IBM Software
Thought Leadership White Paper

Speed SOA development and time to value with IBM WebSphere Enterprise Service Bus Registry Edition

Achieve flexibility, reduce costs, promote service reuse and provide governance
Overcoming the challenges of building a service oriented architecture

Many organizations have made the first foray into service oriented architecture, lead by the promise of greater business flexibility, services reuse, and the ability to quickly respond to new business opportunities, only to experience disillusionment. Instead of agility, decoupling has produced disjointed services that are difficult to see and impossible to manage. Rather than being published in a service catalog, services are tracked on a spreadsheet. Business-critical services are inadvertently removed, or updated without communicating the update.

These things happen for a variety of reasons. In the early days of SOA adoption, many organizations opted to begin an SOA at the departmental level, hoping to demonstrate ROI and gain executive buy-in. Instead, services were not reused, the SOA project was deemed a failure, and funding needed to expand SOA disappeared. Other organizations decided to begin connecting services with an enterprise service bus (ESB), with a goal of implementing a registry and repository at some later date. This led to a series of problems, such as duplication of services, the inability to plan for growth and an inability to reuse services.

Meanwhile, business processes in enterprises today continue to undergo rapid change. Maintaining a competitive edge requires the flexibility that SOA can provide. This white paper examines the business and IT benefits of simultaneously implementing an ESB alongside a service registry and repository. When the capabilities of both are leveraged from the outset, organizations can experience:

- A flexible and agile integration infrastructure.
- A reduction in the costs and complexity of integration.
- Promotion of service reuse with control.
- Visibility into the SOA through governance and management.

Overcoming the challenges of building a service oriented architecture

Many organizations have made the first foray into service oriented architecture, lead by the promise of greater business flexibility, services reuse, and the ability to quickly respond to new business opportunities, only to experience disillusionment. Instead of agility, decoupling has produced disjointed services that are difficult to see and impossible to manage. Rather than being published in a service catalog, services are tracked on a spreadsheet. Business-critical services are inadvertently removed, or updated without communicating the update.

These things happen for a variety of reasons. In the early days of SOA adoption, many organizations opted to begin an SOA at the departmental level, hoping to demonstrate ROI and gain executive buy-in. Instead, services were not reused, the SOA project was deemed a failure, and funding needed to expand SOA disappeared. Other organizations decided to begin connecting services with an enterprise service bus (ESB), with a goal of implementing a registry and repository at some later date. This led to a series of problems, such as duplication of services, the inability to plan for growth and an inability to reuse services.

Meanwhile, business processes in enterprises today continue to undergo rapid change. Maintaining a competitive edge requires the flexibility that SOA can provide. This white paper examines the business and IT benefits of simultaneously implementing an ESB alongside a service registry and repository. When the capabilities of both are leveraged from the outset, organizations can experience:

- A flexible and agile integration infrastructure.
- A reduction in the costs and complexity of integration.
- Promotion of service reuse with control.
- Visibility into the SOA through governance and management.
SOA adoption is widespread but limited in scope

First, let’s examine the current market outlook for SOA. In many organizations today, SOA has been widely adopted, but the SOA itself is typically limited in scope. Adopters often relegate SOA to certain types of projects and haven’t implemented it across the enterprise—hoping to achieve short-term IT benefits rather than long-term business transformation. While it is understandable that IT would hope to reap its own benefits in the form of services reuse, the real gains from SOA are much broader and are only successful if they carry a business focus. The reality is that an SOA requires organizational changes and significant investment and planning—all of which necessitate support at the business level. Building the architecture correctly from the outset, using a top to bottom approach—is key to SOA success.

It is also true that more than half of SOA implementations do not meet expectations to support business changes and reduce costs. This typically stems from an SOA project that was doomed to failure due to budget limitations, and a dearth of research into the components needed to achieve SOA’s benefits. In fact, it is not uncommon for an organization to begin implementing what they believe to be an SOA, only to find that it is not. Services were not created, linked and published properly, and are therefore of little or no use. Shortcuts, in essence, are a death knell for an SOA.

Organizations need a way to smoothly—and rapidly—implement an SOA in a way that will help them achieve the greatest time to value. Key to this smooth implementation is deploying both an ESB and a service registry and repository, simultaneously. IBM SOA solutions represent the key to an SOA “quick start” that can make the difference between an expensive—but-failed experiment and a business-transforming architecture.

Why SOA requires an enterprise service bus

ESBs are the means by which SOA-based organizations connect service consumers with service providers, enabling them to route service calls, transform messages, mediate between protocols and distribute business events. In fact, ESBs are an excellent tool for helping organizations move closer to some of the broader goals of an SOA, including reuse, visibility, flexibility and business alignment. Today’s ESBs are also used to enforce service level agreements (SLAs) and provide service assembly—and therefore require the capability to be dynamic.

The ESB is unquestionably an essential component of the SOA. Without it, organizations have no ability to measure the health of SOA services. Instead, they are faced with:

**Duplicated connectivity efforts**—Lacking the ability to reuse services in an SOA, IT is perpetually relegated to rewriting code and duplicating transformation efforts.

**One-off and custom connections**—Instead of connecting service providers and consumers through an ESB, applications must be connected point-to-point—and hard-coded.

**Lack of control**—Service connections and transformations, rather than streamlined through an ESB, are haphazard due to the point-to-point connectivity.

Why SOA requires a service registry and repository

A service registry fulfills an entirely different—yet complimentary—set of tasks within the SOA, helping to deliver the maximum benefit to business services. It is a key component for delivering the value of SOA to consumers. In
fact, without a good registry and repository, implementers risk deterring adoption and growth of SOA within the organization. Designed to be leveraged in a runtime environment and to support policy enforcement, the service registry enables organizations to:

**Perform service discovery**—To see which services are available, what state the services are in, whether they have been implemented or are actually in testing, and what versions of services are available.

**Manage subscriptions**—To know who is using which services, and which services are not being used.

**Catalog services**—Publishing services in a service catalog enables organizations to increase both service awareness and service reuse. It also helps determine who is using the assets, and to identify duplicate services.

**Promote reuse**—Increasing the visibility of services assets encourages their reuse for building new services and processes, thereby maximizing limited IT budgets and resources. It also helps to minimize redundant services, which reduces costs and increases staff productivity.

**Govern the entire service lifecycle**—To introduce, promote and retire services, ensure that services are properly funded, owned and resourced, and to notify users if a service is retired, or a version is depreciated.

**Control the service assets development process**—A service registry also helps to deliver new/updated services quickly to address new business opportunities; to enforce SOA policies consistently to help achieve reliability and compliance; implement recommended practices with out-of-the-box policies; ensure service ownership and funding exists; and to address requirements from all potential consumers.

Conversely, an SOA lacking a service registry faces duplication of services and the inability to plan for growth, since the organization cannot see who is actually using the services. Reuse of services becomes difficult, and service consumers are unaware of changes to services. Lastly, without a registry, the ESB is less dynamic, since mediation changes require changes to the actual application code.

The benefits of deploying a service registry and repository are well documented. In fact, organizations that deployed IBM WebSphere Service Registry and Repository (WSRR) reported that on average, they experienced:

- A 30 percent increase in software reuse.
- A 25 percent reduction in integration costs.
- A 40 to 60 percent improvement in application maintenance productivity.

The average ROI after deployment ranges from 300 percent to 700 percent—within 9 to 13 months.1

**Leverage a bundled solution: IBM Enterprise Service Bus Registry Edition**

Without the structure and control—and specifically the checks and balances that an ESB and registry provide—organizations are unable to realize the full benefits of SOA, ultimately leaving IT with an array of services that are not reusable, that they cannot see, and that have no meaningful groupings. In addition, when the registry is considered a “phase 2” implementation in the SOA, momentum is lost, and the battle for funding starts again as IT struggles to show ROI that does not exist. Not only that, when the registry is implemented later, the ESB must also be “retrofitted” to support the registry component, and processes must be re-engineered, a time and cost burden that few organizations can afford.
When the registry is deployed alongside the ESB, IT is able to effectively track services they are building, to establish a service catalog in which to publish those services and to establish the foundation to support the complete service lifecycle. Together, the products provide the ability to track service consumers, to see who is using services, how they are being used, and overall service quality. Building on this knowledge, organizations can immediately gain visibility into service usage, to prove ROI by showing service how services can be reused in the future, and to demonstrate that are currently being reused.

**Dynamically update service locations for greater operational flexibility and agility**

Without a registry, most service-based applications must be hard-coded to the service location and version. So a change in location means the organization must redeploy application code. By using the capabilities of the service registry, the service location can be dynamically accessed and changed—to take effect immediately—providing a significant savings in the time and costs of development, testing and deployment, as well as providing improved service availability.

**Dynamically choose service providers and endpoints**

In a typical environment, the ESB decouples the service consumer from the service provider, but the ESB is still coupled to the provider, creating a brittle ESB with expensive mediation code changes when simple endpoint operation changes occur. Some organizations may attempt to manage this via a properties file that resides on a file system or within a database. However, that approach means organizations will lack control regarding who can make changes, and they will not be able to record when code changes or maintenance occurred.

By combining the ESB with the service registry, organizations are able to dynamically choose the service provider and the endpoint in this way:

- The service registry holds the metadata about the available service providers, their versions and the endpoints at which they can be invoked.
- The ESB leverages that information at runtime in the mediation flow through configurable mediations primitives that can retrieve the metadata from the registry.

Without the combined abilities of the ESB and registry, mediation code changes introduce both cost—due to additional modification, test and deployment cycles—and risk.

**Select application and translation logic**

When a service has multiple incompatible versions, it is necessary to apply transformation logic to make the service perform for the consumer. Combining the ESB and registry capabilities enables organizations to design mediations to expect that changes will occur and correct the incompatibility by externalizing some of the application logic as policies governed in the service registry. The policies can then be applied as changes occur or the content of the message flowing through meets the policy criteria.

**Provide service level agreement support**

Without a service registry, SLA information is not easily accessible for services that are being executed. The registry enables organizations to store SLA information as policies and enforce those policies, helping to ensure limited enterprise resources are used efficiently and enabling IT to check the existence of an SLA at runtime.
Manage and enforce consumer entitlement

Providing SLA support is especially important when managing consumer entitlements, another common issue within the SOA that can be managed within the registry and enforced by the ESB. The consumer entitlement helps to ensure that the consuming application trying to invoke the service has an approved SLA in place, thus guarding against rogue consumers. Managing and enforcing consumer entitlements helps to ensure that the service provider can maintain the qualities of service for other consuming applications and protect against users seeking to build mission-critical applications that leverage a service without obtaining the appropriate permissions. Together, the registry and the ESB can check if the consumer has an SLA in place, and is therefore authorized to access the service.

Enhance connectivity and increase runtime flexibility for applications

Pre-built integration points within WebSphere Enterprise Service Bus Registry Edition enable applications to query the registry for service endpoints during runtime, and the registry, in turn, can provide associated metadata for those services. Applications can then invoke the services that best match the application’s needs. Having the registry helps to ensure that that service endpoints and associated metadata are current, enabling the ESB to be more dynamic, flexible and adaptable.

Building on these capabilities, WebSphere Enterprise Service Bus Registry Edition helps to speed time to value of the SOA by enabling organizations to:

- Publish services for dynamic lookup.
- Develop and test integration solutions.
- Modify and enforce mediation policies.
- Report on service consumers to champion the SOA.

Publish services for dynamic lookup

When organizations begin building service connectivity into the SOA, the first step is to establish what services the organization has. This helps to show integration developers which services are available and to set controls regarding how they are developed, their lifecycles and how the services can be modified. Without these controls, the issues here are both cost and risk. By establishing control over service changes, organizations avoid the risk of making changes to business-critical services. One way to address this is to establish an internal regulation that every exposed service must have a security policy.

There are also additional operational costs associated with lack of visibility into existing services, with the inability to keep services current and the inability to track existence and usage—all of which result in service duplication. Also costly is maintaining, rather than decommissioning services, for lack of insight into service usage.

IBM enables organizations to build a catalog of trusted, high-quality services, which provides:

- Multiple methods to publish services.
- Customizable ontologies to classify services aligned with the business domain.
- Powerful queries to find best-fit services.
- A standards-based application programming interface (API) support to access content, including REST interfaces (Web 2.0).
- Service discovery of deployed services on .NET, Oracle/BEA, JBoss and IBM WebSphere Application Servers.
- Faceted search for a natural, user-friendly way to refine search using attributes, document types or classification.
Employ service federation
It is not unusual for organizations to implement departmental SOAs, creating silos of services that cannot be reused—and therefore get recreated, again and again. The IBM Service Federation Management feature pack enables organizations to securely share and federate services across domains, to reuse existing services and to avoid the cost and waste of creating redundant services.

By using IBM Enterprise Service Bus Registry Edition, organizations are able to promote service reuse with control, including governance to changes, and to establish service federation management.

Develop and test integration solutions
The cost of creating integrations by hand and maintaining integrations can be prohibitive. IBM Enterprise Service Bus Registry Edition enables organizations to develop and test integration solutions using declarative integration developer tooling for transformation and routing, with integration patterns and a rich set of connectivity options. Once connectivity patterns are instantiated, organizations can then expose new functionality within the service, which further helps to establish business agility—and speed time to market.

Modify and enforce mediation policies
With a service registry, organizations can easily move mediation policy information out of the ESB and into the registry, enabling them to implement a dynamic ESB. In addition, the registry allows users to change policy data using a business space (Web 2.0) widget instead of re-coding, helping to increase flexibility and lower operational costs. And, since the registry can access and change applications directly, without recoding, this greatly reduces the costs of change management.

Report on service consumers to champion the SOA
The failure of SOA implementations is often due in large part to lack of support within the business. Without metrics showing service usage and reuse, IT simply cannot prove the ROI needed to expand SOA throughout the enterprise. Projects stall, funding disappears, and the promise of SOA business transformation dies as well. One of the biggest advantages of combining the ESB and registry is the ability to monitor and leverage reports on service health and service traffic. This enables IT to immediately capture consumer and provider information, provide impact analysis and speed problem resolution. Further, IT can use these metrics to provide real, demonstrable business benefits that champion the SOA to business decision makers.

SOA increases efficiencies and reduces costs for two IBM customers
One IBM customer, a large insurance holding company, was struggling with inefficient claims processes, which in turn were driving up administrative expenses. By implementing an SOA based on IBM WebSphere solutions, the company was able to not only increase reuse but also achieve better strategic alignment of IT through effective governance. The result was increased efficiency and improved service quality. Today, the company has a globally optimized claims process with reduced processing costs, faster claims cycle times and improved visibility, resulting in a projected $50 million in annual savings.

In another case, a major state-run government agency needed to establish a faster response to its unemployed constituents and to reuse existing services with secure and reliable access. Working with IBM, they were able to develop and test integration solutions using declarative integration developer tooling for transformation and routing, with integration patterns and a rich set of connectivity options. They are now able to ensure scalable, secure and policy-based transactions and access to external and internal users. Business outcomes include massive scaling to support more than 1 million constituents, as well as reduced complexity and efficient reuse of data and applications.
Conclusion
Combining the capabilities of an ESB with a service registry and repository builds agility into service creation, transformation and reuse. It opens the door for IT to prove the ROI of the SOA and to capture new business opportunities through services reuse. Through thousands of customer engagements, IBM is helping organizations around the world to fully leverage their SOA architecture. The bundled offering of WebSphere Enterprise Service Bus Registry Edition is a demonstration of IBM's ongoing commitment to help customers achieve business transformation within a robust, service oriented architecture.

For more information
To learn more about how IBM WebSphere Enterprise Service Bus Registry Edition can help you build agility, services reuse, cost savings and governance into your SOA, contact your IBM representative or IBM Business Partner, or visit: ibm.com/software/integration/wsrr

Additionally, financing solutions from IBM Global Financing can enable effective cash management, protection from technology obsolescence, improved total cost of ownership and return on investment. Also, our Global Asset Recovery Services help address environmental concerns with new, more energy-efficient solutions. For more information on IBM Global Financing, visit: ibm.com/financing

IBM, the IBM logo, ibm.com and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. 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 IBM trademarks is available on the web at “Copyright and trademark information” at ibm.com/legal/copytrade.shtml

References in this publication to IBM products and services do not imply that IBM intends to make them available in all countries in which IBM operates. Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change 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. IBM products are warranted according to the terms and conditions of the agreements (e.g. IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided.

The customer is responsible for ensuring compliance with legal requirements. It is the customer's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law or regulation.

1 Source: Cross-industry averages based on estimates by IBM's Business Value Assessment Team. Actual results may vary by customer and industry. ibm.com/software/integration/wsrr/nonflash.html