

# CBDI Report

## ***Modernizing Application Integration with Service Oriented Architecture***

**Abstract:** Whilst investment in Application Integration initiatives over the last decade has undoubtedly improved the sharing and synchronization of information across the diverse resources of the typical organization, this report considers how the use of Service Oriented Architecture (SOA) enables a more agile integration approach that is more responsive to new business requirements.

<b>Contents</b>	<b>Page</b>
Drivers for Change	1
The Role of SOA and Web Services	5
Infrastructure for Application Integration with SOA	8
Roadmap Planning	17
Summary	22



**Independent Insight for Software Oriented Practice**

## Introduction

Application Integration is not new. The last decade in particular has seen enormous investment in Application Integration initiatives and in the messaging infrastructure to support it. Whilst this has undoubtedly improved the sharing and synchronization of information across the diverse resources of the typical organization, the continued pressure to optimize business processes together with the needs of the business to become more responsive to change requires a fresh look at application integration.

In this report we consider

- **Drivers for Change:** The business and IT drivers that are driving a need to modernize application integration.
- **The Role of SOA and Web Services:** The ways in which Service Oriented Architecture (SOA) and Web Services provide a new approach for application integration.
- **Infrastructure for Application Integration with SOA:** Takes a closer look at the infrastructure capabilities required for taking a Service Oriented approach to integration, including the Enterprise Service Bus (ESB), and considers the role of IBM products.
- **Roadmap Planning:** Finally, the report considers how organizations might plan the introduction of SOA based application integration, and considers the ROI.

## Drivers for Change

Organizations will already have many technologies available to them for Application Integration. Besides simple file transfer approaches, there are more sophisticated solutions such as distributed computing using remote procedure calls (RPCs) or object request brokers (ORBs). There is message-oriented middleware (MOM), sometimes used together with Enterprise Application Integration (EAI) products, or there are EDI and B2B Gateways used for external connectivity. However, CBDI Forum believes there are number of closely related business requirements that follow which are driving a need to re-evaluate how these Application Integration approaches are used.

### Information Visibility and Accuracy

Improving the visibility and accuracy of information is key to the organization from a number of perspectives

- improving customer satisfaction – for example ensuring information on the availability of stock is accurate so that customers have confidence in placing orders that will be fulfilled in time
- improving employee decision making – such as monitoring stock levels, transaction rates and buying trends to ensure stock is replenished in time
- optimizing the supply chain – ensuring that the production and distribution of goods is matched to demand

### Real Time Behavior

To enable the visibility of accurate information often requires more real time behaviour in systems. Organizations may strive to become a Real Time Enterprise (RTE) where events are responded to as they happen. Acting in real time often requires the adoption of a Straight Through Processing (STP) approach to remove time wasting delays in the process. There is also little doubt that optimizing the

enterprise's use of resources through Just In Time (JIT) approaches are beneficial to the bottom line, and JIT principle can be applied to information resources too.

Many current integration and connectivity scenarios today rely on batch processes and file transfers that inhibit real time behavior.

### **Process Optimization**

In turn, the RTE also requires Business Process Optimization (BPO) on a broad scale. There is little point for example in implementing JIT Manufacturing if the sales order and inventory management processes are not also optimized to support the whole end-to-end process.

Moreover, few businesses are an island and they work with many partners up and down the supply chain. So BPO more often equates to Supply Chain Optimization (SCO). Ensuring real time behavior does not stop at the organizational boundary.

### **Increasing Demand for External Connectivity**

This highlights an increasing demand for external connectivity across supply chains and ecosystems whereby suppliers and customers can also be integrated into a seamless end-to-end business process. In parallel, the organization must consider how the real time business process will be executed by an increasingly mobile workforce, or parts of the process that are outsourced to a business partner.

Today, organizations may use different integration solutions for internal and external connectivity with a break in the flow of information. However, these are increasingly inseparable, and the need to exchange information with external participants is itself driving the need for improved internal connectivity as the information may be currently dispersed across several independent silo applications.

### **Diversity**

As well as the growing number of participants, IT is also presented with an ever expanding range of resource types in terms of devices, platforms, packaged applications and technologies that must be connected. As new types emerge, they rarely negate the need to continue addressing those already in use. It is not just diversity but also the sheer volume of terms of the number of instances that must be addressed. For example large global businesses may have literally tens of thousands of business partners. Integration solutions used today may be specific to narrow ranges of resource types.

### **Agility**

Though many of the previous initiatives are focused on improving efficiency which remains a key objective in order to continue to manage costs, the focus of business objectives is also showing evidence of change. Following a period of sustained cost cutting, many organizations are now looking for growth. Increased agility to drive and respond to change is seen as important in achieving this.

Agility presents one of the key drivers for modernizing integration. It is not just connectivity that is required, but the flexibility to change connections with the same efficiency and timeliness as the enterprise is now demanding in the execution of its business processes. Not only do Service Requests need to be routed dynamically to the appropriate resource, but the rules that govern the routing need also to be capable of being dynamically updated in response to real time changes in business conditions or service level delivery.

However, changes to current integration solutions by developers often take an order of magnitude longer than that required by the business process.

## **Governance**

Increased governance and compliance is also high on the business agenda. Not only should enterprises ensure that any integration activity complies with organizational policies, they must also ensure that the execution of the composite application that results from the integration also complies with policies and regulatory requirements. Because of the real time behavior described above, aspects of this must be built into the process itself to ensure that policies are enforced at the time of execution, not after the event when it may be too late.

## **Organizations Large and Small**

Smaller organizations do not escape these issues. Whilst it might be easier for them to exert more control over internal diversity the issues of external connectivity are largely the same. Even the smallest of organizations will find themselves having to respond to Web Services because the larger participants in their ecosystems will drive their adoption. For example, dominant retailers might force small suppliers to use Web Services to improve the flow of information so they in turn can improve the quality of information they provide to their own customers.

However, smaller organizations may also be under even greater cost pressure as they cannot benefit from the economies of scale, and application integration activities may appear proportionately more expensive.

A key reason for the use of Web Services given to CBDI by a number of global organizations is the ability to quickly connect large numbers of small business partners into their ecosystem to improve the responsiveness of the supply chain, but in such a way that application integration cost is not a barrier to those partners.

## **Beyond Applications**

Applications are not the only things that require integration. Nor is every process step going to be automated. Furthermore, there is a significant amount of information held by organizations in files that are not accessed through applications with a clean API. It might for example be held off-line, or in files maintained by end-users who cut and paste or simply re-key data into unstructured documents, yet at the same time may be key to their daily activities and decision making, and hence the smooth running of the business.

To become a truly real time enterprise organizations need to take a more holistic view of integration that also includes human activities such as self-service, collaboration and decision making, and how these are factored into the end-to-end business process in such a way that these manual steps in the process are also designed to achieve the same overall real time behavior. This requires for example event and notification Services that trigger these activities at the appropriate point in the process.

## **Evolving Integration**

Of course there is no shortage of integration and connectivity solutions available today as listed earlier. Many organizations will already have several. The challenge for IT is that these solutions do not always work well together.

For example each integration solution might:

- Use different technology, transports, and integration products.
- Be proprietary or have limited platform coverage.
- Require technical skills specific to the product or technology.
- Create technology dependencies at either endpoint of the integration.

Moreover, current integration solutions often each span only a single part of the overall scenario. For example, the EAI solution used to integrate a set of internal resources, may be inappropriate for sharing information with external resources. As such, delivering seamless end-to-end connectivity and business process automation can be complex as it requires the use of multiple integration technologies.

It is also evident that even with the Enterprise Integration Middleware available today, many organizations continue instead to use custom built integration solutions that are inflexible and cost more to implement, and are typically unsuitable for the requirements outlined earlier. A study by Software Strategies<sup>1</sup> also estimates that up to 35% of the integration links in Fortune 500 companies are still to be addressed, rising to a massive 75% in small businesses.

### **Tight Coupling**

Today's integration approaches may result in tight coupling, where dependencies are created when two or more resources are linked together by requiring either a common platform, middleware or integration product. Integration solutions will also often be application specific, even down to the release level. Tight coupling is also created through the use of inflexible message formats that cannot be changed without impacting the resources that exchange the message. These factors constrain agility as the

- virtualization of resources can be difficult to achieve because the resource cannot be relocated without rebuilding all of its hard-wired dependencies
- Ability to change resources, or use an alternative provider is reduced because of the tight coupling to the current resource and/or its technology platform.

### **Direct Business Impact**

The bottom line is that inadequate integration prevents the organization from achieving optimized business processes and can have a direct business impact.

Consider that if a customer wants to know from a retailer the availability of a product, who in minimizing their own inventory has it shipped to order by an outsourced logistics company, and in turn may it also be built to order by the manufacturer, then there needs to be a very clear flow of information across that supply chain if the customer is going to be given an accurate and timely response at the point of contact. Too many of us will have experienced the annoyance of placing orders via ecommerce systems that promise "next day delivery", only to subsequently receive an email days later apologizing for the product not being in stock.

---

<sup>1</sup> Enterprise Integration Challenge. Software Strategies, April 2005

## The Role of SOA and Web Services

CBDI believes that the solution to these challenges lies in the adoption of both Service Oriented Architecture (SOA) and the use of Web Service as the basis for integration activities.

Web Services are a set of protocols that offer several advantages for use in integration scenarios;

- They are based on platform independent open standards that benefit from universal support by all major vendors.
- They leverage ubiquitous Internet technologies as well as other transports, and are valid for both internal and external usage, for the delivery of new systems and the reuse of existing assets.
- Have a low entry and implementation cost plus a small technology footprint, making them attractive to all sizes of organizations, and applicable to low cost devices used in pervasive and mobile scenarios
- The extensible set of protocols offers solutions for federated security, transactions, reliable messaging and the provisioning of resource, that do not impose technology or product dependencies on endpoints – other than compliance with the protocols.

### Service Oriented Architecture

Service Oriented Architecture (SOA) is an architectural style that provides a framework through which organizations describe services and their behavior, the endpoints at which they are provided, and the policies that govern their use. It is important to recognize that SOA can be applied equally when modeling the business, designing the software to support the business, or describing the run-time deployment of the services and associated resources. As such, one of the key opportunities for SOA is to improve business and IT alignment and enhance governance across the full lifecycle.

SOA can also be seen as set of principles and practices by which organizations manage the delivery of the optimum set of Web Services from which flexible composite solutions can be assembled to best meet the needs of the business.

Together, Web Services and SOA provide more flexible coupling of resources that in turn enables business and IT agility, and reduce the impact of change.

CBDI Forum separates the two as there is no prerequisite to use Web Services protocols to implement SOA. Other technologies (preferably also based on open standards) are also appropriate within some contexts yet still deliver similar benefits providing key principles of SOA are upheld. Conversely, neither does the use of Web Services demand the more formal architectural style of SOA – they could be used simply to get the benefits of loose platform coupling. However, the benefits can be maximized when both are used together.

### SOA Provides Common Integration Framework

Most importantly in the context of the earlier discussion is that Web Services and SOA can provide a common framework for all integration and connectivity scenarios. Web Service protocols are already used today to provide solutions in areas which might traditionally be thought of as EAI, B2B, Distributed Computing, Mobile Computing, and so on. Consequently, delivering a truly integrated end-to-end solution becomes much more straightforward.

## Changing Integration Layers

Today integration is performed at various layers within the IT architecture. At the technology layer integration is focused on how the operational platforms are connected and the transport technology used, for example how will a message sent from an application running on Microsoft .NET be received by J2EE? The application layer contains the applications and components deployed to the technology layer. Here integration might for example consider how information is moved from SAP ERP to Siebel CRM. This has been the main focus of much EAI activity.

The challenge at these layers is that integration efforts are often focused on connecting the specific instances the organization is currently using, such as connecting J2EE to .NET, or SAP to Siebel. If these change then the integration must change too. Assets integrated in this way are considered tightly coupled because they are bound not only by the endpoints, but also by the integration approach.

The Service Layer abstracts services away from these underlying layers. Instead of considering how the process connects to SAP and Siebel, the focus shifts to connecting Orders and Customers – regardless of the underlying applications and technology in use.

Finally, in the Business Process Layer composite application solutions are then constructed by requesting the Services, not by directly coupling to the underlying applications and technology.

This is highlighted further when the requirement for business process optimization that spans multiple participants is considered. Integration is difficult to achieve using traditional approaches at the technology and application layers because this requires common technology, or integration products are shared by all participants.

More importantly, change will happen independently within each participant. No participant will want to be constrained when making changes to their own implementations by the technology used by its partners, or the technology used for integration.

This might not have troubled larger organizations who may have justified investment in EDI for example. However, for the same reasons given earlier EDI itself must now change to provide aspects of real time behavior and to reduce the cost of implementation so that smaller organizations can also participate in an automated supply chain process.

## Separating the Service Provider and Consumer

The principal of loose coupling is that dependences between the Service Consumer or Requestor and the Service Provider are minimised to enable agility. The key benefit of loose coupling is agility. Tight coupling constrains the ability to adapt to changing business and technology requirements.

The Service Layer helps to isolate change in the providing or requesting applications. For example, the frequency of change in the business process, or the device and technology diversity used in the requesting application might be more frequent than that in the core back end systems that are used by the Service Provider.

From a technology perspective, Web Services can be a key enabler for loose coupling application links as this removes platform and middleware dependencies. The only dependency is that each endpoint understands Web Services and can dispatch the requests to the appropriate resource. Though it should be understood that Web Service protocols are not the only technology that could be used for integration where the participants at each endpoint are in agreement.

The Service Layer therefore becomes the new Integration Layer. However, the design of Business Services must also be carefully considered. Just rendering an existing API as a Web Service is rarely enough as

- It often contains implementation specific detail or behaviour, so tight coupling still remains
- Is likely to be at a granularity that is different to that required by the business process, forcing the Service Consumer to make multiple requests to different Services to execute a single business activity

A key action therefore is aggregating and abstracting meaningful Business Services that are independent of the underlying resources, and ideally can be reused in many different business processes. To avoid re-engineering existing resources and to maintain flexibility, Business Service enablement is a new useful role for the integration layer.

### **IBM SOA Guidance**

The application of SOA and Web Services is broader than just Application Integration. They are not just applied at the solution assembly stage of a project. The concept of SOA should be applied right across the lifecycle from business planning to IT execution, in terms of identifying and managing Services as business, design and deployable assets.

Applying SOA across the lifecycle requires a consistent approach to ensure the right assets are delivered to support the business, and where relevant they are applicable across the enterprise.

The development of SOA best practices is just emerging and CBDI Forum believes that IBM are in a strong position with the recent introduction of a number of capabilities they have refined in the field through real customer engagements. Included in this,

- IBM is delivering a comprehensive approach to SOA called Service Oriented Modeling and Architecture (SOMA)
- IBM Global Services also offer a number of SOA Services<sup>2</sup> that assist organizations at each stage of the Service lifecycle.

There isn't scope in this report to look in detail at all aspects of SOA best practice. The following box highlights some examples of guidance from IBM.

---

<sup>2</sup> <http://www-1.ibm.com/services/us/index.wss/it/igs/a1002583>

- |  |  |
|--|--|
| <ul style="list-style-type: none"><li>• Establish a champion or executive sponsor for SOA projects.</li><li>• Divide the business into business components (cohesive activities which collaborate with other business components).</li><li>• Develop a SOA strategy which defines the business context, pain point, reference architecture and a living roadmap for adoption for a lines of business and/or enterprise.</li><li>• Assign service domain owners (functional, technology, application).</li><li>• Implement governance mechanisms to ensure that corporate SOA/ESB strategy gets implemented, delivered or acquired.</li></ul> | <ul style="list-style-type: none"><li>• Extend systems development methodology to address creation of business services with corresponding design attributes for services.</li><li>• Encapsulate key existing/legacy functionality, as appropriate, Take advantage of Middleware capabilities.</li><li>• Compose atomic services into coarse-grained business services.</li><li>• Build for consumability via an ESB approach; refactor services so that they are as broadly applicable, reusable and practical.</li><li>• Use top down and bottom up analysis to create business services which removes redundancy and creates opportunities for services. The middleware construct decision can play a major role.</li></ul> |
|--|--|

**Examples of IBM SOA Best Practice Guidance**

## Infrastructure for Application Integration with SOA

With any change in approach, the first questions many organizations will ask is what impact this has on their existing infrastructure. Is it capable of supporting the new requirements or does it need to be extended or even replaced? With regard to SOA and Web Services, organizations need to consider such issues as,

- What elements of their existing infrastructure need upgrading to provide explicit specific support for Web Service protocols
- Is support for XML enough or does it need specific support for Web Service protocols
- Can the same infrastructure support both internal and external Web Services
- Will the infrastructure be able to support more advanced Web Service protocols that are emerging such as those for transaction, reliable messaging and business process execution

Looking at the requirements outlined earlier, it also becomes apparent that organizations will often need to address these issues on an enterprise-wide basis. For example

- Coarser grained Business Services will need to be aggregated from resources that might currently be in disparate project silos
- To improve governance organizations need to ensure consistency of approach, management, and security of SOA across the enterprise
- Extract aspects of Service Infrastructure and Service Management from individual projects and applications so they do not reinvent the wheel in each project, but more importantly so that they do not implement them within their application solution itself such that it constrains future agility

### The Enterprise Service Bus

The infrastructure layer to facilitate SOA has become commonly referred to as the Enterprise Service Bus (ESB). Today, there is no precise definition of ESB that is agreed by the industry and is most often attached as a label by vendors to a middleware product that provides varying levels of support for Web Service protocols plus associated integration and connectivity technologies, for which they claim an SOA

bias. Nor is there any consensus on the functional scope of ESB, and as such the capabilities included by vendors in their ESB product set can vary widely.

CBDI prefers to consider ESB firstly as an architecture of infrastructure Services as shown in Figure 1. Once organizations have established the architecture for the ESB, then they can consider the products and technologies that provide the capabilities required. The ESB doesn't need to be a single product though there are other merits in dealing with a single vendor. More likely it is an architecture of Service-aware application integration, messaging and message broking, platform technologies, Service management and security products that are carefully chosen to work together.

In a federated environment you would expect each participant to have their own ESB managing incoming and outgoing Service Requests, and acting as the key infrastructure element between them and the application resources.

CBDI believes there are number of desirable characteristics that should be important when considering ESB products. The ESB should:

- Not force product dependencies on any Service Endpoints.
- Be the basis of an extensible framework into which other infrastructure capabilities can be plugged.
- Provide support for the full range of emerging Web Service Protocols, not just SOAP and WSDL.
- Be Policy and Meta-data driven. To enable easy configuration without programming, and ensure compliance with corporate policies.
- Co-exist with the existing Middleware and Application Integration infrastructure

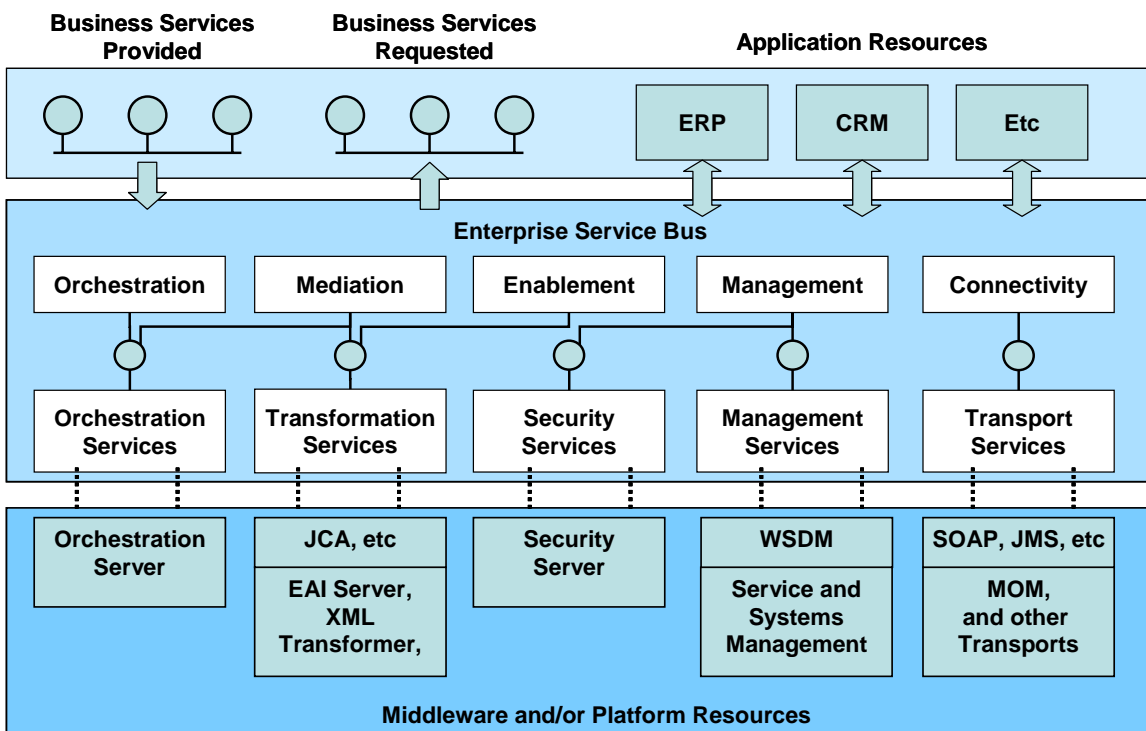


Figure 1 - The Enterprise Service Bus

## Anatomy of an Enterprise Service Bus

What the above suggests is that there isn't a minimum or maximum set of capabilities that scope the ESB. As stated, it should be seen as an architectural pattern, into which any number of Infrastructure capabilities could be plugged. Having said that, it is worth examining the main functionality that might be expected to form an ESB and also comment on the relationship of the ESB to other middleware as illustrated in Figure 1.

Many of the following functions will sound familiar. After all, similar capabilities are built into EAI, MOM and other existing application integration and middleware solutions. The key difference is that now the functions must be "service aware". For example,

- Service Mediation
  - Deal with abstract endpoints, rather than fixed binding to absolute addresses. This enables dynamic routing and broking
  - Within the constraints of security policies, be able to examine Service Messages to make content based decisions for routing, transformation, etc.
  - Enable aggregation and composition of new Business Services from other Services. Including the augmentation and enrichment of message contents by reference to other Services.
- Service and Process Orchestration
  - Orchestration of multiple Business Services
  - Management of events that trigger Service activities
- Service Creation and Enablement
  - Creation of new Business Services from existing resources and their APIs.
- Service Security
  - Federated authentication of Service Providers and Requestors.
  - Encryption of Service Messages

Table 1 lists the common functions of an ESB together with relevant protocol standards. It also shows relationships to existing application integration products and technologies that might be expected to play a role in implementing these functions.

Many organizations will of course need to integrate non service-based resources as many existing systems and files will not support Web Services or will have APIs that are inappropriate to be considered service-based. Even where the Application Server or container in which they are hosted supports Web Services, the resources will often not have been updated to use these and the maintenance effort and testing to do this may not be easily justified.

Therefore, the ESB architecture should include the ability to access these resources, and where appropriate could be used as the Service Creation and Enablement layer so that the resources can be included in an SOA without incurring the development cost of upgrading the resource itself. The Service Enablement functions included in Table 1 could be used to wrap the existing resources with Web Service protocols, but equally to transform them to other technologies such as MOM or an ORB that could be used in a Service Oriented manner. As such existing EAI products that include the adaptors for countless existing applications and file formats can still play a useful role within the ESB.

Functional Area	Functions	Relevant Protocols and Standards	Relationships
Service Mediation	<ul style="list-style-type: none"> <li>• Broking</li> <li>• Routing</li> <li>• Transformation</li> <li>• Aggregation and Composition</li> </ul>	<ul style="list-style-type: none"> <li>• WS-Addressing</li> <li>• XSLT</li> </ul>	<ul style="list-style-type: none"> <li>• MOM</li> <li>• EAI</li> </ul>
Service and Process Orchestration	<ul style="list-style-type: none"> <li>• Process Execution</li> <li>• Event Management</li> </ul>	<ul style="list-style-type: none"> <li>• BPEL</li> <li>• WS-Coordination</li> </ul>	<ul style="list-style-type: none"> <li>• Workflow</li> </ul>
Service Enablement	<ul style="list-style-type: none"> <li>• Wrapping</li> <li>• Protocol conversion</li> <li>• Transformation</li> <li>• Access non service based resources</li> </ul>	<ul style="list-style-type: none"> <li>• BPEL</li> <li>• WSDL</li> </ul>	<ul style="list-style-type: none"> <li>• EAI</li> <li>• Application Server</li> <li>• Development Tools</li> </ul>
Service Connectivity	<ul style="list-style-type: none"> <li>• Transport</li> <li>• Assured Delivery</li> <li>• Protocol Conversion</li> </ul>	<ul style="list-style-type: none"> <li>• SOAP</li> <li>• JMS</li> <li>• WS-ReliableMessaging</li> <li>• WS-Coordination</li> </ul>	<ul style="list-style-type: none"> <li>• MOM</li> <li>• Distributed Computing Technology (e.g. ORB)</li> </ul>

**Table 1 – Common Enterprise Service Bus Functions and Infrastructure Relationships**

### IBM ESB Capabilities

To support these requirements, IBM provides a comprehensive suite of products as listed in Table 2. To ensure cohesion, there is a single Eclipse-based console with common tooling for management and configuration across the ESB.

Functional Area	IBM Products
Service Mediation	<ul style="list-style-type: none"> <li>• WebSphere Business Integration Event/Message Broker</li> <li>• WebSphere MQ</li> </ul>
Service and Process Orchestration	<ul style="list-style-type: none"> <li>• WebSphere Event/Message Broker</li> <li>• BPEL support                             <ul style="list-style-type: none"> <li>○ WebSphere Business Integration Modeler</li> <li>○ WebSphere MQ Workflow</li> </ul> </li> </ul>
Service Enablement	<ul style="list-style-type: none"> <li>• WebSphere Business Integration Adaptors</li> <li>• WebSphere Application Server</li> </ul>
Service Connectivity	<ul style="list-style-type: none"> <li>• WebSphere MQ</li> </ul>

**Table 2 - IBM Products for ESB**

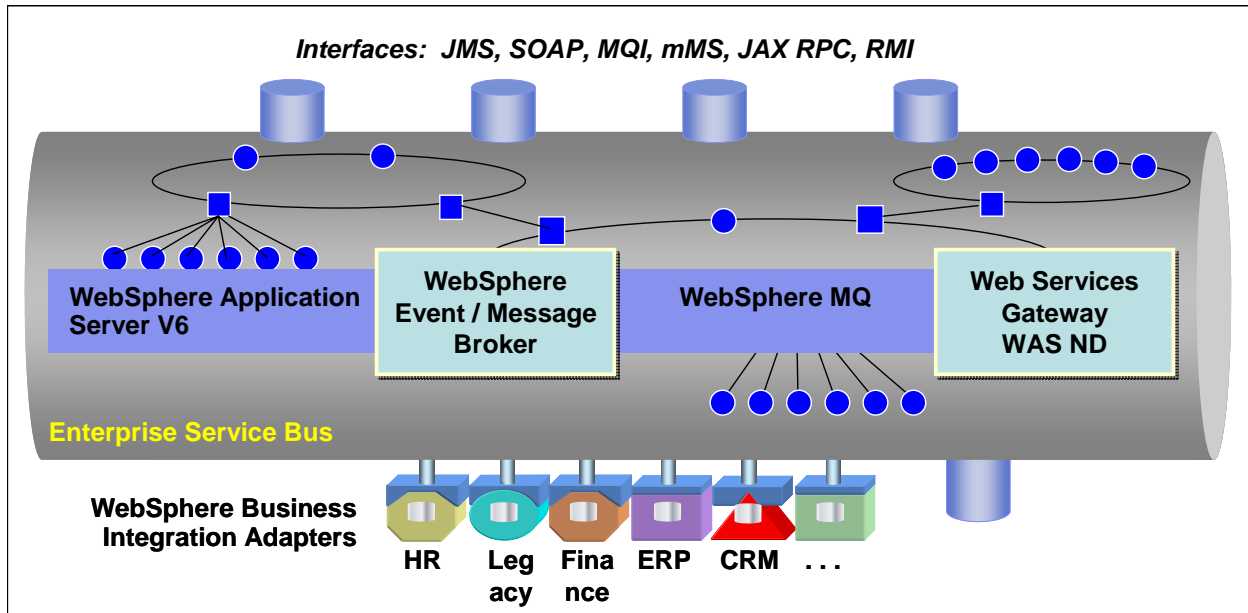


Figure 2 - IBM Products for ESB (source IBM)

As illustrated in Figure 2, the recently announced Version 6 of WebSphere MQ provides the foundation of the ESB.

WebSphere Business Integration Event/Message Broker provides message transformation and database integration, content and topic-based routing capabilities, plus message filtering and message warehousing. It also provides event Services including publish and subscribe, with support for numerous transport protocols.

The Web Services Gateway, available as an integrated component within WAS ND (WebSphere Application Server Network Deployment), provides a single point of control for access, validation and filtering of web services requests, helping to ensure the security of external Web Services.

## Service Management

Some ESB products include elements of Service Management. Though this might be thought of as a separate infrastructure domain, the drivers outlined earlier require Service Integration and Service Management to work closely together to deliver a more dynamic environment where decisions such as Service Routing can be made in real time based on Service Management feedback.

In the Service Management domain we can consider the following service aware activities listed in Table 3 that are related to application integration.

- Service Management - Enabling Service Mediation decisions to be made on the basis of Service availability and performance
  - Manage Services and Service Messages as resources in their own right, not just the other resources which form part of their implementation (and which might be shared across many services)
  - Management of Service Level Agreements, and monitoring compliance
  - Provisioning of Services and associated resources

- Business Performance Management - Enabling Service Mediation decisions to be made on the basis of the performance of business activities
  - Business Services at the right level of granularity and abstraction can enable monitoring of business level activities.
- Participation Management - Enabling Service Mediation decisions to be made on the basis of the type of participant and their access levels.
  - Managing access to Services, which may extend down to the operation, or message content level
  - Managing Service Level Agreements at a per Service/Participant level
  - Managing commercial aspects of Service usage where appropriate, such as metered (per request) billing, or mediation based on commercial contracts
- Service Security – Ensuring federated security across a end-to-end integration scenario
  - Closely related to Participation Management are the security functions to identify and authenticate participants.

Functional Area	Functions	Relevant Protocols and Standards	Relationships
Service Management	<ul style="list-style-type: none"> <li>• Service Monitoring</li> <li>• Service Level Agreement Compliance</li> <li>• Provisioning Service Resources</li> </ul>	<ul style="list-style-type: none"> <li>• WSDM</li> <li>• WS-Policy</li> <li>• WS-RF</li> </ul>	<ul style="list-style-type: none"> <li>• Systems Management</li> <li>• Web Service Management</li> </ul>
Business Performance Management	<ul style="list-style-type: none"> <li>• Business Activity Monitoring (BAM)</li> </ul>		<ul style="list-style-type: none"> <li>• Business Intelligence</li> </ul>
Participation Management	<ul style="list-style-type: none"> <li>• SLA Management</li> <li>• Usage Contracts</li> <li>• Commercial</li> </ul>		
Service Security	<ul style="list-style-type: none"> <li>• Authentication</li> <li>• Encryption</li> </ul>	<ul style="list-style-type: none"> <li>• WS-Policy</li> <li>• WS-Security</li> </ul>	<ul style="list-style-type: none"> <li>• Directory</li> <li>• Security/Identity</li> </ul>

**Table 3 - Service Management Functions and Integration**

### IBM Service Management Capabilities

IBM products that support service management in the context of Application Integration are shown in Table 4.

Functional Area	IBM Products
Service Management	<ul style="list-style-type: none"> <li>• IBM Tivoli Monitoring for Transaction Performance 5.3</li> <li>• Tivoli Monitoring for Web Services</li> <li>• WebSphere 6.0</li> </ul>
Business Performance Management	<ul style="list-style-type: none"> <li>• Websphere Business Integration Modeler/Monitor</li> </ul>
Participation Management	<ul style="list-style-type: none"> <li>• Web Services Gateway provides filtering and security</li> <li>• WebSphere Business Integration Connect</li> <li>• Tivoli Federated Identity Manager</li> </ul>
Service Security	<ul style="list-style-type: none"> <li>• WebSphere 6.0</li> <li>• Tivoli Access Manager for e-Business 5.1</li> <li>• Tivoli Federated Identity Manager</li> </ul>

Table 4 - IBM Products for Application Integration Service Management

### Business Integration Reference Model

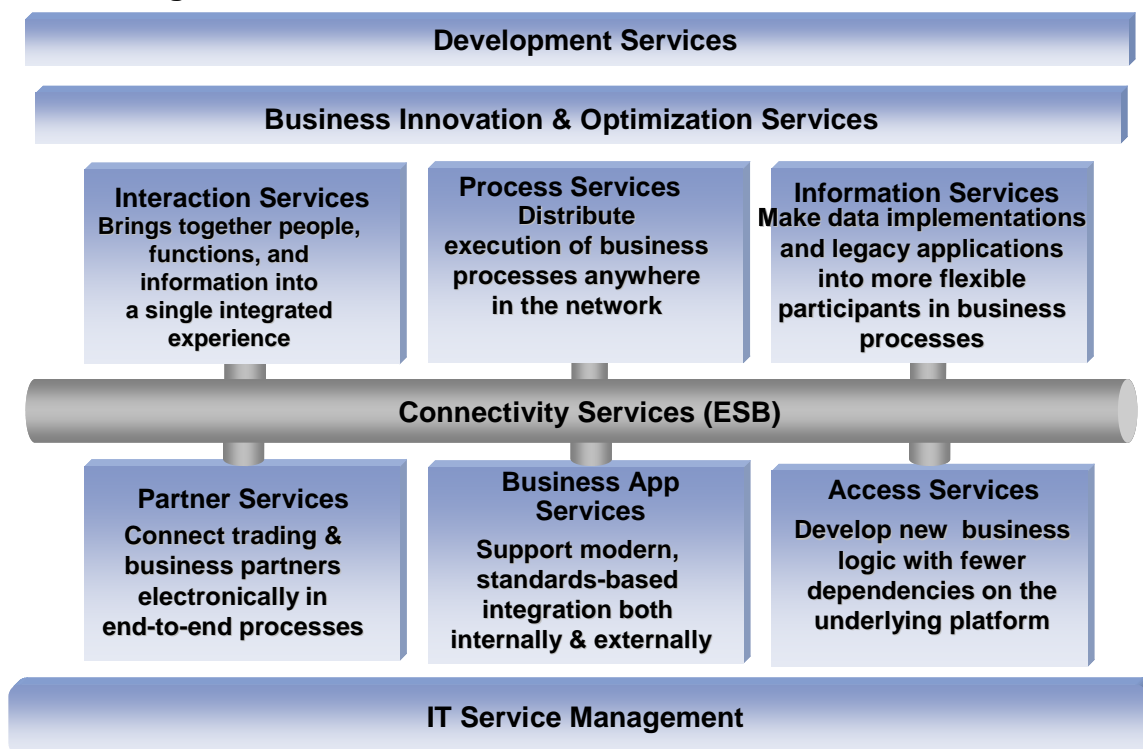


Figure 3 - IBM Business Integration Reference Model (source IBM)

These IBM capabilities sit within a broader Business Integration Reference Model illustrated in Figure 3. This model addresses many of the requirements outlined in the first section of this report recognizing the

need to include people, partners and processes as well as application resources in the broader scope of integration.

It is beyond the scope of this paper to detail each service domain shown. However this model illustrates a key role of the ESB – to move information to where it is needed, when it is needed.

## **Service Oriented Infrastructure**

The previous sections outline the infrastructure for application integration that needs to be made “Service Aware”. That is, it must support the delivery and the runtime execution of Business Services that are part of an SOA and facilitate the message passing within the business solutions that are integrated or assembled from the Services within that architecture.

This is not the same as the application integration infrastructure itself being Service Oriented – i.e. the middleware itself exposes a set of Infrastructure Services, which exhibit the same principles such as loose coupling as are demanded of the Business Services they support. As shown in Figure 1, the middleware and infrastructure resources ideally expose their own set of Services that form the ESB.

The importance of this is two-fold. Firstly, it means that the infrastructure itself can be virtualized and federated in the same way as application resources can via SOA. Secondly it provides an important mechanism by which existing infrastructure resources can be reused in an ESB – again in the same way that existing application resources can be reused in an SOA.

Table 1 and Table 3 highlight the relevance of existing application integration capabilities. Many organizations will have significant investment in relevant products and the solutions they have built with them. Ideally, they are not usually looking to replace this infrastructure, at least in the short term. More likely they are looking for ways in which they can extend it to support new requirements. Key to this will be vendors adding Service interfaces to their existing infrastructure products where relevant so that they can not only be configured through a standard UI and tool framework such as Eclipse, but also programmatically configured through Web Service interfaces.

Today, application integration infrastructure products such as IBM WebSphere support Java based interfaces such as Java Messaging Service (JMS) and Java Connector Architecture (JCA) that enable WebSphere products to be configured and extended in a standard way. Increasingly in future CDBI Forum also expects to see standardized Web Service protocol based Infrastructure Services emerging, such as WSDM (Distributed Management) that provides a mechanism by which federated Systems Management tools and platforms can share information and expose management functions – the key benefit being that Web Services do not impose a Java or other dependency on the infrastructure endpoints.

## **Service Governance**

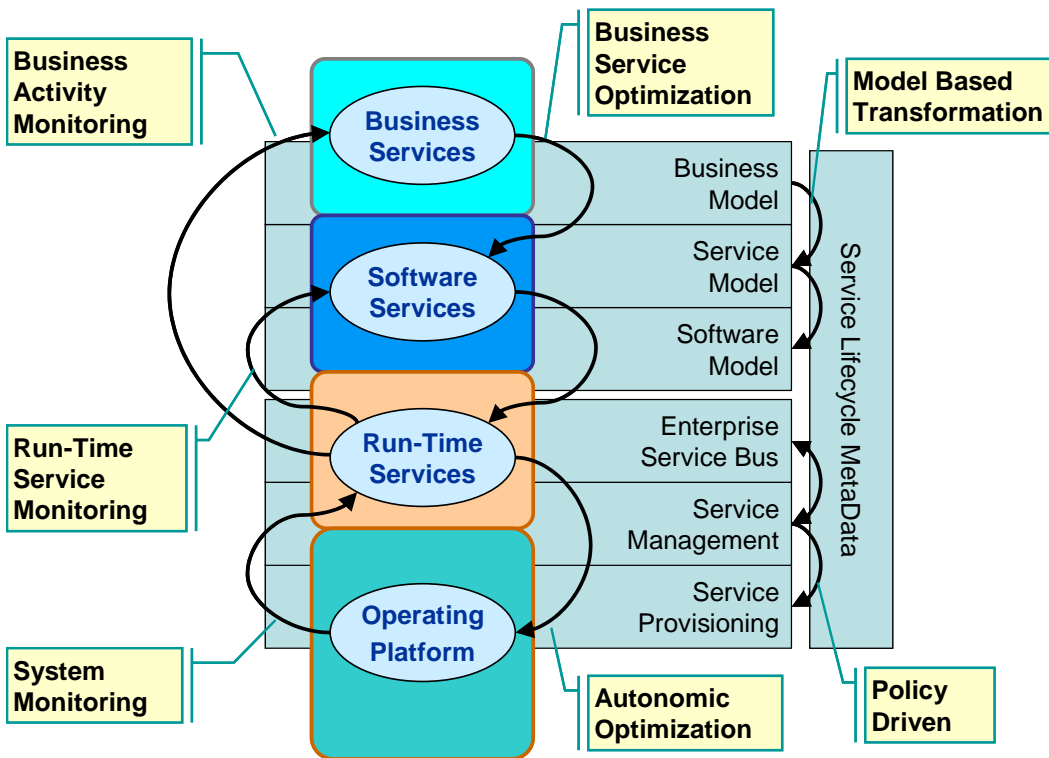
Governance can be seen from many perspectives. SOA both requires governance, and can help deliver improved governance. Table 5 illustrates some SOA governance requirements and corresponding improvements

Governance Requirements	Governance Improvements
Governance of the Service development process is required to ensure that the right Services are delivered to support the business, and the deployed Service is a correct implementation of requirements	Well designed meaningful Business Services enables monitoring of business level activity
Governance of Service execution is required to ensure that policies are effective	Flexible policy compliance can be built into Service execution to ensure changes in compliance are implemented immediately

**Table 5 - SOA Governance Requirements and Improvements**

Figure 4 illustrates how the proper application of SOA can enable full lifecycle governance. The ESB can play a central role in this by

- Managing flexible policy deployment
- Run-time Service Monitoring to check policy compliance, and provide audit trails
- System and Service monitoring
- Facilitating feedback loops to systems management for resource optimization
- Facilitating feedback loops to the business for process optimization



**Figure 4 - Improving Service Governance**

## Roadmap Planning

Acquiring or upgrading the full range of infrastructure capabilities required for SOA is not something that organizations need to address via a “big bang” approach. Instead organizations should:

- Grow the capability in line with their actual need and the SOA scenarios in use. E.g. they have no need to worry about managing external participants if the Services are only used internally
- Implement capabilities in line with their level of maturity, as indicated in Figure 5
- Expect to reuse existing infrastructure capability where applicable

To facilitate this CDBI Forum recommend the use of a roadmap planning approach, which is a useful device to coordinate many activities that are often widely dispersed in terms of time, geography and accountability. CDBI Forum suggests a roadmap framework such as that illustrated in Figure 5 which can be used to plan the introduction and rollout of SOA and Web Services. The y axis provides a structure for thinking about timing and interdependency which are simply clustering mechanisms, which specifically remove any organizational relevance. We refer to these as streams.

The x axis then represents stages of maturity:

- Early learning is undertaken in a deliberately uncoordinated manner, with the minimum of formality.
- The integration phase then requires certain matters to be managed in order to achieve sensible levels of consistency, which will hopefully reduce overall cost to the organization as well as establish common foundations for matters such as classification, core infrastructure services etc.
- The reengineering phase is then focused on creating enterprise level services and platforms
- Maturity is all about convergence of the IT Services and the business products and processes.

<b>Plan &amp; Manage</b>	<i>Development and coordination of common policies and practices</i>			
<b>Infrastructure</b>	<i>Cluster of topics and deliverables to establish new infrastructure for SOA</i>			
<b>Architecture</b>	<i>Establishing enterprise architecture and frameworks</i>			
<b>Process</b>	<i>Establishing the Service Oriented delivery process for the Service Lifecycle</i>			
<b>Projects</b>	<i>Recommended project profiles and patterns</i>			
	<b>Early Learning</b>	<b>Integration</b>	<b>Reengineering</b>	<b>Maturity</b>

**Figure 5 - CDBI Forum Roadmap Planning Framework**

For each stream, organizations then consider what state the organization should be in according to industry benchmarks and its own goals, and then plot the activities required to move from one level of maturity to the next.

We highlight a broad view of Roadmap planning in this paper because modernizing the integration infrastructure to support SOA is not going to be something that happens in isolation. Rather it is one

stream in a parallel set of activities designed to move the whole organization forward into SOA and enabling a more responsive business and IT environment.

Core to the success of introducing a significant new approach such as SOA is dealing with the potential organizational impact. For example managing responsibility for the delivery of new Business Services that might cut across existing projects that are currently responsible for silo application resources.

### IBM Business Integration Adoption Model

To assist organizations in planning their SOA roadmap, IBM have recently introduced a Business Integration Adoption model. Table 6, illustrates the business and IT drivers for each of the adoption levels in the model. The value in this model is that it enables organizations to frame their new and existing projects within the context of their business integration and SOA vision.

Adoption Level	Business Domain	IT Domain
<b>Connect:</b> Assure reliable and flexible information flow between diverse applications & systems	<ul style="list-style-type: none"> <li>• Guarantee transactions</li> <li>• Integrate partners</li> <li>• Streamline processes</li> </ul>	<ul style="list-style-type: none"> <li>• Verifiable, reliable information exchange</li> <li>• Secure application access</li> <li>• Reduced integration costs</li> </ul>
<b>Integrate:</b> An integration framework that supports interoperability among heterogeneous environment – removing barriers to building an integrated architecture powered by web services and non-web services applications and integration approaches	<ul style="list-style-type: none"> <li>• Rapid offering introduction (e.g., online retail lending)</li> <li>• Expedited payments exception resolution</li> <li>• Web access to core systems</li> </ul>	<ul style="list-style-type: none"> <li>• Technology standards as a business enabler</li> <li>• Reduced IT complexity &amp; headcount</li> <li>• Reusable IT systems &amp; data</li> </ul>
<b>Automate:</b> The orchestration of business and IT processes to align IT with the business goals to grow revenue while containing costs.	<ul style="list-style-type: none"> <li>• Regulatory change</li> <li>• Post-merger integration of firms</li> <li>• Automation of loan origination process</li> </ul>	<ul style="list-style-type: none"> <li>• Reduce process inefficiencies</li> <li>• Scale to meet increased demand</li> </ul>
<b>Optimize:</b> A holistic approach to transform and manage a business by aligning strategic and operational objectives with business activities and supporting IT services	<ul style="list-style-type: none"> <li>• Differentiated service</li> <li>• Managed risk &amp; liquidity</li> <li>• Automated, informed processes</li> </ul>	<ul style="list-style-type: none"> <li>• Asset control &amp; alignment</li> <li>• Greater performance visibility</li> <li>• Optimized resources</li> </ul>

**Table 6 - Levels of Adoption - IBM Business Integration Adoption Model**

So that organizations can take stock of where they are in terms of adoption and the types of activity they might consider in order to achieve their vision, IBM has also introduced a SOA Self Assessment Service<sup>3</sup>. This is an online diagnostic, co-developed with IBM Global Services SOA Center of Excellence in which organizations answer questions about practices in key SOA categories of process, architecture, application, and infrastructure.

<sup>3</sup> [http://www.ibm.com/websphere/soa\\_assessment](http://www.ibm.com/websphere/soa_assessment)

From this, the assessment service generates a custom report that is mapped to the Business Integration Model and illustrates to organizations:

- Their level of SOA sophistication described in clear, straight-forward terms
- The benefits and advantages that are already attainable with their existing integration infrastructure and current SOA readiness
- Some of the benefits they can expect in the future as they evolve their integration capabilities and service-orientation
- Recommendations for achieving greater business flexibility through improved SOA readiness

### Application Integration Infrastructure Roadmap

Against the backdrop of adoption levels, organizations need to make the choices as outlined in Table 6 for the modernization of each component of the Application Integration infrastructure.

Action	Requirement	Example
Retain	Can continue to play role in SOA as is	MOM
Upgrade	Must be upgraded by vendor to make Service aware	Message Broker
Extend	Modules or components that compliment or plug in to existing capability to add SOA capability	Service Management extensions to Systems Management
Replace	Infrastructure cannot be made Service aware	Products no longer supported by vendors
Acquire	New functionality that does not existing in current infrastructure	BPEL Orchestration Engine
Retire	Redundant infrastructure no longer required	Products superseded by new SOA infrastructure

**Table 7 - Application Integration Infrastructure Roadmap Actions**

A basic high level roadmap for the project tasked with modernizing the Application Integration infrastructure project would be

- Identify infrastructure roadmap actions as outlined in Table 7 to provide support for
  - Web Services – identify the steps to upgrade to support both basic and emerging advanced protocols.
  - SOA – in terms of how the infrastructure helps in better abstracting the business service away from the back end resources. For example, can the infrastructure products such as the ESB provide mediation capabilities that removing the need for low-level development work, and improve loose coupling?
  - Service Management – assess whether upgrades to existing Systems Management tools are appropriate, or do new Service Management products need to be acquired.

- Consider the upgrade approach. For example, is the infrastructure upgraded in stages in line with roadmap planning? Is it upgraded in step with emerging needs and the evolution of Web Service protocols? This should be balanced against the potential benefits of investing in an infrastructure component that perhaps meets current and emerging needs in one investment.
- Assess the organization impacts, for example understanding the different responsibilities of the individual projects vs. central IT. Clarify which of them are responsible for putting which pieces of the Application Integration infrastructure in place
- Build the ROI case and acquire funding for infrastructure upgrades where required
- Establish applicable enterprise wide policies for Services, such as SLA and security.

### **Service Based Application Integration ROI**

Some of the general measures by which the ROI on modernizing application integration can be ascertained are listed in Table 7. Some of these will be straightforward to derive from IT costs and savings such as the cost of software upgrades or reduction in developer effort. Others might require more complex measurement of business performance. Some business benefits may be hard to generalize in the overall ROI as they are situation specific, however they will still factor in building the ROI case for the individual solution projects and justification for using a Service Oriented integration approach.

The study by Software Strategies referenced earlier concluded that around 13% of the IT budget could be saved each year through the use of Enterprise Integration Middleware such as IBM WebSphere compared to alternative approaches that rely mainly on custom built solutions that are developed and maintained in-house.

CBDI Forum believes that this saving could be further improved through the adoption of an SOA approach for the reasons outlined in Table 8 and elsewhere in this paper, whilst at the same time enabling a more agile integration environment that delivers further savings over time by removing the need to constantly re-build integrations because of business and IT change.

IBM customers already provide evidence that these benefits can be achieved. For example

- Xerox required an ESB to provide agile integration in line with industry standards. They are using WebSphere MQ, WBI Message Broker and WebSphere Application Server to process over 2 million messages a month linking 50 applications and have achieved greater than 99.9% availability. They have seen a reduction in TCO due to the reuse of standardized interfaces, loose coupling of resources, and the ability to reuse existing resources. This has delivered a reduction in the time to implement changes by 75%.
- Traveler, one of the biggest names in the global foreign exchange market has seen an 85 percent reduction in the time required for customer processing of foreign exchange transactions by implementing a Straight Through Processing approach using WebSphere Business Integration to connect their front and back end systems. Furthermore, the consolidation of application infrastructure is expected to reduce application support costs by 20%<sup>4</sup>.

---

<sup>4</sup> <http://www-306.ibm.com/software/success/cssdb.nsf/CS/JSTS-64MQR2?OpenDocument&Site=wsssoftware>

Business Value
Reduction to business process execution time and introduction of real-time behavior.
Ease of integration makes it attractive to new partners and customers.
Availability of Web Services enables new channels and business models
IT Value
Reduced TCO
Lower development through consistent programming model exposed by infrastructure services.
Lower developer training costs with a single consistent approach for all integration scenarios.
Reuse of ESB and infrastructure across the enterprise, rather than re-invention in each project.
A fully service aware ESB can also facilitate resource virtualization to reduce the cost of IT operations.
Reduction in Time to Solution
Common framework, based on open standards means connections can be quickly established.
Reduction in Time to Change
Policy driven approaches enable rules to be quickly effective across all relevant integrations without having to adjust each one individually
Provide certain levels of autonomic behavior to enable systems to respond automatically to changes.
Improved Governance
The use of an ESB to ensure consistency of service provision and service management across the enterprise.
Active WSM and SLA monitoring will reduces cost of problem resolution, and avoid penalty payments for non-compliance with SLA.
Business Activity Monitoring can be used to improve business awareness of the state processes and activities – so they can avoid business process problems – not just system problems.
Policy driven approach enables corporate policies to be quickly effect
IT Cost
Development
Understanding domain wide requirements, and designing Business Services that could be reused across the domain will initially cost more than Business Services that are based solely on the needs of the specific project. This can be clawed back through reuse.
Infrastructure
Upgrades may be required to the existing application integration infrastructure to support SOA
There may be acquisition and implementation costs for new components
There will be rollout and training in SOA approach

**Table 8 - The Value and Costs of Service Oriented Application Integration**

## Asset Modernization

Organizations typically have a high expectation that they will reuse existing application assets in new SOA scenarios. One of the basic principles of SOA is that loose coupling should enable existing implementations to continue operating behind a new Service façade, and a key role of the ESB is Service enablement as listed earlier.

Using IBM WebSphere as the basis for an ESB therefore enables it to play a dual role in providing Service Mediation together with Service Enablement thanks to a rich palette of adaptors for both packaged applications and multiple technologies that provide access to existing resources.

However, organizations must also consider that existing applications as they are implemented today may not always be ideal for the new agile, on demand scenarios they intend to deliver. For example the process embedded in the application might run contra to the behavior of the new scenario, or the internal dependencies within the application and with other applications might create unwelcome Service dependencies that negate some benefits of loose coupling.

The modernization of existing assets is beyond the scope of this report. However, we suggest that organizations initiate a modernization roadmap for these assets in parallel to their infrastructure roadmap outlined above. In many cases new interfaces will be sufficient, but in others some reengineering may be required.

## Summary

It is difficult to see how the behavior outlined at the beginning of this report can be delivered without modernizing application integration. SOA provides the approach, and Web Services the means by which this can be achieved.

However, keeping pace with changing requirements is never easy. The current transition to SOA and the use of Web Services places added pressure on maintaining the infrastructure because it presents a shift in approach, not just new technology. It is much easier to add Web Service capability to the infrastructure when it is supported by a vendor such as IBM who is leading the development and standardization of Web Service protocols and is able to provide implementations at the earliest opportunity.

IBM's ESB provides a blend of innovative technology and products whilst at the same time continuing to provide support for existing customers and giving them a clear path towards enabling a more responsive organization with SOA. This approach minimizes investment by facilitating the reuse of both existing application assets and the integration infrastructure already in place. For example: An application implemented using J2EE components may be plugged into the ESB using advanced Web Service protocols such as WS-ReliableMessaging or through a Java Message Service (JMS) interface using a MessageDrivenBean. Similarly a Microsoft .NET application can participate in the SOA by using the same Web Service protocols, or both could leverage IBM WebSphere MQ as a common MOM transport layer in the ESB if that was already in place. Meanwhile, in another part of an organization, applications that perform nightly batch processing based on the bulk dump, load, and transfer of data may now connect to the ESB using an FTP or file based interface, and take advantage of the ESB's mediation capabilities to break up the bulk data into its individual parts for consumption and use by other endpoints on the ESB. This can be accomplished by using open standards based middleware and services.

Delivering the behavior outlined in this report requires an enterprise-wide response to integration if the requirements are to be met. End-to-end business process optimization is difficult to achieve whilst silo projects continue to use incompatible technologies and approaches. At the same time, SOA and Web Services provides a mechanism by which individual projects can still focus on their specific requirements and leverage whatever technology they feel is appropriate to their solution as long as they commit to using the ESB as the common integration backplane. As such CBDI Forum emphasizes the need for roadmap planning to move the whole organization forward in a consistent manner.

Author: Lawrence Wilkes, CBDI Forum - [lawrence.wilkes@cbdiforum.com](mailto:lawrence.wilkes@cbdiforum.com)

This report was commissioned by IBM Inc.

## Links

IBM WebSphere Business Integration:

<http://www-306.ibm.com/software/info1/websphere/index.jsp?tab=products/businessint>

IBM SOA and Web Services

<http://www-306.ibm.com/software/solutions/webservices/>

IBM SOA Self Assessment

[http://www.ibm.com/websphere/soa\\_assessment](http://www.ibm.com/websphere/soa_assessment)

CBDI Web Services and SOA Roadmap:

<http://roadmap.cbdiforum.com>



## Independent Insight for Software Oriented Practice

### Subscribe to the CBDi Forum

*The CBDi Journal is published monthly. In addition to the Journal, subscription includes electronic access to current and all back numbers that now represent a significant resource library. There are individual and corporate subscription schemes. Corporate subscription includes access to Powerpoint libraries and our workshop materials.*

*For more details and to  
subscribe see:*

*[www.cbdiforum.com](http://www.cbdiforum.com)*

### CBDi Raison d'être

We aim to provide unique insight on component and service oriented technologies and processes for the software industry and its customers. To provide high quality analysis and other information resources on best practice in business software creation, reuse and management. To maintain the highest level of independence.

### Modus Operandi

The CBDi Forum has a number of channels:

- Subscription services – provision of continuous commentary and information.
- Workshops and seminars – providing indepth briefing and guidance on advanced architectures, processes and practices.
- Consulting – including related product management and marketing advice, application audit and guidance, technical and business evaluation for investment

### How we compare with others

We are not a mass market, media oriented organization. All material published by the forum is unique. The majority of material is published by our own analysts, or commissioned from others, and under strict editorial control to ensure accuracy. We rigorously exclude spurious marketing.

We provide depth insight which covers a narrow topic footprint in a deeper way than the other analysts, and in particular cover not just the technology, but also the architectures, usage, practices and processes.

Also we are unusual as analysts; we do not simply echo what the vendors say, we are a think tank, identifying new ideas, opportunities and providing stimulus for thinking. We are thought leaders providing ideas generation and a rich source of conceptual thinking based on practical, real world feedback.

### Who Reads the CBDi Journal

Technical leaders including Technical and Application Architects, Business Analysts, Consultants, CTO's, Designers, Product Strategists, Senior Developers, Project Managers, CIO's etc. Subscription is split roughly 40% USA and 50% Europe.

### Contact Us

For further information on any of our services contact us at:  
[info@cbdiforum.com](mailto:info@cbdiforum.com) or +353 28 38073 (international)

**IMPORTANT NOTICE:** The information available in CBDi publications and services, irrespective of delivery channel or media is given in good faith and is believed to be reliable. CBDi Forum Limited expressly excludes any representation or warranty (express or implied) about the suitability of materials for any particular purpose and excludes to the fullest extent possible any liability in contract, tort or howsoever for implementation of, or reliance upon, the information provided. All trademarks and copyrights are recognised and acknowledged