Developer
  - defines design time information of a Web Service

- Web Service Configurator
  - defines runtime information of a Web Service

Consumer-side

- Web Service Client Developer
Agenda

Web Service Overview
- What are Web services?
- Why do we need them?

SAP NetWeaver
Web Service Technology

SAP Web Application Server 6.40
- Providing Web Services
- Consuming Web Services

Summary

The Web Service Creation Wizard
- Background -

Creating a Web Service in less than 1 minute

- The Web Service Creation wizard
  - proves that Web services are easy and out-of-the-box
  - allows to expose an existing endpoint as a Web Service with few mouse-clicks
  - Only shows the most important settings in the wizard
    - Default values for other settings
  - hides technical details
  - implicitly creates all necessary objects
Steps for the Web Service Wizard

1. Implement Business Logic
2. Start Web Service Wizard
3. Deploy Web Service

.publish Web Service (optional)

Virtual Interface
Web Service Definition
Web Service Config.

UDDI

Demo

Web Service Creation Wizard
Step 1 – Implement Business Logic

Function module: Display_Z_RFC_CREDIT_LIMIT_CHECK

Step 2 – Start Web Service Wizard

Function module: Display_Z_RFC_CREDIT_LIMIT_CHECK
Step 2 – Web Service Wizard (1)

This Wizard allows you to create objects required for a Web Service. The objects you need are as follows:

- A virtual interface
- A Web service definition

Afterwards the Web Service is automatically activated so that it can be called.

Step 2 – Web Service Wizard (2)

Creating a Virtual Interface

The virtual interface maps the names of the SAP object to those in the Web service. The wizard creates a default mapping automatically. To edit the virtual interface, use the Object Navigator (SE80).

Enter a name to the new virtual interface. Specify a short text and choose an endpoint type.
Step 2 – Web Service Wizard (3)

Specify an object with the appropriate endpoint type.

Step 2 – Web Service Wizard (4)

Creating a Web Services Definition

The Web service definition refers to the actual interface you just defined. Enter the name and short text to the web service definition.

Web service-specific features are stored in the Web service definition. Using various feature profiles you can select a predefined set of features.

The Web service definition can be processed further in the Object.

Name: CreditSLI41Check
Description: Web Service Definition of CreditSLI41Check
Profile: Internet Public
Profile: The interface is available on the Internet via SOAP and is not secured.
### Step 2 – Web Service Wizard (5)

As soon as you choose “Complete”, the following objects will be released:
- Virtual Interface CreditInitCheckVHF
- Web Service CreditInitCheckWSD

### Web Service Wizard Results

**All necessary files are created**

- Virtual Interfaces
- Web service Definition
- Web Service Configuration
- Enterprise Application Project (optional)
The Web Service Homepage

- is available for each activated Web Service
- shows the documentation for the Web Service
- allows to retrieve WSDL descriptions in different styles
- offers testing capabilities

Demo

Web Service Homepage
The Web Service Creation Wizard
- Background -

The Step-by-Step approach

- Create all objects by hand
- Full flexibility and options
- More expenses
- ... but still configuration only

The Step by Step Approach

1. Implement Business Logic
2. Define Virtual Interface (VI)
3. Create Web Service Definition (WSD)
   3a. Publish WSD (optional)
4. Configure Web Service
5. Activate Web Service
   5a. Publish Web Service (optional)
The Web Service Creation Wizard
- Background -

The Step-by-Step approach

- Create all objects by hand
- Full flexibility and options
- More expenses
- ... but still configuration only

Support for Web Service Developers (1)
- Background -

The Web Service developer

- knows/defines the application logic
- knows/defines the Web Service Interface
  - which should be accessible for clients
- knows/defines the “behavior” of the Web Service
  - e.g. “Stateless / Stateful”, Security requirements, etc.
- does not need to have detailed knowledge about
  - technical protocols which implement abstract behavior
  - concrete application servers on which the Web Service will be running

→ Defines Design-Time Information of a Web Service
Support for Web Service Developers (2)
- Tasks of a Web Service Developer -

Step 1: Implementing the business logic

- Supported by the standard IDE
- Takes place in “Endpoint-specific” standard Tools
  - Java: SAP NetWeaver Developer Studio
  - ABAP: SE80

Support for Web Service Developers (3)
- Tasks of a Web Service Developer -

Step 2: Definition of Virtual Interface

- Web Service Interface = Interface visible to clients (via WSDL, UDDI, … )
- Supported by the Web Service Infrastructure via Virtual Interface Editor

- A Virtual Interface
  - provides abstraction from concrete implementation (=endpoint)
  - allows to publish a “view” on existing implementations as Web Service Interfaces
    - i.e. renaming, hiding of parameters/methods, default values
  - allows to define how the Web Service Interface is represented in the SOAP message
Virtual Interfaces

Request

Response

creditLimitCheck

Default Value: BUP001

idNumber (I)

score (O)

credit Limit (O)

limit Currency (O)

valid To (O)

FSCM_Cr_Limit_Check

Implementation Layer

i_credit segment (I)

i_partner (I)

e_score (O)

e_limit (O)

e_currency (O)

e_valid_to (O)

<i>SOAP-ENV:Envelope

<i>SOAP-ENV:Body

<ns1:creditLimitCheck>

<ns1:idNumber>1001</ns1:idNumber>

</ns1:creditLimitCheck>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<i>SOAP-ENV:Envelope

<i>SOAP-ENV:Body

<rpl:creditLimitCheckResponse>

<rpl:Response>

<tns:score>980</tns:score>

<tns:creditLimit>500000.00</tns:creditLimit>

<tns:limitCurrency>USD</tns:limitCurrency>

<tns:validTo>2005-01-01</tns:validTo>

</rpl:Response>

</rpl:creditLimitCheckResponse>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

Rename Methods

Virtual Interfaces

Request

Response

creditLimitCheck

Default Value: BUP001

idNumber (I)

score (O)

credit Limit (O)

limit Currency (O)

valid To (O)

FSCM_Cr_Limit_Check

Implementation Layer

i_credit segment (I)

i_partner (I)

e_score (O)

e_limit (O)

e_currency (O)

e_valid_to (O)
Virtual Interfaces

Request

<SOAP-ENV:Envelope>
  <SOAP-ENV:Body>
    <ns1:creditLimitCheck>
      <ns1:idNumber>1001</ns1:idNumber>
    </ns1:creditLimitCheck>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

Response

<SOAP-ENV:Envelope>
  <SOAP-ENV:Body>
    <rpl:creditLimitCheckResponse>
      <rpl:Response>
        <tns:score>980</tns:score>
        <tns:creditLimit>500000.00</tns:creditLimit>
        <tns:limitCurrency>USD</tns:limitCurrency>
        <tns:validTo>2005-01-01</tns:validTo>
      </rpl:Response>
    </rpl:creditLimitCheckResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

FSCM_Cr_Limit_Check
Implementation Layer

idNumber
(score
credit Limit
limit Currency
valid To)

i_credit
segment (I)
i_partner
(e_score
credit Limit
limit Currency
valid To)

Virtual Interface

creditLimitCheck

Default Value: BUP001

Rename Parameters

Hide parameters and / or provide default values

Response

<SOAP-ENV:Envelope>
  <SOAP-ENV:Body>
    <rpl:creditLimitCheckResponse>
      <rpl:Response>
        <tns:score>980</tns:score>
        <tns:creditLimit>500000.00</tns:creditLimit>
        <tns:limitCurrency>USD</tns:limitCurrency>
        <tns:validTo>2005-01-01</tns:validTo>
      </rpl:Response>
    </rpl:creditLimitCheckResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
Demo

Virtual Interfaces

Virtual Interface Editor
- Creation of a „default“ Virtual Interface -

Virtual Interface ändern

Virtual Interface: MAPI_ACE_Credit
Modul: FUNCTION

EndpunktTyp: Funktionston

- ID
- CREDIT_LIMIT
- CURRENCY
- VALID_TO
- STATUS
- SCORE
- CREDIT_LIMIT
- CURRENCY
- VALID_TO

- Lokalkollektiv: 5
- Filter

- Stapelfeld:
  - MAPI_ACE_Credit
  - CreditLimit
  - CreditLimitCheck
  - CreditLimitCheckW
  - CreditLimitCheckM

© 2003 SAP AG
Virtual Interface Editor
- Rename operation -

Virtual Interface Editor
- Parameter Mapping -
Virtual Interface Editor
- Type Mapping (3) -

Virtual Interface Editor
Type Mapping (4)
Virtual Interface Editor
Type Mapping (5)

Virtuals Interface: VCreditLimitCheck
Endpunkt: Funktionsbaustein

**Originaltypen**
- STATUS [STRINGVAL]
- SCORE [STRINGVAL]
- CREDIT_LIMIT [STRINGVAL]
- LIMIT_CURRENCY [STRINGVAL]
- VALID_TO [STRINGVAL]

**Kopierte Typen**
- CREDIT_INFO
  - Status [STRINGVAL]
  - Score [STRINGVAL]
  - CreditLimit [STRINGVAL]
  - LIMIT_CURRENCY [STRINGVAL]
  - VALID_TO [STRINGVAL]

**Feldnamen**
- STATUS
- LIMIT_CURRENCY

- Validieren auf Originalfeldname: aktiviert

Virtual Interface Editor
Type Mapping (6)

Virtuals Interface: VCreditLimitCheck
Endpunkt: Funktionsbaustein

**Originaltypen**
- CREDIT_INFO
  - Status [STRINGVAL]
  - Score [STRINGVAL]
  - CreditLimit [STRINGVAL]
  - LIMIT_CURRENCY [STRINGVAL]
  - VALID_TO [STRINGVAL]

**Kopierte Typen**
- CREDIT_INFO
  - Status [STRINGVAL]
  - Score [STRINGVAL]
  - CreditLimit [STRINGVAL]
  - LIMIT_CURRENCY [STRINGVAL]
  - VALID_TO [STRINGVAL]
Virtual Interface Editor - Type Mapping (7) -

Original type

Virtual Interface Editor - Type Mapping (8) -

Choose new type
Virtual Interface Editor
- Type Mapping (9) -

Support for Web Service Developers (4)
- Tasks of a Web Service Developer -

Step 3: Creation of Web Service Definition

- Web Service Definition = Assignment of behavior to a Web Service Interface via features

Motivation
- a Web Service is not only described by its interface but also by its behavior with respect to
  - stateless/stateful communication,
  - transactional behavior,
  - security requirements, ...

Idea:
- describe this kind of behavior via abstract features
- leave it to the Web Service Configurator to assign technical protocols to these abstract features

Example:
- Feature: Stateful
- Technical Protocols: Stateful via HTTP-Cookies, Stateful via URL-Extension
Support for Web Service Developers (5)
- Tasks of a Web Service Developer -

Benefits

- The same Web Service definition can be activated on various systems which may differ in their technical capabilities
- Web Service Client implementations do not depend from technical server information
  - Clients are generated based on implementation-independent Web Service definitions
  - Technical details are configured separately in the Web Service Client Runtime

Supported by the Web Service Infrastructure via

- Web Service Definition Editor
- UDDI Publication of Web Service Definitions
- WSDL Generation for Web Service Definitions

Demo

Web Service Definition
Web Service Definition - UDDI Publishing

Start UDDI publishing

UDDI Publishing – Authentication

UDDI Registry Information

Select UDDI Registry

User: Administrator
Password: [enter password]
Next
**UDDI Publishing – Service Definition Details**

Publish Service Definition

**Information by UDDI Registry**

- **Name**: The service definition is not published yet
- **UDDI Registry**: A Test Registry

**Language**

- **English (America)**

**Service Definition Details**

- **Overview URL**: http://F44D71-99000000+32x2C62f%20%3D%2011.9886e_7%3A943k1%3C1%20%7F%1D%2D%2D%20%20%20%20%064760006526

**New Description**

- **Publish**

---

**UDDI Publishing – Result View**

- **TModel Published Successfully**
- **Votes**: 1000000-0000-1141-0451-090020059122
Support for Web Service Configurators (1)
- Background -

The Web Service Configurator

- knows the technical protocols which implement the abstract features
- knows the System Landscape
- knows the Web Service capabilities of the single Application Servers

→ Defines Runtime information of a Web Service

Support for Web Service Configurators (2)
- Tasks of a Web Service Configurator -

Step 1: Technical configuration and Activation of a Web Service Definition

- Assignment of technical protocols to the abstract features
  - based on the Web Service capabilities of the Application Server

- Configuration of additional Web Service Runtime features
  - Logging, Tracing, Monitoring
  - Security Roles
  - ...

- Saving and automatic registration in the Web Service Runtime Registry

- Supported by Web Service Infrastructure via
  - Web Service Configuration Transaction (WSCONFIG)
  - Web Service Administration Transaction (WSADMIN)
    - WSDL Generation for deployed Web Services
    - UDDI Publishing
Demo

Web Service Configuration and Administration

Transaction „WSCONFIG“

Release Web Services for SOAP Runtime

<table>
<thead>
<tr>
<th>SOAP Application</th>
<th>BOAP Application for RFC-Call to Function module</th>
<th>CreditLimitCheckWS</th>
<th>Web Service</th>
<th>CREDITLIMITCHECKWS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

default_host/sap/bc/itc/sap/CREDITLIMITCHECKWS
**Web Service Configuration**

- Create new Configuration (Overall Settings) -

**Release Web Services for SOAP Runtime**

- Configure Transport Settings

- Create new Configuration (Transport Level Config) -
Web Service Configuration
- Create new Configuration (Transport Level Config) -

- Create new Configuration (Operation configuration) -

Release Web Services for SOAP Runtime

Configure Security Settings
Web Service Administration
- List of Available Web Services -

- Configure Web Service -
UDDI Publishing – Authentication

UDDI Registry Information

Select UDDI Registry

UDDI Registry: A Test Registry

Enter User Information

User: Administrator
Password: 

Next

UDDI Publishing – Service Publication: Names

Service Publication

Information by UDDI Registry

Business Entity: SAP TechEd Basel
Service title: The service is not published yet
UDDI Registry: A Test Registry

Next

Page 1 / 1

© 2003 SAP AG
Session ID, Speaker 43
UDDI Publishing – Service Publication: Binding Templates

Service Publication

Information by UDDI Registry

- Business Entity: SAP NetWeaver Data Company
- Service Key: The service is not published, yet
- UDDI Registry: A Test Registry

<table>
<thead>
<tr>
<th>Address Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Language ID</td>
<td>en-US</td>
</tr>
<tr>
<td>Character Set</td>
<td>UTF-8</td>
</tr>
</tbody>
</table>

Agenda

- **Web Service Overview**
  - What are Web services?
  - Why do we need them?
- **SAP NetWeaver**
  - Web Service Technology
- **SAP Web Application Server 6.40**
  - Providing Web Services
  - Consuming Web Services
- **Summary**
Support for Web Service Consumers (1)
- Background -

The Web Service client developer

- implements the client application
  - knows how to use the Web Service client proxy in the client application
- knows how to get the description of a Web Service
  (UDDI, WSDL, …)
- knows how to generate the Web Service client proxy

Support for Web Service Consumers (2)
- Tasks of a Web Service Consumer -

Step 1: Retrieve description of Web Service

- Retrieval of URL of WSDL description
  - either manually,
  - or using UDDI Browsing
- Supported by the Web Service Infrastructure via
  - Web Service Proxy generation
  - UDDI Client Browser
  - Exchange Infrastructure IFR - browsing
Support for Web Service Consumers (3)
- Tasks of a Web Service Consumer -

**Step 2: Generate Web Service Client Proxy**

- Start proxy generator based on imported WSDL Document
- Supported by the Web Service Infrastructure via
  - Web service Proxy Definition

**Step 3: Implementation of Client Application**

- Use generated client proxies
- Supported by the Web Service Infrastructure via
  - SAP Development Workbench (SE80)

---

**Demo**

**Web Service Proxy Project**
Display generated Client Proxy (1)
- Main properties -

Display generated Client Proxy (2)
- Generated entities -
### Proxy Runtime configuration - Global settings -

<table>
<thead>
<tr>
<th>Label</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProxyPort</td>
<td>schlacht_catalog_out</td>
</tr>
<tr>
<td>LogcounterPort</td>
<td>LPI</td>
</tr>
<tr>
<td>Description</td>
<td>Proxy runtime configuration of Proxy &quot;schlacht_catalog_out&quot;</td>
</tr>
</tbody>
</table>

#### General settings

- **Label**
  - HTTP Destination
  - HTTP Destination Prefix
  - URL
  - Ad of Bindings
  - http://www.w3.org/2000/09/w3c/tns/notifications/HTTP

#### Application-specific settings

<table>
<thead>
<tr>
<th>Label</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message ID</td>
<td>schlacht_catalog_out</td>
</tr>
<tr>
<td>Global Configuration</td>
<td>schlacht_catalog_out</td>
</tr>
</tbody>
</table>
Proxy Runtime configuration
- Operation-specific settings -

Summary

The Enterprise Service Architecture (ESA), SAP’s approach for building services-oriented business applications utilizes Web Services as vital part of its communication and integration strategy.

SAP’s rich business functionality, in conjunction with state-of-the-art technology, enables you to establish cross-company business processes as an integrated part of your development efforts based on Web Services.

SAP Web Application Server 6.40 offers an easy, convenient way to build Web services.
Questions?

Q&A

Feedback

Please complete your session evaluation and drop it in the box on your way out.

Thank You!

The SAP TechEd '03 Basel Team
Corporate Colors for PPT

You can use these colors with gradations