



IBM Software Group

WebSphere® Service Registry and Repository

Version 6.2 - Overview



@business on demand.

© 2008 IBM Corporation
Updated June 12, 2014

This presentation provides an overview of IBM WebSphere Service Registry and Repository.

Goals

- Introduce service oriented architecture
- Describe the benefits of a service oriented architecture
- Outline the governance challenges presented by the adoption of a service oriented architecture
- Describe the value and features of WebSphere Service Registry and Repository
- Provide an overview of the WebSphere Service Registry and Repository architecture

The goals of this presentation are:

- To give an introduction to the key concepts and goals of a service oriented architecture
- To describe the benefits of a service oriented architecture
- To outline the governance challenges that must be met when a service oriented architecture is adopted
- To describe some of the key features of WebSphere Service Registry and Repository, and the value it gives to the service oriented architecture life cycle
- To provide an overview of the architecture of the WebSphere Service Registry and Repository product

Service oriented architecture (SOA)

- An architectural style / design philosophy
- Businesses need to
 - ▶ Be flexible
 - Develop loosely coupled re-usable services with well-defined interfaces; removing environment dependency
 - ▶ Be responsive
 - Link existing services together to build new business processes
 - ▶ Reduce time to market
 - Provide closer alignment between your IT systems and your business processes



Service Oriented Architecture, or SOA, is an architectural style, and a design philosophy, for business process development.

IT systems within an organization can be prone to inflexibility due to having tightly coupled applications, which makes them difficult to change and difficult to re-use.

Typically, a business process is composed of numerous more fine-grained tasks, or services.

In a service oriented architecture, business processes are decoupled from these fine-grained services as much as possible, so that common fine-grained services can be re-used by different business processes, rather than re-inventing them each time they are required by a different department.

To be re-usable, a service must be loosely coupled and have well defined interfaces. Web services provide such a capability.

Web services enable SOA through open standards, and are widely adopted in the IT community.

New business processes are built by linking together existing services.

Promised benefits of SOA



Business process vitality



New value through reuse of assets



Improved connectivity



Closer alignment of IT to business



Business Flexibility

If the IT system architecture is based on the use of multiple services, it becomes easier to change a business process, or introduce a new business process, because a service can be replaced by another, or a new business process can be developed by linking together existing services.

If a service is re-used, only one copy of the code needs to be changed to introduce a new function, and only one copy of the code needs to be tested and debugged. It also means that the business can operate on consistent data, which is likely to improve the experience for the customers of the business.

By using open standards such as Web services, many more software and hardware systems can be connected and can interoperate.

If the IT systems in an organization are closely aligned to the business processes that need to be performed, then those business processes are likely to run much more smoothly. It need not be the case that business requirements are hampered by what the IT systems can perform. It should be the case that the IT systems are flexible enough to allow the business to do what it needs to be successful.

When business processes can be quickly pulled together from existing re-usable services, the business becomes much more flexible, decreasing time to market and increasing competitive advantage.

Governance issues

How do I eliminate “rogue services” and ensure control of my SOA?

How do I govern services as part of my SOA?

How do I manage the services life cycle?

How do I enable enforcement of policies across all internal and external services?



How can I help my ESB issue in the right context?

How do I help services interact efficiently and dynamically with each other?



With the introduction of a service oriented architecture come governance challenges that must be addressed.

As discussed previously, the key to a successful service oriented architecture is ensuring that the right set of repeatable business tasks are made available as services. For this to happen, there must be a governance structure in place that ensures that the right people make the decisions as to which services will be available, which department will own the service, fund the service, maintain the service and so on.

Another area where governance is needed is around the service life cycle. With services being shared across organizational boundaries, it is important that proposed changes to the service are well understood. Some questions to consider would be; who has authority to use the service, who will the change affect, who needs to authorize the change? How do you ensure that services are re-used and not redeveloped? Where would a user look to find the services that are available?

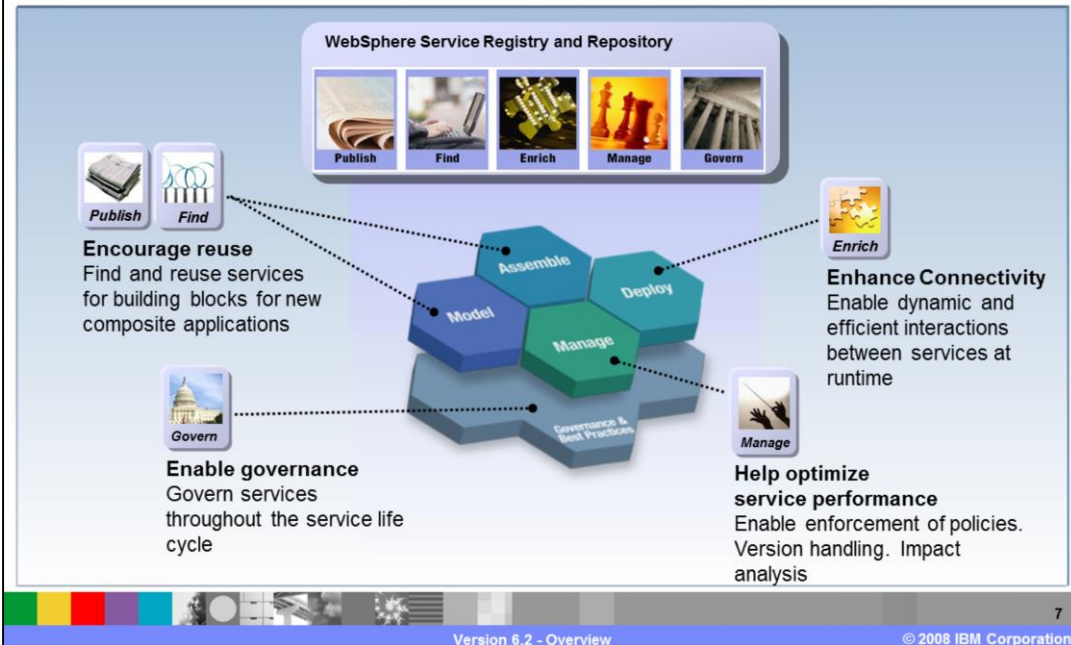
Achieving SOA governance

- From this
- To this



Having a collection of services in your organization does not necessarily mean that you have a service oriented architecture. The services need to be the right set in order for them to be of value across different business processes. Having the right SOA governance model helps to ensure this.

The value of WebSphere Service Registry and Repository in the SOA life cycle



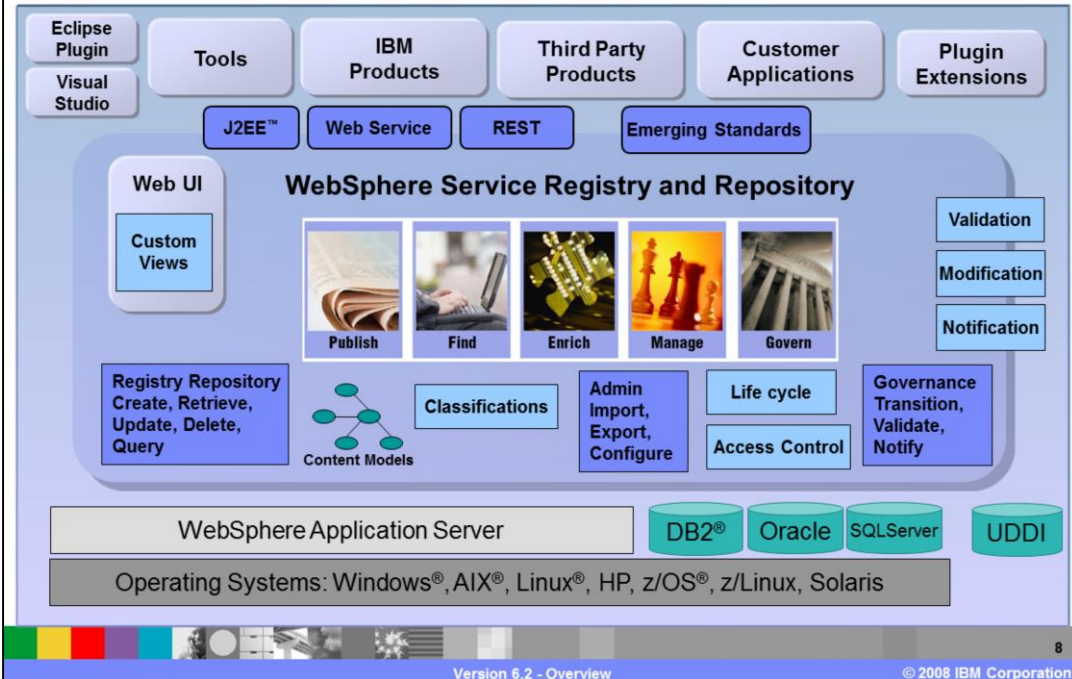
The SOA life cycle has four main stages. The first is the model stage; when requirements are gathered and the business process is modeled and designed. Second is assemble; when appropriate re-usable services are discovered, and the business process is constructed and tested. Third is deploy; when the process is moved to the production environment. Fourth is manage; when the process is managed and monitored, and policies are enforced.

Governance and best practices support the complete life cycle. WebSphere Service Registry and Repository is an integrated service metadata repository that helps you govern and manage services through the service life cycle. Service metadata is information about your services, such as descriptions, owner details, classifications, and so on.

With WebSphere Service Registry and Repository, you can use this metadata information to select, invoke, govern, and reuse services as part of a successful and dynamic service oriented architecture. By promoting visibility and consistency, reducing redundancy, and by managing the service life cycle, WebSphere Service Registry and Repository helps you get the most business value from your service oriented architecture.

The features provided by WebSphere Service Registry and Repository include: Publish and find capabilities, governance policy management transitioning of services through the life cycle, impact analysis to determine the effect of service introduction, deletion or alteration, role based access control and the ability to integrate with other IBM products, such as WebSphere enterprise service bus and WebSphere Message Broker.

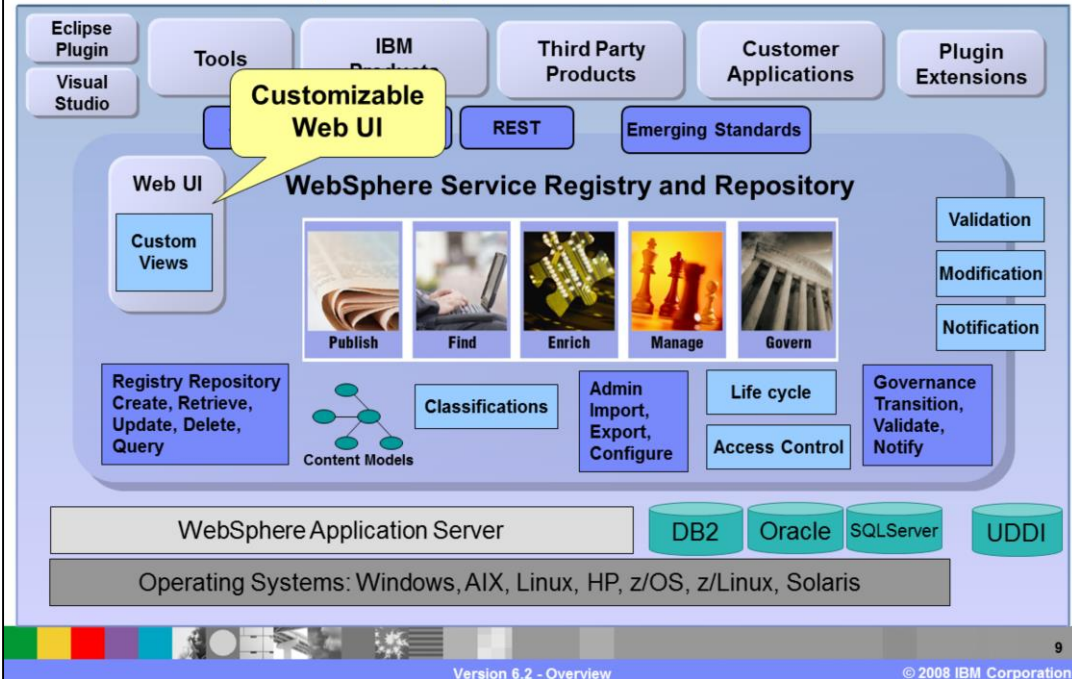
Architecture (1 of 8)



This slide shows the architecture of the WebSphere Service Registry and Repository product. It is built on WebSphere Application Server, and supports a range of database types for the storage of your services and service metadata.

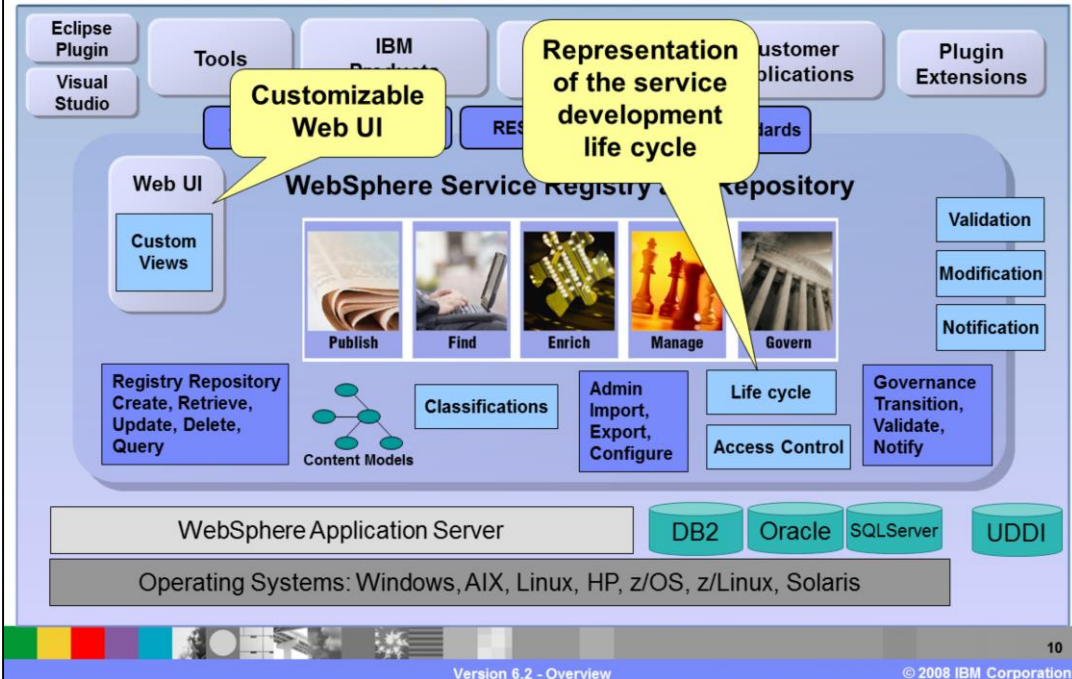
This presentation will now describe the key components of the architecture.

Architecture (2 of 8)



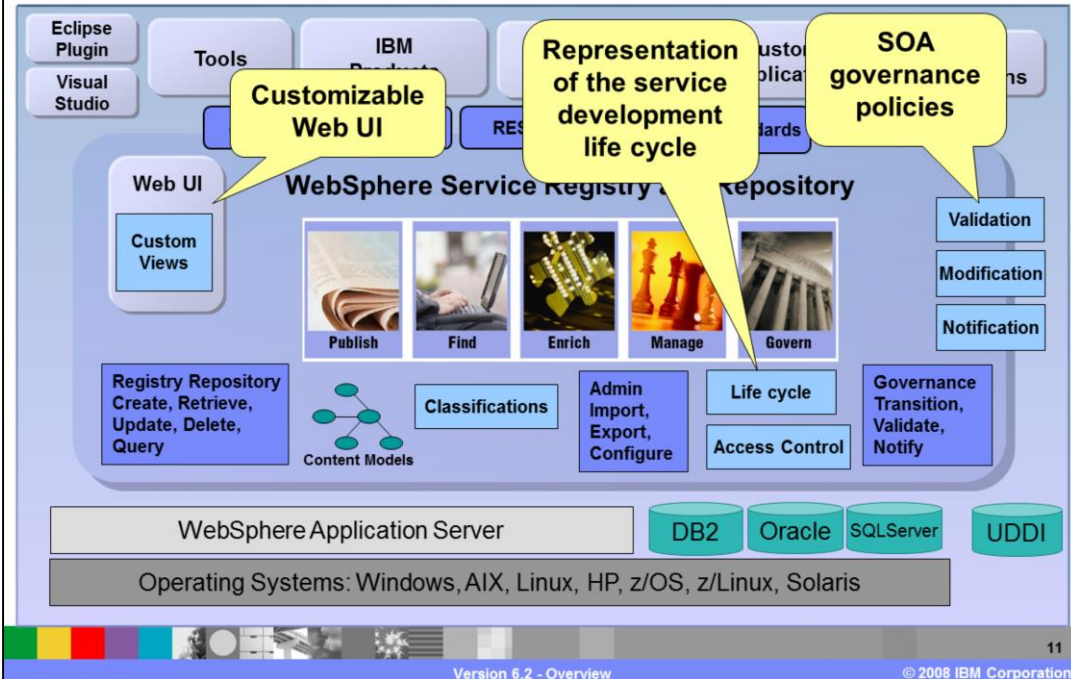
You use the Web UI to manage your services. You can publish and find services, attach metadata, define policies, govern your services through the life cycle, perform impact analysis, and more. You can customize the Web UI to fully meet the needs of your business environment.

Architecture (3 of 8)



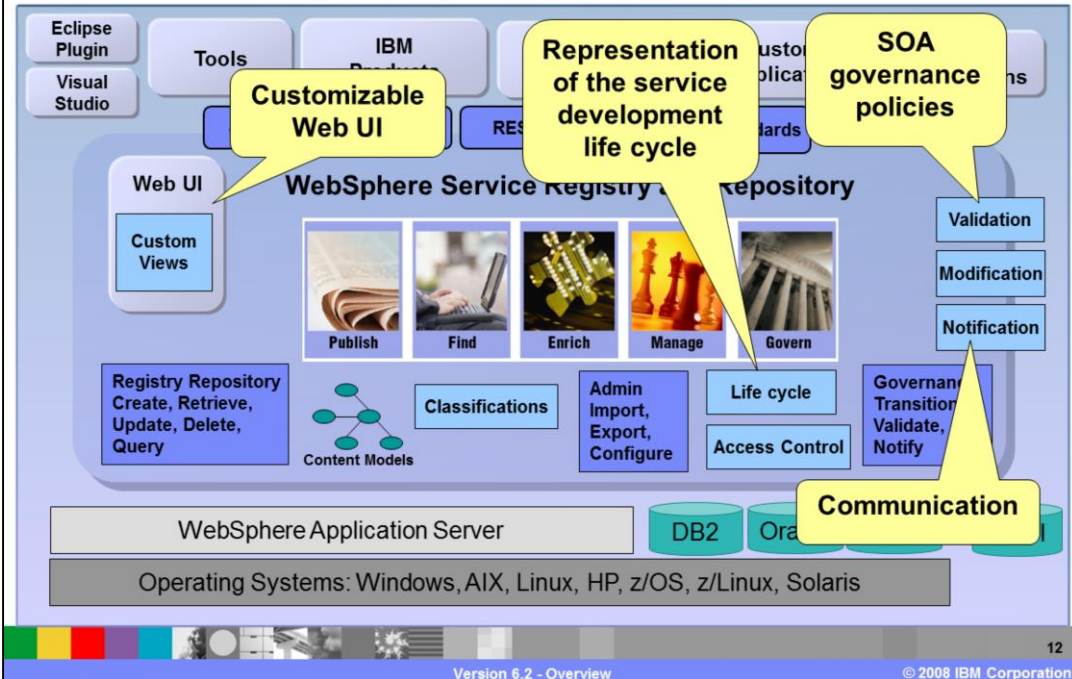
The service development life cycle specifies the complete set of possible states during the development of a service, and the transitions between them. A default life cycle is provided with WebSphere Service Registry and Repository, but you can define your own.

Architecture (4 of 8)



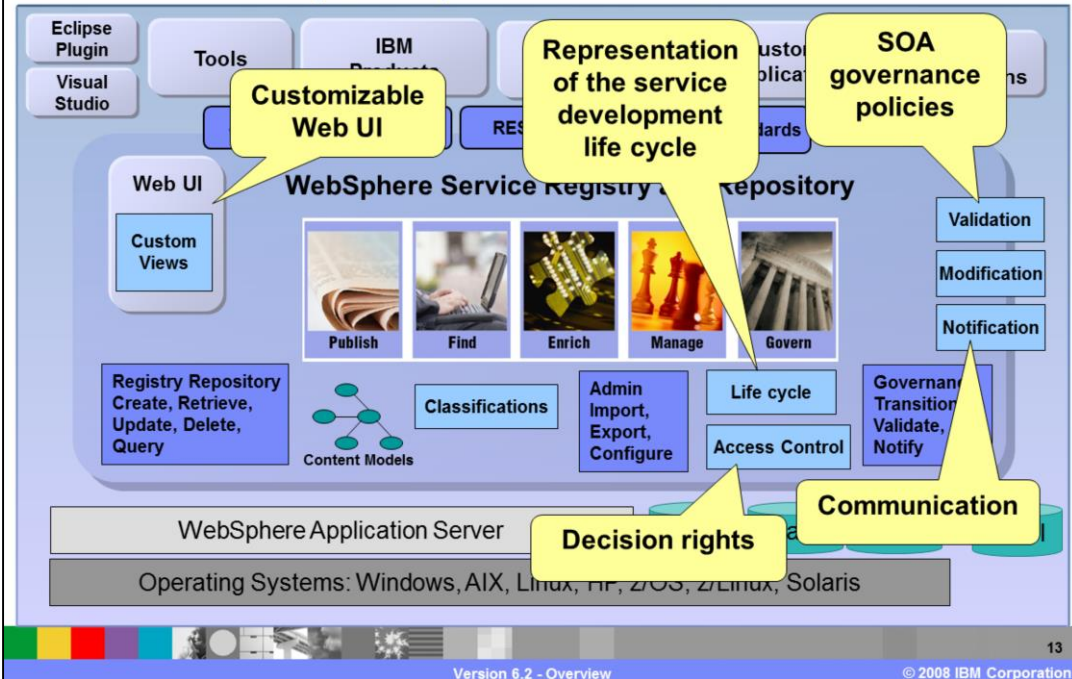
The WebSphere Service Registry and Repository validation and modification mechanism can be used to enforce governance policies at runtime. For example, you can prevent a service from being deleted unless it is in the “Retired” state.

Architecture (5 of 8)

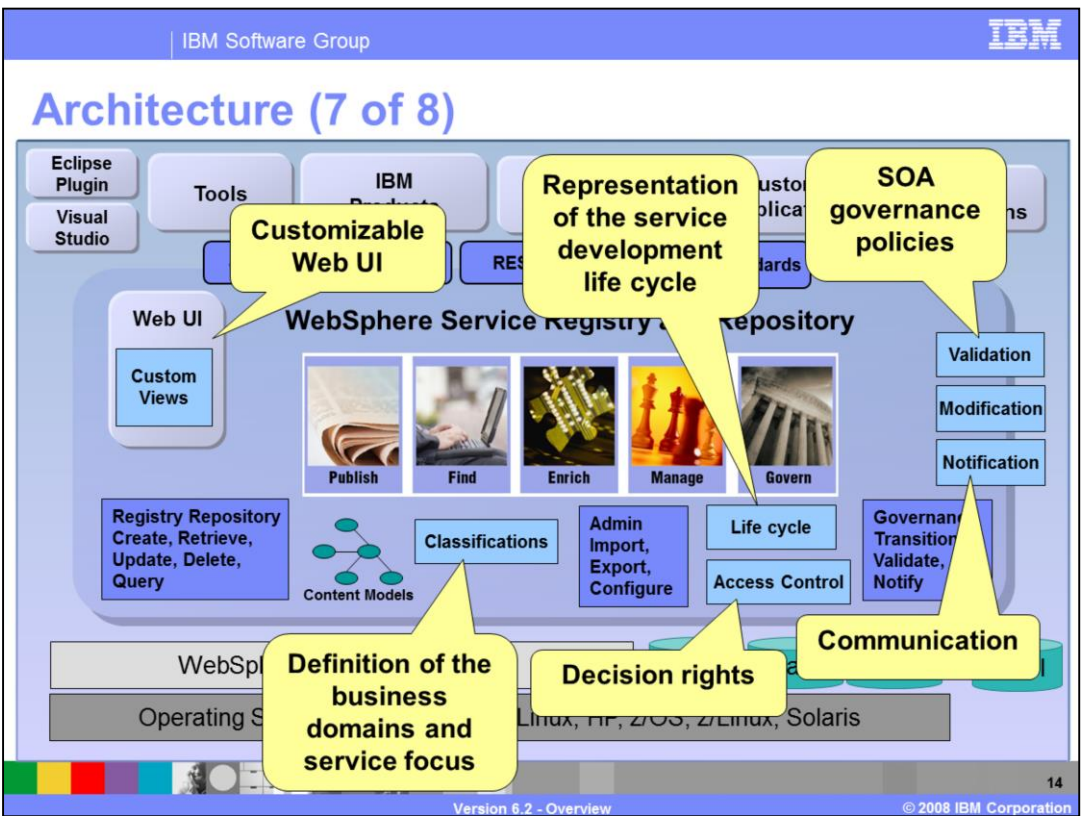


The notification mechanism can be used to communicate details of events relating to services in the registry. For example, you can be notified whenever updates are made to particular types of service.

Architecture (6 of 8)

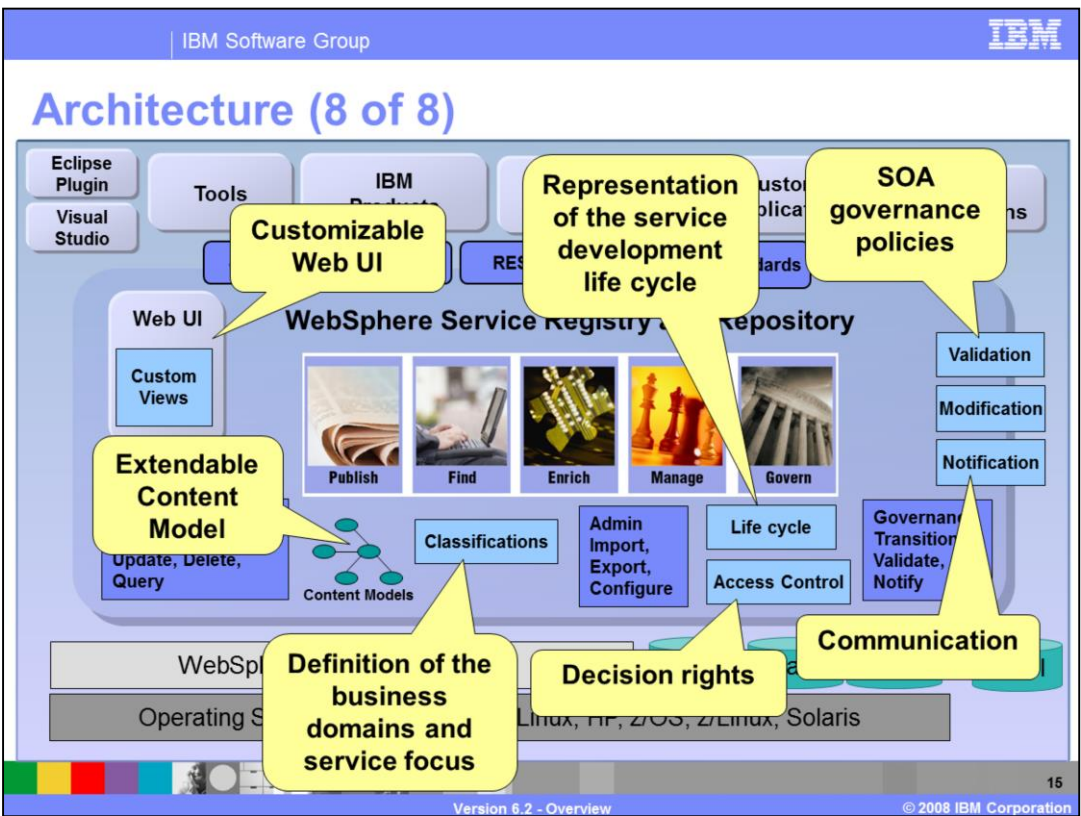


The role based access control mechanism defines decision rights to control which users are able to perform which actions on which entities.



Good classification systems are a key factor in the success of your service oriented architecture. A good classification system simplifies the task of finding services, making it easier to reuse and compose them.

For example, you might use one classification system to represent the organization and another to represent the types of services deployed in your service oriented architecture. Then you could search, say, for the "customer information services" in your "retail banking line of business".



The WebSphere Service Registry and Repository content model defines the structure of all the different types of entity that can be stored in the registry, the types of metadata that can be attached to them, and the relationships between them.

For example, when you load a WSDL document, any XML schema upon which the WSDL depends will also be loaded and the necessary relationship between them will be established. Furthermore, separate objects will be created representing the WSDL entities defined in the WSDL document, such as interfaces, operations and bindings.

The content model can be extended if necessary.

Summary

- A service oriented architecture enhances business flexibility and vitality through the re-use of existing services
- When a service oriented architecture is adopted, effective governance is essential
- With WebSphere Service Registry and Repository, you can publish and find your services, and manage and govern them through the service oriented architecture life cycle



In summary, a service oriented architecture enhances business flexibility and vitality through the re-use of existing services. When a service oriented architecture is adopted, effective governance is essential and with WebSphere Service Registry and Repository, you can publish and find your services, and manage and govern them through the service oriented architecture life cycle.

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX DB2 WebSphere z/OS

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

J2EE, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.