Service oriented architecture
White paper

WebSphere® software

Smart, flexible SOA for all seasons.

By Mark Simmonds, IBM, worldwide product marketing manager, WebSphere MQ

May 2008

Contents

Introduction

One of the key challenges that CEOs face is their ability to change in order to drive innovation. The global economy is forcing companies to adapt to new and changing dynamics in the marketplace, such as rapidly growing consumer markets in China and India, global integration of technology and standards, and competitive business and technology models that support business agility.

In 2007, IBM conducted a CIO survey. The results showed that 87 percent of CIOs believe that fundamental change is required over the next two years to drive innovation. Change is inevitable, so it helps if you can:

- *Leverage and reuse your existing and any future investments.*
- *Start quickly and simply to meet your more immediate needs, and then grow as the business dictates – without having to rearchitect.*
- *Use the right qualities of service for the various types of business transactions.*
- *Leverage future technologies that can be integrated with your existing investments.*

Service oriented architecture underpinning innovation

When business and IT pull together, extraordinary results can be achieved. Service oriented architecture (SOA) bridges the gap, enabling IT to render existing and future assets as a set of highly reusable business services that can remain stable for as long as the business requires.

The business view of the world is, of course, different from that of IT. The business has to drive innovation that matters, deliver top-line growth, and achieve operational excellence and compliance while, at the same time, reducing costs and gaining market share. While attempting to achieve these goals, the business correctly aims to reuse the investments it has made over the previous years in IT and to deliver change at a pace and cost that outwits the competition.

The IT manager, on the other hand, has a different set of challenges. These include a complex mix of systems with applications that have processes buried deep inside them, and many interfaces that are proprietary or difficult to adapt and change because they were never envisaged beyond their original use. Consequently, the IT department spends a large percentage of its budget on maintenance instead of new investments to support rapid business innovation.

So how does an IT department untangle the spaghetti of complexity while leveraging and maximizing the return on investment (ROI) it already has? Enter SOA.

Simply SOA

What SOA provides is a way of defining higher-level, service-based interfaces that can remain stable for as long as the business needs them to, while protecting them from lower-level technology changes, such as the old life-insurance-claims system being replaced by a new insurance-claims system, running on a different operating platform.

IBM WebSphere® MQ software is a messaging backbone for your SOA that can integrate almost any commercial IT system, allowing data to flow between all your applications and data sources. It forms a flexible infrastructure that allows reuse of existing assets and systems that can be readily available and orchestrated to participate in new, agile business processes.

Some people are still wary of the term SOA: "Another buzzword– do I have to start again? Do I have to reinvest? Is it just another fashion, and how long will it last? Do I even have the skills?"

But fear not. IBM offers the IBM Smart SOA™ strategy, which enables you to start simple on an SOA foundation and progress along a continuum as business needs dictate, leveraging you existing investments.

IBM offers the Smart SOA™ strategy, which enables you to start simple on an SOA foundation and progress along a continuum as business needs dictate. You can start anywhere on the continuum as well as from any of the five entry points to SOA: connectivity, process, reuse, information and people. This strategy enables you to tackle your most pressing challenges first while allowing you to evolve your foundations to meet future goals when they arrive. Process integrity is an area that IBM has been known for—from integrity of low-level transactions right through to the integrity of the process at the level of business process management (BPM).

Get started quickly, using the skills you already have

WebSphere MQ fits into the connectivity entry point. It's easy to get started with. It can be installed and configured in 6 to 10 minutes, depending on your platform. What's great about the product is that it includes a lot of sample code in various programming languages, which can be reused as part of your solution. These samples represent a high proportion of the connectivity code required.

Regardless of where you install WebSphere MQ—whether on IBM AIX®, Linux®, Microsoft® Windows®, or IBM z/OS® platforms, for example—you have a single point for administration and configuration across the entire infrastructure.

And you have a choice of using different styles of interfaces. WebSphere MQ provides a powerful multilanguage Message Queuing Interface (MQI). Or for those of you with big Java™ investments, WebSphere MQ provides a fully compliant Java Message Service (JMS). It also provides Multi-Language Message Service (XMS) messaging, similar to JMS but for languages that are not based on Java (see Figure 1).

In addition to WebSphere MQ, within the connectivity entry point, we have a choice of federated but integrated enterprise service buses (ESBs) and prebuilt adapters for a range of business applications and technologies that can discover services buried deep inside the applications. All of these assets integrate with WebSphere MQ.

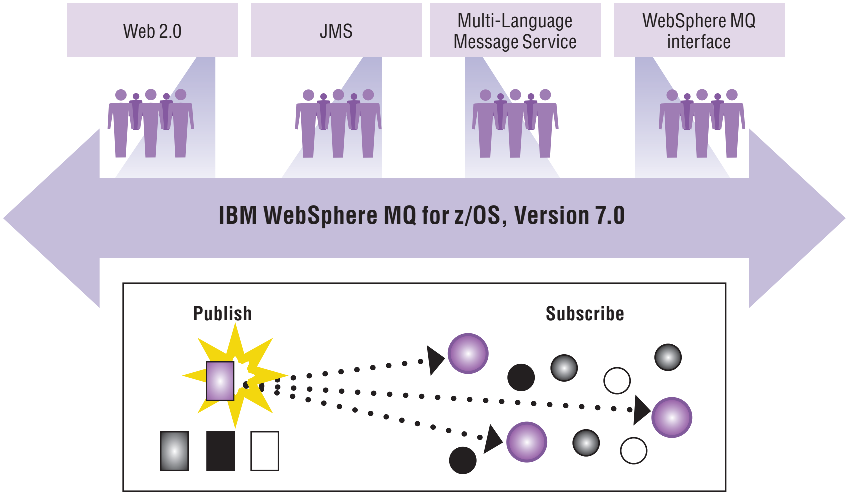

Web 2.0
JMS
Multi-Language Message Service
WebSphere MQ interface
IBM WebSphere MQ for z/OS, Version 7.0
Publish
Subscribe

Figure 1. WebSphere MQ, Version 7.0 delivers enhanced ease of use and choice for developers with integrated publish-and-subscribe messaging.

WebSphere MQ provides a reliable messaging backbone that spans virtually any IT system, and it does so with assured end-to-end delivery. All business transactions that have to ripple across the furthest reaches of the organization are preserved in transit by WebSphere MQ, regardless of how many IT systems are involved in processing each transaction. This means that wherever business data has to be shared and moved, its integrity can be preserved.

WebSphere MQ remains relevant in the SOA world with support for the latest industry standards and provides the foundation for an ESB. Its strength and breadth are demonstrated by its support for virtually any IT commercial platform, including native z/OS support, along with its wide support for programming languages, including Java, C/C++, C#, .NET, COBOL, Fortran, Assembler and REXX. This means you can integrate anything you have and use the skills you already have.

WebSphere MQ provides you with choice and flexibility. Just use the interface to match your skills and needs and reach all of your assets, no matter where they are, how the information is stored or what languages the applications are written in. You can start simple by connecting just the key applications needed to address a particular set of business needs, and then use WebSphere MQ to incorporate more and more applications into the messaging backbone, knowing that it is extremely scalable and highly secure.

The magic of WebSphere MQ

WebSphere MQ provides an always-connected infrastructure. Applications connect to the WebSphere MQ infrastructure using a simple and consistent application programming interface (API) comprising some 20 verbs that can be quickly grasped by a developer. Queue managers reside on platforms that house business applications. The business applications are extended unobtrusively using the WebSphere MQ verbs. Applications send and receive messages through MQPUT and MQGET verbs to service interfaces called *queues*. The queue manager is responsible for the integrity of the data stored in its queues as messages, as well as for reliably distributing those messages to other queue managers.

Developers don’t have to worry about the physical location of the destination service or application. They just send or receive the message through MQPUT or MQGET commands to or from the service interface name (the queue name). WebSphere MQ takes care of the rest, abstracting all the complexities of the underlying and potentially changing infrastructure.

The asynchronous nature of WebSphere MQ and its queuing paradigm has several advantages. One service or application can send data to the destination service without that destination being available. Even if the service sending or receiving the message fails after the message was sent, or the network itself fails, or both services become temporarily unavailable, the message and its integrity are always preserved. Client testimonials confirm this. One client stated, “We’ve used MQ for 10 years, and we’ve never lost a piece of data, ever.”

Units of work

When conducting business, your SOA needs to be able to deliver end-to-end transactional integrity. A business transaction can comprise multiple subtransactions spanning multiple distributed systems across and beyond your organization.

WebSphere MQ provides transactional integrity. It can coordinate XA-compliant resources, or it can participate under the control of a transaction manager, such as IBM CICS® software. What this means is that multiple transactions can be part of a "unit of work" so that if any one part fails, all the transactions involved can be rolled back if necessary.

For example, a travel company's system must communicate with multiple IT systems to make a booking, reserve flights, hold a hotel room, secure a rental car, and so on. If any of these reservations fail, the whole booking needs to be undone or it will be partially made and the data in these systems cannot be reconciled. This capability is built into WebSphere MQ.

Specifically, the technology of the asynchronous programming model used by WebSphere enables this parallel processing of multiple threads so that numerous transactions can be reconciled under a unit of work.

Business scalability and resilience

So what happens if your business suddenly needs to process large volumes of data – maybe because of a merger and acquisition?

WebSphere MQ provides what is known as *clustering*. This means that the workload can be split across multiple back-end systems, enabling much higher throughput and higher availability in support of 24x7 operations.

Interruptions in availability can be devastating. The report *Business resilience: Ensuring continuity in a volatile environment* from the Economist Intelligence Unit, sponsored by ACE, IBM and KPMG, states that "according to the U.S. National Archives and Records Administration, 25 percent of the companies that experienced an IT outage of two to six days went bankrupt immediately."

Basically, a cluster is a network of WebSphere MQ queue managers that are logically associated in some way. The queue managers in a cluster can be physically remote. For example, they might represent the branches of an international chain store and be physically located in different countries.

Configuration of the cluster is simple, because relationships between each queue manager in the cluster are automated. Though you can have multiple queue managers on a single server, the idea of clustering is to ensure that even in the event of a server (hardware) failure, queue manages on other servers can share the workload. Different workload balancing schemes can be applied, from simple *round robin* (each queue manager in turn) to sophisticated, prioritized and weighted algorithms.

In addition to this clustering capability, multiple instances of WebSphere MQ applications can serve particular queues and be dynamically started and stopped based on workload.

All in all, WebSphere MQ provides the scalability, security and high availability that you require for your mission-critical SOA backbone.

Market adoption and usage

WebSphere MQ is the undisputed market leader for messaging, with more than a decade of IBM investment, and has broad support from many independent software vendor (ISVs) and systems integrators. In its *Worldwide Mission Critical Messaging Market Opportunities, Market Forecasts and Market Strategies, 2007-2013* report, Wintergreen Research states that “IBM leads the mission-critical messaging markets.”

It is used by more than 10 000 customers in all industries across the world. Ninety percent of the Fortune 100, 300 of the Fortune 500 and all top 20 GLOBAL 500 banks depend on it. One banking client moves transactions worth up to US$35 trillion each day using WebSphere MQ, while a government client moves more than 675 million messages a day over WebSphere MQ.

Knowing your data is in safe hands

More companies in a range of industries are being confronted with regulations from industry bodies, suppliers and governments. These regulations impose real penalties on senior individuals (such as imprisonment) and companies (such as fines or dramatic effects on stock valuation). Most regulatory compliance obligations are based on the need to ensure that financial records are timely and accurate. This is very difficult to do and demonstrate if critical business data about transactions moves through the organization in a way that is not secure, auditable and trustworthy. Because WebSphere MQ can help ensure that records meet regulatory requirements, many clients find that they can reduce their risk of failing compliance by using WebSphere MQ instead of the alternatives.

Ask yourself these questions:

- *Do I have a complete, end-to-end audit trail for data?*
- *How sure am I that no data being exchanged is lost or tampered with?*
- *What is the risk, even if one application isn't reconciled?*
- *How can I expect to comply if I lose data?*
- *How can I preserve the integrity of applications, security of data and my auditability?*

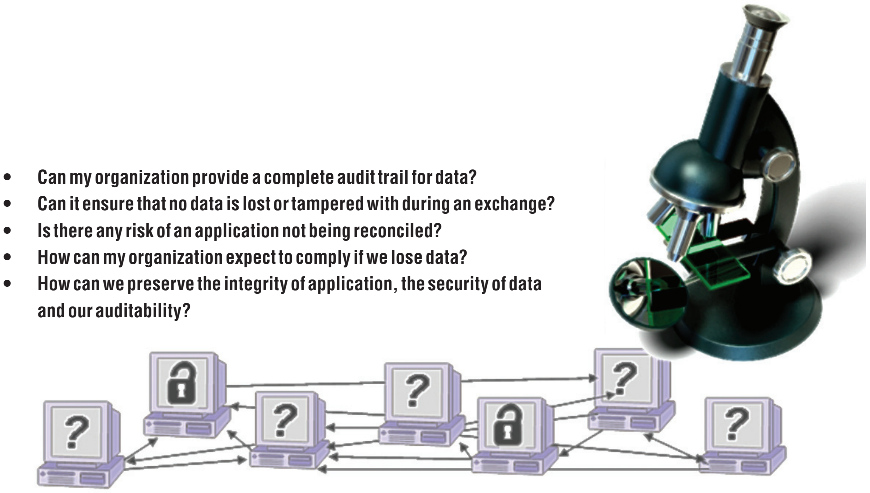
Many readers of this paper will think, "These regulatory concerns don't apply to my organization." If you are using File Transfer Protocol (FTP), problems related to these concerns occur more often than you might realize. FTP is used to move 70 percent of data in an organization but, staggeringly, it fails 20 percent of the time. Here are some of the issues associated with using FTP:

- *Transfers aren't flexible. Checkpoint restart is not always available if a transfer fails; you generally have to start over. All resources must be available concurrently, so any failure means you have to restart the transfer from the beginning. Only one FTP-based transfer can run at a time, and transfers cannot be prioritized.*
- *Data security is limited or nonexistent. Often user names and passwords are sent with the data, frequently as plain text. Nonrepudiation, privacy and authentication support are often lacking.*
- *Visibility and traceability of data is poor. Even if you manage to send the data, you can't always prove it. There are no centralized monitoring or management capabilities. There is no logging of data movements to satisfy audit requirements.*
- *Data integrity is often compromised. Transfers can be incomplete, resulting in the receipt of partial files. Partial files can potentially be processed by applications, causing problems with the integrity of applications, data and business processes downstream. Files get corrupted (such as during ASCII or binary transfer), get lost and are not transactional in nature.*

Secure, managed file transfer or just file transfer?

Increasingly, businesses of all sizes need to comply with more and more financial and other regulations, especially in response to a series of high-profile scandals that were in the news. Proving what was done, when it was done and who did it is becoming increasingly important (see Figure 2). File transfer can affect how well you can meet these requirements; however, as previously mentioned, it often fails.

The message persistence and logging capabilities of WebSphere MQ make it ideal for managed file transfer. IBM and many of IBM's partners provide value-added managed file transfer solutions based on WebSphere MQ because of its robust, yet simple, architecture and market leadership.



- **Can my organization provide a complete audit trail for data?**
- **Can it ensure that no data is lost or tampered with during an exchange?**
- **Is there any risk of an application not being reconciled?**
- **How can my organization expect to comply if we lose data?**
- **How can we preserve the integrity of application, the security of data and our auditability?**

Figure 2. Concerns about the security of data and regulatory compliance trouble many organizations.

PM4Data is one example of an IBM Business Partner offering (from Metastorm) that enables FTP applications to leverage WebSphere MQ reliability and manageability without changing the connectivity approach used by those applications. It assures that each file is sent correctly and arrives intact, and proves the integrity of the transaction using the log information and persistence characteristics. If security is important, IBM WebSphere MQ Extended Security Edition assures who can put and get messages on and from a queue and apply security policies, without requiring changes to your existing WebSphere MQ applications (such as the FTP applications just mentioned). So, the combination of WebSphere MQ Extended Security Edition and PM4Data is powerful for regulatory compliance.

In summary, WebSphere MQ abstracts all of the complex infrastructure, such as platform difference, code page sets and data representation conversion, and it assures delivery, providing clear audit trails of what happened. Because WebSphere MQ covers a wide range of platforms, files can be shipped safely and securely, literally anywhere in your virtual infrastructure, even across Hypertext Transfer Protocol (HTTP).

When latency matters

Latency means a period of inactivity—the delay between an attempt to carry out an action and that action actually taking place. In messaging terms, it is the time lag between two points in a network—the time it takes for a message to get from its point of origin to its destination. In order to respond quickly, the lower the latency, or delay, the better. In financial markets, a delay of microseconds between a good opportunity to trade an equity and the ability to spot and carry out that trade can translate into a massive difference in the profit or loss. In this market, equity traders are essentially in an arms race, in which the trader with the shortest delay and the best and fastest algorithm to spot equity opportunities wins. Clearly, the speed of the messaging infrastructure is only one piece of the puzzle, but it is significant. Latency is typically measured in units of time. In the low-latency messaging market, microseconds, written *μs*, is the commonly used unit of time. In the past, submillisecond (1000 μs) latency was considered good. Now the benchmark is tens of microseconds to a few hundred microseconds. IBM WebSphere MQ Low Latency Messaging can achieve latencies as low as 30 μs. Tests have shown the product can handle in excess of 8 million messages a second.

WebSphere MQ Low Latency Messaging, Version 2.0 extends the WebSphere MQ product line with a new transport designed for low-latency, high-throughput messaging. It adds to the existing range of transports a messaging product optimized for the very high-volume, low-latency requirements typical of financial services firms and other industries where speed of data delivery is paramount. Although WebSphere MQ continues to provide the premier solution for rock-solid, assured, time-independent message delivery, the addition of WebSphere MQ Low Latency Messaging to the WebSphere messaging portfolio augments this comprehensive suite of transport protocols to address an increasingly broad range of quality-of-service requirements.

IBM WebSphere MQ Low Latency Messaging delivers:

- *Very high messaging throughput with low latency*
- *One-to-many multicast messaging*
- *Point-to-point unicast messaging*
- *Support for User Datagram Protocol (UDP) and Transmission Control Protocol (TCP)*
- *Positive or negative message acknowledgement*
- *Stream failover for high availability*
- *Flexible, fine-grained message filtering*
- *Traffic rate and congestion control*
- *Robust monitoring of application and network statistics, including internal and external latency*

Consolidating investments in a multivendor environment

The Java Message Service specifies how Java developers move data asynchronously using a standard approach across all Java Enterprise Edition (JEE) vendor platforms. Java is widely used but is one of many programming languages and technologies used by clients. There are many vendors who provide JEE platforms that comply with the specification. So why do we need WebSphere MQ? Two very good reasons.

First, Java is developed through the Java Community Process, allowing multiple vendors to design in agreement. Although it specifies the interface, classes and structure of Java-based messaging, the Java Community Process does not specify the degree of quality of service or transactional integrity that products such as WebSphere MQ provide. This means that developers or vendors have to create custom integration code or adapters between each application-server island. Though there is consistency at the programming level, there is no interoperability at the wire level. For example, the message structures and headers from two different Java vendors are probably not compatible, which means an application server from vendor A cannot process a message generated by an application server from vendor B.

Second, WebSphere MQ is also valuable as a fully compliant JMS engine, integrated as part of the queue manager. It provides what is known as JEE Connector architecture (JCA) support. All JEE-compliant application servers must support JCA. By configuring the application server, WebSphere MQ can be used as a JCA resource at run time. This is transparent to the Java developer. Every message is generated with a standard WebSphere MQ message header, helping to ensure interoperability between any JEE application servers and any other services not hosted on application servers.

Truly reliable Web services

When it was introduced, the concept of Web services triggered a huge interest, strong following and mass adoption. Its simplicity was its initial attraction. A set of specifications defined how to find, publish and invoke services. Not only that, but WS-I.org provided interoperability testing to ensure that implementations from multiple vendors could truly interoperate. The basic principles of Web services were formed around SOAP, Web Service Definition Language (WSDL) and Universal Description Discovery and Integration (UDDI). Many vendors were quick to include Web services support in their products. Even application vendors shipped Web services interfaces, in an effort to make their monolithic, closed applications more open. As time progressed and Web services were extensively promoted, clients were fueled by vendors' visions of what Web services could do in the future. A raft of Web services specifications rapidly appeared in the marketplace trying to deliver every aspect of integration – from connectivity to process orchestration.

Assured delivery of information over Web services is a key requirement for using Web services for processes that involve business transactions or other movements of valuable, sensitive or private information. Today much Web services traffic depends upon readily available Web technologies, and flows over HTTP or its more secure form HTTP Secure (HTTPS). While HTTP is simple to use as a transport for Web services, it cannot guarantee the delivery of information and does not have the transactional qualities needed to enable use of Web services for business-critical applications. The synchronous nature of HTTP requires that communicating Web services are simultaneously available when information is exchanged, imposing a restriction of when Web services can exchange information, and tying up Web services during the transfer.

Web services standards are beginning to emerge that are aimed at addressing this challenge of reliable Web services interaction. Web Services-Reliable Messaging (WS-RM) and Web Services Reliable (WS-R) are two such competing approaches

Reliable Web services interactions are possible today, using proven technologies that have provided a reliable data transport mechanism for over 15 years. Furthermore, this mechanism is entirely compatiable with the use of emerging Web services standards around reliability.

If you want your SOAP-formatted data to be delivered with the quality of service and benefits of a messaging backbone, you need to use a messaging product for the underlying data transfer. WebSphere MQ can send and receive SOAP data within a Web services implementation – most common within the Java 2 Platform, Enterprise Edition (J2EE) environment – and often referred to as SOAP over JMS. This approach enables Web services to take advantage of the benefits of a messaging backbone.

As Web services standards in this area mature, IBM's messaging backbone delivered by its WebSphere MQ family of products, will be able to translate this standards into technology that can be applied to real-world problems. In the meantime, WebSphere MQ - as the reliable, transactional, proven alternative to HTTP - can continue to transport SOAP robustly to support business-critical Web services. WebSphere MQ enables you to combine today's proven, reliable middleware with the benefits of Web services: standardized message formats, rapid application development and composition, and self-describing service definitions.

Web 2.0

Web 2.0 is the latest buzzword conveying the promise of rapidly created dynamic applications based on Asynchronous JavaScript and XML (AJAX), integration through Representational State Transfer (REST)ful Web services, and a whole host of other technologies. Simplicity is its core value proposition. Simple commands – such as Put, Post, Get and Delete – allow the developer to interact with resources called uniform resource identifiers (URIs). However, simplicity is only one element of technology adoption. Real-world applicability is the ultimate test.

Once again, WebSphere MQ provides the answer. If you need to create a rich Web experience with AJAX and REST, the WebSphere MQ bridge for HTTP offers a simple way to access business data from core applications and then present it to Web users, as well as to collect data from Web applications (see Figure 3). Essentially, it maps HTTP requests structured in the form of a URI as part of a REST command and maps them to WebSphere MQ resources on the other side of the bridge. The Web 2.0 developer does not need know or understand anything about how WebSphere MQ technically works but can leverage the full scalability, platform reach and qualities of service provided by WebSphere MQ, thereby preserving the Web 2.0 programming model.

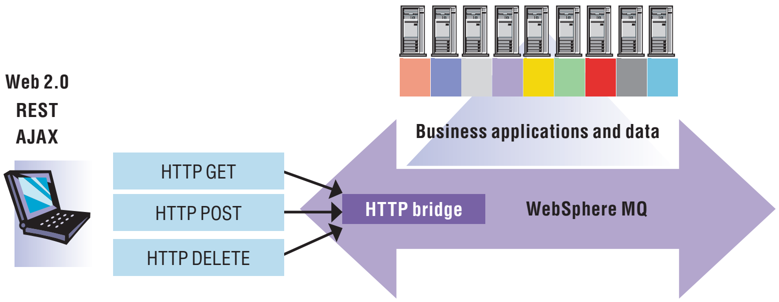


Figure 3. WebSphere MQ, Version 7.0 connects Web 2.0 with core enterprise systems, enabling a richer user experience and unlocking the value of business data.

In short, the WebSphere MQ bridge for HTTP:

- *Speeds and eases integration of new Web applications with enterprise applications and data using the qualities of service provided by WebSphere MQ*
- *Helps deliver richer content to Web users*
- *Provides a publish-and-subscribe data-movement model for Web applications*
- *Requires no WebSphere MQ skills to access data on WebSphere MQ*

If you deal with large numbers of applications that require only very simple access to WebSphere MQ, the bridge for HTTP offers a way to connect these applications without first installing WebSphere MQ clients: *zero client access.* This can help you to:

- *Simplify deployment and maintenance of large-scale, distributed applications*
- *Eliminate the need to install and set up WebSphere MQ clients*
- *Eliminate the WebSphere MQ client footprint*

Underpinning your ESB

So, we have seen how versatile WebSphere MQ is as the messaging backbone for your SOA, providing the necessary qualities of service required for your mission-critical business. That it can scale to meet unexpected business growth, help you with regulatory compliance, provide solid security, help consolidate multivendor environments into a unified messaging backbone and allow Web 2.0 developers to leverage all of this without having to know about WebSphere MQ programming constructs.

But how do you ensure that information stored or represented in different formats within these different systems can be freely exchanged without having to put a lot of transformation, routing and enrichment logic into the endpoints? This is where the ESB comes in. The ESB mediates between the endpoints, decoupling applications and reducing dependencies between them. It inspects requesting services and, based on the type or actual content of the request, can transform, enrich and route in-flight information.

When it comes to ESBs, one size does not fit all. This is why IBM today provides three ESBs: federated but integrated so that you can start quickly to address your immediate needs and embrace further capabilities when needed. The three ESBs are :

- *IBM WebSphere Message Broker*
- *IBM WebSphere Enterprise Service Bus*
- *IBM WebSphere DataPower® XI50 Integration Appliance*

All three ESBs leverage the WebSphere MQ messaging backbone and its qualities of service.

Conclusion

No matter what platforms and data you have dispersed across your heterogeneous infrastructure, WebSphere MQ is the backbone that carries it all, securely, reliably, with scalability and transactional integrity. No matter what skills you have today, no matter what skills you must have tomorrow, no matter what new fads or technologies appear on future horizons, WebSphere MQ enables you to leverage it all. So, start simple, start smart and reap the benefits for your business today.

For more information

To learn more about WebSphere MQ, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/webspheremq

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

Printed in the U.S.A.
05-08