

Interoperability of the IBM WebSphere Message Broker and the SAP NetWeaver Process Integration



Applies to:

IBM WebSphere Message Broker, SAP NetWeaver Process Integration, IBM WebSphere Service Registry and Repository, SAP NetWeaver Enterprise Services Repository and Services Registry. For more information, visit the [Service Bus-based Integration homepage](#).

Summary

In larger enterprises, system landscapes comprise quite commonly SOA products from different vendors, in particular different service registries and Enterprise Service Buses (ESBs). When developing business scenarios that run across the entire enterprise, it is essential that the different registries and ESBs interoperate seamlessly. In this article, we want to provide a comprehensive overview of options to integrate the IBM WebSphere Message Broker and the SAP NetWeaver Process Integration as well as the IBM WebSphere Registry and Repository (IBM WSRR) and the SAP Services Registry (SAP SR).

Authors: Peter Kulka (SAP AG), Martin Herzog (IBM)

Company: SAP AG / IBM

Created on: 5 October 2010

Author Bio



Peter Kulka received his PhD in Computer Science / Computer Graphics from the University of Auckland, New Zealand. After teaching Java / C++ programming courses at the University of Auckland, Peter joined SAP AG in 1999. Until November 2005 Peter was responsible for the product definition of the SAP NetWeaver Application Server, SAP's Java EE 5 compliant application server. Currently Peter is a Solution Architect in SAP's Global Ecosystem and Partner Group, where he advises SAP partners on how to architect their Java solutions based on SAP NetWeaver. Peter presented at leading industry conferences such as JavaOne, SAP TechEd, and OOP.



As an IT Architect within the IBM SAP International Competence Center (ISICC) in Walldorf/Germany Martin Herzog (mherz@de.ibm.com) is responsible for the WebSphere brand. Being with IBM for more than twelve years Martin started his career in Voice Technology development. In the ISICC Martin now focuses on interoperability topics between the IBM WebSphere product portfolio and SAP's NetWeaver stack.

Table of Contents

Introduction	3
Typical SOA Landscapes with both ESBs	3
Potential Interoperability Options	4
Service Registry Interoperability	5
Registry Synchronization	5
Functionality of the IBM WSRR-UDDI Synchronization Module.....	5
Quality of the synchronization	6
The IBM WSRR-UDDI Synchronization Module in development, test and productive environments.....	6
Lifecycle states of SAP Enterprise Services	7
ESB Interoperability	8
Business Semantics.....	9
Discussion and Recommendation	10
Service Discovery	10
Endpoint Look-up	10
Service Development / Deployment.....	11
Service Modeling.....	11
Conclusion	12
Related Content	13
Copyright.....	14

Introduction

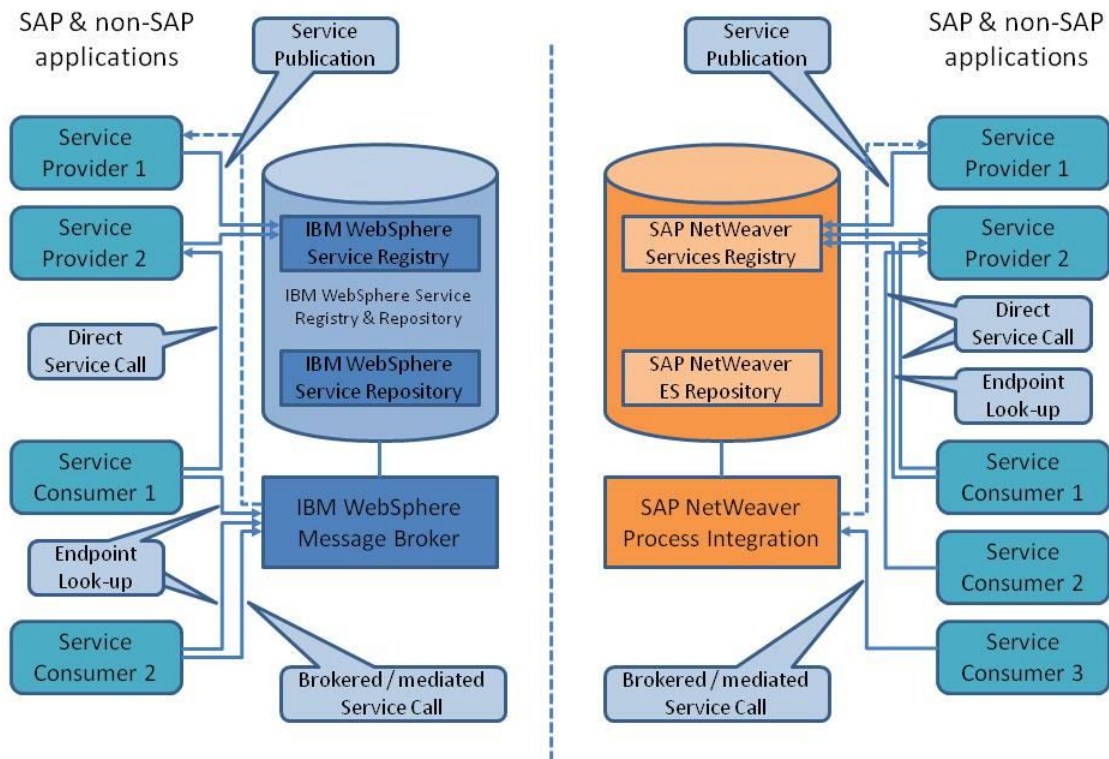
SOA-based enterprise applications and integration scenarios are not just marketing anymore; they are reality. It is quite common in larger enterprises that different lines of business started local SOA projects and implementations independently using SOA products from different vendors, in particular different repository and registry facilities and Enterprise Service Buses (ESBs). A consolidation of these products into one central SOA infrastructure at a later point of time is in many cases not feasible due to the already developed integration content and skill sets. It may also be the case that no single SOA product can provide all required features and functions to serve all use cases. There is also an organizational and political dimension, which makes a consolidation quite difficult. As a result, many IT departments have accepted to run heterogeneous SOA landscapes.

When developing business scenarios that run across the entire enterprise, it is essential that the different registries and ESBs interoperate seamlessly. Since the topic of heterogeneous SOA landscapes is certainly too complex to deal with in one single document we focus here on the interoperability of two ESBs: IBM WebSphere Message Broker (IBM WMB) and the SAP NetWeaver Process Integration (SAP NetWeaver PI), including the corresponding registries and repositories, the IBM WebSphere Service Registry and Repository (IBM WSRR) and the SAP NetWeaver Enterprise Services Repository and Services Registry (SAP NetWeaver ES Repository and SR).

A general introduction to SAP's SOA products can be found on the SAP Community Network (<http://www.sdn.sap.com/irj/sdn/nw-soa>). A general description of typical SOA Middleware components and their mapping to SAP products can be found [here](#). The following Web page contains a description of the equivalent IBM SOA philosophy and its products: <http://www-01.ibm.com/software/solutions/soa/>.

Typical SOA Landscapes with both ESBs

As a starting point, we assume that there are two separate SOA landscapes, where the IBM WebSphere Message Broker and the SAP NetWeaver Process Integration do not interoperate. The figure below shows this situation:



Typical SOA landscape without ESB interoperability

The left hand side shows the IBM SOA landscape. The service providers 1 and 2 publish their services using IBM's API for service publication. Both, consumers 1 and 2 look-up the endpoints of the services in the IBM WebSphere Message Broker first and they are then able to make use of the endpoint caching provided by the IBM WebSphere Message Broker / IBM WSSR integration in subsequent calls.

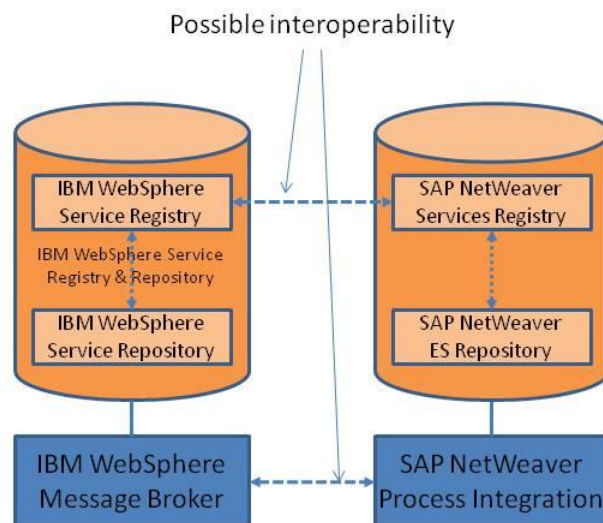
The right hand side of the figure displays a typical SAP SOA landscape. Service providers publish their ABAP or Java services in the SAP NetWeaver SR. The SAP tools to publish services, which are written in ABAP or Java, use an SAP API. Since the SAP NetWeaver SR supports UDDI, it is additionally possible to use UDDI for service publication.

On the SAP side, service endpoints can be looked-up either dynamically or during the configuration. Consumer 1 on the right hand side of the figure above, for example, determines the service endpoint dynamically and then calls the service. The SAP SR additionally provides the possibility to arrange services in Service Groups, where all services that belong to a Service Group are configured together. In this case the endpoints are looked-up only once and then they are cached for performance reasons. Subsequent service calls can then retrieve the endpoints directly from the cache. More information about Service Groups can be found [here](#).

The service consumer 2 in the figure above receives the endpoint during the configuration of the consumer proxy, i.e. the endpoint look-ups are not dynamically done during run time. It is quite common to configure the endpoints of synchronous services in the consumer proxies statically. Finally it is of course also possible to perform brokered or mediated service invocations, as shown by consumer 3 in the figure above.

Potential Interoperability Options

The figure below shows the possible interoperability options for the IBM WMB, IBM WSRR, SAP NetWeaver PI and SAP NetWeaver SR:



The interoperability options for service registries and the two ESBs are described in detail in the following sections.

Note that the interoperability of the SAP NetWeaver ES Repository and IBM WSRR is directly not possible at the repository level, since the repositories contain very different data, yet it is possible at the registry level. The SAP NetWeaver ES Repository contains the business semantics for Services, including SAP and custom built Enterprise Services and their related SOA assets, such as service definitions and data types. The IBM WSRR, however, contains metadata for SOA governance, though it can also support storing BPEL documents and other Meta data through OWL-based user-defined semantics classification.

Service Registry Interoperability

The SAP NetWeaver SR consists of a UDDI v3.02 compliant registry, a Classification service and a Free text search. Since the SAP NetWeaver SR supports UDDI, it is possible to replace the UDDI part of the SAP NetWeaver SR by a 3rd party UDDI compliant registry. In this case all published WSDL files are stored in the 3rd party registry and the services are looked-up in the 3rd party registry. The UDDI part of the SAP NetWeaver SR is not used anymore. Detailed information, how to replace the UDDI part of the SAP NetWeaver SR by the HP Systinet Service Registry, can be found in the blog "[How to Set Up an Interoperable Scenario with SAP NetWeaver Services Registry and HP SOA Governance Systinet Registry Foundation](#)".

As IBM WSRR is both a service registry and repository, IBM has adopted a different implementation from UDDI, since the current UDDI specification does not provide a standard repository capable of storing artifacts, nor governance capabilities for managing the end-to-end life cycle of the various types of artifacts related to services. The IBM WSRR does have a synchronization module that enables it to synchronize data with UDDI v3 compliant service registries.

Registry Synchronization

Applications that interact with the IBM WSRR use IBM's APIs for the service publication and service look-up, whereas SAP tools for service publication use SAP's APIs for this purpose. This means that SAP applications can publish SAP Enterprise Services and look-up endpoints only in the SAP NetWeaver SR. Non-SAP applications can publish their services and look-up endpoints either in the SAP NetWeaver SR or in the IBM WSRR.

Since the service publication and endpoint look-up depends on the usage of the particular API, the only way to achieve a full interoperability, i.e. every application (SAP or non-SAP) can look-up every service (Web Service or SAP Enterprise Service), is a synchronization of the registry content of the SAP NetWeaver SR and the IBM WSRR. This synchronization can be achieved via a synchronization module provided by IBM. IBM's WSRR-UDDI Synchronization Module requires a service registry, which is UDDI v3 compliant. The SAP NetWeaver SR that comes with SAP NetWeaver CE and SAP NetWeaver PI 7.1 supports UDDI v3.02. More information about the IBM WSRR can be found [here](#) for the version 7.0, [here](#) for the version 6.2 and [here](#) for the version 6.3.

Functionality of the IBM WSRR-UDDI Synchronization Module

The IBM WSRR-UDDI Synchronization Module can exchange the following registry objects:

- **WSDL files, Concepts, XSD documents:** Bidirectional exchange of WSDL files including secured WSDL files is possible. Also concepts, XSD documents and all other WSRR artifacts are pushed to the SAP NetWeaver SR in the synchronization process.
- **Web Service Classifications:** The exchange of classifications from the SAP NetWeaver SR to the IBM WSRR is done automatically. The other direction, i.e. the exchange of classifications from the IBM WSRR to the SAP NetWeaver SR, requires the manual creation of a mapping file that maps between IBM WSRR classifications to the ones created in the SAP NetWeaver SR. The Web Ontology Language (OWL) is a W3C standardized XML based document format for describing Web ontologies. The IBM WSRR exploits only the taxonomy part of the OWL language to describe classifications.
- **Web Service Artifacts / Documents:** Any Web Service artifacts or documents that are compliant with the standard [Technical Note 2.0.2](#) can be bidirectional exchanged between both registries. Support is also provided for the mapping of policy expressions, and policy attachments, that comply with the [Web Services Policy 1.2](#) or [Web Services Policy 1.5](#) frameworks.

Filters for these objects can be defined, which are applied during the synchronization.

Different classifications of the IBM WSRR and the SAP NetWeaver SR can be mapped during synchronization. To control the synchronization of classifications, the IBM WSRR uses an OWL file plus a mapping file that can be edited to meet individual requirements. The editing of the mapping file has to be done only once, thus it can be seen as an initial configuration step.

Quality of the synchronization

The synchronization of registry content can be initiated either scheduled or it can be called directly to run immediately. Initial data synchronization as well as delta synchronization runs are possible. Note that existing WSDL documents in the IBM WSRR will not be synchronized with the SAP NetWeaver SR until an update occurs on them.

When the synchronization process stops in the middle of the synchronization, any uncompleted synchronization steps will be rolled back and only complete synchronization steps will be incorporated into the IBM WSRR. When the systems are ready for synchronization again the IBM WSRR-UDDI Synchronization Module will automatically check for objects that have not yet been synchronized and will proceed with this data.

When the content of the registries, for whatever reason, is not consistent anymore, the IBM WSRR-UDDI Synchronization Module is designed to resynchronize everything that it has been configured for. It will automatically detect data that is inconsistent and will resynchronize it by checking what has already been synchronized as WSRR keeps a record of what entities in WSRR match to which UDDI entity keys.

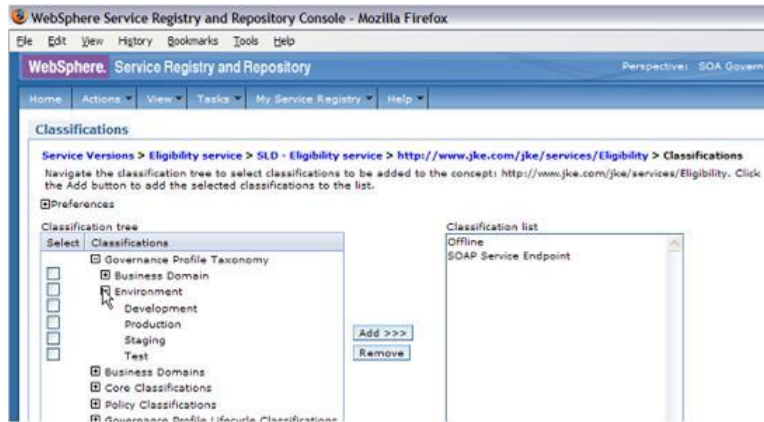
The IBM WSRR-UDDI Synchronization Module in development, test and productive environments

Through the lifecycle SAP typically uses three different system landscapes for development, testing and production. Any development object, e.g. coding or configuration, is created and maintained in the development landscape. When the development of an object has finished, it is transported to the test landscape for testing. After a series of successful tests the objects are then transported into the production landscape. Corrections are performed only in the development landscape and transported again to the test landscape.

Web Services or SAP Enterprise Services consist of several artifacts, which may be spread over different systems or components. This includes, for example, the storage of service definitions in the SAP NetWeaver ES Repository, while the corresponding service implementations are typically deployed in the SAP backend systems. The runtime information of a service, i.e. a WSDL file containing the endpoints, can be discovered from the SAP NetWeaver SR.

When developing Enterprise Services or Web Services, only the design time artifacts, i.e. service definition and its implementations, are transported. The content of the SAP NetWeaver SR is not transported. In the target system landscape there are batch jobs running, which will publish new or changed services to the corresponding SAP NetWeaver SR. By default, the batch jobs run once a day, but other time intervals can be configured. The automatic publication through batch jobs is available for the SAP NetWeaver Application Server ABAP 7.0 EHP 1.

In case of multiple SAP system landscapes, e.g. for development, test and production, every SAP NetWeaver SR in the SAP system landscapes synchronizes with one single IBM WSRR, which acts in this case as a governance master. The IBM WSRR creates a WSDL file for every SAP NetWeaver SR. These WSDL files would be correlated together by the governance master using Correlators, such that a common object would link all WSDL files. The endpoints could then be marked by the environment as part of the governance process and eventually promoted out to the runtime IBM WSRRs as appropriate (see the screenshot below).



The synchronization with each SAP NetWeaver SR (can be more than one at the same time) is being done by the IBM WSRR-UDDI Synchronization Module with a push-and-pull model. Notification events are generated in the registry of the IBM WSRR when documents are created, updated or deleted. The synchronization model repeatedly polls the IBM WSRR registry and processes these events by conducting the appropriate operations in the SAP NetWeaver SR. In the same way the IBM WSRR-UDDI Synchronization Module carries out repeated polls of the SAP Service Registry, which lead to corresponding changes in the IBM WSRR Registry.

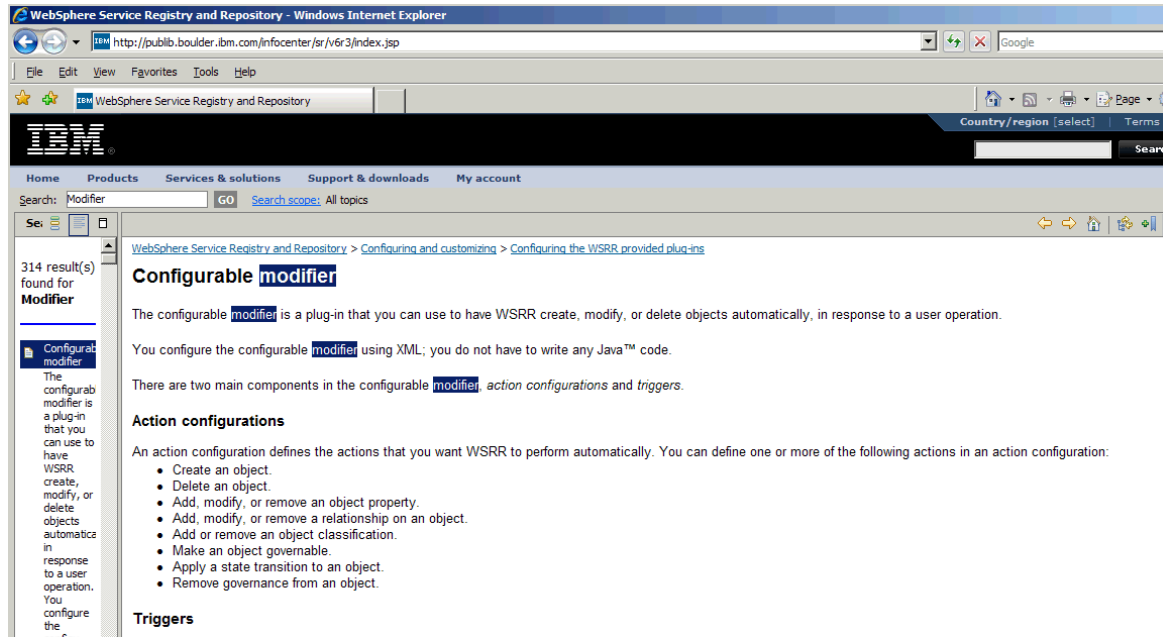
Lifecycle states of SAP Enterprise Services

There are two independent classifications systems available to reflect the current lifecycle states of services: One for the entire service and one for the interface only. The classification for the entire service consists of the following states

- **“Modeled”** – the service interface is modeled in the SAP ESR
- **“Activated/Deployed”** – the corresponding service implementation is available and deployed in a backend
- **“Configured”** – the provider proxy is configured. I.e. this service is ready to be consumed.

While the lifecycle states for the entire service are set automatically, the state for the service interface (called “release state”) needs to be set manually. This is typically done in the ABAP Development Workbench (transaction “SE80”) in the ABAP backend systems or in the SAP NetWeaver ES Repository. Note that if the interface state changes from “Deprecated” to “Revoked” the corresponding services are not automatically removed from the SAP NetWeaver SR. Here another manual step is necessary to change the provider proxy in the backend system in order to un-publish a revoked service.

When the lifecycle status of a service is changed in one service registry, this change is also synchronized, so that the corresponding service is updated accordingly in the other registry. In the IBM WSRR, the updated services are marked so that changed status can be identified. In order to get the correct lifecycle status assigned automatically for a modified service in the IBM WSRR, it is necessary to either write (very little) Java code (IBM WSRR 6.2) or to use the custom modifier functionality of the IBM WSRR 6.3 (see the screenshot below). However, this process works only unidirectionally from the SAP NetWeaver SR to the IBM WSRR.



The SAP NetWeaver SR contains a tab called “Used by” to show the existing services of Service Groups. Service Groups can be used by composite applications in the SAP NetWeaver CE to simplify the configuration of multiple services or service consumers. Just to avoid any misunderstanding: this tab cannot be used to maintain subscription lists or service dependencies.

ESB Interoperability

The interoperability of the IBM WMB and the SAP NetWeaver PI is relevant only for brokered or mediated service calls, since direct service calls do not require any ESB functionalities, such as mapping or routing. However, it may make sense to route service calls through an ESB for monitoring purposes.

Web Services or SAP Enterprise Services are typically consumed using Simple Object Access Protocol (SOAP) messages. Since Web Services or Enterprise Services (SAP or custom built) can be used for the communication between the SAP NetWeaver PI and IBM WMB, SOAP messages need to be exchanged between the two buses. This means that a communication protocol must be used that is capable of transporting SOAP messages and that is supported by both ESBs. The following transport protocols are very common:

- **File system:** Today data exchange via the file system is mainly used for batch transfer of bulk messages. The data transfer happens asynchronously. One drawback of this type of data exchange is the limited monitoring capability in particular during the time period after one process has written the data into the file system and before another process picks-up that data, though this has achieved to a certain extent by Managed File Transfer software. For instance, the integrated monitoring can be implemented in the ESB framework through WMB and WebSphere MQ File Transfer Edition software.
- **SOAP over Hypertext Transfer Protocol (HTTP):** In case of online or real time communication, SOAP over HTTP is the preferred protocol for synchronous communication.
- **SOAP over Java Message Service (JMS):** SOAP over JMS is the common protocol for asynchronous communication and it is used if guaranteed delivery of message is required. Due to the asynchronous nature of this protocol, only near real time communication is possible and a delay during the data transfer may happen.
- **Web Service Reliable Messaging (WS-RM):** For asynchronous communication WS-RM is the preferred protocol, however it must be carefully evaluated, if and which scenarios are supported by the involved communication parties.

The choice of the most suitable protocol depends whether synchronous or asynchronous communication is required. As stated above, in case of synchronous communication SOAP over HTTP is the preferred protocol, whereas in case of asynchronous communication SOAP over JMS is commonly used. However, for asynchronous communication WS-RM should also be considered. If the required scenarios are supported by the communication parties, i.e. the two ESBs, then WS-RM should be used.

Synchronous communication protocols are often regarded as unreliable and should only be used for information retrieval operations. Asynchronous protocols are generally recommended due to the secure and reliable nature.

The IBM Redbook "[Connect WebSphere Service-Oriented Middleware to SAP](#)" contains very detailed information about the service-oriented communication between IBM middleware products and the SAP NetWeaver PI. Since this Redbook is quite comprehensive (nearly 1500 pages), the following chapters of the book seem to be most relevant in this context:

Synchronous communication via HTTP		
Scenario	IBM WMB consumes services provided by SAP NetWeaver PI	Chapter 4.2.13 and 5.5
	IBM WMB provides services that are consumed by the SAP NetWeaver PI	Chapter 4.2.16 and 5.6
Asynchronous communication via JMS		
Scenario	IBM WMB consumes services provided by SAP NetWeaver PI	Chapter 4.2.23, 5.7 (SAP configuration) and 8.4 (IBM configuration)
	IBM WMB provides services that are consumed by the SAP NetWeaver PI	Chapter 4.2.24, 5.8 (SAP configuration) and 8.5 (IBM configuration)

Note: Web Services and Enterprise Services (SAP or custom built) can be synchronous or asynchronous. This should not be an issue unless synchronous Web Services or Enterprise Services (SAP or custom built) are consumed via asynchronous communication protocols. In this case some extra work might be necessary to correlate the service requests to the corresponding responses, which is described in detail in the How-to guide "[How to Correlate JMS Messages](#)". Most SAP Enterprise Services are synchronous.

Business Semantics

Another aspect in the ESB interoperability is the business semantic defined by SAP's SOA methodology. SAP has defined a harmonized set of data types underlying all SAP Enterprise Services. The core of these data types is based upon proven concept of the [UNCEFACT / CCTS](#) standard using CCTS Core Data Types. Examples for such elementary core data types are "Amount", "Number", "Date" or even "BinaryObject". The "Date" core data type, for instance, describes in detail how a date data type has to look like, since there are many possibilities to represent a date, e.g. "25.12.2009", "12/25/2009" or "2009-12-25". If the format of data types is not defined, mappings even on field level between the SOAP messages are necessary.

SAP has used this open standard to pre-define data types with higher-level business semantics, the Global Data Types (GDT). Re-using these data types in the definition of the service messages is key to harmonize the service signatures across the SAP community, not only to ensure maximum comprehensibility, but as well to simplify integration on the message element level. For more information on GDTs, please have a look at the [SAP Online Help](#), at the blog "[SAP Global Data Types Catalog](#)" or at the blog "[Get those Global Data Types into your ESR](#)".

Beside the format of data types, their interpretation by the applications is equally important. One example is the GDT "Log", which contains the application logs. Assume an application calls an Enterprise Service to create a new customer in an SAP ERP backend system. Unfortunately the end user, who entered the customer data, accidentally provided the wrong ZIP code in the address. Now the validation in the SAP ERP

backend will reject the creating of the customer due to the wrong ZIP code. For the consuming application the underlying Web Service call will be successful, since no technical errors occurred.

The information that the customer was not created, however, is kept in the GDT "Log". The way, how the "Log" data structure must be interpreted by applications, is described in the ES Workplace (<http://www.sdn.sap.com/irj/sdn/esworkplace>) and not carried in the WSDL files, which are exchanged between the IBM and SAP registries. This means that non-SAP applications must be aware of the business semantics and that also application errors of non-SAP applications must be mapped to the "Log" data structure in order to provide a common error handling between non-SAP and SAP applications.

Discussion and Recommendation

So far, we described the technical interoperability of the two registries and ESBs. Using IBM's WSRR-UDDI Synchronization Module it is possible to keep the content of the two registries synchronized. One question that still needs to be discussed is the role of the registries, in particular, if there is a leading registry and, if yes, which one leads. The answer to this question depends heavily on the concrete situation. For example, which functions or features are needed? Which skill sets are available? And of course, is there an enterprise-wide policy to use either IBM or SAP products? There is no generic answer to these questions and it depends heavily on the particular situation. However, since both registries contain the same content, we may rephrase this question to, which task should be performed or owned by each registry? The following paragraphs discuss the role of the registries during the service discovery, endpoint look-up, service development / deployment and service modeling.

Service Discovery

During application development it is important to find the most appropriate service for consumption. Since the two registries contain the same information about services, it makes technically no difference which registry to use for the service discovery.

For the discovery of SAP Enterprise Services, however, the SAP NetWeaver SR is a good choice for business oriented users, since SAP Enterprise Services are already fully classified following a taxonomy that contains business semantics and is in accord with the associated business scenarios. A Free text search is also available. This will help application developers or SAP Enterprise Service consumers to discover relevant services quicker.

Endpoint Look-up

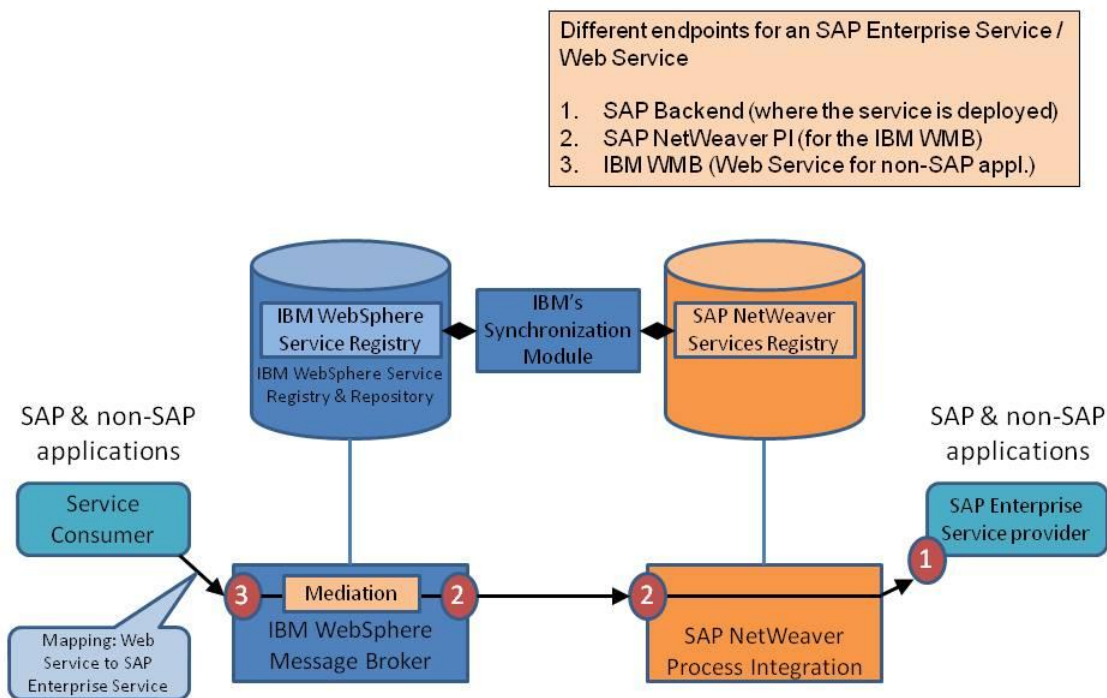
Applications that are going to consume services can obtain the corresponding service endpoints either statically, during the configuration, or dynamically via endpoint look-ups in a service registry. For service consuming SAP applications the endpoints are usually configured statically in the consumer proxies. Dynamic endpoint look-ups, however, can be done in both registries. The IBM WSRR provides an open API for that purpose and the SAP SR supports the UDDI standard.

Since the IBM WSRR-UDDI Synchronization Module ensures that all service endpoints are available in both registries, it is not mandatory to have one leading registry for the endpoint look-up. So, if desired any of the two registries can be chosen to be the leading registry for endpoint look-ups.

The service endpoints determine whether a service call is mediated / brokered or not. If the endpoint is an ESB then the service call is brokered otherwise not. For monitoring purposes or for the mapping between Enterprise Services and Web Services, it makes sense to route a service call through an ESB.

Enterprise Service (SAP or custom built) calls from a non-SAP application that looks-up endpoints in the WSRR, for example, may be realized as follows: the SAP Enterprise Services that should be consumed by the non-SAP application are published in the SAP NetWeaver SR with the SAP NetWeaver PI acting as an additional endpoint, i.e. these Enterprise Service calls will go through SAP NetWeaver PI. Due to the synchronized registries, the IBM WMB can call Enterprise Services by looking-up the endpoints in the IBM WSRR. The IBM WMB can then provide Web Service interfaces for the non-SAP applications and perform all required mediation tasks to convert the Web Service calls into Enterprise Services. In this way the IBM WMB acts as a Web Service provider for the non-SAP application on the one hand, and as an Enterprise Service consumer on the other hand. In this case the SAP NetWeaver PI acts as an Enterprise Service provider proxy to the IBM WMB. Note that the SAP NetWeaver SR will in this case contain two different

endpoints for each Enterprise Service, one to be used by the IBM WMB and another one to be used for communication between SAP applications.



Provide three endpoints for SAP Enterprise Services: one for inter-SAP communication, one for the IBM WebSphere Message Broker and one for the non-SAP service consumer

SAP NetWeaver SR is available with SAP NetWeaver PI 7.1 or higher releases and hence service publication is available only for these SAP NetWeaver PI releases. However, if the SAP NetWeaver PI should act as an endpoint for the Enterprise Services to non-SAP applications, the services can also be published manually.

Service Development / Deployment

The development of new services requires an interaction with a service registry, since all integrated development environments for Web service development generate code from the corresponding WSDL files. Since both registries contain the same service definitions, any development environment that can either interact with the IBM WSRR or SAP SR can be used for the development.

Service Modeling

Even though there is no interoperability between the service repositories possible as stated above, there is one use case that can be highly relevant for IBM-SAP interoperability scenarios.

Since SAP has already a strong SOA methodology in place, it certainly makes sense to evaluate the SAP SOA methodology. Particular beneficial could be the usage of GDT as an enterprise-wide canonical data model for business application, since field mappings and structure mappings will not be necessary anymore. The available set of GDTs ranges from simple structures like the "Date" data type to very complex aggregated data types like "Communication". These GDTs are then used to type the elements of the individual business objects, such as customer or sales order.

In case SAP's GDTs are used as an enterprise-wide canonical data model for custom developed business applications, the SAP NetWeaver ES Repository would be a natural choice for service modeling. After a service is modeled it is published into the SAP NetWeaver SR (without an endpoint, since no implementation exists yet). Through the synchronization of the SAP and IBM registries, the modeled service will be available

in both registries, so that the service can be developed and deployed with either IBM or SAP development tools.

Conclusion

In this document we described possible interoperability options to integrate SOA landscapes from IBM and SAP, in particular landscapes that comprises the IBM WebSphere Message Broker, the IBM WebSphere Service Registry and Repository, the SAP NetWeaver Process Integration including the SAP NetWeaver ES Repository and the Services Registry. Any integration of the two registries requires the IBM's WSRR-UDDI Synchronization Module, which keeps the content of both registries synchronous. Since the IBM WSRR-UDDI Synchronization Module requires UDDI v3, an upgrade to the SAP NetWeaver PI 7.1 or better to SAP NetWeaver PI 7.1 EHP 1 is a prerequisite for the registry interoperation.

The choice for either the IBM or the SAP registry and the appropriate development environment really depends. It is not primarily driven by the characteristics of the systems themselves but rather governed by questions like 'How big is the proportion of SAP or Non-SAP systems in my landscape?', 'What should the software stack in my company look like?' or 'Is there skill for one or the other software already available?'. The real choice that needs to be made is about the general structure of the SOA environment and not just the set of SOA governance tools.

Related Content

<http://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/11000>

<http://www.redbooks.ibm.com/abstracts/sg247220.html?Open>

For more information, visit the [Service Bus-based Integration homepage](#).

Copyright

© Copyright 2010 SAP AG. All rights reserved.

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.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, 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 other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.