IBM WebSphere MQ: The right information in the right place and time

By Leif Davidsen
IBM WebSphere MQ Product Manager

Contents

- Accelerate the flow of messages throughout your organization and extend business value outward to smart devices and mobile phones.
- Create a more-responsive business by supporting rapid, reliable, and secure transport of information between your applications, systems and services.
- Reduce cost when you diminish complexity and reduce maintenance.
- Take action to drive revenue by connecting to new solutions and to new technologies—from the mainframe to the mobile enterprise.

Connect your business

Starting simple

The history of information technology (IT) in the enterprise has been the history of connecting things together. Whether IT is connecting people, applications, systems or data, as years have passed, the number of connections between every aspect of the business and the infrastructure has increased, almost beyond measure.

A key driver of the growth in connectivity has been the increase in the quantity of information being generated and the ensuing use of that information. It is this growth in information use and interconnectivity that has presented new challenges to enterprise leaders.

If one were to look at just a subset of the enterprise IT infrastructure in isolation, one application at a time, there may not seem to be any major problems. Any one application and any one user might seem to be simple.

After all, connecting one application to another and exchanging data is not inherently complex, at least not if everything works. But as the saying goes, “the devil is in the details.” And the details for virtually every business today are in a huge number and variety of applications and systems, generating immense quantities of data, all of which might be of value to the business. And all of these pieces of data may need to be preserved, tracked and reviewed without loss, risk or delay. And as time passes, the connections of any one application may multiply, change and evolve as business opportunities come and go, as new business partnerships are created, and as customers become more interactive and demanding.

This then creates an increasing challenge for your business, in the face of a global competitive environment. You must meet the changing needs of your business, partners and customers by providing the right information, to the right place, at the right time. You must accomplish this without compromising information security and without losing data or data integrity. And you must ensure that the burden of providing this IT infrastructure is not so complex that it drives cost up—or drives down the responsiveness of your IT organization to the many changes demanded of it through the year.

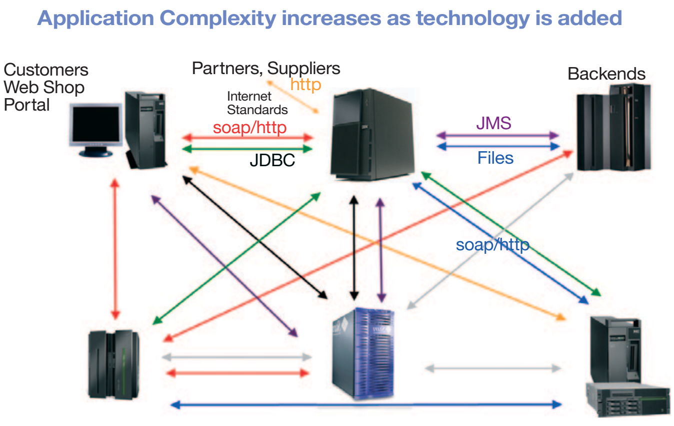

Figure 1 illustrates a growth in complexity as business infrastructure changes from simple to complex.

Increasingly complex

In order to improve today’s sophisticated IT environments, there must first be an understanding of the problems, which are becoming more difficult as systems connect and grow. These are problems that have been increasing and becoming ever more urgent over many years. Go back far enough, and enterprise computing was just a few monolithic applications, acting as processors, and data stores, performing mostly simple repetitive routines to well understood data structures. Information was made available sporadically, in batches, for overnight processing of daily activities that had been stored in databases. Understanding and control were straightforward, as were the systems.

However, with greater numbers of computing users came a greater desire for more up-to-date information, with resources and data more distributed—putting computing and the results of computing at the “beck and call” of users—heralding the beginning of the end of batch-processing. Fast forward to today, and we are coming to the end of this era. No longer are users of the enterprise IT infrastructure a subset of employees, but are almost certainly every employee, and users extend out to include business partners and customers. All of these individuals want access to everything from anywhere, on any device. Many businesses today still have batch jobs, where all the day’s work is held for processing overnight, but the batch-processing approach is becoming much less common. And this is driven by the fact that businesses no longer shut down overnight. When systems are taking work and processing it all day and night, there is no longer the opportunity to hold work during the day and then to process it at night. Not only is time-critical processing important, but the computing resources are never sitting idle, ready for batch-driven workloads.

The problems of complexity

Why is connecting so many different endpoints and users so complex, given that a single connection could be described as relatively straightforward? Obviously the scale of connections is an issue, but the other part of the problem is the risks of failure.

Connecting systems and applications is easy if everything works. The complexity comes when there is either failure, or the possibility of failure. And it is this possibility that causes the problem. If everything was understood, and all failure modes were well-defined, then again things become relatively easy. However *every* program and *every* connection in an infrastructure needs to cater for the possibility of any type of failure mode, which means that the programmers must be far more creative in imagining failure than they have generally been in the definition of programmatic success. There are just so many different possible causes of failure, and the greater the number of connections, the possible failure modes increase almost exponentially.

A good example of the need to design for failure can be seen in some of the businesses that have deployed in public clouds. Increasingly, these hosting infrastructures are becoming commonly used and can be excellent ways to extend the amount of computing resource available. However, it needs to be understood that these environments can be subject to failure, and to potentially much higher failure rates than the typical in-house data center that is under the complete control of the owning business. Where customers have deployed critical parts of their business in the cloud and have not built resilient applications that can handle failure in part of their deployment, then the entire application ecosystem might fail when just a single piece fails. So the need to design for failure and to tolerate failure puts an extremely heavy burden on application programmers, making them far less focused on their business logic and thus far less productive.

Another example is in the move to connect physical assets. Connecting to physical assets that run outside the data center means that a far greater rate of failure must be expected. Failures could extend to virtually any component or linkage. Therefore, the applications that expect to receive data must be built to expect failure, and the applications must be designed to monitor, tolerate and recover from any type of failure mode. Failure to decouple the links between the applications and the sources of data, especially when they are likely to be so numerous, would be catastrophic. Applications must not be designed to expect data, especially transient data, but the application and the device must be expected to intercommunicate—but to execute completely independent of each other.

A last example to consider is the accelerating pace of business itself. Back-office systems have been designed and built for years to run with extremely high performance and throughput. These back-office systems have been measured by transactions per second, and response times to the submission of an individual piece of work. As teams move to accessing and even running these solutions outside the enterprise, they continue to expect the reliability and the speed of response seen within the enterprise domain. To try to achieve this will require substantial evaluation of how to implement this connectivity to try to balance these requests. When it comes to responding to requests, delays of even a second or more may lead to users to believe that there is be a problem, and a negative perception can lead to rapid dissatisfaction with the solution.

Accelerating change and potential solutions

Many of today's back-office systems are likely to have been running for a number of years. The systems which drive those enterprise applications and databases are likely to be far more recent. This is due not just to the transformation of new technology, but by a rapidly increasing number of new business opportunities, which tends to be the primary driver of change for infrastructure.

The massive change and growth in new applications—whether new sources of data, or new ways to interact with new or existing employees, customers or partners—has produced a growth in workload for many systems. Some of that growth is seen in peak-time spikes of work, driven perhaps by external events or social media. Other growth is in a widening of the peak load of applications. In the past, some applications would have been only busy for a few periods in the day. Now, with mobile connectivity or with connected devices, more data might be coming through at a more continuous rate. This constant usage is one reason that the batch-processing approach is ending, with the other reason being the desire for instant processing of all transactions—something that is not possible with batch.

When business infrastructure, including applications and systems, was less driven by the need to change, modifications, migrations and updates were done with long advance planning. Stability was the watchword. However, being ever more responsive, in every sense, is now essential. Both the business and its infrastructure must now be able to be targeted at a new opportunity and must be capable of going to market with the opportunity without delay. This means that there is no chance to build an entirely new set of applications, and that there is no way to interrupt existing applications to make disruptive changes to them. Everything must be capable of being seamlessly modified to handle new ways of running, in order to allow the business to respond to new opportunities. This demands that the applications must be written and connected together to support success in this area.

Accelerating change with mobile connectivity

One place in which business leaders are seeing new opportunities is the massive growth in mobile computing. There are, of course, many ways in which this opportunity is being seen. There are opportunities to engage with mobile customers to drive more existing business. There are also opportunities to take advantage of engaging with mobile customers to create completely new business opportunities. Either of these opportunities could be driven simply by tailored access to existing consumer-facing business access that is suitable for mobile use, either with a browser-based solution or with a mobile application. There is also increasing use of location-based sensing leading to new business opportunities. With the customer's client device able to directly engage with businesses based on physical location or other attributes then completely new and timely offers to drive business can be made. All of this will require interfaces to the mobile environment and to the existing back-end business systems. And the solution (or solutions) will need to be highly dynamic, changing to reflect the ways in which the customers are using the capabilities—and those customers and mobile endpoints could number in the millions or higher. These are not environments in which you write an application once, and then maintain it for 10 years. The ways in which your business engages with the customer will have to change rapidly, but your back-end systems must to continue to run with enterprise security, reliability and efficiency.

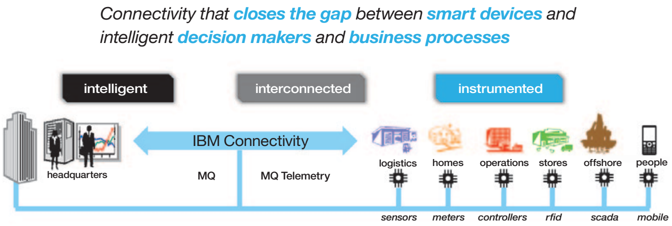

Figure 2 illustrates growth in the mobile environment and in the "Internet of things" and reveals the importance of connecting from these devices into your back end systems.

Accelerating change with the "internet of things"

Another new set of opportunities is related to the "internet of things." For many years, business infrastructure has been focused inwards on directly connected IT systems within the business environment, even though in many businesses there were thousands of other devices deployed across the enterprise which were generating and consuming data—but doing so outside of the IT environment. Therefore, the data these devices were generating was rarely used within business, and if the data was used it was rarely consumed and acted upon in a timely manner. Now, however, business leaders see the possibilities and opportunities of gaining access to this data, and taking action on the events identified by that data. This can create opportunities for identifying potentially useful new business events, or for cost savings, perhaps identifying when devices are running unnecessarily or are consuming too much energy. Again, many of the back end systems will need to continue to run and process the data that is coming in, although this might drive increased workloads, and increased numbers of connections. And of course the devices that are at the edge of the infrastructure—devices that are to be connected to the back-end systems—will need to run "low footprint" code. These devices must make use of a lightweight transport protocol to ensure that the solution can be deployed widely, into the environments in which the devices are deployed and used.

Accelerating change by deploying into the cloud

Until the last couple of years, business leaders had really two options for their IT infrastructure. They could buy lots of IT hardware, and hire a multi-skilled team of specialists to build, deploy and maintain both the hardware and software running their business. Or they could outsource virtually all of their IT to a large IT services provider—a provider who had both the staff and the hardware; these cases typically involved a long-term contract to determine the usage and service levels. The circumstances of the in-house IT resources could vary, from being heavily over-committed to sitting idle. Externally, the IT services provided under contract could sometimes be seen as an "anchor" holding back the innovation and rapid changes needed, with the contract being defined to meet the needs of today instead of tomorrow. The growth of cloud computing has created a change in this picture. *Cloud* is not a single thing, but in fact can be used to describe the more-dynamic allocation and use of in-house IT resources, or *cloud* can mean the use as needed of publically available IT resources on a usage-based model. Or *cloud* could be any combination of these—and even other—deployment types.

Business leaders are likely to be considering which of their IT systems, and which of their applications, are suitable for deployment in the cloud—whatever *cloud* might mean to them. Perhaps this might be for cost-saving reasons, or it might be for more flexible deployment in response to a growth in usage, or it might simply be the fastest way to get something done. Whatever the reason, applications will need to become ever more agnostic about their runtime environment, and the connections they have to the rest of the enterprise. They will need to be loosely connected, but reliably and securely connected, enabling business teams to make decisions on deployment completely independent of the application architecture itself.

Accelerating change driven by Big Data

The growth of IT usage has been driven by, and mirrored by, the growth in data. Almost by definition, business IT is the consumption of data in a business context to produce a business outcome. And the more that business makes use of IT to operate, and to go after new business, then the closer the organization's relationship becomes with data, and the more the volume of data itself increases. However, as described in the paragraphs above, leaders are now increasingly extending the reach of their businesses. This reach is spreading to the mobile world, including physical devices, each connection, each transaction, each customer interaction creating more and more data. And for every piece of data, and for every business opportunity, there is more processing required, more rules to be evaluated and more actions to be taken. Plus, of course none of the data is discarded. In total, this activity is generating staggering amounts of data. This Big Data would overwhelm any existing

IT infrastructure—and clearly any data that is produced and kept needs to be valuable and usable in some context, or it is a pointless waste of resources.

Applications and the business IT infrastructure itself needs to be robust enough, and must perform fast enough, to consume and take action on all the data and all the events, in a timely manner. Other data can be held and processed to extract the remaining value. Indeed, the applications and infrastructure must be both high-performing and "smart." Applications and infrastructure must not only be able to recognize the data that was previously important, but must change to be able to identify and act upon some of the rest of the Big Data, creating the new business opportunities that Big Data represents, even while Big Data creates new challenges to the IT infrastructure—the systems, the applications and the connections between them.

Connecting your business systems and applications

Every business already has dozens, if not hundreds or even thousands of applications. These applications are deployed on many different IT environments—whether predominantly deployed in a data center, or scattered through many branch offices, warehouses or throughout multiple countries. Some of these systems will be connected together. Others will run in isolation, or will be connected to some subset of other applications. Business leaders are likely to be evaluating not just how to connect new applications to address new business opportunities (see examples above), but also whether the connections between existing applications and systems are meeting today's challenges and are ready for tomorrow's.

Applications are likely to be connected in a variety of ways, and it makes sense to review some of these options. Each option tends to have reasons why it has been chosen in the past, reasons why it may be chosen today and also why it might be replaced going forward.

File transfer

One approach that is used to connect many business applications today is one that has been used for years—even decades: file transfer. File transfer is typically seen as straightforward, as the applications write data to the file system, and then a file transfer routine moves the file to another system to make it possible for another application to use the data. This approach moves the data without requiring the applications themselves to be connected in any way. Also, for virtually all systems, the generic file transfer protocol (FTP) capability is always there, and it can be very simple and quick to write a script to move the data. However, although this process is seen as straightforward, there are many points of failure. The data that is being stored in the file does need to be in the right format. And the file transfer can fail, leading to old or incorrect data being used by the receiving application, and there is little or no security or management-by-standard in most file transfer solutions. The file that is being moved can be incomplete or corrupt. Although file transfer is still widely used, and is very quick and easy to set up, most business leaders want to have more reliability, security and reporting. It is all too easy for file transfers to fail or go to the wrong place with no record or logging of this. So although it is common to see this scenario, there are a lot of ways in which the approach can be improved.

HTTP

Another approach that has grown up along with the browser-based front ends to web-server based applications is to use HTTP as a standardized and general connectivity solution. Again, this approach is used because it is widely available on virtually any platform, is familiar to most developers and therefore can provide a way to move data between systems quickly to meet development-project goals. Both users and developers are familiar with HTTP, and people feel secure using HTTP if it is protected by Secure Socket Layer (SSL). However, although HTTP has all these benefits, there are a number of drawbacks to this approach. HTTP is designed to provide tightly coupled connectivity. It is not appropriate for time-independent processing in which you can “fire off “a request without waiting for a response or having to retry if you fail. HTTP requests are not able to persist over network failures, or to move data under transactional control, or to be anything other than point-to-point connections. Many of these restrictions do not matter for a lot of data movements, but in cases in which business critical data is being moved and a number of other factors need to be considered, then these are key issues with the HTTP approach.

Embedded JMS

Assuming that applications are being written using Java application servers, yet another approach is to use Java Message Service (JMS) application programming interfaces (APIs) and make use of the Java Message Service that is embedded in many application engines. This approach will be simple and straightforward for Java programmers who are comfortable using their preferred Java application server, and when all the applications involved are running within that specific Java engine. However, the limitations mentioned above point to the possible problems with this approach. Not all applications are written in Java, and the applications do not necessarily all run in the same types of Java application server, which is a problem for most JMS implementations since different implementations cannot exchange messages. So using embedded JMS providers to connect applications is perfectly valid for only a limited number of use cases.

Enterprise messaging middleware

One final way to move data between applications is to use a dedicated middleware layer for enterprise messaging. This is a similar approach to the embedded JMS messaging mentioned above, but avoids the limitations of being Java only, and of only connecting to other instances running in the same application server environment. The enterprise messaging approach also avoids some of the limitations of HTTP, since you are able to use more loosely coupled requests, including both time-independence and transactional integrity. And although it is different from a file transfer solution, enterprise messaging can be used to enhance, replace or update an existing or a new file transfer-based approach without application disruption. One of the leading solutions in this space is IBM® WebSphere® MQ middleware, which has been used by IT leaders to address this solution for nearly 20 years. This paper will now review this IBM offering in more detail, highlighting how WebSphere MQ can provide a solution to multiple different customer requirements for data movement, and how WebSphere MQ matches up against the other solution approaches for different deployments.

What is IBM WebSphere MQ?

Messaging-oriented middleware

At its simplest, WebSphere MQ provides messaging-based middleware (MOM), sometimes described as *enterprise messaging*, enabling integration between applications, systems and services. The principle behind MOM is that, instead of connecting and directly exchanging information with each other, applications send information over an indirect middleware layer. This layer packages the information that is to be exchanged as a message and moves the message through a queuing system to send the message to the receiving application.

With messaging-oriented middleware such as IBM WebSphere MQ an application uses a simple application programming interface (API) to send a message by moving the data into the messaging middleware environment. The receiving application then uses the same API to retrieve the data from the MOM environment. Thus, the application programmer can rely on the messaging environment to deal with aspects of transmission failure and error-handling, and the application can focus purely on the business logic. And the API is relatively simple to learn and is standardized throughout virtually every platform.

The nature of the messaging system also enables the programming logic to be asynchronous. Applications can be freed from waiting for a response or from checking to determine whether the receiving application is available or has confirmed that it has received the message. With all this "checking for success" now managed by the messaging middleware, programming resources are freed up and applications are simplified, all while strengthening reliability and manageability for the connection. If all applications, networks and data are available, there is no latency disadvantage to an asynchronous programming model, and real benefits accrue quickly. And for the numerous cases where there is a problem, a delay or a failure, then this asynchronous approach enables applications to handle failure without programming complexity, and without constantly polling to check for success.

Using an indirect connection, through the messaging layer, means that the sending and receiving applications can be coded very differently, using a much simpler approach to architecture. Instead of business logic for each business function being tangled with the inbound and outbound connectivity interface, each function can be coded cleanly. The function can then be invoked independently by multiple different applications, systems and services. Clean coding vastly increases the possibilities of reuse, which can increase the business value of the application, support the return on investment and also create the ability of the business to be more responsive to change.

Gain business value quickly with IBM WebSphere MQ. Appliances are pre-configured and can be deployed in days.

WebSphere MQ—architecture and capabilities

Since 1993, WebSphere MQ has been the leading choice for MOM, offering assured, once-and-once-only delivery of messages between applications and systems on virtually every commercial IT platform. This IBM middleware is used as the fundamental messaging backbone for mission-critical environments by thousands of IBM clients in a wide range of regions and industries. WebSphere MQ is available on more than 80 platform configurations, offering standards-based APIs and proprietary approaches for maximum programming flexibility.

WebSphere MQ enables businesses to connect applications using a point-to-point messaging approach. Applications can be coded specifically to move messages from one application to another. Coders can take advantage of WebSphere MQ to simplify the connectivity interface and the other supporting logic that otherwise would be required. Many thousands of organizational leaders have been doing this to remove business risk from their application connectivity. And the nature of their use of WebSphere MQ within their applications has enabled them to be more suited to greater componentization and reuse, as part of service-oriented architecture (SOA) or as part of a move to the cloud.

Transactional exchange of data

One of the fundamental capabilities of WebSphere MQ is how it acts as a transaction manager. As described previously, WebSphere MQ is used to send and receive data between applications. However, take the case of a bank that is moving money between two accounts, or the case of a travel agent who is booking a seat on a plane. Either of these circumstances requires a transactional exchange of data.

It is critical that 1) information moves and both sides of the exchange are updated with the new state, or 2) nothing happens.

As a business leader, you do not, want to move money from one account to another, and have the same money credited to both accounts. You also do not want to try to reserve a seat on a plane, and fail, yet have the plane's ticketing system see a seat reserved. Avoiding such errors requires a transaction manager. WebSphere MQ is built on top of a transaction manager, so that messages are moved as part of a transaction, and resources under the transactional control of a resource manager can be updated as part of a transactional unit-of-work. This gives the business a level of assurance that the message will be moved once and once only, with no degree of uncertainty, no need for duplication and no requirement for additional logic in the application for verification.

Although a high level of transactional management is appropriate or necessary for many forms of information exchange, many other message types are transient and do not require transactional control. For such messages, WebSphere MQ can be configured to allow a lighter approach to assured delivery and to apply message persistence. For example, when an account balance is being requested, or airplane seat availability is being queried, there is no need for any transactional locking, and if required, the same message could be sent repeatedly. Therefore, WebSphere MQ can take a more lightweight and simple approach to the exchange of messages so that the message is sent faster, with less overhead and with less impact on overall system performance.

Persistence

When any transaction can be critical for your business or your customer, you need absolute assurance that even system failures will not affect the handling of the customer data. As part of the transactional support provided by WebSphere MQ, each message can, if required, be persisted, or written to disk, as a way to maintain the integrity of the information during the movement of the message. This preserves the message and completes the transaction without losing data if a failure occurs at either end during the transaction. Persistence is fundamental to the ability of WebSphere MQ to assure once-and-once-only delivery of messages without burdening the application with complex error-handling code. Persistence can become very important when the data being moved is critical, when the systems through which the data is moving are subject to failures, or when it would be complex to recreate or resubmit the data. In many environments, even though failures are rare, they will always occur, and then the costs and efforts of recovery can determine whether it would have been more efficient to use persistence to ease recovery.

Publish/subscribe and multicast

As businesses seek to extend the application simplicity and flexibility that result when WebSphere MQ provides the connectivity layer, some want to take the next step to even greater flexibility by using the publish/subscribe approach rather than a request/response mechanism. This publish/subscribe capability completely disengages the links between the sending and receiving applications. When you use this approach, an application sending a message does not send it to a specific receiving application but rather publishes the message with a topic description, and any interested applications can then subscribe to that message topic. Any number of applications from virtually anywhere in an enterprise can then receive and use the data, with no effect on the originating application. The Publish/Subscribe capability can significantly reduce required maintenance updates to support new applications and can increase the potential for reuse still further.

Publish/Subscribe networks can grow very large, and when there are thousands or more subscribers it can be important to ensure that all subscribers receive the information simultaneously, rather than having some network latency and subscription-list latency. If this reduction in latency for large numbers of subscribers is important, then multicast is a form of publish/subscribe that allows information to be delivered to very large numbers of subscribers at the same time. The WebSphere MQ team added a multicast capability as a part of the IBM WebSphere MQ V7.1 release in the fourth quarter of 2011.

Managed File Transfer over a WebSphere MQ network

Many files, as described above, contain critical business data. Applications write the data to a file, and then File Transfer Protocol (FTP) is used to copy the file to another system, to allow the data to be used in the other system, typically by another application. However, as described earlier in this whitepaper, this approach can be unreliable and unsecure—and unmanaged, even if additional layers are written to support the FTP scripting. WebSphere MQ enables files to be sent as messages over the WebSphere MQ infrastructure. Not only does this increase the reliability of delivery by taking advantage of the inherent robustness of WebSphere MQ, but all the other capabilities of WebSphere MQ are also available, along with the security aspects and all the management, tracking and logging aspects are able to be used. This can transform file transfer into a fully capable method of moving large amounts of data, without any requirement to change the existing business applications that continue to write to files and read from files.

With the release of WebSphere MQ V7.5, the Managed File Transfer function is available as a part of the WebSphere MQ package as WebSphere MQ Managed File Transfer, subject to separate entitlement.

IBM has been the market share leader of the AIM market for more than a decade, and it is the market share leader in business process management suite (BPMS), enterprise service bus (ESB) (including AIM appliances), stand-alone B2B gateway software, message-oriented middleware (MOM) and transaction processing monitor (TPM).[1]

By providing Managed File Transfer capability running over WebSphere MQ, your message traffic, whether application messages or files packaged as messages, can be tracked to completion using the single control dashboard that is provided as a part of WebSphere MQ. This approach provides real benefits for users, especially administrators. These individuals can review what is moving through their business with a single glance, whether the contents are a file or an application message. All types can be treated and managed in the same way.

A comprehensive MFT solution with IBM WebSphere MQ

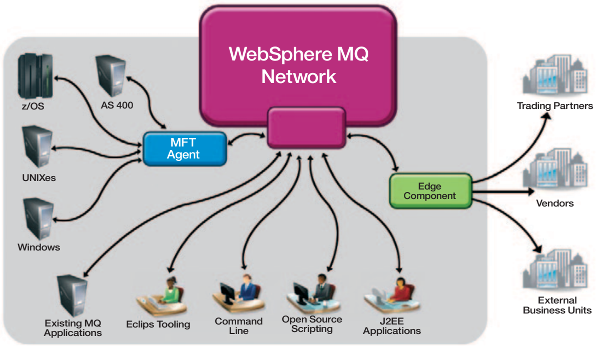

Figure 3 shows how you can move files throughout your enterprise and beyond when you use IBM WebSphere MQ.

Securing WebSphere MQ and securing message contents

Whether they are operating completely within the enterprise, or are connected to the wider internet, all systems today need to be secured. There are many aspects to security for an enterprise messaging system. Given that WebSphere MQ moves a large amount of critical business data from system to system, WebSphere MQ itself must be designed for security, with authentication to connect to the system, ensuring that unauthorized users cannot gain access and change the configuration. Then there is security of the messages flowing over WebSphere MQ. Here, there are a number of security features provided by WebSphere MQ. There is improved security on the connections between systems using WebSphere MQ because you can take advantage of SSL and other connection security. This helps to ensure that the WebSphere MQ messages that are flowing over the wires are secure. However, due to the message queuing approach, messages may be written to a queue at a Queue Manager as the messages flow through the network, for persistence. If the message contents are not encrypted, then when the message gets written to disk at the MQ Queue Manager, this means it is no longer encrypted by the SSL. Clearly, the applications could be rewritten in order to send encrypted message contents, but in some cases the applications will be unchangeable. WebSphere MQ has an additional component available in the WebSphere MQ V7.5 package, subject to entitlement, that enables the message contents to be encrypted throughout the process, even when the message is persisted in a queue. There is no change required to the applications at either end, and the message contents will be encrypted from application to application.

Messaging clients for extending access and transactions throughout the infrastructure

Business leaders who deploy WebSphere MQ generally install and maintain code for the systems that need to connect together using WebSphere MQ with WebSphere MQ following a client/server architecture. The server runs as the Queue Manager and all queues are on the server. WebSphere MQ clients can be remote from the server or can be co-located. Message persistence is supported from the moment when a message is written to a queue on the server, so depending on the nature of the data being moved, either clients or servers may be deployed through the infrastructure.

Some data that is being moved from a client may need to be part of a transactional unit of work. Here, the data from the client is being moved from a managed resource such as a database on the client, and through the WebSphere MQ Transaction Manager. For this type of solution, an Extended Transactional Client can be deployed in place of the standard client. This approach enables coordination with an external resource manager to make it possible for the client connection to move data as part of a transactional unit of work, providing excellent protection for the data movement and for the business driving it.

Using WebSphere MQ as a JMS provider

As discussed above, many business leaders use applications coded in Java, running in an application server. Part of the Java standard is JMS—providing a messaging service as a part of the programming model. Using JMS is therefore a natural way for such applications to move data into an application and out of an application, and many application servers include a JMS provider to “listen” for JMS requests and execute them. However, this will only work if both the sending application and the receiving application are running a common JMS provider. Although JMS is a standard API, the wire format is not standard, so different JMS providers cannot exchange messages. WebSphere MQ can act as a JMS provider in more environments than can other messaging provider, which makes it possible for Java programmers to use JMS in more places. Programmers can send and receive messages with any other

WebSphere MQ application, whether the application uses JMS or not, and whether the Java programs are running in the same type of application server or not. This helps to ensure that programmers are free to gain the benefits of messaging without the complexity of interoperability concerns.

Automatic failover for high availability with IBM WebSphere MQ

Keeping production environments running and connected is critical for many businesses. A failure of connectivity is just as important as an application failure if connectivity failure prevents work from being done. Most hardware platforms offer some form of high-availability solution, but this high availability can be complex to set up and can be expensive. WebSphere MQ supports these hardware-based solutions and also offers a software-based solution to provide automatic failover for WebSphere MQ Queue Managers, providing High Availability without hardware dependency. In the case of a Queue Manager failure, another instance of the Queue Manager takes over the work and the transactions of the failed system, with no intervention needed.

Connecting physical assets and mobile devices

For years, businesses have been focusing on their internal IT infrastructure, based around servers and tightly coupled clients. Now there is an explosion of interest in extending the business IT infrastructure out to the growing set of mobile connected devices—typically smartphones and tablets. There is also a growing recognition of the new set of opportunities around the data that is sent and received by physical assets—which might be sensors, GPS locators, utility meters, medical devices or just about any other device which can produce or consume data. Mobile devices and physical assets tend to be limited in terms of what is supported for installation, and mobile devices and physical assets also operate in environments in which there may be both limited power and limited bandwidth for connectivity.

WebSphere MQ supports a solution that enables connections to these types of devices: a transport protocol called IBM WebSphere MQ Telemetry Transport. This open-standard protocol allows for small lightweight clients to be written that can run in a very small footprint on a mobile device or a remote physical device. The WebSphere MQ Telemetry Transport protocol is designed to operate over an unreliable low bandwidth connection, and therefore is very concise, helping to ensure that little power is consumed on the device from running the client or from sending the message. This protocol also helps to ensure that the message takes up very few transmission bytes, and that the message is only active when needed. Applications that use WebSphere MQ Telemetry Transport then connect to a WebSphere MQ Queue Manager that is using the WebSphere MQ Telemetry component, which is included in the WebSphere MQ V7.5 package but requires separate entitlement if used.

By using this protocol, businesses can extend applications to gain better awareness of what is happening beyond the corporate data center, and they can respond faster to events that are identified by these physical devices. Also, new and existing business opportunities can be driven by new users and in new places and times through mobile device connectivity and revenue can be gained from the new opportunities that are created by mobile access to the business.

Using WebSphere MQ on IBM z/OS

For many businesses today, IT infrastructure is widely available throughout the enterprise, but critical business applications continue to run on mainframes—specifically IBM System z® hardware running the IBM z/OS® operating system. These applications clearly need to connect to other parts of the business, and WebSphere MQ exists on both z/OS and other platforms to provide this capability. There are some important differences on the z/OS platform which are reflected in the

WebSphere MQ offering on that platform. A key point is that WebSphere MQ on z/OS is written natively for the z/OS platform. This helps WebSphere MQ to run extremely efficiently, and helps to ensure that WebSphere MQ is able to exploit the many key features native to the z/OS platform. A recent release of WebSphere MQ on z/OS demonstrated the middleware exploiting the coupling facility more efficiently when using large messages, and improving recovery when using the coupling facility. Also, WebSphere MQ on z/OS, with the latest release is able to scale to one million messages per second (2KB non-shared) on a single Queue Manager on a 30-way IBM zEnterprise® z196 server.[3]

Extending messaging beyond connectivity

WebSphere MQ moves data and files without reading, understanding or changing the contents of the message that is being moved. In many parts of the business, that is all that is required. Data on one system, under the control of one application, is simply required in the same format on another system for an application designed to use the same data.

In the same way, even if a business moves a file with WebSphere MQ, the file will be moved with assurance, but there is no assurance that any application on a remote system will be able to make use of the file or the data held within. However, increasingly in the dynamic business environment, messages need to be transformed, understood and acted upon, and not just simply connected. As messages or files get moved throughout a business, it is useful or necessary to do additional work on the data before it reaches the receiving application or system. This work could involve reformatting, enriching or truncating the data. Having a flexible, configurable integration environment working as an enterprise service bus (ESB) that can perform these types of integration functions is an essential part of the integration solution for many businesses to gain the benefits from the integration, and who wish to realize the value of the data being moved.

Choices for deploying WebSphere MQ in cloud environments

In recent years, leaders have seen a huge growth of interest in virtualization and the cloud. For many businesses *cloud* means a publically hosted environment, paid for when used. For many other businesses *cloud* can refer to a form of virtualization to maximize use of existing hardware and software on-premise. The reality is a blend of both. Cloud deployments will likely lead to businesses selecting some services to be hosted on a public, external shared system, and other environments will be on internal hardware, configured and deployed as needed.

Accessing the applications, services and data on any of these environments requires a reliable, secure connectivity mechanism. WebSphere MQ is purposely designed for use in these scenarios, because it is in the nature of cloud deployments, especially in externally hosted environments, to be subject to events beyond the control of any individual user. WebSphere MQ can help to deal with service interruptions and failures. The IBM offering is also available as a WebSphere MQ Hypervisor Edition to be deployed into virtual machines within the business environment used as private clouds. One example of this is in providing WebSphere MQ messaging as a system pattern for the IBM PureApplication™ System.

What does a smartly interconnected enterprise look like?

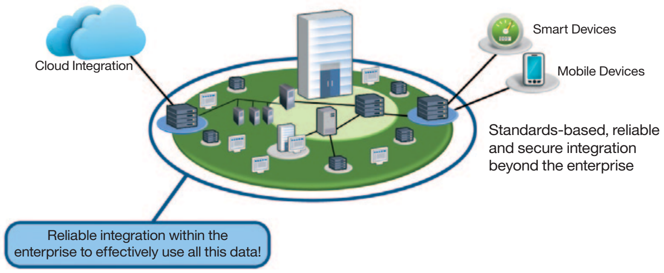
Cloud Integration
Smart Devices
Mobile Devices
Standards-based, reliable and secure integration beyond the enterprise
Reliable integration within the enterprise to effectively use all this data!

Figure 4 the value of MQ integrating all endpoints across the organization.

Reasons to use WebSphere MQ in key scenarios

Why use WebSphere MQ instead of HTTP to connect mobile or remote physical devices?

- Is the connectivity between the remote device and the backend infrastructure brittle and prone to interruptions and failure?
  - HTTP connections can easily be interrupted and can lose data when connectivity is inconsistent. WebSphere MQ offers a more-reliable and robust transport that delivers data once and once only.
- Is the back end application required to add complex error handling?
  - To cope with the higher rates of failure that are expected in remote connectivity, HTTP-based applications need to have complex failure handling. WebSphere MQ removes some of the complexity from applications, keeping them simpler and focused.
- Can the mobile infrastructure perform push-notification?
  - HTTP applications rely on connectivity that is initiated by the client. More-flexible connectivity provided by WebSphere MQ allows for two-way connectivity.
- Can publish-subscribe applications be built to connect to remote devices?
  - HTTP-based applications are locked into synchronous communications. WebSphere MQ allows for adaptable publish-subscribe connectivity to exchange data
- Are there concerns over bandwidth consumption?
  - For remotely connected devices the bandwidth available can be limited or costly. WebSphere MQ uses WebSphere MQ Telemetry Transport to provide an extremely efficient protocol to reduce bandwidth consumption to optimal levels.
- Are there concerns over power consumption?
  - Mobile devices have limited battery lives. HTTP can use high amounts of power in communicating with the remote server. WebSphere MQ, using WebSphere MQ Telemetry Transport, minimizes the power consumption through efficient applications and connectivity.

Why use WebSphere MQ instead of an embedded JMS provider?

- Is there only a single Java application environment in the infrastructure?
  - JMS providers may offer the same API, but the providers are not able to interact and exchange messages, so a platform-neutral JMS provider such as WebSphere MQ is a strong solution for connecting all Java applications.
- Are there any non-Java applications that need to be connected?
  - Most JMS providers can only connect with other JMS-enabled applications. WebSphere MQ offers common message-based connectivity between JMS, Java and non-Java apps.
- Does the embedded JMS provider deliver full transactional integrity in the case of a failure?
  - When moving critical data and updating transactions, can the JMS provider offer the robust messaging integrity that is built into WebSphere MQ?
- Can large messages be handled by the embedded JMS provider?
  - Some Java environments can struggle with large message sizes. WebSphere MQ has demonstrated for many years its ability to move large messages.[2]
- Is there an integrated dashboard to show and to manage the flow of all data moving in the infrastructure?
  - Does an embedded JMS provider in a Java application environment offer a dashboard for ease of configuration, operation and visibility to all data and activity moving through the message layer, as does the single dashboard of WebSphere MQ?

Why use WebSphere MQ instead of File Transfer?

- Are FTP scripts used, and how do they cope with changes and maintenance?
  - With networks, applications and data constantly changing, FTP scripts can be a hidden cost and a business risk. WebSphere MQ provides a single point of control for moving files in a managed way without complex and risky updates and maintenance needs.

- Is there a record of success or failure for file transfer?
  - Without knowing whether files have successfully reached their destination, business processes can start to operate in old or incomplete data. WebSphere MQ delivers files more reliably and offers visibility through an audit trail to the success or failure of delivery.
- How can FTP scripts cope with different failures, given that FTP can fail up to 20 percent of the time?
  - File transfer can fail in many ways. FTP scripting can be highly complex, leading to increased costs, risks of failure and a demand for expert skills. WebSphere MQ helps to remove the complexity from file transfer.
- Are there issues with the security of files and file movement?
  - Data that is held in files can be valuable and confidential. WebSphere MQ can provide security-rich, more-reliable delivery, including encryption of file data being moved.
- Is the data in files used by other applications in a timely manner?
  - File data is moved to be used. Delivering files to a file system can be inefficient if the applications are not aware of the new data. Moving files through WebSphere MQ can either trigger action—or even move the data directly into a consuming application.

Why use WebSphere MQ instead of open source messaging or home-grown connectivity?

- How much of the IT budget is spent on maintaining connectivity between applications?
  - WebSphere MQ reduces total cost of ownership by reducing the time and effort needed to maintain the integration layer.
- Can application changes be made quickly or are changes and updates delayed due to complexity?
  - Applications connected using WebSphere MQ can focus on business logic without unnecessary complexity for more-rapid changes, helping business teams to respond faster to new business opportunities.
- What happens to in-flight data when there is a failure?
  - WebSphere MQ offers robust transactional integrity and provides assured message delivery without requiring additional programming. “Home-grown” connectivity, and even some open-source messaging solutions, struggle to maintain data or transactions when failures occur.
- Can you use a single dashboard to view data movement and to control the environment for all the data that is flowing through the infrastructure?
  - WebSphere MQ comes with powerful tooling to configure, deploy and manage the flow of messages and files over the WebSphere MQ infrastructure, providing a single point of visibility and control.
- Is there built-in High Availability support or must you have hardware-based High Availability?
  - WebSphere MQ supports High Availability through native capabilities, offering automatic failover without requiring hardware-based functions.

Business value

WebSphere MQ is designed to provide you with the most-complete solution for connecting applications, systems, and services to move data and files more reliably, more securely, more rapidly and more simply. Available for the widest possible set of platforms and programming environments, this IBM solution helps to remove the complexity from applications and enables a business teams to focus on core business functions. WebSphere MQ provides critical connectivity infrastructure for virtually every part of your business, extending from the mainframe to the mobile.

WebSphere MQ has been connecting applications in companies for nearly 20 years. Start to discover what IBM WebSphere MQ can do for your business today.

Next steps

Download the trial version of WebSphere MQ

Read the redbooks

Watch demo

For more information

To learn more about IBM WebSphere MQ, please contact your IBM representative or IBM Business Partner, or visit the following website: http://www-01.ibm.com/software/integration/wmqfamily/

Additionally, IBM Global Financing can help you acquire the software capabilities that your business needs in the most cost-effective and strategic way possible. We'll partner with credit-qualified clients to customize a financing solution to suit your business and development goals, enable effective cash management, and improve your total cost of ownership. Fund your critical IT investment and propel your business forward with IBM Global Financing. For more information, visit: **ibm.com**/financing

Messaging middleware is software that provides an interface between applications. This interface makes it possible for applications to send data back and forth to each other asynchronously. Data sent by one program can be stored and then forwarded to the receiving program when the receiving program becomes available to process the data.

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
October 2012

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary. It is the user's responsibility to evaluate and verify the operation of any other products or programs with IBM products and programs.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

The client is responsible for ensuring compliance with laws and regulations applicable to it. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the client is in compliance with any law or regulation.

Actual available storage capacity may be reported for both uncompressed and compressed data and will vary and may be less than stated.

[1] Gartner Market Share Analysis: Application Infrastructure and Middleware Software, Worldwide, 2011, Document G00233638, April 23, 2012

[2] IBM wins Royal Academy of Engineering MacRobert Award for IBM WebSphere MQ http://www.raeng.org.uk/prizes/macrobert/winners/win2004.htm

[3] Peter Toghill and Tony Ford, *IBM WebSphere MQ on z/OS*, November 15, 2011

Please Recycle