SAML for Web Services
Interoperability with .NET

SAP NetWeaver Product Management Security

June 2008
Agenda

1. Web Services Security – A short primer
   Interoperability and Standards
   WS-Security Basics

   Single Sign-On Technologies & Standards
   WS-Security SAML Token Profile

3. Web Services Single Sign-On with SAP NetWeaver
   SAML Token Profile support in current SAP NetWeaver releases
   Roadmap

4. Interoperability in Practice: Web Services SSO between SAP NetWeaver and Microsoft .NET
   Web Services Security in the Microsoft .NET Framework
   Implementation of an interoperable Web Service Consumer in .NET
   Configuration of an interoperable Web Service Provider in SAP NetWeaver
Definition of Interoperability

The IEEE defines interoperability as:

The ability of two or more systems or components to exchange information and to use the information that has been exchanged*

Interoperability – The Value Of Standards

Interoperability is achieved by standards, supports the seamless exchange and use of business information.

**Semantic Interoperability**
Applications understand and correctly use the information being exchanged.

**Technical Interoperability**
Systems are connected and can exchange information.
Web Service (WS) standards define the format of the message in transit to guarantee the interoperable exchange between service consumer and provider on a technical level.

Web Service Standards don’t specify any infrastructure- or application-specific aspects, such as:
- APIs or programming languages that applications must use to send or deliver messages – these are always platform-specific
- Runtime architecture and components
The SOAP protocol on its own does not provide any security mechanisms for:
- Message Integrity & Confidentiality
- Authentication
- Non Repudiation of origin or receipt
- But: SOAP can be extended to provide additional features

Up to the year 2002, best practice was to secure Web Services using Secure Sockets Layer (SSL)

But SSL provides transport – not application-level security
- SOAP Messages secure point-to-point, not end-to-end
- Messages stored unencrypted in files or databases at intermediaries
- not independent of underlying transport protocol

WS-Security submitted to standards body (OASIS) in Sept 2002 and approved as an OASIS Standard in April 2004
WS-Security Overview

- The OASIS WS-Security Standard extends a SOAP message by one or more **WS-Security Headers** (wsse:Security) which contains security information for each recipient.

- This new SOAP Header contains all relevant security metadata to secure a SOAP message, such as:
  - **Security Tokens** to carry security information (e.g. user authentication data, X.509 certificates)
  - A **Timestamp** to protect against Replay Attacks
  - **Signatures** to protect against message tampering*
  - **Encrypted Keys** and **Data** to protect confidential information

- Single Sign-On is provided by using e.g. SAML Security Tokens

* The act of altering something secretly or improperly
WS-Security is a rich framework to secure SOAP on the message layer

WS-Security provides a general-purpose mechanism for associating security tokens with message content

No specific type of security token is required. The specification is designed to be extensible, so as to support multiple security token formats
Agenda

1. Web Services Security – A short primer
   Interoperability and Standards
   WS-Security Basics

   Single Sign-On Technologies & Standards
   WS-Security SAML Token Profile

3. Web Services Single Sign-On with SAP NetWeaver
   SAML Token Profile support in current SAP NetWeaver releases
   Roadmap

4. Interoperability in Practice: Web Services SSO between SAP NetWeaver and Microsoft .NET
   Web Services Security in the Microsoft .NET Framework
   Implementation of an interoperable Web Service Consumer in .NET
   Configuration of an interoperable Web Service Provider in SAP NetWeaver
Definition of Single Sign-On (SSO)

Wikipedia defines Single Sign-On as:

Single Sign-On (SSO) is a method of access control that enables a user to authenticate once and gain access to the resources of multiple software systems*.

Basic Architectural Pattern for Single Sign-On (SSO)

- **Issuing Authority**: A system entity that issues security-related information about individual users. Usually this includes at least identity information about the user (e.g. a user name or E-Mail address)

- **Relying Party**: A system entity that decides to take an action based on the security information provided by the Issuing Authority. The Relying Party must have a trust relationship with the Issuing Authority

- **User**: A natural person who makes use of a system and its resources

1. User authenticates at Issuing Authority and request the security data that is required to access a protected resource at the Relying Party
2. Issuing Authority responds with the security information about the user
3. User authenticates with the issued data at the Relying Party to access a protected resource
4. Relying Party authenticates the user based on the security information issued by the Issuing Authority and sends response
Important Characteristics of Single Sign-On Technologies and Standards

- **Cross-Domain**
  Is it possible to use the SSO technology only within a security domain (i.e. the corporate Intranet) or can it be used across different domains (e.g. in a B2B scenario)?

- **Cross-Platform**
  Which platforms are supported by the SSO technology? Is it a widely adopted standard in the industry or a vendor-specific technology?

- **User Agent**
  Which type of user agent (e.g. Web Browser, Web Service Consumer, Mobile Clients) is supported by the SSO technology?
### Single Sign-On Technologies and Standards Supported by SAP

<table>
<thead>
<tr>
<th>Standard / Technology</th>
<th>Cross-Domain</th>
<th>Cross Platform</th>
<th>User Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Logon Ticket</td>
<td>No</td>
<td>Yes</td>
<td>• Web Browser</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Web Service Consumer</td>
</tr>
<tr>
<td>OASIS Security Assertion Markup Language (SAML)</td>
<td>Yes</td>
<td>Yes</td>
<td>• Web Browser</td>
</tr>
<tr>
<td>OASIS WS-Security SAML Token Profile</td>
<td>Yes</td>
<td>Yes</td>
<td>• Web Service Consumer</td>
</tr>
<tr>
<td>SPNego / Windows Integrated Authentication</td>
<td>No</td>
<td>No</td>
<td>• Web Browser</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Web Service Consumer</td>
</tr>
<tr>
<td>X.509/PKI</td>
<td>Yes</td>
<td>Yes</td>
<td>• Web Browser</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Web Service Consumer</td>
</tr>
</tbody>
</table>

For Web Services, **SAML** and its associated **token profile** for WS-Security is the most widely adopted standard in the industry.
Benefits of the Security Assertions Markup Language (SAML)

- **Interoperable security solution** to allow systems integration with great ease and minimal resources

- SAML is a protocol for **encoding security related information** (assertions) into XML and exchanging this information in a request/response fashion

- Provides **standard based mechanisms** to exchange security information using SOAP, HTTP(s)

- SAML is an **OASIS** standard
SAML Based Scenarios

Web Service based SSO
- SAML Token Profile
- Web Service Security Standard

Web Browser based SSO
- SAML Browser/Artifact Profile
- SAML Standard
Security Assertion Markup Language (SAML)

Building Blocks

- **Assertions**: statements about a subject. This could be an authentication, attribute information, or authorization permissions.

- **Protocols**: SAML defines request/response protocols for obtaining assertions.

- **Protocol Bindings**: defines how SAML protocols map to transport and messaging protocols, e.g. SAML SOAP Binding.

- **Profiles**: define how assertions, protocols, and bindings are combined for particular use cases.
A **SAML Assertion** can consist of

- **Authentication Statement**: Piece of data that represents an act of authentication performed on a subject (user) by the SAML Issuing Authority

- **Other Statements**: Attribute Statement, Authorization Decision Statement
Relationship Between WS Security SAML Token Profile and the SAML Standard

Profiles
- SAML Confirmation Methods

Bindings
- Assertions and Protocol
  - SAML Assertions

OASIS WS-Security
- SOAP Message Security
- Username Token Profile
- SAML Token Profile

references

© SAP 2008 Page 18
The **SAML Token Profile** defines the use of **SAML Assertions** as **Security Tokens** in the **WS-Security Header**.

The SAML Token is used by the service provider to authenticate the user based on the identity information in the SAML Assertion in incoming requests from service consumers.
The Token Issuer or **Security Token Service (STS)** is a distinguished Web service that **issues, exchanges and validates** security tokens.

The STS has broad applicability in that it can be used to **issue** security tokens in a **wide range of formats** (e.g. Certificates, SAML assertions).

Basic operations supported by an STS:
- **Issue** a new token
- **Renew** token
- **Validate** a token
- **Cancel** a token

Role of the Token Issuer (aka Security Token Service, STS) in Web Services SSO

1. Web Service Consumer
2. **Authenticate user**
   - **Generate requested Token**
3. Security Token Request
4. **Security Token**
5. Security Token Response
Web Services SSO with SAML – General Message Exchange

1. Web Service (WS) Consumer authenticates at the Token Issuer (Security Token Service, STS) and requests a SAML Token

2. Token Issuer authenticates the User and issues a SAML Token to the WS Consumer

3. WS Consumer uses the SAML Token for authentication at the WS Provider

4. WS Provider must trust the assertion in the SAML Token to authenticate the WS Consumer and sends back the response

The SAML Token profile addresses **two major questions**:

- How can the SAML Token be bound to the SOAP message so that the service provider can be sure that they belong together?
- How can the service provider be sure that the sender of the message is really the subject in the assertion?
**Sender-Vouches (SV) Subject Confirmation Method**

- The WS Consumer cryptographically binds the assertion to the body of the SOAP message by signing both with its **private key**
  - The WS Provider compares the identity information from the message signature with the subject information in the assertion

**Holder-of-Key (HoK) Subject Confirmation Method**

- The **assertion holds a key** that is used by the WS Consumer to cryptographically bind (sign) the assertion and the body of the SOAP message
  - The WS Provider uses the same key to verify the signature. The subject in the assertion is the party that can demonstrate that it is the holder of the key.
1. User **authenticates** at the Token Issuer (STS) and requests a SAML Token with the WS-Trust protocol

2. Token Issuer authenticates the User and **issues** a SAML Token in the response to the WS Consumer with the WS-Trust protocol

3. WS Consumer uses its private key to **create a signature** over the SAML Token and the message body

4. To **confirm** the WS Consumer identity, WS Provider verifies the signature and compares the identity information in the SAML Token with the identity information of the WS Consumer’s Public Key certificate

**Prerequisites:**
- Pre-established trust relationship between WS Provider and WS Consumer
- WS-Consumer must possess a signature key pair
1. User **authenticates** at the Token Issuer (STS) and requests a SAML Token with the WS-Trust protocol. Optionally, the user provides key material to the Issuer for the short-lived key.

2. The Token Issuer **generates** the short-lived symmetric key, encrypts it with the WS Provider’s public key. The key is added to the SAML Assertion which is then signed by the Token Issuer with its signature key.

3. The Token Issuer **issues** the SAML Assertion as a SAML Token in the WS-Trust response message to the WS Consumer, along with its key material used to generate the symmetric short-lived key.
4. The WS Consumer also **generates** the short-lived symmetric key based on both parties key material.

5. The WS Consumer **signs** the SAML Token and the message body with the previously generated short-lived symmetric key and sends a request to the WS Provider.

6. The WS Provider **verifies** the Token Issuers signature in the SAML Token and decrypts the short-lived symmetric key contained in the SAML Token using its private key.

7. The WS Provider verifies the WS Consumer‘s (i.e. the key holder‘s) signature by using the decrypted short-lived symmetric key. The Token Issuer **confirmed** that the holder of the key is the subject in the assertion.

**Prerequisites:**

- Pre-established trust relationship between WS Provider and Token Issuer.

---

**Diagram:**

- **Web Service Consumer**
- **Web Service Provider**
- **Token Issuer (STS)**
- **SAML Token (HoK)**
- **Token Issuer Public Key Certificate**
- **Trust Relationship**
- **Short-lived Key**
The **WS-Security header** contains the following authentication information:

The user with the identifier **TechEd08\stefanie** has been successfully authenticated at

7:35 pm on Sept. 9th, 2008 using her **password**.

The issuer **TechEdAuthority** confirms that the subject of the assertion is the party that signed the message.
How are the Common SSO Issues Addressed by WS-Security and the SAML Token Profile?

- **Syntax and Semantics** of the security information
  - How is the security information serialized on the wire?
  - What is the syntax and semantics of this serialized security information about an end user (identity)?
    - Name, User ID
    - Roles
    - Groups
    -...

- **Confirmation** of the user’s identity
  - How can the Relying Party be sure that the request or message sent by the user is associated with the identity data in the issued security information?

- **Protocol** to transfer the security information between the parties
  - How does a user request the security information from the Issuing Authority?
  - How does a user transfer this security information to the Relying Party?

---

© SAP 2008 Page 27
Summary

- **SAML** is a proven and widely adopted standard in the industry for interoperable, cross-domain SSO

- The **SAML Token Profile** specifies how SAML is used to support SSO for Web Services

- SAML Token Profile defines two subject **confirmation methods**: Sender-Vouches and Holder-of-Key

- Both confirmation methods differ mainly in **trust relationship setup** and **key management**
Agenda

1. Web Services Security – A short primer
   Interoperability and Standards
   WS-Security Basics

   Single Sign-On Technologies & Standards
   WS-Security SAML Token Profile

3. Web Services Single Sign-On with SAP NetWeaver
   SAML Token Profile support in current SAP NetWeaver releases
   Roadmap

4. Interoperability in Practice: Web Services SSO between SAP NetWeaver and Microsoft .NET
   Web Services Security in the Microsoft .NET Framework
   Implementation of an interoperable Web Service Consumer in .NET
   Configuration of an interoperable Web Service Provider in SAP NetWeaver
SAML 1.1 Sender Vouches (WS-Consumer, WS-Provider)

- Support for signed and unsigned Sender-Vouches SAML assertions
- Only for **authentication** purposes, i.e. no attribute-based authorization
- For WS-Consumer, only a local token issuer is supported (i.e. no external STS)

Supported in
- SAP NetWeaver 7.0 >= SP14 (ABAP)
- SAP NetWeaver 7.1 (Java, ABAP)
SAML Token Profile Support in SAP NetWeaver Support in Current Releases 7.0 and 7.1 (2/3)

- **SAML 1.1 Holder of Key (HOK) (WS-Provider)**

- Only support for **symmetric keys** in SAML 1.1 Holder of Key Tokens (i.e. no support for asymmetric keys)

- Optional **user mapping** from external (non-SAP platform) username to SAP username in ABAP table **USREXTID**

---

**Planned** for SAP EHP1*, SP2 for SAP NetWeaver 7.1:
- SAP NetWeaver 7.01 SP2 (ABAP) (Oct 2008)
- SAP NetWeaver 7.11 SP1 (Java, ABAP) (Dec 2008)

* Enhancement Package
SAML 1.1 Holder of Key (HOK) (WS-Consumer)

- SAP NetWeaver Web Service Consumers can request a SAML Holder-of-Key Token from an external Token Issuer (STS)

**Planned** for SAP EHP2* for SAP NetWeaver 7.1
- SAP NetWeaver 7.02 (ABAP) (Q3 2009)
- SAP NetWeaver 7.12 (Java, ABAP) (Q3 2009)

* Enhancement Package
Agenda

1. Web Services Security – A short primer
   - Interoperability and Standards
   - WS-Security Basics

   - Single Sign-On Technologies & Standards
   - WS-Security SAML Token Profile

3. Web Services Single Sign-On with SAP NetWeaver
   - SAML Token Profile support in current SAP NetWeaver releases
   - Roadmap

4. Interoperability in Practice: Web Services SSO between SAP NetWeaver and Microsoft .NET
   - Web Services Security in the Microsoft .NET Framework
   - Implementation of an interoperable Web Service Consumer in .NET
   - Configuration of an interoperable Web Service Provider in SAP NetWeaver
WS-Security Support in Microsoft .NET

- **.NET 2.0** supports core Web Service standards, such as WSDL 1.1 and SOAP 1.1/1.2

- **Web Services Enhancements (WSE)** for Microsoft .NET 2.0 is a supported add-on to Microsoft Visual Studio .NET and the Microsoft .NET 2.0 Framework providing support for WS-Security and other advanced Web Service protocols

- With **.NET 3.0**, these advanced Web Service protocols became an integral part of the .NET Framework, which is now called the **Windows Communication Foundation (WCF)**, formerly known as 'Indigo')
The WCF programming model **unifies** the existing communication technologies for distributed computing in .NET 2.0 (e.g. Web Services/WSE, .NET Remoting, Distributed Transactions, Message Queues) into a **single** Service-oriented programming model.

A Service in WCF is composed of three parts:
- a Service Class (e.g. written in C#) that implements the Service
- a Service Host Environment to host the Service (e.g. IIS, Self-Hosting)
- one or more Endpoints to which Clients can connect

An **Endpoint** defines the **Contract** (What ?), the **Address** (Where ?) and the **Binding** (How ?) of a Service.
.NET WCF Bindings

- Bindings define the **Transport** (e.g. HTTP), **Encoding** (e.g. Text Message) and **Protocol** (e.g. Web Services) required to communicate with the Service.

- WCF is shipped with predefined, **System-Provided Bindings**. These bindings can be configured declaratively.

- Developers can also create their own **Custom Bindings** with the WCF API that provide full control over the messaging stack when one of the system-provided bindings does not meet the requirements of a consumer or provider.

  **WCF Binding**
  
  ![WCF Binding Diagram]

  **Transport**
  - TCP
  - HTTP
  - MSMQ
  - ...

  **Encoders**
  - Text
  - Binary
  - ...

  **Protocol**
  - Security
  - Reliability
  - .NET
  - ...

© SAP 2008 Page 36
Web Services Security Support in .NET WCF System-Provided Bindings

- Support* for Web Services Core- and WS-Security Standards in WCF System-Provided Bindings:

<table>
<thead>
<tr>
<th>Interoperability</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>basicHTTPBinding</td>
<td>1.1 1.1 1.0</td>
</tr>
<tr>
<td>wsHttpBinding</td>
<td>1.2 1.1 1.1</td>
</tr>
<tr>
<td>wsFederationHttpBinding</td>
<td>1.2 1.1 1.1</td>
</tr>
</tbody>
</table>

- Support for Web Services Core- and WS-Security Standards in SAP NetWeaver

<table>
<thead>
<tr>
<th>SOAP</th>
<th>WSDL</th>
<th>WS-Security</th>
<th>Security Token Profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>1.1</td>
<td>1.0</td>
<td>Username, X.509, SAML 1.0</td>
</tr>
</tbody>
</table>

SAML Token Profile Support in .NET WCF

The system-provided `wsFederationHttpBinding` in WCF uses the SAML Token Profile for Web Services SSO scenarios. This binding has the following constraints:

- SAML 1.1 Assertions with `Holder-of-Key` subject confirmation method

- **Web Services Security 1.1** and **SOAP 1.2**

WCF has no system-provided binding that matches the supported standards in SAP NetWeaver (SAML Token Profile 1.0, WS-Security 1.0, SOAP 1.1)

With the WCF APIs, developers can create SAML 1.1 Assertions with Sender-Vouches Confirmation Method

```csharp
```

However, WCF does not support the **STR-Transform** algorithm which is required to sign SAML Tokens with Sender-Vouches confirmation method

To implement an interoperability scenario between SAP NetWeaver and .NET, a **WCF custom binding** is required to support the use of SAML Token Profile 1.0, WS-Security 1.0 and SOAP 1.1 on the WS Consumer side. In addition, Holder-of-Key support in SAP NetWeaver is a prerequisite.
Interoperability Scenario between SAP NetWeaver and Microsoft .NET WCF

1. The user logs on to the Windows domain with his Windows Credentials
2. The WS Consumer authenticates at the Token Issuer with Windows Integrated Authentication and requests a SAML HoK Token that contains the domain identity
3. The Token Issuer issues the SAML Token
4. The WS Consumer sends the request using the Custom Binding (WS-Sec 1.0, SOAP 1.1, SAML Token Profile 1.0)
5. The WS Provider maps the Windows User identity to the ABAP User identity
6. WS Provider sends response
Configuration Steps for the Interoperability Scenario with SAML Token Profile (HoK)

- Implement the **Custom Binding** for .NET/WCF
- Configure an endpoint of the Web Service Provider to support **symmetric key encryption/signature** using SAML-bases message authentication
- Maintain **mapping** of external user id (e.g. Windows Domain Name) to internal SAP user id

- **.NET/WCF Code**
- **SE38 - RSUSREXTID**
- **SOAMANAGER**
SymmetricSecurityBindingElement secBinding = new SymmetricSecurityBindingElement();
secBinding.SecurityHeaderLayout = SecurityHeaderLayout.Strict;
secBinding.IncludeTimestamp = true;
secBinding.SetKeyDerivation(false);
// set WSS 1.0 and SOAP 1.1
// don't encrypt signature
secBinding.MessageProtectionOrder = MessageProtectionOrder.SignBeforeEncrypt;

... using WS-Security 1.0 ...

... based on this algorithm suite ...

... without encrypting the signature in the WS-Security header.

... using WS-Security 1.0 ...

... based on this algorithm suite ...

Use a symmetric key to protect the message ...

... without encrypting the signature in the WS-Security header.
... IssuedSecurityTokenParameters itp = new IssuedSecurityTokenParameters();
itp.ReferenceStyle = SecurityTokenReferenceStyle.Internal;

itp.InclusionMode = SecurityTokenInclusionMode.AlwaysToRecipient;

itp.TokenType = "urn:oasis:names:tc:SAML:1.1:assertion";

itp.KeyType = SecurityKeyType.SymmetricKey;

itp.KeySize = 128;

itp.RequireDerivedKeys = false;

itp.IssuerAddress = new EndpointAddress(new Uri("http://localhost:8000/samlsts"));

// set sts binding

itp.IssuerBinding = new WSHttpBinding("stsBinding");

secBinding.ProtectionTokenParameters = itp;

Get an issued token to protect the message ...

... using the SAML Token Profile 1.0 with a SAML 1.1 assertion ...

... that holds the symmetric key to protect the message ...

... from the Token Issuer (STS) with this URL.
Configuring SAML Holder-of-Key Token for the ABAP Web Service Provider (1/3)

1. Invoke Transaction SOAMANAGER

2. Switch to the tab Application and Scenario Configuration and select Single Service Administration
3. **Search** for the service, select it in the search results list and click on **Apply Selection**.

4. Click on **Create Service** to create a new service or select an existing entry and click on **Edit**.
For SAML Holder-of-Key Authentication, select Symmetric Message Signature/Encryption in Transport Guarantee → Communication Security

Single Sign-On using SAML in Authentication Settings → Authentication Method → Message Authentication

Click on Save
Configuring the SAML User Mapping in the AS ABAP (1/4)

1. Start the ABAP Editor with transaction SE38

2. Enter RSUSREXTID and click on Execute (F8)
Enter the **SAP user name** and select **SA** for the **External ID type**. Optionally, enter the **prefix** (e.g. Token Issuer/STS name + "::" + Windows Domain Name) and/or **suffix** that is present in the external name. In addition, enter the DN of the Token Issuer's (STS) certificate.

Save the new mapping with **Execute (F8)** and review the changes made in the mapping table.
5. Display the current SAML user mappings with the Data Browser (SE16)

6. Enter VUSREXTID for the Table Name and press F7

7. Select SA for the External ID type and press Enter
8 Display the external SAML Mapping

Display View "Assignment of External ID to Users": Overview

<table>
<thead>
<tr>
<th>External ID type</th>
<th>SAML NameIdentifier (PartnerID:NameQualifier:Name)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Assignment of External ID to Users

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>urn:wcf.SAMLSTS::SAP_ALLD023939</td>
<td>D023939</td>
<td>✓</td>
</tr>
<tr>
<td>urn:wcf.SAMLSTS::SAP_ALLD039113</td>
<td>D039113</td>
<td>✓</td>
</tr>
<tr>
<td>urn:wcf.SAMLSTS::SAP_ALLD044724</td>
<td>D044724</td>
<td>✓</td>
</tr>
</tbody>
</table>

Position... Entry 23 of 32
SAML Token Issued by the .NET WCF Token Issuer and Used for Authentication at SAP

<User Identity Information from Windows Integrated Authentication>

[User Identity Information]

[Subject Confirmation Method (HoK)]

[Encrypted symmetric key]

[Token Issuer signature]
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

SAP, R/3, xApps, xApp, SAP NetWeaver, Duet, SAP Business ByDesign, ByDesign, PartnerEdge and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned and associated logos displayed are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG. This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP on any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice. SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Einige von der SAP AG und deren Vertriebspartnern vertriebene Softwareprodukte können Softwarekomponenten umfassen, die Eigentum anderer Softwarehersteller sind.

SAP, R/3, xApps, xApp, SAP NetWeaver, Duet, SAP Business ByDesign, ByDesign, PartnerEdge und andere in diesem Dokument erwähnte SAP-Produkte und Services sowie die dazugehörigen Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und in mehreren anderen Ländern weltweit. Alle anderen in diesem Dokument erwähnten Namen von Produkten und Services sowie die damit verbundenen Firmenlogos sind Marken der jeweiligen Unternehmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länderspezifische Unterschiede aufweisen.


Die gesetzliche Haftung bei Personenschäden oder Produkthaftung bleibt unberührt. Die Informationen, auf die Sie möglicherweise über die in diesem Material enthaltenen Hotlinks zugreifen, unterliegen nicht dem Einfluss von SAP, und SAP unterstützt nicht die Nutzung von Internetseiten Dritter durch Sie und gibt keinerlei Gewährleistungen oder Zusagen über Internetseiten Dritter ab.

Alle Rechte vorbehalten.