



**WebSphere** software

## **Building an ESB without limits.**

*By Leif Davidsen, IBM Software Group*

---

<b>Contents</b>
<b>2 Introduction</b>
<b>3 The connectivity challenge</b>
<b>3 Addressing connectivity with an ESB</b>
<b>6 Can an ESB solve all your IT problems?</b>
<b>8 Your ESB selection could limit your business</b>
<b>11 The extent of integration required in your SOA</b>
<b>14 Qualities of service</b>
<b>15 ESB offerings compared</b>
<b>17 Selecting your ESB and your ESB vendor</b>
<b>19 ESBs from IBM</b>
<b>20 ESB without limits—extending the bus for a complete SOA solution</b>
<b>27 IBM clients choose an ESB without limits</b>
<b>27 For more information</b>

## **Introduction**

Connectivity across the IT environment for your enterprise is now increasingly critical for integrating existing and new applications, processes and services in an efficient and cost-effective manner. Reliable connectivity, without the need for complex additional programming in the applications to verify or assure the exchange of information, is required for your service oriented architecture (SOA). Without addressing this need, ever-increasing IT complexity, along with costs for development and maintenance, can end up exceeding your ability to push for the innovation and more-flexible integration you need to enable your SOA to respond faster to new business opportunities.

Connectivity is just one of the five key entry points to SOA. Others are people, process, information and reuse. All have a part to play in a successful SOA deployment. However, connectivity and reuse are IT-focused entry points to SOA and as such, affect all aspects of an SOA deployment – particularly because SOA becomes more widely used across the whole enterprise, requiring the IT infrastructure to support the business activities.

As your business becomes more service-oriented, your connectivity solution needs to enable more parts of your enterprise to be connected seamlessly, reliably and securely – from handheld devices, sensors and application servers to appliances and mainframes, as well as providing the framework for the other SOA entry points.

Connecting all your systems, applications and services, and using your business logic and enterprise data, can help ensure that you have the most-connected SOA available, providing the infrastructure for your business to be as flexible and responsive as you need it to be.

### **The connectivity challenge**

As with so many other aspects of business life, for IT the only thing that is certain is change. Even the simplest application or function can gradually become more and more complex, as routine changes are made to the application itself or the environment, or as connections to other parts of the business and other applications are made and altered.

Eventually your business environment can be overwhelmed by IT complexity – with most of your IT budget dedicated to just keeping things connected and working in your IT environment, rather than developing and enhancing your business. Indeed some businesses might find that around three quarters of their IT budget is simply devoted to maintaining their environment rather than being spent on new projects.

With businesses wanting to extend the reach and responsiveness of their operations as part of SOA projects, there is an increasing need to help ensure that connectivity does not become an inhibitor to a widely deployed SOA, including interconnected processes and applications that are flexible, scalable and secure. Without addressing the connectivity needs of such an environment and an SOA, then the goals of your SOA deployment are unlikely to ever be achieved.

### **Addressing connectivity with an ESB**

A cursory inspection of many papers on SOA would suggest that the solution required for connectivity in an SOA is an enterprise service bus (ESB).

However, this white paper dives deeper into this area, to answer why you need one in an SOA, and what the ESB will do for you. It also addresses whether it is the only thing you need, and if all ESBs are pretty much the same, delivering a standard set of functions for a standardized environment.

As Figure 1 shows, today's connectivity environments can be tremendously complex. Trying to ignore this complexity and the impact on your business won't help you to address it—in fact, it will only get worse. Many previous approaches to this connectivity problem have added to the complexity, so any solution to your current problems needs not only to address the fundamental complexity issue, but also to ensure that it can handle all the past environments. But how can businesses use an ESB to address this problem?

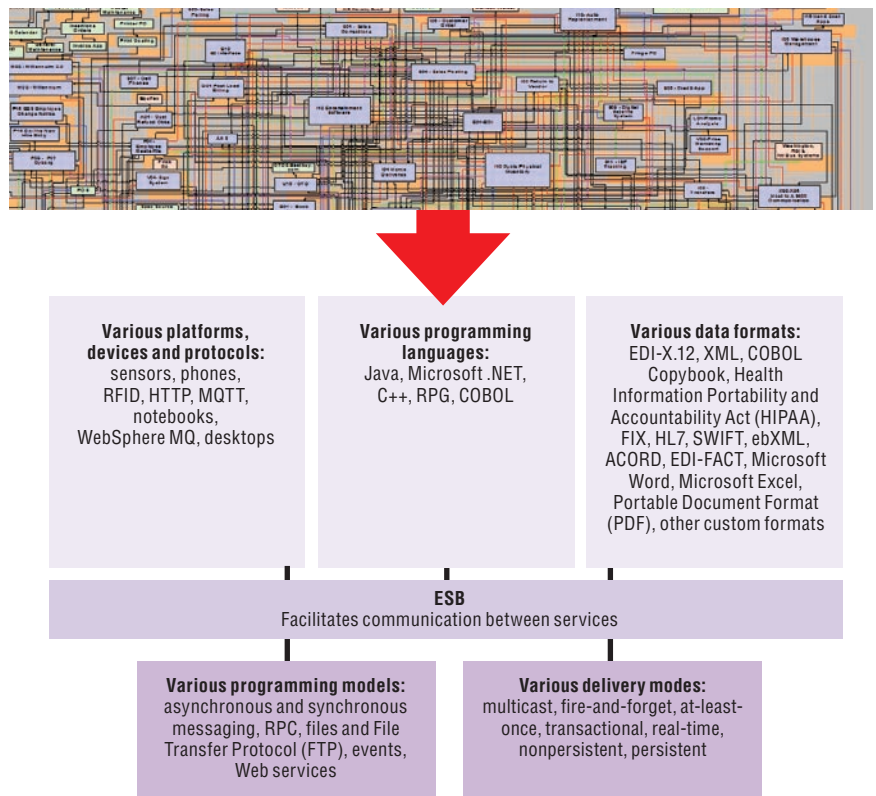


Figure 1. An ESB can help you handle the complexity of your organization's IT infrastructure.

An ESB does a number of tasks to meet the connectivity needs of applications and services. It matches and routes messages between services. It converts transport protocols between requesters and service providers. It transforms message formats between requesters and service providers. And it distributes business events to and from disparate sources.

An ESB does all this in an intermediary layer sitting between applications, systems and services, helping to ensure that not only do applications not get more complex overall due to their connectivity interfaces, but in many cases that the complexity could be removed from applications, or at least bypassed, decoupling all the hardcoded links between applications and systems (see Figure 2). The goal is for all applications, services and other IT assets to only connect to the ESB, rather than to connect directly with any other part of the infrastructure.

If you do this for all applications and for all connectivity invocations, then applications would only require a single interface connecting the application itself to the ESB for all interactions. And any other asset also connected to the ESB could interact with this application through this interface, with every other asset also only requiring a single connectivity definition to connect it to the ESB. And these interfaces, where any processing or complexity is required, are predominantly defined in the ESB, and can be maintained there, leaving the applications unchanged as the environment around them changes.

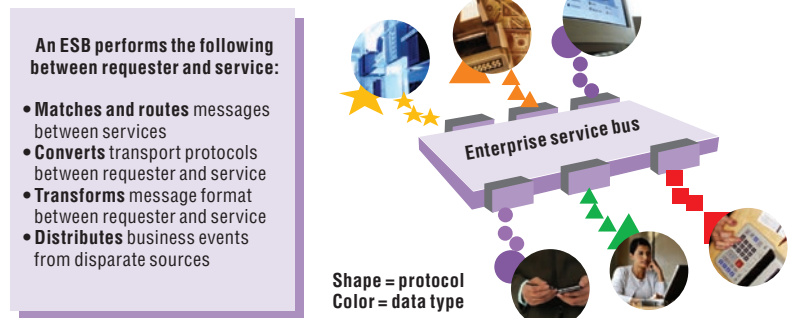


Figure 2. An ESB provides a variety of connectivity functions.

This decoupling of connectivity helps ensure that interfaces within applications become far less complex, which in turn helps reduce the cost and risks of changing those applications, and makes it far easier, quicker and less risky to use them in new ways, particularly enabling more applications and functions to be accessed and reused as services. This accelerated progress towards an SOA can enable your business to become more flexible and responsive than ever before.

**Can an ESB solve all your IT problems?**

Despite the great promise of ESBs, and even though the theory of decoupling your applications is sound, the connectivity problems of your business are not overcome by simply buying and implementing an ESB. In fact, many ESBs today only address a limited subset of the connectivity problems your business is likely to be experiencing. ESBs, typically designed for specific SOA environments or typical services-related IT problems, can be limited in their abilities to perform some of the basic tasks your business needs them to address, or they can be limited in how they work with the rest of your business. The ESB might be ideal for a specific project but that same ESB might not provide the flexible IT-connectivity infrastructure you were looking for across your wider business, and the ESB might not be able to support the other existing parts of your infrastructure and the rest of your SOA projects.

These failures in scope, scale and overall capabilities can mean real difficulties for your business if you become dependent on a single limited ESB to solve all your business-connectivity needs. If you can't easily connect all parts of your business and all your business and IT assets, then how will you be able to provide your staff, business partners and customers with all the information they need to do business with you? Will the promise of reuse and flexibility under SOA prove to be baseless?

When you are able to spend your IT budget on new and innovative projects to enhance your business or to address new opportunities, it is likely that you might choose to design and build these projects to use Web services programming methodology and XML-type data structures. If you are connecting these new systems using XML and Web services to your existing infrastructure, can you be sure they work together well, or will the different architectures, programming constructs and data formats cause new problems for your business? There is a risk that instead of enhancing your business by using the latest standards-based approach, you have instead added to your problems by creating a new set of incompatible assets that are unable to connect to your existing assets. If you have implemented an ESB to connect your newly built assets together, is the ESB also able to connect those new assets to the older assets, running on different systems, written in different languages, connecting over completely different protocols and exchanging very different data with different qualities of service expected? Although almost all IT problems can be reduced to a simple matter of programming, it is vital that your ESB implementation does not get bogged down with building new infrastructure code that is going to be as complex and inflexible as the very set of interfaces the ESB is designed to replace.

If you take a step back from the issues directly connected to your IT assets and the implementation of an ESB, what are the wider effects? As things get more connected and more interdependent, can you be sure you know what is happening in your business, even when information is flowing faster than ever before and with changes being requested and implemented faster as well? Is this flow of information between applications, systems and services as secure as possible? And just as important, can you prove that to auditors? With a more-connected business providing instant, up-to-the-minute access to all your business information, suddenly a failure or breach in any one part of your business could affect everything. Just as the rewards for success in implementing your connected SOA are high, the risks of failure can be severe if the solution fails to deliver, or is unable to cope with the challenges of your IT environment.

Though connectivity is critical in your business and for the benefits you want to achieve through SOA, connectivity by itself is not going to be the whole solution. An SOA is delivered through the implementation of many components. Your business is going to be concerned with process, management and governance – all part of SOA, but not necessarily delivered as part of a standard ESB, or even supported by one. This is a key point. Simply implementing an ESB to address some of the connectivity needs of perhaps a single project could affect your business on a far wider scale if it does not fit into your wider infrastructure. You don't want to slow down your ability to make changes to your business and infrastructure because your choice of ESB won't let you. Whether your ESB is selected to be the only ESB required for all of your enterprise, or whether your ESB is a local decision selected to meet the needs of a small project, the wider needs of the business can suffer from an ESB that is unable to cope with the overall architecture in the business.

**Your ESB selection could limit your business**

ESBs are clearly important, and if you don't pick the right ESB, you could be creating more problems for your business – constraining yourself by adding complexity and inflexibility to your infrastructure instead of making your business more dynamic and responsive.

Making the decision about your ESB could be the most-important strategic decision you make for your business IT infrastructure. After you make your decision, can your business continue to grow? Can it respond to the challenges of tomorrow – never mind next year? Even if you want to use Web services and other standards in the future, how important are your existing systems? How long will you continue to rely on them, all day, every day?



In almost every business, newly developed standards-based applications are swamped in number and complexity by existing applications and systems, developed and maintained over years, and through which the bulk of the business transactions flow. Some of these might be scheduled for updates, such as being adapted and reused as services, but with many of them likely to be back-office applications that run in production uninterrupted, they are unlikely to be a high priority for updates. However, the overall challenges to the IT infrastructure imposed by SOA are likely to drive changes where there might not have been a need for change before.

A critical business function that can easily be overlooked when overhauling architectures and designing new systems is business reporting. You want to be able to report on your business – for your status meetings or compliance audits. Any of these areas can be affected by your choice of ESB, and by the ways in which it delivers value to your business, without adding complexity. Your ESB should enable you to connect your business together, not simply connect a few select systems, based on standards, with any systems for reporting needing to be tacked on afterwards, creating additional work and nullifying the goals of SOA. A complete connectivity solution should ensure that an ESB provides flexibility in choosing which aspects of the business require monitoring at both an individual task level, and monitoring the business status. The monitoring, though capable of using the ESB to access the data flowing around the business, should be separate from the ESB itself, because the end-to-end functioning of the ESB is just as critical to the business as any application, and businesses are likely to need to monitor and report on the ESB activity status in the same way as they keep track of other critical systems.

A commonly tracked and reported aspect of the business is the status of transactions. You need to track a business transaction on a number of different levels, closely mirroring the different components of your SOA. At the higher business and process level, a transaction can be the successful creation of a new insurance policy, or the final payment on an insurance claim. However, from an IT level, those same transactions can be broken down into individual transactions involving individual exchanges of information between applications and databases so that they reflect the IT needs of the business that implements those processes. These individual IT units of work have to be tracked and action taken in the case of failure, in much the same way as action would be taken in response to the failure in creating a new insurance policy, or a directive to order some new stock, or any other business event. This decomposition of business activities into specific IT events and transactions reflects the different entry points to SOA described previously, and illustrates the fundamental importance of the IT entry points of reuse and connectivity to any aspect of SOA, further demonstrating that the ESB needs to be a highly flexible and capable infrastructure at the heart of your SOA. It is through your ESB that all your business transactions flow, whether through individual units of work that are exchanging information between applications or through the orchestration of processes that is coordinating those processes. The orchestration capability makes it possible to track the changing states of key activities that are critical to the overall business-process completion.

A single failure to deliver any of these pieces of information, or in securing the information as it flows across the environment, could compromise the success of the business and your ability to trust in the ESB. To be used throughout the business, and to be relied upon by all parts of the business, the ESB needs to be always available, reliable and secure, delivering information to meet the needs of all parties, whatever the quality of service needed – reflecting the importance of the data being moved. A failure of your ESB to handle the throughput required by the business, or a failure to provide the capabilities to access to a key system, could lead to your infrastructure not meeting the flexibility needs the business is expecting. If this happens, then your ESB is limiting your business, rather than enabling it, and your SOA will not be able to live up to its potential for your business.

### Every transaction is important

Sometimes in business, some of the most important points can be overlooked — because they look small. Even though an individual transaction can look small and insignificant, your business systems need to be able to treat all your transactions as important. A transaction needs to be able to update resources and complete its unit of work, or key information can be lost in the event of a problem or failure. The failure of a single transaction as part of a sequence of events could lead to the overall failure of entire business processes. And transactions that can't complete all their operations, such as locking and coordinating the updating of resources across systems, can lead to more-complex business environments that can't keep track of their successfully completed business activities accurately.

For example, if you are moving money between accounts, then you really want to ensure that both accounts are updated or the original states are maintained, or someone, somewhere could end up losing those funds. And what about booking a holiday — reserving a flight, hotel and car? You want to be sure that when customers make those reservations, those transactions are recorded promptly and accurately.

So your ESB needs to be able to use transactional coordination of resource updates where possible, either by taking advantage of underlying services or by providing its own coordination.

### The extent of integration required in your SOA

When considering the integration that your SOA needs to provide to your IT infrastructure, and that needs to be delivered by your ESB, you should be certain that your ESB is accessible from all of your systems and environments. You want to access and move data — any data from anywhere — to anywhere. Increasingly, this means not just from a UNIX® box to a Microsoft® Windows® or Linux® box. And it doesn't just involve a Java™ application, a Web services application and maybe a C program.

If you want your business connected, then you have to consider all of your assets — whatever and wherever they might be. To create a truly connected infrastructure, you might want to have immediate access to data, even if it is coming from sensors, actuators and handheld devices. This type of connectivity, linking these types of assets directly to business systems for immediate input processing, has never been performed easily without sophisticated programming, and might provide your business with new opportunities and challenges to use that data to be more responsive than ever before. It is possible to intervene when a sensor shows a temperature getting beyond nominal or to track goods using radio frequency identification (RFID) tags as they move around your business or through a manufacturing site. This type of information has the potential to create added value across the value chain, providing customers with more information about their orders, enabling your business partners to have more insight into the need for resupply and helping your business reduce the potential for spoilage or other negative consequences.

Along with tangible assets whose movement or properties could be tracked or monitored, many business assets are not only intangible but currently underutilized. Probably the strongest example of this type of asset is the business information stored in files. Information stored in files, scattered throughout the file system of a typical business, can prove to be very hard to access in a timely, secure or reliable manner. Access to this type of information is almost always very tightly coupled to the business application that reads and writes to the file itself. The record or other data structure used to hold the information might well be understood only by a specific application, and access to the data is thus tightly locked to that application. Access to the file directly, or from a remote system, is unlikely to be welcomed or easy to set up in any regular environment.

Other than the applications tied to the files, the only other common access method for file-based data access is through File Transfer Protocol (FTP). Even though probably half of all data in a business is moved using FTP, it is hard to think of a method for moving business data that is less reliable, less secure or more in need of careful monitoring and management than FTP. Although FTP failures in a typical business are likely to be common, there is unlikely to be much visibility of the costs these incur because the line-of-business (LOB) system that owns the applications would typically absorb this impact as a cost of maintaining and operating their applications as part of normal, day-to-day business. However if an ESB is to span the enterprise, delivering the connectivity required and enabling SOA to deliver flexibility, then the valuable information currently locked away in files should be made as accessible to all parts of the business as any other data is, and it goes without saying that this should happen without added cost or complexity to the existing applications.

When thinking of the applications in your business, it is probably obvious that it is not just applications developed in-house that are critical. Your business is likely to have at least one or more packaged applications, for enterprise resource planning (ERP), customer relationship management (CRM) or other key aspects of the business. Quite often, you have made substantial investments in acquiring and configuring these assets, which only return value if you can integrate enough of the rest of the business into these applications. Again, an ESB can assist here, by helping to reduce the application-specific interfaces to be built, because all parties can connect into the ESB, and the ESB can access the packaged application through a well-defined interface.

Across all aspects of data movement, between any device or any application, a common criteria is likely to be security. The information that flows across the business, through the ESB, must always meet the standards of security appropriate, or mandated, for that data. The ESB must not only be able to provide those criteria, but might need to recognize and enforce those security aspects on each message as it passes through. Of course, to meet your overall business needs, this level of security must not add complexity, even as it adds protection and governance to the business.

Focusing across all these asset areas, assuming the ESB is capable, can help your ESB to increase the return on all your IT investments, delivering SOA everywhere and for everything – from the smallest sensor to the most-complex packaged applications. You should include all of these components in the SOA life cycle and in the access that your ESB should give to your business.

If in your business you want to update a number of resources within a transaction, then you need to make sure they can take place within a unit of work managed by the environment itself – and you are going to have to handle a great deal of additional complexity as you try to write rollback scripts to cope with any scenario. And complexity is what you need to avoid; it is one of the reasons to use an ESB in the first place.

### Qualities of service

One of the key drivers behind moves to SOA is business responsiveness – helping to ensure that your business can be more flexible and better able to respond to the needs of the customer and the changing needs of the business. This flexibility should be demonstrated by the enhanced ability of the business to provide the highly adaptable qualities of service to meet the needs of different customers, as well as the different levels required across the business. These qualities of service should be available at all times, not restricted or impeded by other activities in the infrastructure or the configuration and deployment options used. Customers and businesses are rarely sympathetic to performance that fluctuates with demand, or services that are only available to some parts of the business due to hardware selection by either requester or provider.

Implementing an ESB should enable your business to decouple the qualities of service required from the provision of the services across the enterprise. Hardware implementations should be a detail that becomes irrelevant to users. And the ESB should scale to meet the fluctuating needs of the business, delivering requests to the service providers without delay or bottlenecks.

In any case, resources are never as boundless as we would like to imagine. Your business must constrain use of resources to help ensure that those requests that are the highest priority for the business are given the most-effective use of IT resources, rather than trying to deliver the same level of quality of service to all requests – which could end up satisfying no one. Some requests, such as a noncritical query, might not be given high priority, either in resource cycles or in persisted delivery. If a failure occurs, the request can always be sent again. Others need to be delivered across systems, with the highest level of assurance and integrity, because they involve the movement of valuable assets as part of a key business transaction.

The best ESBs can identify these high-value transactions from the content and context, and allocate the appropriate level of resources, without adding to the level of programming complexity. Your ESB, like your business, needs to be adaptable to the changes in demand placed on it. Imagine a transportation and logistics firm. It might start out delivering everything in a local area in a small truck, but as the business scales, different logistics and different modes of transport are needed. Sometimes information can be sent in regular mail. Sometimes a signature is required. Some packages can be shipped around the world and others need to be put on a jet to help ensure swift delivery. The same configurable level of service applies to your ESB.

Some ESBs are good in many areas of service delivery and data handling, but let themselves down – literally. Your ESB is responsible for the efficiency of so much of your business, but the ESB itself is also a critical part of your business. To help ensure that the ESB itself is not a problem, its operational characteristics must also provide high levels of quality of service. This includes not just day-to-day operations, but also in responding to its own change, allowing for hot deployment and powerful version control to help ensure that you really understand what is running from day to day. If you don't have full control over your ESB, then your business can slip out of your control.

#### **ESB offerings compared**

For many other vendors, an ESB exists to connect Web services to other Web services and perhaps some other applications. Some vendors might provide just that and no more, maybe with some variations on the number or type of different applications that can be connected. This approach can create substantial confusion in the market about what an ESB does, or should do. As this white paper has discussed so far, simply providing connectivity for standards-based applications or services doesn't begin to meet the real ESB needs of a business that wants to move to SOA. You must not limit your business by deploying an ESB and then be left cobbling together additional software or adding complexity back into your applications or infrastructure to meet your full SOA connectivity needs.

Instead of an ESB that has limits – and that can limit your business – you need an ESB *without* limits. One that can deliver all it needs to, in connecting platforms, applications and services. One that is always reliable and assured in delivering information, and that will grow to meet your needs, instead of restricting your growth. And one that allows you to make the best use of the rest of your infrastructure.

When you choose an ESB without limits, your business is not limited by the IT infrastructure, and your SOA deployment can be successful, without being constrained as you move forward. However, you should also recognize that for an SOA deployment to be successful, you need more than just a highly capable ESB for a complete SOA solution. You need to add other capabilities to your ESB to provide a complete connectivity solution for your SOA. It is important to note that these capabilities would not fit well as native or core ESB capabilities. Many of them are highly specific to use cases or deployment models and would be incompatible with the mostly stateless application- and service-connectivity model provided by the ESB.

In addition, an ESB already has many capabilities that are essential to performing its functions. As with so many other things, the ESB should not be overly complicated with extra capabilities that are not required for every deployment. As explained later in this white paper, that is why IBM has more than just a single ESB. Business needs for integration and connectivity between applications and services are so diverse that IBM believes the ESB space is best addressed through a set of different, but compatible ESBs, offering tailored sets of functions for different sets of requirements. These ESBs help to support the widest possible range of businesses and help to deliver the most complete and connected SOA environments. (See Figure 3.)



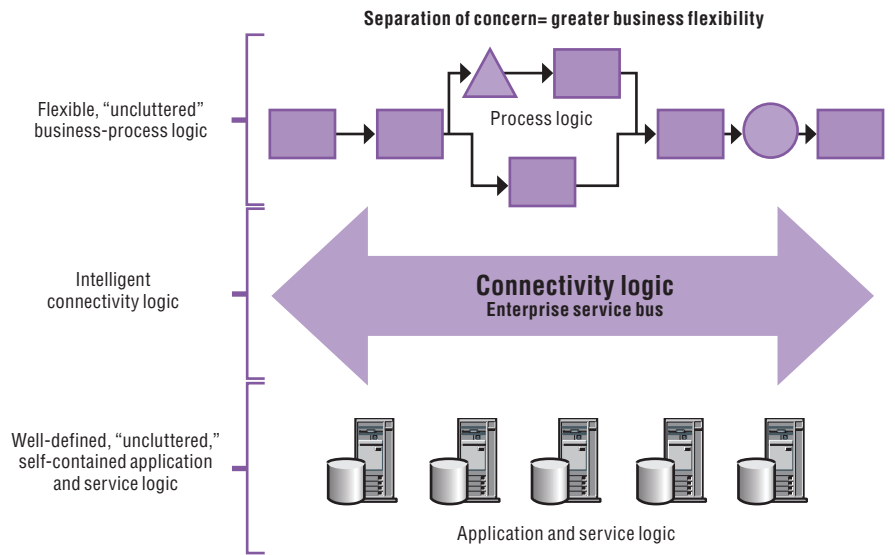


Figure 3. Using an ESB as the connectivity layer for your business

### Selecting your ESB and your ESB vendor

Along with fundamental attributes that are critical to your decision, such as whether a specific ESB is available on your required hardware platform, you should consider a number of other aspects when making a decision to select an ESB that best meets the needs of your business. Your ESB selection is almost certain to be critical to the success of your SOA deployment, so you should not decide based on a single factor or with a single project in mind. So far, this white paper has covered a number of factors that are likely to be important in selecting an ESB.

IBM considers all these factors. Platform coverage. Support for Web services. Support for existing application environments. Configurable, built-in support for transaction coordination. Security. Scalability. Throughput. Performance. XML and Web services off-load. Evaluating an ESB that offers every permutation of all these factors would be overkill for many environments, when a simpler solution is required, because of simpler integration needs. Equally, a simple solution is not what many businesses need when trying to integrate existing complex or bespoke applications or to offer them as services.

Taking all these factors in mind, IBM believes it is impossible to build a single ESB offering to meet all your expectations in terms of functionality, as well as keeping the ESB offering easy to configure and use for all implementations. Business environments are so diverse in terms of technology types, network load and available skills, that no single product can be provided to deliver an ideal match for every use case.

Instead, IBM offers a range of ESBs to better suit the individual deployment needs of customers. And these ESBs are designed to provide seamless connectivity and compatibility across the whole enterprise and between any deployed combination of ESBs. Customers don't have to make an ESB decision that is right for the individual project, and then have to move to a different ESB. Any and all of the ESBs can be run alongside each other to allow for tailored deployments of different ESBs across the enterprise.

IBM can deliver this flexible, powerful and robust approach through the use of core IBM WebSphere® technologies at the heart of these ESBs, providing a rich and reliable foundation that then enables you to deliver the qualities of service you require.

All of IBM's ESBs are able to make fully integrated use of IBM WebSphere MQ – the messaging backbone for your ESB – a market leader for more than 10 years, and providing highly secure and reliable messaging on virtually every business platform. IBM ESBs offer built-in support for numerous standards, including the latest Web services standards, industry standards and various proprietary formats and protocols with extensibility to support many more. Today many thousands of businesses across all industries take advantage of IBM ESBs and the wider integration platform for their SOA deployments.

### ESBs from IBM

As mentioned previously, IBM has multiple ESB offerings that you can use on their own or together in any combination to provide the most powerful and flexible ESB to enable your SOA across your entire business. Figure 4 shows three IBM offerings that deliver an ESB

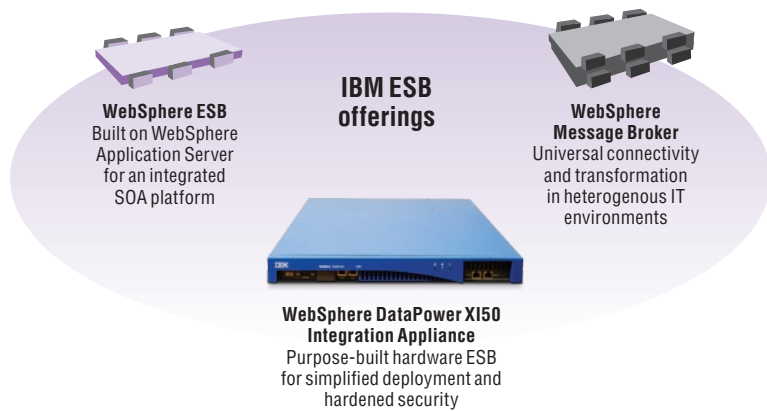


Figure 4. ESBs from IBM

For environments that are predominantly standards-based, and focused primarily on integrating Web services, IBM WebSphere Enterprise Service Bus (WebSphere ESB) is a good choice. It is built on top of IBM WebSphere Application Server, delivering a robust, security-rich and scalable foundation, as well as being able to host services and integrate with other services.

Where businesses have environments that are a mix of different technologies, environments and other infrastructure aspects, encompassing existing applications, as well as potentially new and evolving Web services, IBM WebSphere Message Broker is a highly adaptable ESB, built on WebSphere MQ. WebSphere Message Broker is one of the most-capable integration servers on the market. With each release, it has evolved to continue to meet the changing needs of companies, and it now provides a very comprehensive ESB that can work for virtually any integration requirement, including both services and nonservices assets delivered through a highly flexible and highly performing infrastructure, independent of application-server platform.

Additionally and uniquely, IBM offers an ESB in the form of a hardware appliance – IBM WebSphere DataPower® XI50 Integration Appliance is a rack-mounted network device capable of transforming between different message formats, including binary, traditional and XML, and providing message routing and security. This ESB is particularly effective at off-loading the cost of processing Web services calls and XML processing from servers, especially mainframes where processing cycles are always at a premium. Its unique architecture and capabilities allow it to extend your ESB to make your business more effective, as well as enabling you to deploy the XI50 appliance as an ESB where you would otherwise not be able to deploy an ESB.

Any of these might be used on its own, or in combination, to help implement SOA in your enterprise. With an effective end-to-end ESB deployed in your enterprise, and with specific ESB deployments meeting the needs of all the component parts of your enterprise, your business has the core of the connectivity it is likely to need for its SOA. An ESB by itself is not the complete solution for an end-to-end SOA implementation, no matter how effective an ESB it proves to be. You need an ESB without limits, capable of being extended and tailored to meet the needs of your SOA infrastructure and in supporting the other aspects, or entry points, of SOA.

**ESB without limits – extending the bus for a complete SOA solution**

You might find five key areas of additional function essential as you drive your business and your ESB to be more responsive as part of the journey towards SOA – and to help ensure that your IT infrastructure responds as your business grows and changes. These functions are available today from IBM to extend your ESB, so that you can build an ESB without limits and a comprehensive SOA solution environment. These functions are supported as extensions to IBM ESB offerings – WebSphere ESB, WebSphere Message Broker and WebSphere DataPower XI50 Integration Appliance – helping to ensure that these IBM products deliver an ESB without limits (see Figure 5).

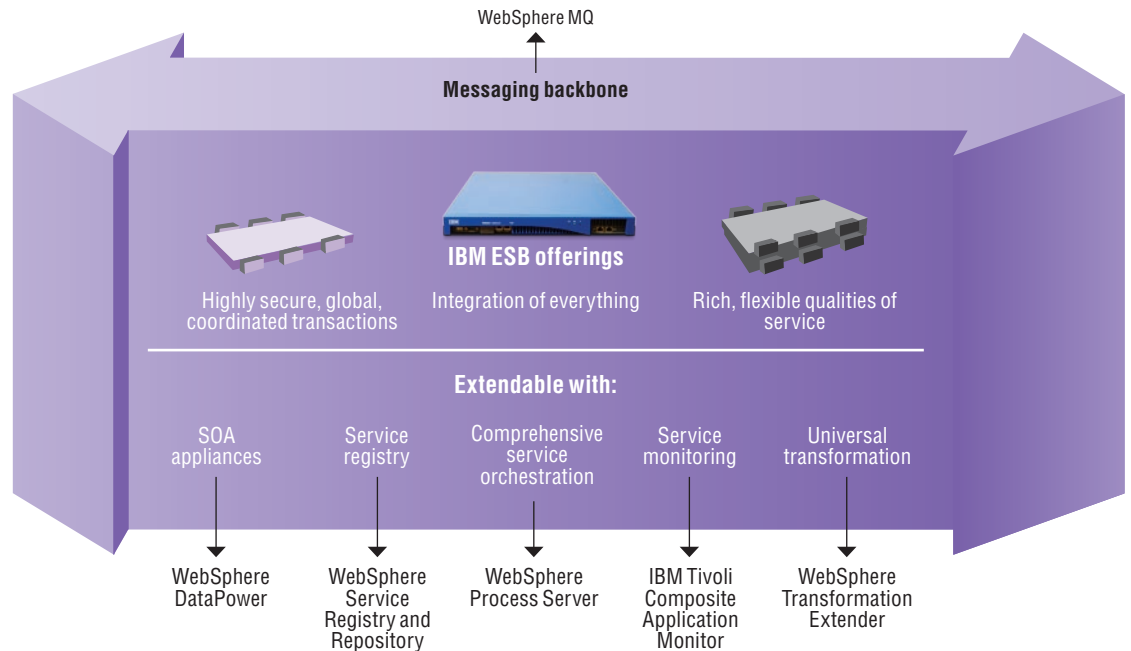


Figure 5. An ESB without limits

### Specialized SOA hardware

In the world of SOA, services can come from anywhere and do pretty much anything – internally or externally. SOA introduces many issues, including serious security issues, that most ESBs aren't equipped to handle, creating limits to the ESB that can drag down the business goals or even the business itself. This situation leads to the previously unthought-of need to add some kind of ESB function into the DMZ, on specialized processors optimized for handling specific functions. But if you are going to do that, you should be sure that you have superior lock-down technology. That's why you need an ESB without limits that can be extended by an appliance strategy. Appliances offer many capabilities for services, including great capability for locking information access down, accelerating key processing and handling key integration functions, thus making sure your ESB is really efficient and bulletproof. Appliances – specialized hardware for your SOA – help minimize the complexity and reduce the cycles of implementing SOA infrastructures, while enhancing performance, security and management of business processes implemented under SOA principles.

This white paper has already mentioned the hardware ESB appliance (WebSphere DataPower XI50), which is capable of acting as an ESB. However, other complementary offerings in the WebSphere DataPower product range include IBM WebSphere DataPower XML Accelerator XA35, which can provide the ability to off-load XML, XML Schema Definition (XSD), XML Path Language (XPath) and XML Stylesheet Language Transformation (XSLT) processing from overtaxed servers and perform that processing at wirespeed. There is also the IBM WebSphere DataPower XML Security Gateway XS40, which in addition to the function offered by the XA35, delivers a comprehensive set of functions including WS-Security, XML encryption, XML/SOAP firewall filtering, XML digital signatures and XML schema validation.

What makes IBM unique in delivering an ESB without limits is that it has recognized the need to move beyond just an ESB. IBM, rather than offering a single, limited ESB, delivers a federated ESB strategy that enables you to put the right function in the right place – appliances – and core transactional ESB systems on multiple platforms, including the IBM System z™ platform, as part of one fabric.

#### *Service registry and repository*

This white paper has covered how important it is to connect assets simply and flexibly as the business moves towards implementing an SOA. Clearly getting assets connected is important. But the further businesses move towards an SOA, the more flexible and dynamic the environment becomes. This situation can lead to difficulties in locating and accessing services, which can change or be replaced at a moment's notice and is part of the whole thought behind Web services. So you don't want to be limited by your ESB being unable to find resources – especially fast-changing services.

And what about controlling access to those services. Is it a free-for-all or do you have to code control and enforcement into the applications and services, again adding to complexity? Do you have to handle access on a case-by-case basis? How much effort would that take to coordinate? And how would you be able to track who had access or who had made changes, and whether all aspects conformed to overall business intentions?

To deliver an ESB without limits, you need integrated support for services, and the dynamic definition and location for those services. Another key capability is policy-based access for use and reuse of assets, with strong management support to promote SOA governance, handling the management of assets and the mitigation of the potential for risk associated with the more-connected and dynamic environment.

IBM recognizes that these are key requirements for an SOA as it becomes more widely deployed. To meet this need, it offers IBM WebSphere Service Registry and Repository, providing clear visibility into service associations and relationships. It helps to encourage reuse of services through better awareness and better information about the services and how to access them. Together with these operational enhancements to SOA, WebSphere Service Registry and Repository can help to promote best practices as well as better control of the business through the management and governance capabilities provided by the offering.

So, IBM offers an ESB without limits that includes seamless integration from the ESB, and more, to WebSphere Service Registry and Repository, supporting the dynamic publish and lookup of services, as well as providing SOA governance through definition and enforcement of policies to govern your SOA, and manage operations.

#### *Service orchestration*

If you are planning to move to SOA, then maybe you have reviewed a number of approaches, and you think that the five SOA entry points – people, process, information, reuse and connectivity – make a lot of sense. But are they unrelated? Should you try to solve your connectivity issues along with, and as a part of, your process plans? If you want to streamline your business through effective and optimized processes, then you need to be sure that your processes are efficiently orchestrated, and effectively linked, to the underlying applications and services. To do this, you need to make sure you do not complicate your processes with connectivity interfaces. You need to be sure that your ESB works seamlessly with your process layer, that process and connectivity are effectively decoupled and that these different environments are separated. The process engine should be able to take advantage of the features of the ESB, not replace them. You don't want your ESB limited by weighing down the process layer with connectivity information that would be better defined in a separate ESB.

What you need your ESB without limits to deliver is a robust, scalable environment that integrates seamlessly with the process engine, the runtime environment, and also with the tooling environment to maximize shared use of assets and greater reuse of skills.

IBM WebSphere Process Server is a high-performance business engine that you can use to design and implement business processes that enable you to meet your business goals. It orchestrates the assets of your business to form highly optimized and efficient processes, whether you need to automate processes in the factory, process claims or financial payments, or run an efficient supply chain while complying with industry regulations.

IBM delivers an ESB without limits through seamless integration with WebSphere Process Server. This provides optimized business process management (BPM), capable of using a specialized ESB connectivity layer that is embedded within WebSphere Process Server.

*Service monitoring and managed access*

Suppose you have been doing well in your SOA planning. You are successfully connecting all your platforms, applications and services. And you have good links between your applications and your processes flowing through your ESB – connecting through your existing infrastructure and new deployments. But now that things are up and running everything seems out of control. How do you know what is going on, as well as who is accessing your systems? Can you track what is happening in your systems, or are you limited to just what your ESB can tell you? In the highly connected and interdependent environment of an SOA, it becomes critical to be alerted to any situation you define – to increase both automation and awareness of business activity, or lack of it because of bottlenecks or failures. You also need to be sure that your systems are built on trust of the integrity and privacy of data.



For an ESB without limits, you need to be able to identify problems and resolve them quickly – drilling down from the symptom to find the real problems. You also need to have a simple, yet effective, model for managing access to your resources to eliminate a potential source of problems. Managing these resources needs to be possible throughout the entire life cycle, and not just in a single environment, but across your SOA, and for multiple platforms. Ideally, everything should be visible and managed through a single powerful console to help ensure that nothing gets missed.

IBM further extends the ESB without limits through strong integration with the IBM Tivoli® portfolio of products, including IBM Tivoli Composite Application Manager for SOA. This product enables your ESB to be fully monitored and managed, and covers other vendor middleware products that might be deployed in the SOA. The portfolio also includes IBM Tivoli Federated Identity Manager, which provides a simple, flexible, loosely coupled model for managing identity and access to resources that span companies or security domains (which could well be the case in a widely deployed SOA). Along with these extensions to the ESB, you can further extend your ESB capabilities, such as with the business dashboard provided by IBM WebSphere Business Monitor.

#### *Universal transformation*

So far, this white paper has covered a number of aspects of SOA connectivity and ESB features, but it hasn't yet touched on one very important area – transformation. For many businesses, mapping of data and transformation of data formats as it is exchanged between services is one of the top requirements for an ESB. After all, if your SOA needs to connect applications flexibly, and the data is not in the expected format, then the transformation has to happen in the application. In that case, you have not only increased complexity, but you have pretty much eliminated any prospects of reuse for that application. So if you don't want to be limited by your ESB, then transformation is going to be a highly important aspect.

For an effective SOA, you need to be able to have data transformed from any format to any other format, whether the formats are simple or complex, highly proprietary or a core industry-standards-based format.

However, equally important to real SOA deployments is not just whether the data can be transformed, but where the transformation takes place. If you need to perform transformation between two local applications or services, you don't want to have to route the message to a remote hub to do the transformation. You need to have mapping capable of running everywhere you need it, even in other vendors' integration products – without adding complexity to your applications or instability to your environment.

IBM WebSphere Transformation Extender offers the most comprehensive and flexible transformation product, capable of running on its own, embedded as part of the application infrastructure, or as an integral part of the ESB. Fast integration with complex applications using WebSphere Transformation Extender can be further accelerated with the Extension Packs, Industry Packs and Enterprise Packs, as well as with IBM WebSphere Adapters. All of these can speed the time taken to effectively use business information otherwise buried in applications.

Only IBM offers an ESB without limits providing universal transformation – with transformation not just of any data format, but available anywhere in your ESB. You can provide transformation as a core capability of the ESB itself, or deploy and access it anywhere within the enterprise and across the ESB. WebSphere Message Broker, WebSphere ESB and WebSphere DataPower XI50 Integration Appliance can all have their transformation capabilities extended by WebSphere Transformation Extender, deployable practically anywhere.

**IBM clients choose an ESB without limits**

As more and more businesses move down their own journey to SOA, thousands have already been using IBM offerings to help transform their businesses to be more robust and more flexible, reusing more assets and connecting their business more completely and with less complexity than ever before. The range of capabilities and the extent of the qualities of service delivered by an ESB without limits from IBM continues to provide excellent value to clients already using it today. Read about some of these clients at [ibm.com/software/success](http://ibm.com/software/success).

**For more information**

Now is the time to get started. And IBM has a wealth of resources available to help. To get code, white papers and demonstrations, and register for events and Web broadcasts, contact your IBM representative or IBM Business Partner or visit:

[ibm.com/soa](http://ibm.com/soa)



© Copyright IBM Corporation 2007

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America  
05-07  
All Rights Reserved

DataPower, IBM, the IBM logo, System z, Tivoli and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries or both.

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

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

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