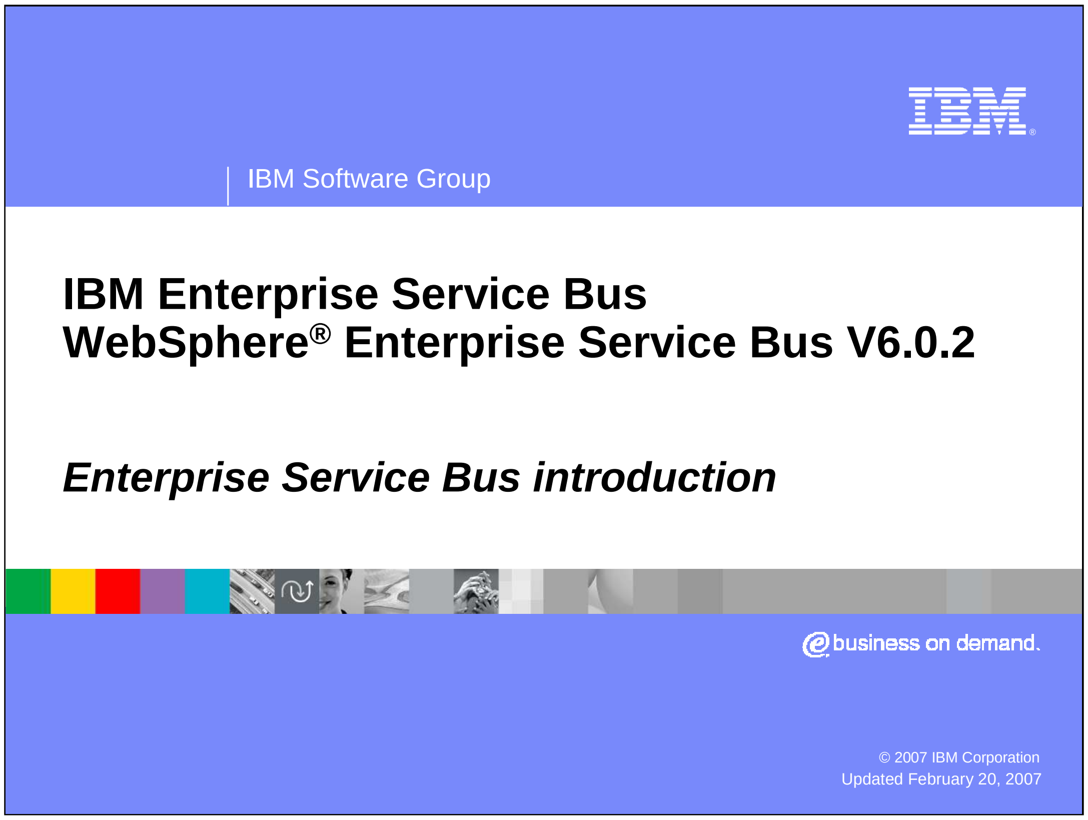


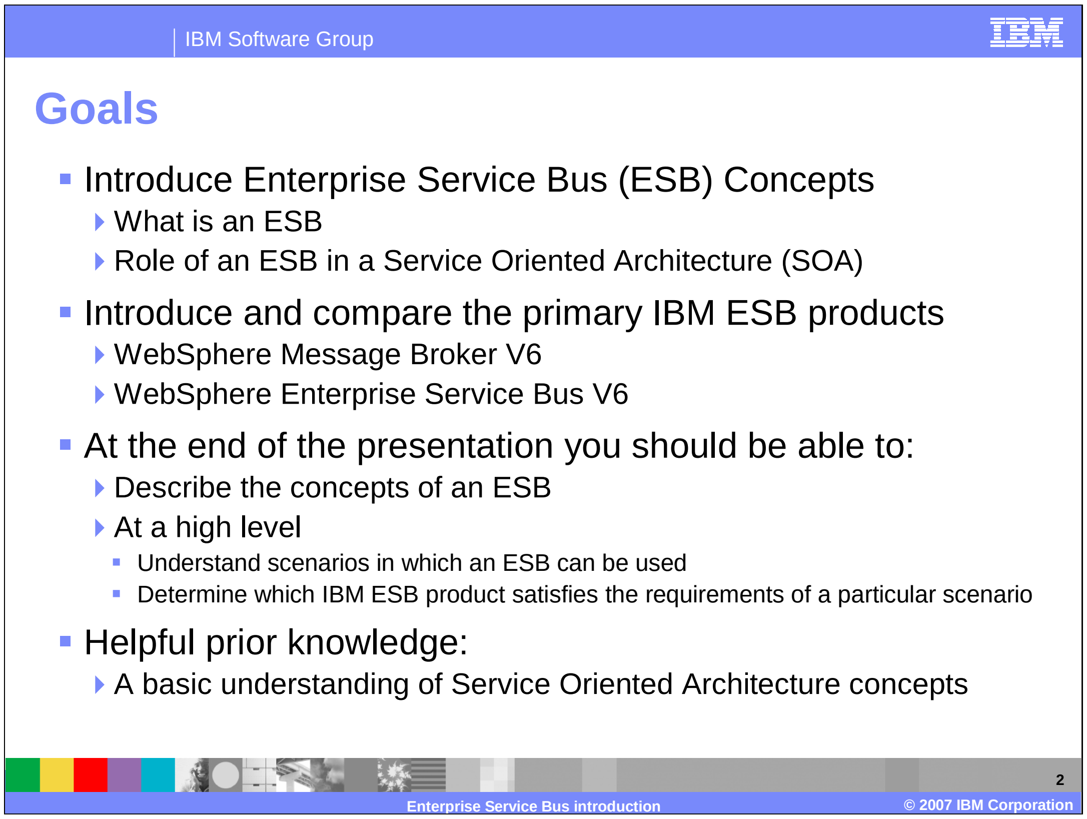
IBM Software Group

IBM Enterprise Service Bus WebSphere® Enterprise Service Bus V6.0.2

Enterprise Service Bus introduction

@business on demand.

This presentation provides an introduction to Enterprise Service Bus concepts and associated IBM products.



Goals

- Introduce Enterprise Service Bus (ESB) Concepts
  - What is an ESB
  - Role of an ESB in a Service Oriented Architecture (SOA)
- Introduce and compare the primary IBM ESB products
  - WebSphere Message Broker V6
  - WebSphere Enterprise Service Bus V6
- At the end of the presentation you should be able to:
  - Describe the concepts of an ESB
  - At a high level
    - Understand scenarios in which an ESB can be used
    - Determine which IBM ESB product satisfies the requirements of a particular scenario
- Helpful prior knowledge:
  - A basic understanding of Service Oriented Architecture concepts

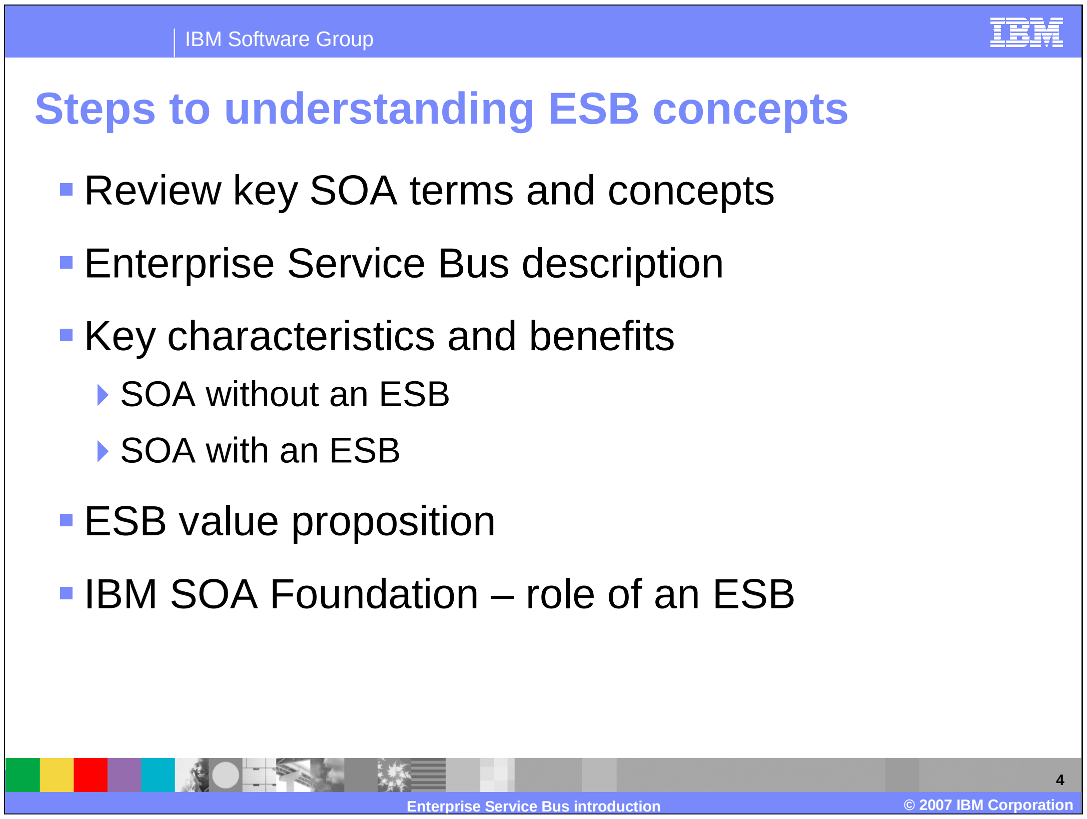
The presentation's goal is to introduce you to the concepts of an Enterprise Service Bus, which is also commonly referred to as an ESB.

You will start out by learning what is the function of an ESB and the role that it plays in a service oriented architecture. IBM has two primary products which are classified as Enterprise Service Bus products, specifically the WebSphere Message Broker version 6 and the WebSphere Enterprise Service Bus version 6. You will see how the functionality delivered by these two products compares. Therefore, you should gain a basic understanding of how to evaluate which product would be a more appropriate fit for a given set of requirements. Although there is a quick introduction to service oriented architecture provided, a basic understanding of service oriented architecture will provide a useful foundation for understanding this presentation.

Section

Enterprise Service Bus concepts

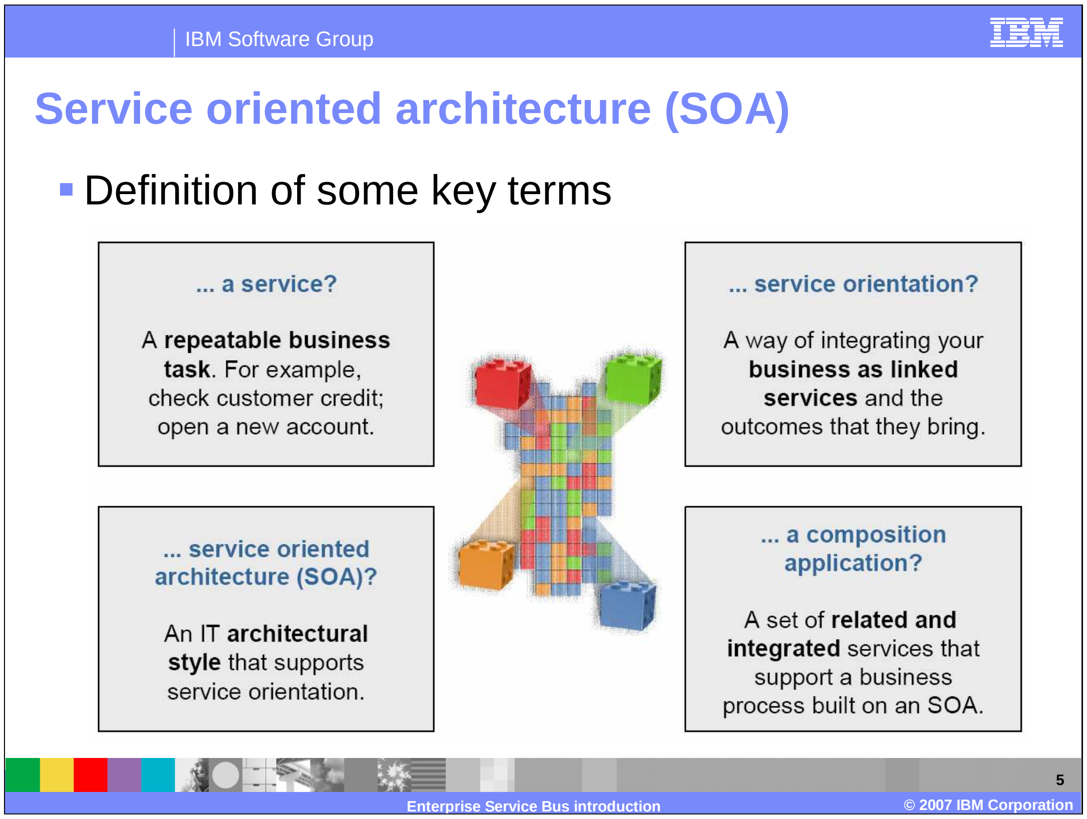
In this section the concepts of an Enterprise Service Bus will be presented.



Steps to understanding ESB concepts

- Review key SOA terms and concepts
- Enterprise Service Bus description
- Key characteristics and benefits
  - SOA without an ESB
  - SOA with an ESB
- ESB value proposition
- IBM SOA Foundation – role of an ESB

You will learn about Enterprise Service Bus concepts by first having a quick review of some key service oriented architecture terms and concepts. A description of what an ESB is and the function that it performs is then presented. With this as a background, you will see some aspects of the benefits of a service oriented architecture without an ESB and then the additional benefits that an ESB brings. This leads to the value proposition for using an enterprise service bus. Finally, the IBM SOA Foundation reference architecture will be shown, highlighting the role played by an enterprise service bus.



Service oriented architecture (SOA)

- Definition of some key terms

... a service?

A **repeatable business task**. For example, check customer credit; open a new account.

... service orientation?

A way of integrating your **business as linked services** and the outcomes that they bring.

... service oriented architecture (SOA)?

An IT **architectural style** that supports service orientation.

... a composition application?

A set of **related and integrated** services that support a business process built on an SOA.

This slide highlights the key terms used to describe a service oriented architecture.

A *service* is representative of a repeatable business task. Services are used to encapsulate the functional units of an application by providing an interface that is well defined and implementation independent. Services can be invoked by other services or client applications.

Service orientation defines a method of integrating business applications and processes as linked services.

Service oriented architecture can be different things to different people, depending on the persons role and whether their view is from a business, architecture, implementation or operational perspective.

From a business perspective, SOA defines a set of business services composed to capture the business design that the enterprise wants to expose internally and externally to its customers and partners.

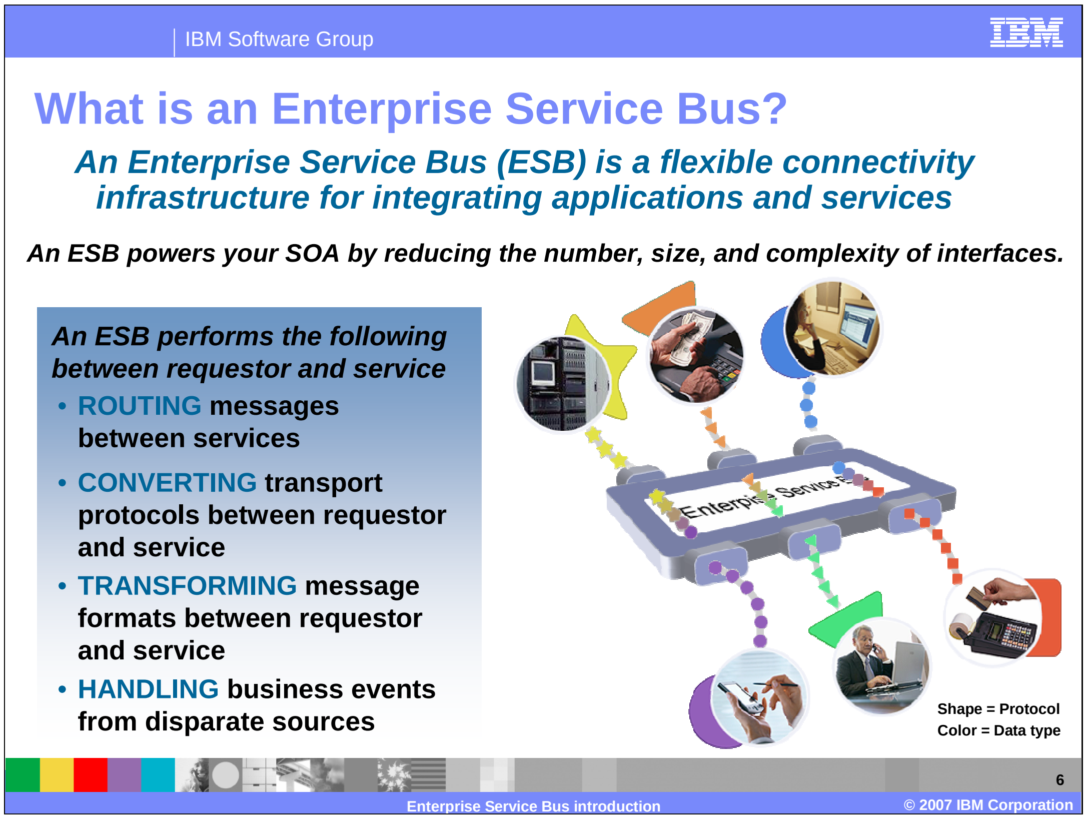
From an architecture perspective, SOA is an architectural style that supports service orientation.

At an implementation level, SOA is fulfilled using a standards based infrastructure, programming model and technologies such as Web services.

From an operational perspective, SOA includes a set of agreements between service consumers and providers that specify the quality of service expected, and enables reporting on the key business and IT metrics.

Summing it up, a service oriented architecture is an architectural style for creating an enterprise IT architecture. It exploits the principals of service orientation to achieve a tighter relationship between the business and the information systems that support the business.

Finally, a *composite application* is a set of related and integrated services that support a business process built on an SOA.


IBM Software Group

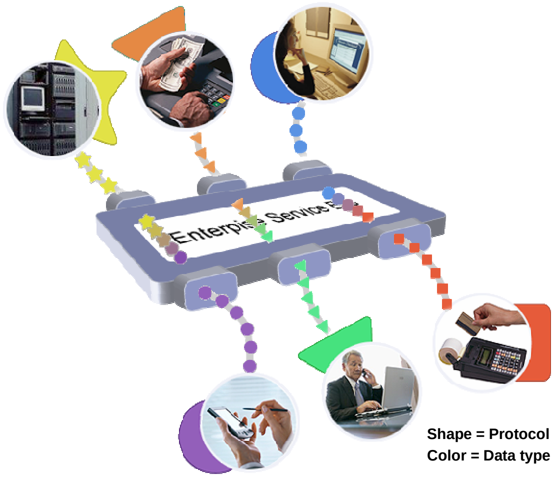
What is an Enterprise Service Bus?

An Enterprise Service Bus (ESB) is a flexible connectivity infrastructure for integrating applications and services

An ESB powers your SOA by reducing the number, size, and complexity of interfaces.

An ESB performs the following between requestor and service

- **ROUTING messages between services**
- **CONVERTING transport protocols between requestor and service**
- **TRANSFORMING message formats between requestor and service**
- **HANDLING business events from disparate sources**



6

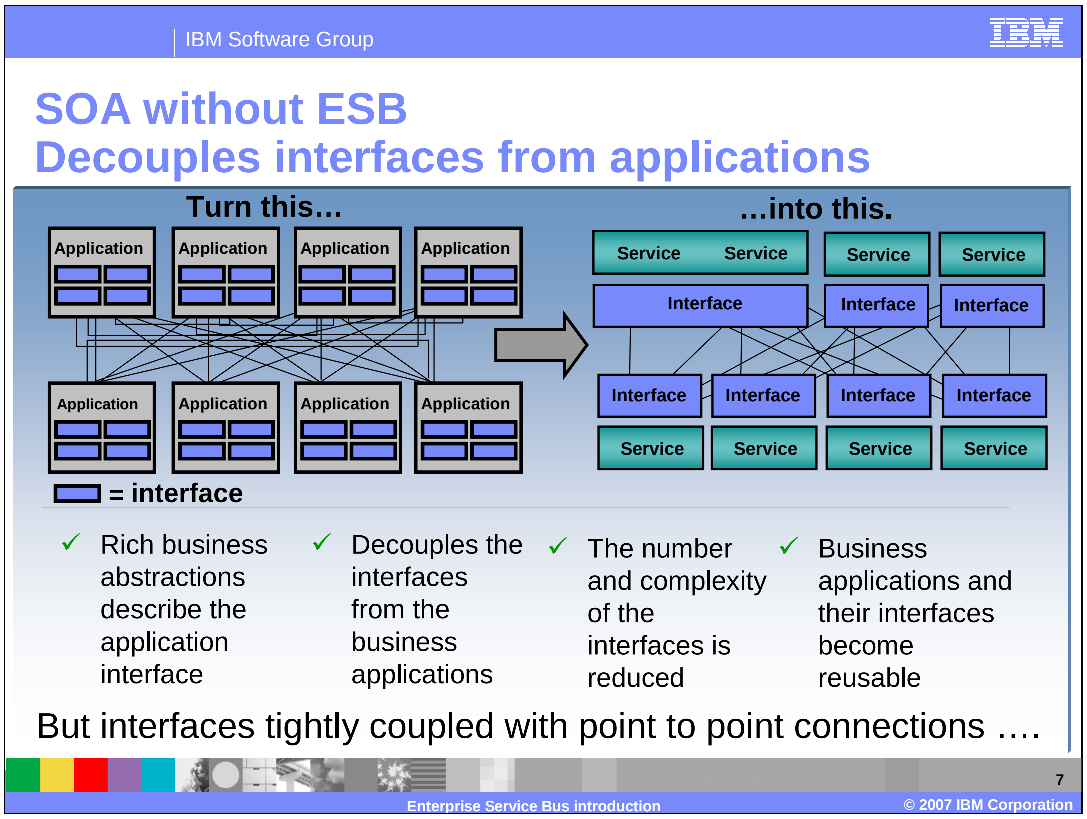
Enterprise Service Bus introduction

© 2007 IBM Corporation

The Enterprise Service Bus provides one of the keys to helping you achieve the goals of a service oriented architecture. It provides a flexible connectivity infrastructure for integrating applications and services, enabling composite applications to be built as a loose coupling of independent services. It is at the heart of your service oriented architecture, reducing the number, size, and complexity of interfaces and connections that must be defined and maintained.

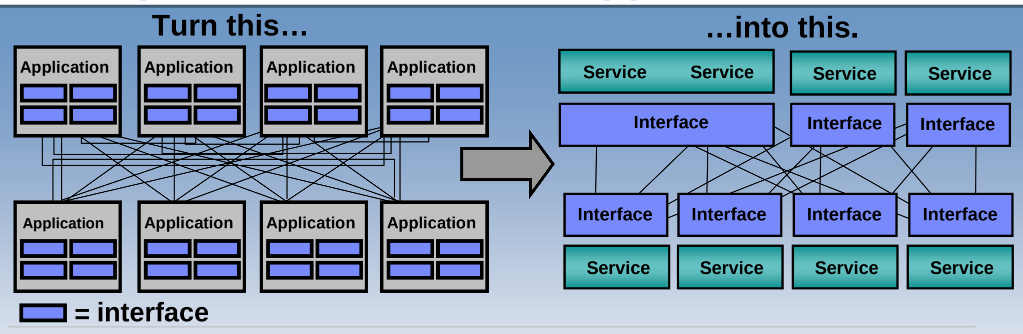
There are four primary functions provided by an enterprise service bus:

- Its first responsibility is the ROUTING of messages. Rather than the service requestor calling directly to the service provider, the requestor sends the request to the ESB, and the ESB then is responsible for making the call on the service provider.
- Secondly, it is responsible for CONVERTING transport protocols. If the service requestor called directly to the service provider, they would need to use the same transport protocol. The ESB enables the service requestor to use one transport protocol while the service provider uses another.
- Thirdly, it is responsible for TRANSFORMING message formats. By eliminating the direct call from the service requestor to the service provider the ESB is capable of modifying the message so that the interfaces used by the requestor and provider do not have to be identical.
- Lastly, the ESB is capable of HANDLING business events from disparate sources. Therefore, the same service provider responsible for performing some particular business function can be indirectly invoked from a variety of application contexts.



IBM Software Group

SOA without ESB
Decouples interfaces from applications



- ✓ Rich business abstractions describe the application interface
- ✓ Decouples the interfaces from the business applications
- ✓ The number and complexity of the interfaces is reduced
- ✓ Business applications and their interfaces become reusable

But interfaces tightly coupled with point to point connections ….

7

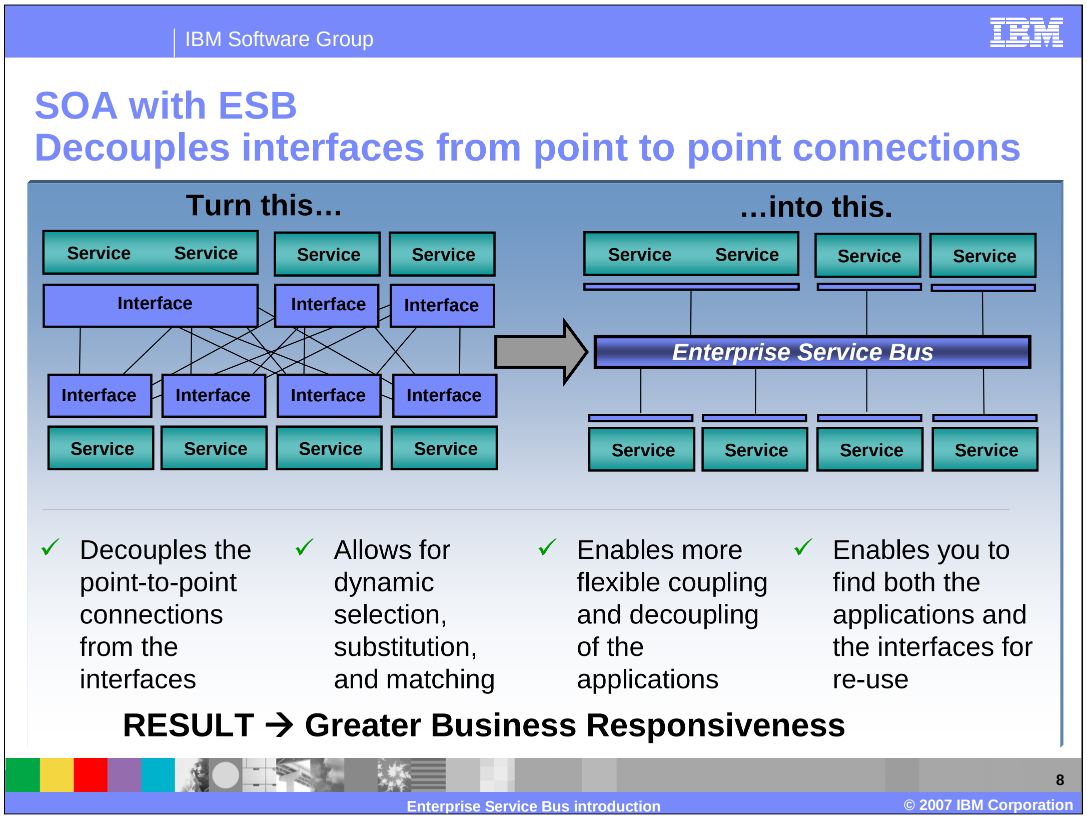
Enterprise Service Bus introduction © 2007 IBM Corporation

In this slide you will see a benefit that a service oriented architecture provides even in the absence of an enterprise service bus.

In a non-SOA environment, the interfaces and connectivity between applications tend to be buried within the applications themselves. As the number of applications increases, the number of interfaces and connections required between them also increases.

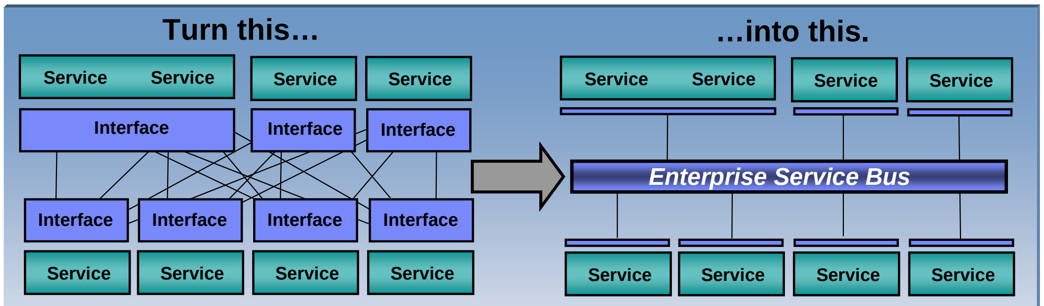
SOA uses a programming model that allows a rich abstraction of the business application, which can be described by an interface. The interface is not contained within the application. Therefore, the application logic can now be packaged as a service supporting the externally defined interface. This decoupling of the interface from the application logic tends to reduce the number and complexity of the interfaces needed to provide the overall business function in what is now a composite application. The services that are described by these interfaces generally become reusable for the creation of additional composite applications which provide additional business function.

However, there is still an issue. The interfaces used between services make use of point to point connections. There can still be a large number of these that must be defined and maintained. As additional composite applications are developed, the number of connections grow even if the number of interfaces does not grow.



IBM Software Group

SOA with ESB
Decouples interfaces from point to point connections



- ✓ Decouples the point-to-point connections from the interfaces
- ✓ Allows for dynamic selection, substitution, and matching
- ✓ Enables more flexible coupling and decoupling of the applications
- ✓ Enables you to find both the applications and the interfaces for re-use

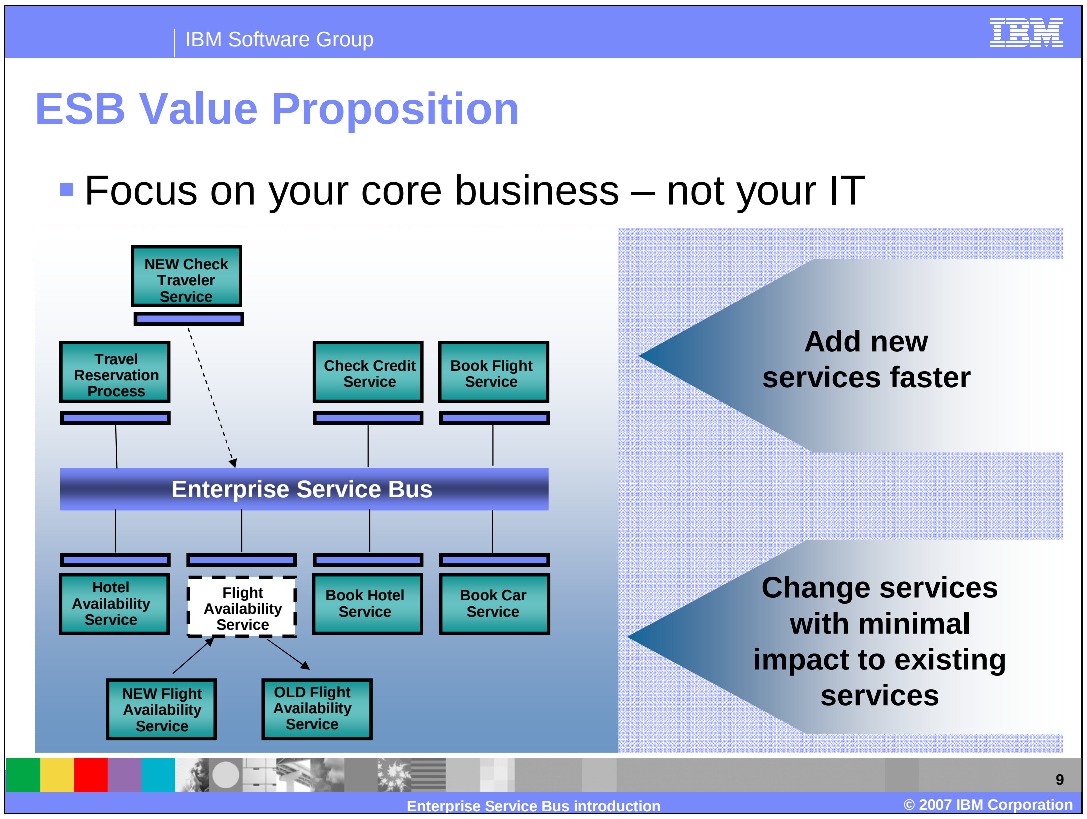
RESULT → Greater Business Responsiveness

8

Enterprise Service Bus introduction © 2007 IBM Corporation

Introducing an enterprise service bus into the service oriented architecture eliminates the need for point to point connections for connecting the interfaces between service requestors and providers. The connections are now all between the interfaces for the services and the enterprise service bus, thus greatly reducing the number of connections required. This enables capabilities such as dynamic selection of services and the increased flexibility in changing individual services without affecting multiple users of the service. It also becomes easier to find an existing service for reuse in a new composite application.

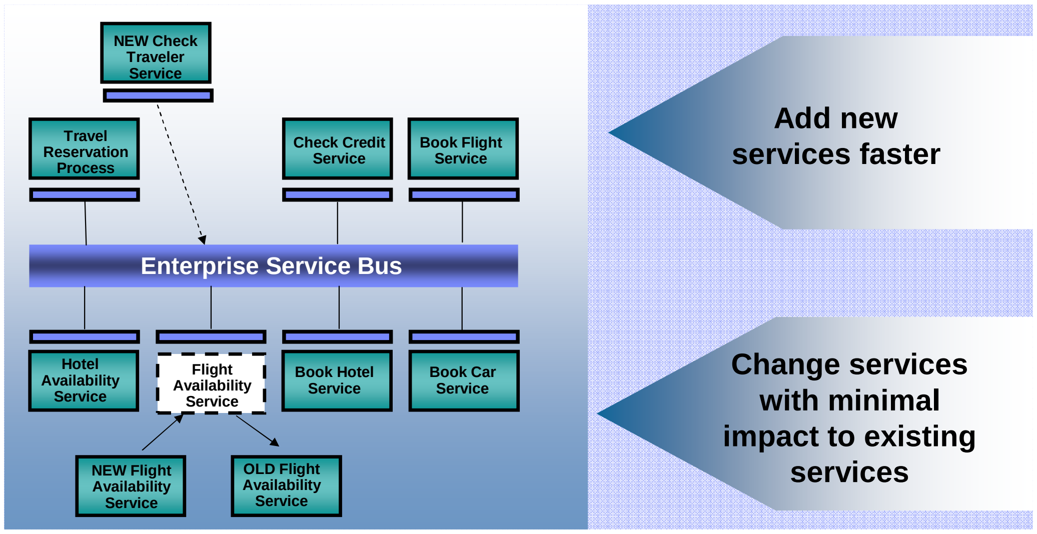
All of this results in achieving greater flexibility and responsiveness to the changing requirements of the business.



IBM Software Group

ESB Value Proposition

- Focus on your core business – not your IT


NEW Check Traveler Service
Travel Reservation Process
Check Credit Service
Book Flight Service
Enterprise Service Bus
Hotel Availability Service
Flight Availability Service
Book Hotel Service
Book Car Service
NEW Flight Availability Service
OLD Flight Availability Service
Add new services faster
Change services with minimal impact to existing services

9

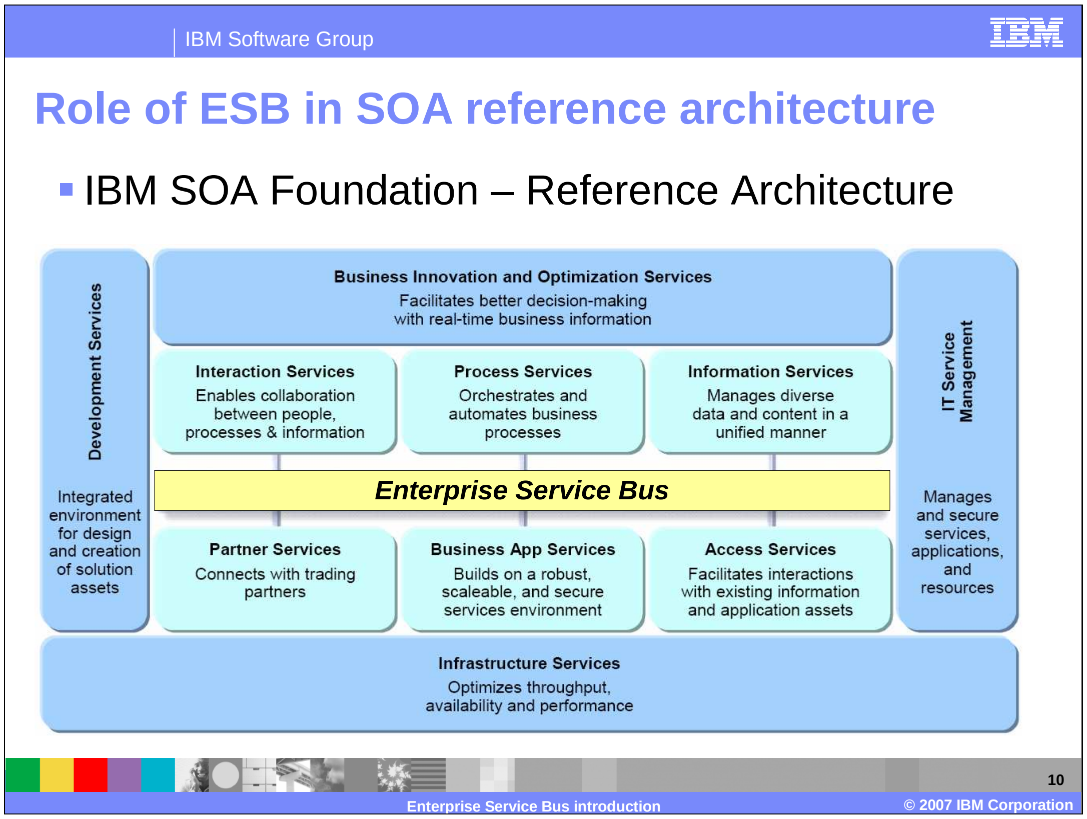
Enterprise Service Bus introduction © 2007 IBM Corporation

As described in the previous slide, an enterprise service bus leads to achieving greater flexibility and responsiveness to changing requirements of the business. This is the real value proposition that an ESB brings. There are many ways that this greater flexibility is manifested, a couple of which are shown on this slide.

New services can be added faster because they only have to be directly known to the ESB. Depending upon what the function of the service is determines if service requestors need to know about this new service or not. If it is a completely new capability, the potential requestors would need to know that the capabilities of the new service now exists. However, the service might be similar to existing services, possibly with a slightly different interface or improved implementation approach. In this case the ESB can take care of routing requests to this new service without service requestors even having to be aware that it has been added to the bus.

Another flexibility is manifested in the minimal impact when an existing service is modified or replaced. For example, a service may be replaced with a new and improved version that contains a new interface to support the additional function. Without an ESB, these changes would ripple back to every user of the service. However, the ESB can handle whatever transformations are needed to meet the new interface and current users of the service will not need to know that a change has happened.

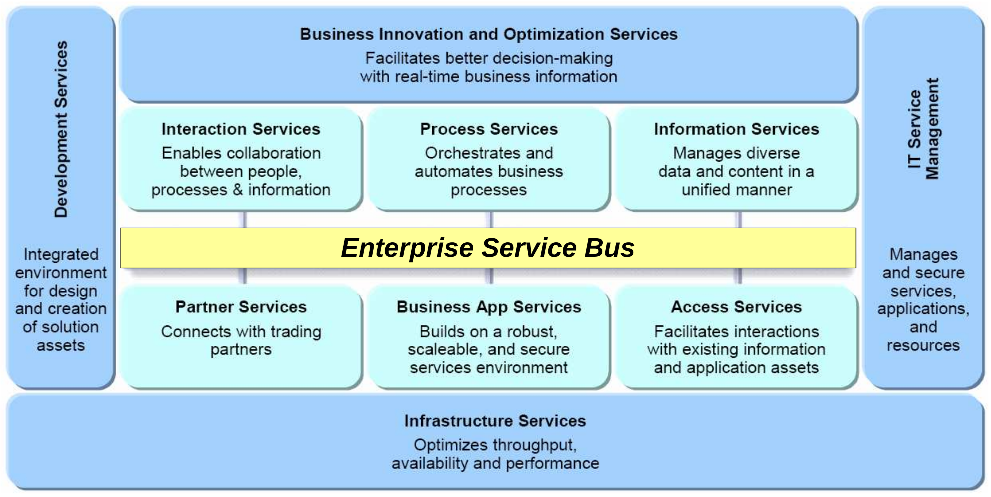
As you can see, the flexibility to make these kind of changes provides great business value by enabling new business requirements to be addressed quickly without adversely affecting the rest of the system.



IBM Software Group

Role of ESB in SOA reference architecture

- IBM SOA Foundation – Reference Architecture


Business Innovation and Optimization Services
Facilitates better decision-making with real-time business information
Development Services
Integrated environment for design and creation of solution assets
Interaction Services
Enables collaboration between people, processes & information
Process Services
Orchestrates and automates business processes
Information Services
Manages diverse data and content in a unified manner
Enterprise Service Bus
Partner Services
Connects with trading partners
Business App Services
Builds on a robust, scaleable, and secure services environment
Access Services
Facilitates interactions with existing information and application assets
IT Service Management
Manages and secure services, applications, and resources
Infrastructure Services
Optimizes throughput, availability and performance

10

Enterprise Service Bus introduction

© 2007 IBM Corporation

This diagram depicts the SOA reference architecture which provides a logical architectural model for SOA as defined by the IBM SOA Foundation. You can see that at the very core of the architecture is the Enterprise Service Bus. Around the ESB in the center of the diagram are the core services used by applications at runtime to provide the application function. Around the outer edge of the diagram are those services that support the core services.

As has been discussed in this presentation, the ESB provides the capabilities needed to allow service consumers and providers to be loosely coupled. This avoids the use of point-to-point connections which can quickly lead to a high level of complexity and inflexibility within the overall SOA environment. The placement of the ESB at the core of the SOA reference architecture suggests that an ESB should be used, at a minimum, for requests made between the core components. Additionally, it makes sense that many requests made within a core component also be routed through the ESB.

As a quick summary, the core components are described as follows:

Interaction services provide the required capabilities for users to interact with the system.

Process services provide the capabilities for managing the flow and interactions of multiple services in ways that implement business processes.

Business application services provide the core business logic and are the basic building blocks of the business design. These services are not further decomposable within the business model.

Information services federate, replicate and transform disparate data sources so that they can be handled in a unified manner.

Access services provide bridging capabilities that enable existing applications which are not SOA based to be delivered as services within a service oriented architecture.

Partner services provide the document, protocol and partner management capabilities required to interact with outside business partners.

The supporting services around the outside of the diagram can be described as follows:

Business innovation and optimization services represent the tools and metadata for encoding the business design, policies and objectives.

Development services represent the entire suite of tools, methodologies and repositories needed to construct an SOA application.

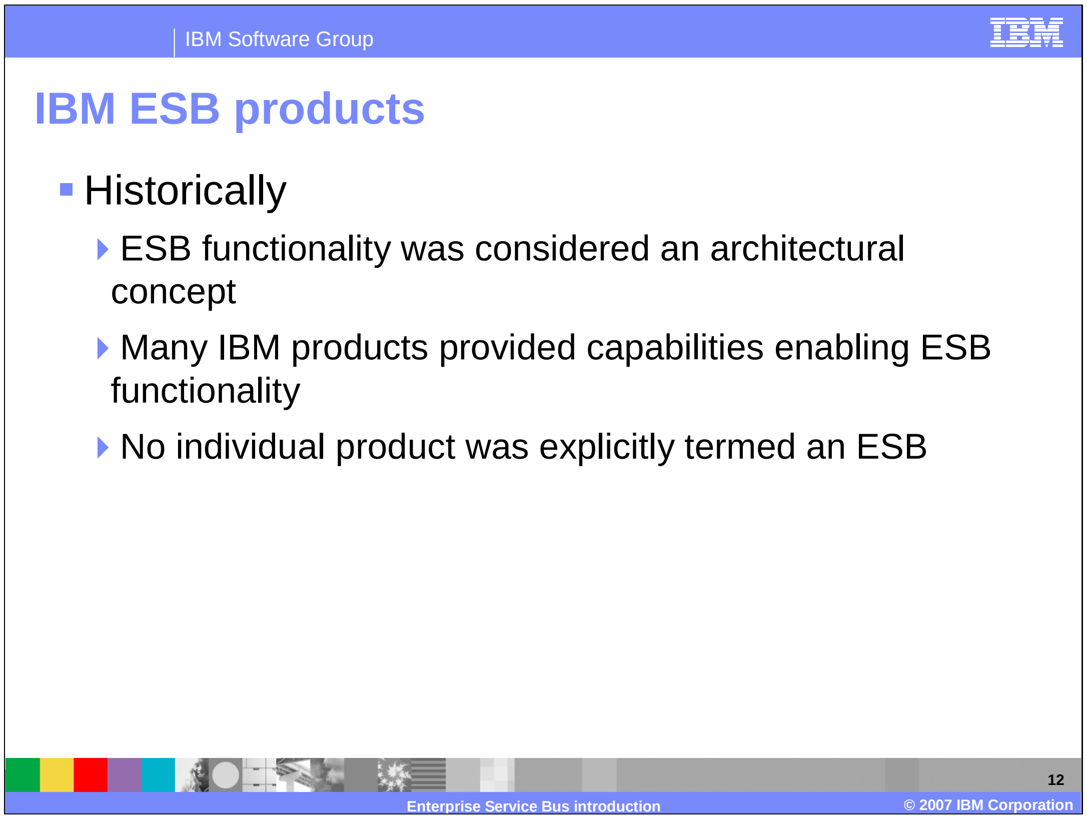
IT service management represents the tools needed to manage your service flows, system utilization, enforcement of policies and other runtime management of the system.

Infrastructure services represent the core of the runtime environment which hosts the SOA applications, providing capabilities for high availability, performance and management of the environment.

Section

IBM ESB products

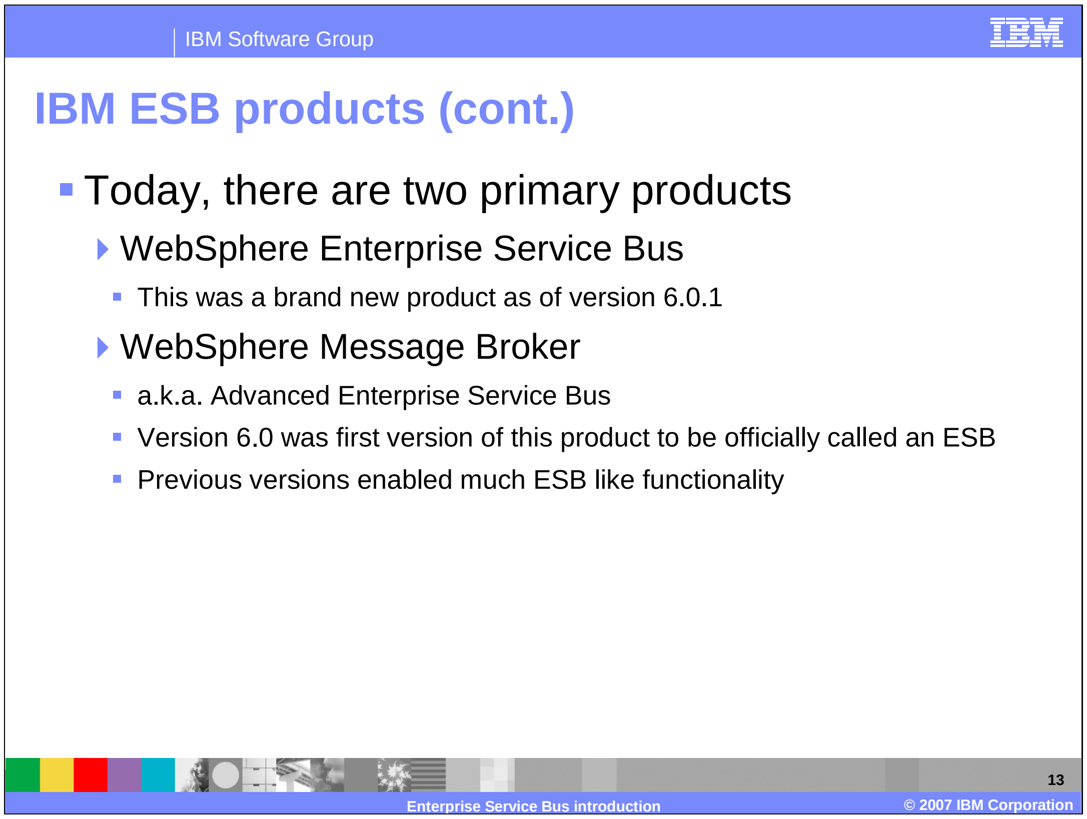
In the following section you will learn about the IBM ESB products.



IBM ESB products

- Historically
  - ESB functionality was considered an architectural concept
  - Many IBM products provided capabilities enabling ESB functionality
  - No individual product was explicitly termed an ESB

Historically, the concept of an enterprise service bus was considered mostly an architectural concept. There have been many IBM products that support functionality needed to build an enterprise service bus but these products themselves were not called ESBs. For example, there is a Redbook that describes how to build an ESB using WebSphere Application Server version 6 in conjunction with WebSphere Message Broker version 5. Neither of these products is characterized as an ESB, but the architectural concept of an ESB could be implemented using the pattern described in the Redbook.



IBM Software Group

IBM ESB products (cont.)

- Today, there are two primary products
  - WebSphere Enterprise Service Bus
    - This was a brand new product as of version 6.0.1
  - WebSphere Message Broker
    - a.k.a. Advanced Enterprise Service Bus
    - Version 6.0 was first version of this product to be officially called an ESB
    - Previous versions enabled much ESB like functionality

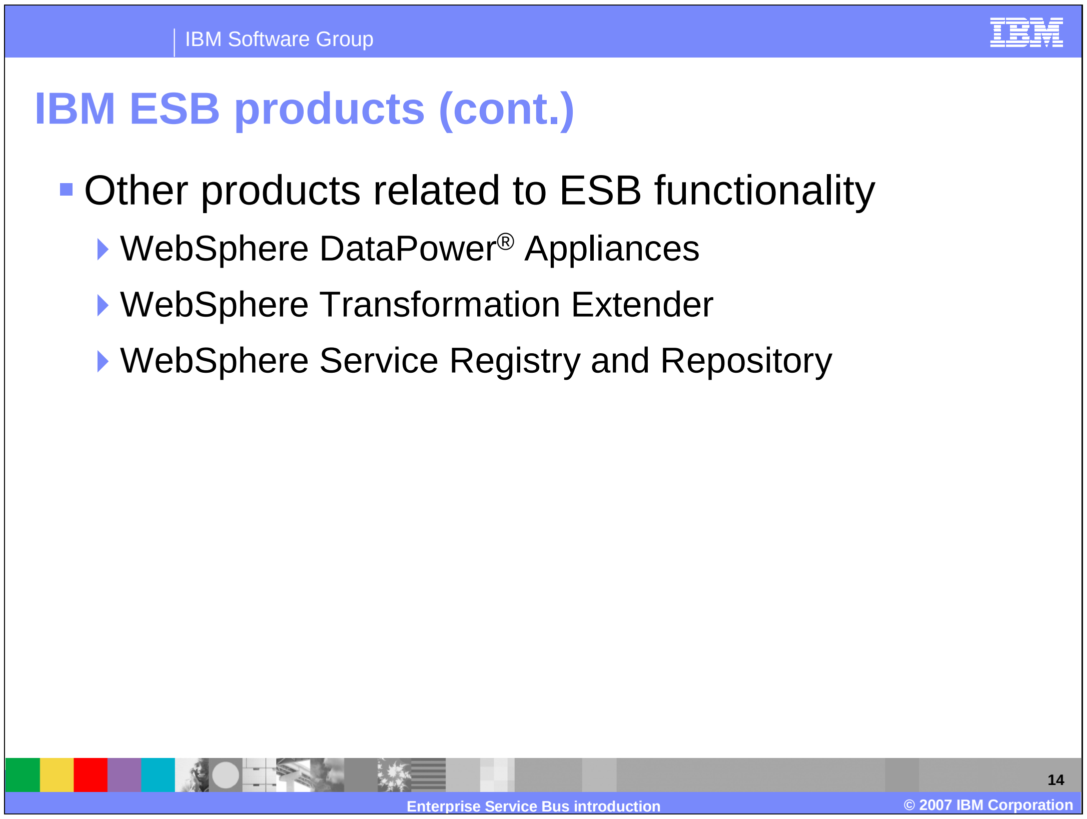
13

Enterprise Service Bus introduction © 2007 IBM Corporation

Today there are two IBM products which are characterized as an ESB and support building an environment that aligns with the architectural concept of an ESB.

The first product is WebSphere Enterprise Service Bus which was released as a new product when it was at version 6.0.1. Although seemingly an unusual version designation for a new product, it was done to synchronize it with the version numbering for WebSphere Process Server which is functionally a superset of WebSphere Enterprise Service Bus.

The second product is WebSphere Message Broker version 6 which is sometimes called the Advanced Enterprise Service Bus. This is a follow on release to WebSphere Message Broker version 5, but version 6 of the product is the first version to be explicitly characterized as an ESB.



IBM ESB products (cont.)

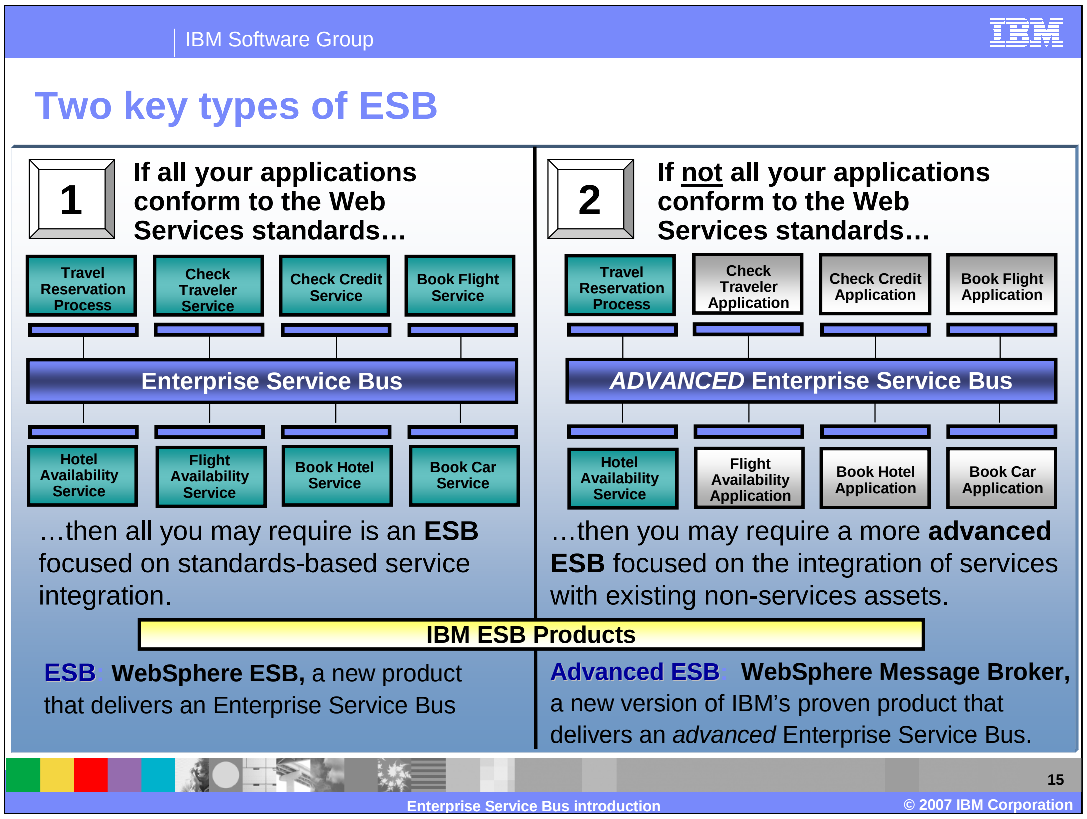
- Other products related to ESB functionality
  - WebSphere DataPower® Appliances
  - WebSphere Transformation Extender
  - WebSphere Service Registry and Repository

There are other products that provide some functionality that is key to enabling an ESB and therefore sometimes get designated as ESB or ESB related product. A few of the key products that fit into this classification are listed here.

Certain models of the WebSphere DataPower Appliance provide routing and transformation capabilities very similar to an ESB and are sometime characterized as an ESB.

The WebSphere Transformation Extender provides data transformation capabilities and can be integrated with ESB products to enhance their transformation abilities.

The WebSphere Service Registry and Repository is an essential piece if you are concerned with providing a high degree of flexibility within your ESB for dynamic selection of service providers. It also is a key piece in enabling governance within your SOA environment.



Two key types of ESB

1
If all your applications conform to the Web Services standards…
Travel Reservation Process
Check Traveler Service
Check Credit Service
Book Flight Service
Enterprise Service Bus
Hotel Availability Service
Flight Availability Service
Book Hotel Service
Book Car Service
2
If not all your applications conform to the Web Services standards…
Travel Reservation Process
Check Traveler Application
Check Credit Application
Book Flight Application
ADVANCED Enterprise Service Bus
Hotel Availability Service
Flight Availability Application
Book Hotel Application
Book Car Application

…then all you may require is an **ESB** focused on standards-based service integration.

…then you may require a more **advanced ESB** focused on the integration of services with existing non-services assets.

IBM ESB Products

ESB **WebSphere ESB,** a new product that delivers an Enterprise Service Bus

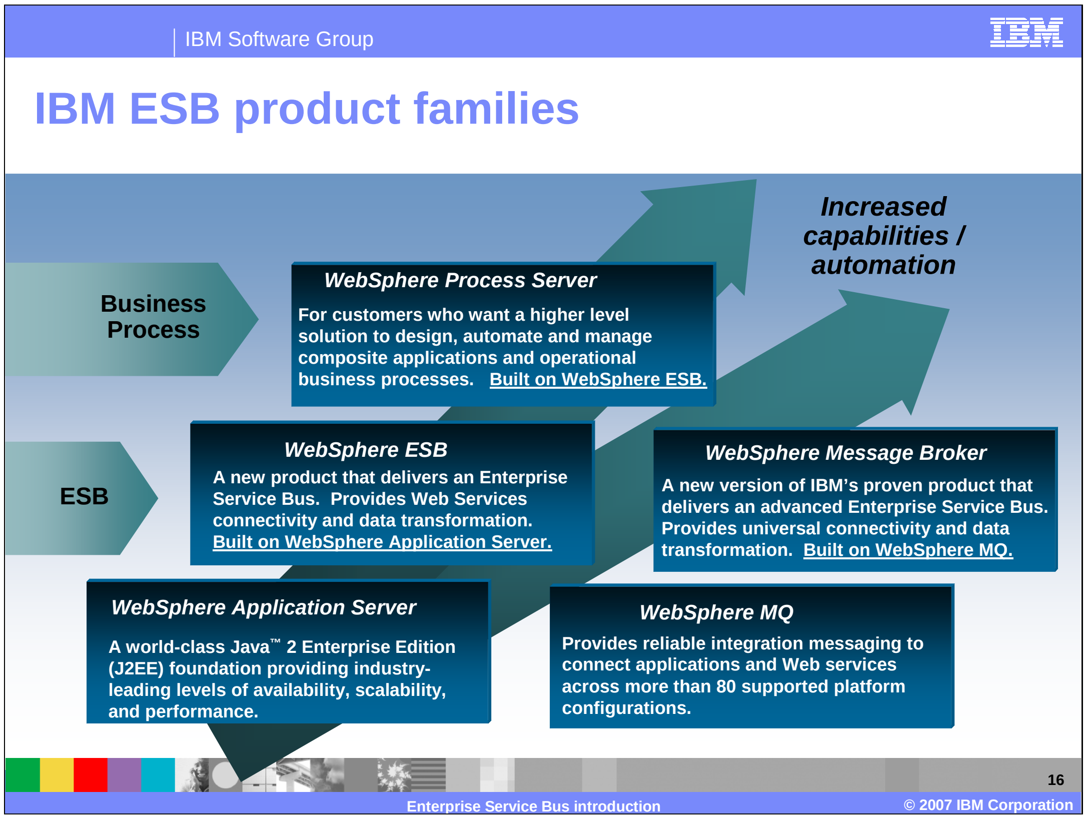
Advanced ESB **WebSphere Message Broker,** a new version of IBM's proven product that delivers an *advanced* Enterprise Service Bus.

The next several slides will be used to compare and contrast the two IBM ESB products, WebSphere Enterprise Service Bus and WebSphere Message Broker.

If all your applications conform to the Web Services standards, you may only need an ESB focused on the integration of these standards-based interfaces. However, if not all your applications conform to the SOA standards, you may need something with a wider range of capabilities that can mediate between the SOA standards and other protocols and data formats.

WebSphere Enterprise Service Bus was introduced as a product which primarily addresses the scenario where the focus is the integration of standards-based interfaces.

WebSphere Message Broker version 6 was introduced as a new version of IBM's proven Message Broker product but with additional features to deliver an advanced enterprise service bus functionality. WebSphere Message Broker is characterized as an Advanced ESB because it addresses a high percentage of the legacy protocols and data formats that are prevalent in many customer environments today.



IBM Software Group

IBM ESB product families

WebSphere Process Server

For customers who want a higher level solution to design, automate and manage composite applications and operational business processes. Built on WebSphere ESB.

WebSphere ESB

A new product that delivers an Enterprise Service Bus. Provides Web Services connectivity and data transformation. Built on WebSphere Application Server.

WebSphere Message Broker

A new version of IBM's proven product that delivers an advanced Enterprise Service Bus. Provides universal connectivity and data transformation. Built on WebSphere MQ.

WebSphere Application Server

A world-class Java™ 2 Enterprise Edition (J2EE) foundation providing industry-leading levels of availability, scalability, and performance.

WebSphere MQ

Provides reliable integration messaging to connect applications and Web services across more than 80 supported platform configurations.

16

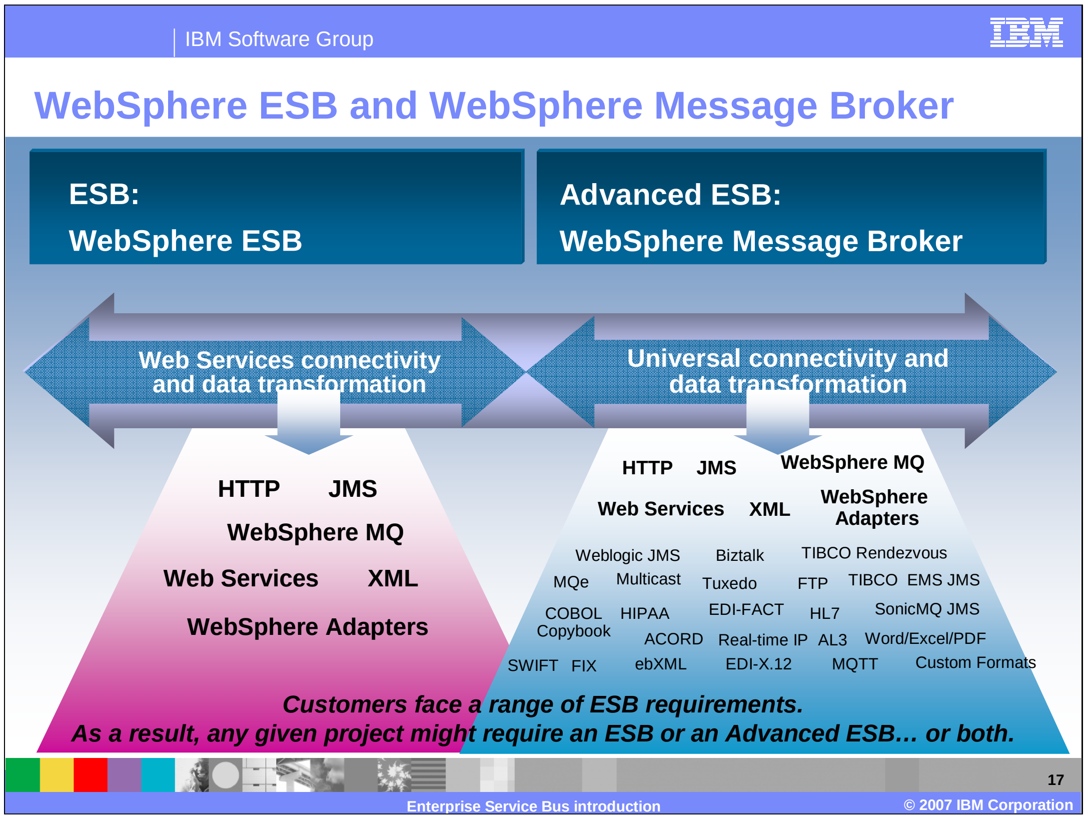
Enterprise Service Bus introduction © 2007 IBM Corporation

This slide introduces how the IBM ESB products fit into the product stacks of different IBM offerings

As you can see on the left, WebSphere Enterprise Service Bus is only one piece of a complete solution that IBM offers for transport, mediation, process, and transformation needs. WebSphere Enterprise Service Bus is built on top of WebSphere Application Server. Therefore, the foundation for WebSphere Enterprise Service Bus is a world class J2EE application serving environment. It provides high levels of qualities of service for security, availability, scalability and performance. Then, built on top of WebSphere Enterprise Service Bus is the WebSphere Process Server, which adds a variety of business process integration capabilities and is the runtime environment for IBM's Business Process Management solutions.

On the right you can see that WebSphere Message Broker is built on top of WebSphere MQ, which provides a proven and highly reliable messaging infrastructure as the foundation. WebSphere MQ provides high levels of qualities of service such as security, availability, scalability and performance and is supported on a wide range of platform configurations.

Finally, although built on different stacks, the IBM ESB products interoperate to provide an even more comprehensive solution.



WebSphere ESB and WebSphere Message Broker

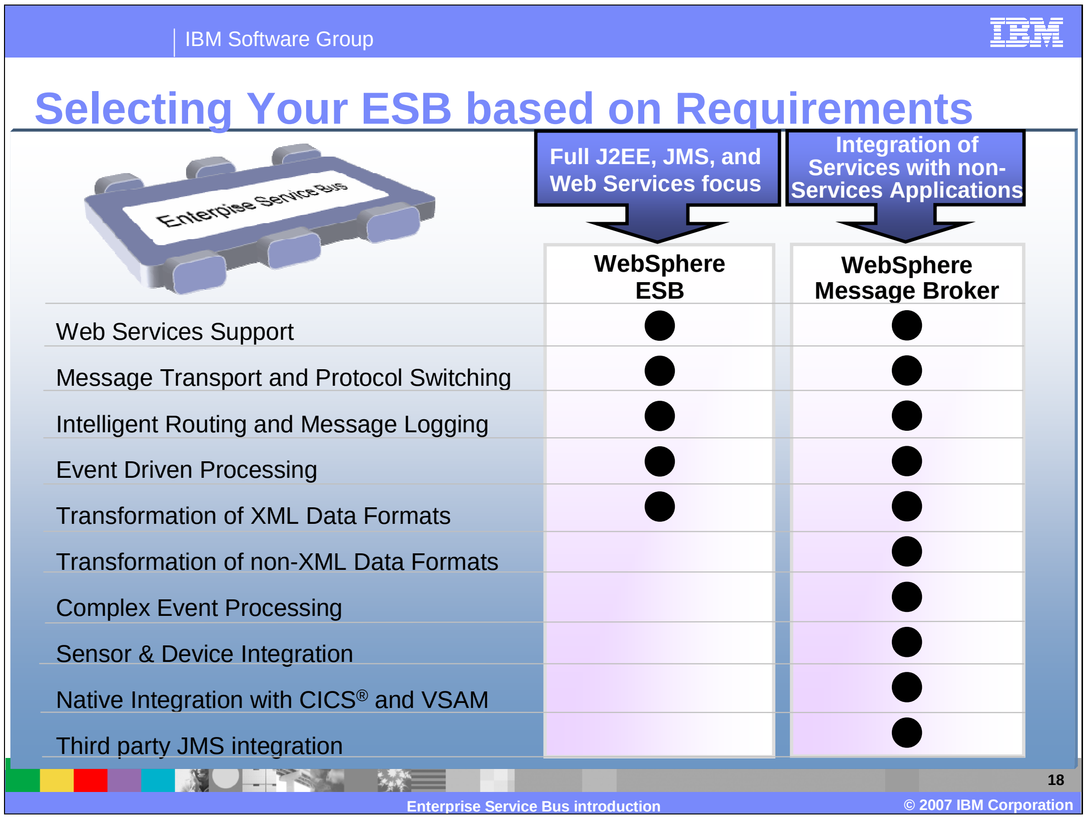
Customers face a range of ESB requirements.
As a result, any given project might require an ESB or an Advanced ESB… or both.

Selecting which of the ESB products is most appropriate to power your SOA depends upon your requirements.

WebSphere Enterprise Service Bus provides ESB functionality through the use of Service Component Architecture for connectivity and Service Data Objects for data representation. Therefore, any protocol or data format that can be used with SCA and SDO can be integrated using this product. This includes Web Services with SOAP over HTTP or SOAP over JMS, WebSphere JMS messaging, native MQ messaging and WebSphere Adapters. Data formats such as XML are handled easily. Other data formats can be handled, but may require user provided code for transformation logic to and from SDO representation.

WebSphere Message Broker offers a much wider range of integration options and is therefore characterized as an Advanced ESB. Universal connectivity is provided through the support of many additional connectivity options, such as FTP and other software vendor JMS providers. Any to any data format transformation provides built in capabilities for a wide variety of the standardized data formats in use today, such as Electronic Data Interchange, commonly known as EDI.

Customers today face a wide range of ESB requirements from the simple to the complex. As a result, any given project might require an ESB, or an Advanced ESB, or both.



IBM Software Group

Selecting Your ESB based on Requirements

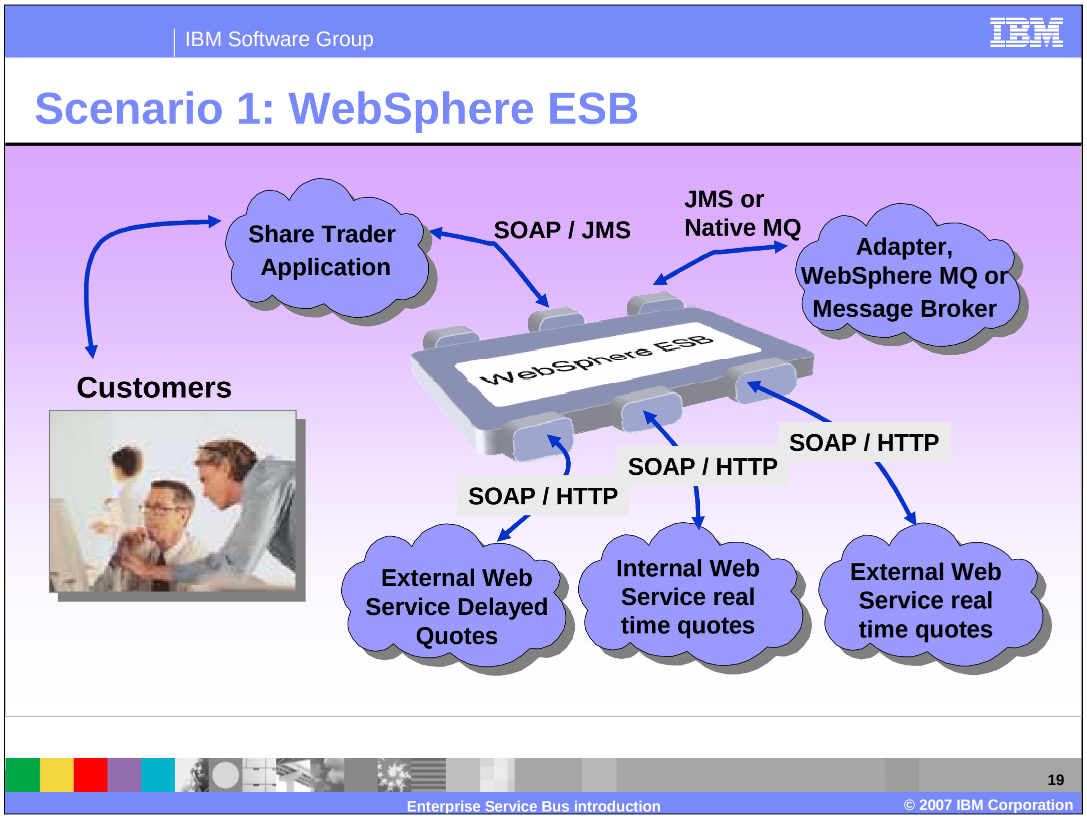
	Full J2EE, JMS, and Web Services focus	Integration of Services with non-Services Applications
	WebSphere ESB	**WebSphere Message Broker**
Web Services Support	●	●
Message Transport and Protocol Switching	●	●
Intelligent Routing and Message Logging	●	●
Event Driven Processing	●	●
Transformation of XML Data Formats	●	●
Transformation of non-XML Data Formats		●
Complex Event Processing		●
Sensor & Device Integration		●
Native Integration with CICS® and VSAM		●
Third party JMS integration		●

18

Enterprise Service Bus introduction

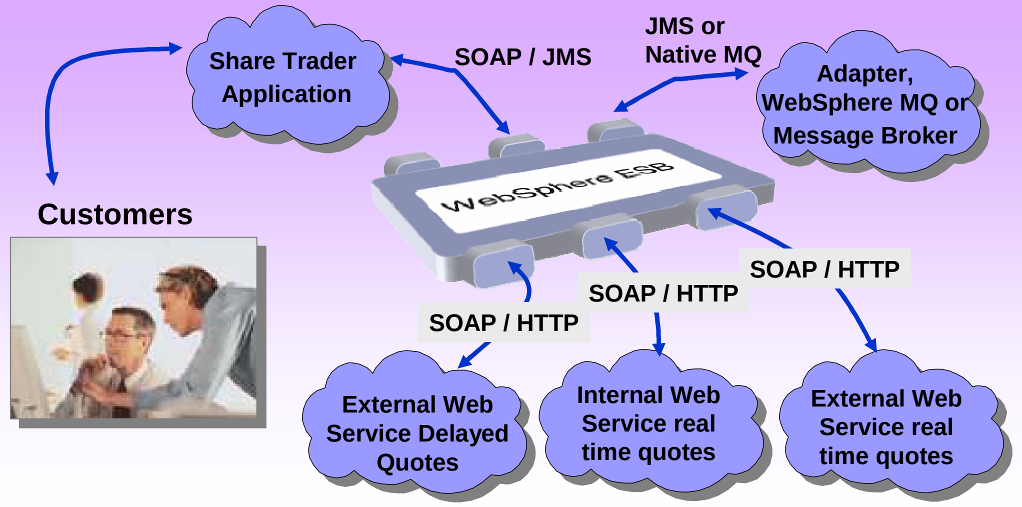
© 2007 IBM Corporation

As already mentioned, selecting the right ESB product depends upon your requirements. This table summarizes the key capabilities of the WebSphere Enterprise Service Bus and the WebSphere Message Broker products.
Both products work in an event driven manner and are capable of supporting Web Services, performing message transport and protocol switching, intelligent routing of messages and message logging. Transformation of XML data formats is also built into both products. If your requirements are within the scope of these capabilities, either of the products can satisfy your needs. Because the development tools and underlying runtime environments are different for both of these products, you may need to factor that in when determining which product would be appropriate for you.
In addition, the WebSphere Message Broker provides some additional capabilities. The first is the ability to natively transform non-XML formats without the use of adapters or user written data transformations. You are able to go directly from one format to another which is very useful if you have data in your enterprise today that does not conform to the XML standard.
Another additional capability is complex event processing. Complex events are events which occur only as a result of some combination of multiple different events occurring according to some pattern. By identifying complex business events your business can act and react much earlier that you might otherwise be able to if additional analysis was required to recognize the occurrence of the complex event.
Currently sensors and mobile devices do not support the SOA standards, but they can be integrated into an SOA architecture using the Message Broker.
The Broker also provides native integration of CICS and VSAM.
Finally, many companies make use of JMS services from different vendors, such as Sonic MQ and Tibco. JMS is an open standard API, but implementations from different vendors do not necessarily talk to one another. WebSphere Message Broker can handle inputs from virtually all these vendor provided JMS systems and therefore can work as an integration hub for various JMS providers. Unlike other vendors, whose JMS implementations can works as transactional clients but not transaction managers, WebSphere Message Broker can act as a transaction manager. This means that it is the only product on the market that can manage a transaction involving multiple messaging products.



IBM Software Group

Scenario 1: WebSphere ESB


Share Trader Application
SOAP / JMS
JMS or Native MQ
Adapter, WebSphere MQ or Message Broker
Customers
WebSphere ESB
SOAP / HTTP
SOAP / HTTP
SOAP / HTTP
External Web Service Delayed Quotes
Internal Web Service real time quotes
External Web Service real time quotes

19

Enterprise Service Bus introduction © 2007 IBM Corporation

In the next three slides you will be introduced to scenarios which make use of the IBM ESB products.

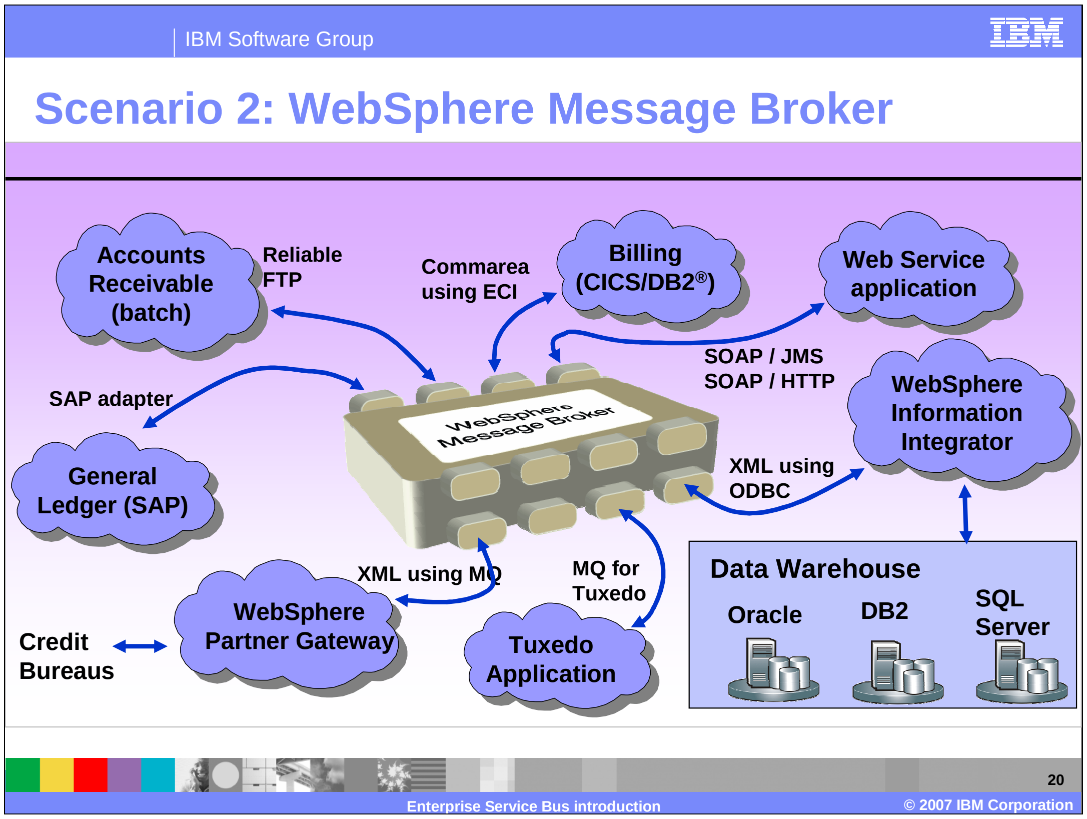
Scenario 1 demonstrates the capabilities of WebSphere Enterprise Service Bus when used alone. It assumes that almost all of the interfaces are web services using either SOAP over JMS or SOAP over HTTP. It can also talk to WebSphere MQ, WebSphere Message Broker or a WebSphere Adapter.

An example of this solution is a Share Trader application utilizing WebSphere Enterprise Service Bus to implement an ESB which performs the following functions:

- Incoming requests for stock quotes are received from a web based share trader application using SOAP over JMS
- The appropriate service to call is determined based on the status of the customer
- Knowing the target service provider, the request is transformed to the appropriate format for the target destination using XSL transformation
- The call to and response from the selected service is handled using a SOAP over HTTP protocol
- The response is transformed to the format expected by the original requestor, again using XSL transformation
- The response is returned to the share trader application using SOAP over JMS

There are another couple of things that can be pointed out with this scenario:

- Stock quote requests could be received from different applications using messaging protocols or through WebSphere adapters, while still making use of the same set of service providers
- Changes in service provider endpoints can be made without impact to the requestors. For example, moving an internally provided stock quote service to a more dependable external provider.


IBM Software Group

Scenario 2: WebSphere Message Broker

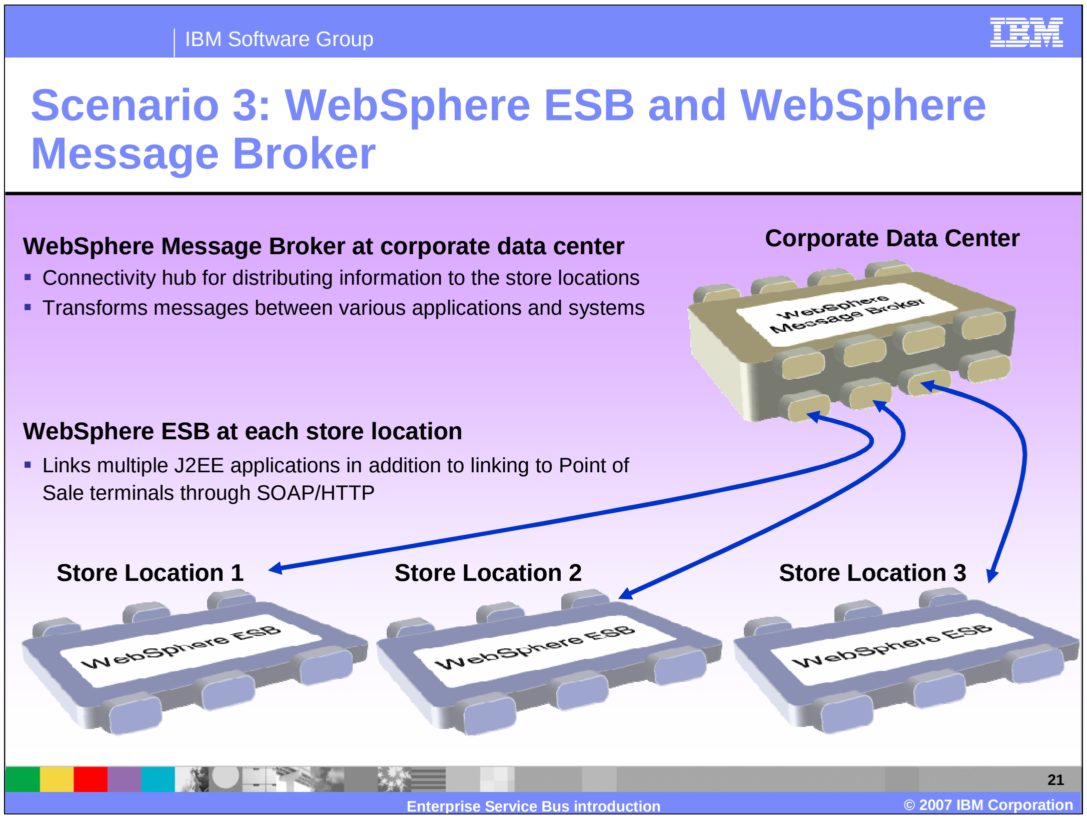
20

Enterprise Service Bus introduction

© 2007 IBM Corporation

Scenario 2 is focused around WebSphere Message Broker and assumes you have to connect to a wide range of business applications which are not packaged as services. These could include applications such as FTP, SAP, Business to Business connections, CICS, BEA Tuxedo, ODBC links to products such as WebSphere Information Integrator, and so on. It also allows you to do Web Service calls using either SOAP over JMS or SOAP over HTTP.

An example of this solution is an enterprise implementing WebSphere Message Broker to establish an extensible enterprise integration backbone to integrate existing applications. The possible interactions in such an environment are too numerous to list. There are several advantages to this environment, such as a reduction in risk because applications are insulated from the complexity of a highly heterogeneous environment. Enterprise-wide consistency is increased by integrating several back end applications using a variety of formats and connection methods. Also, by using the complex event support you can correlate individual messages to detect situations mandated by the business, such as fraud detection.

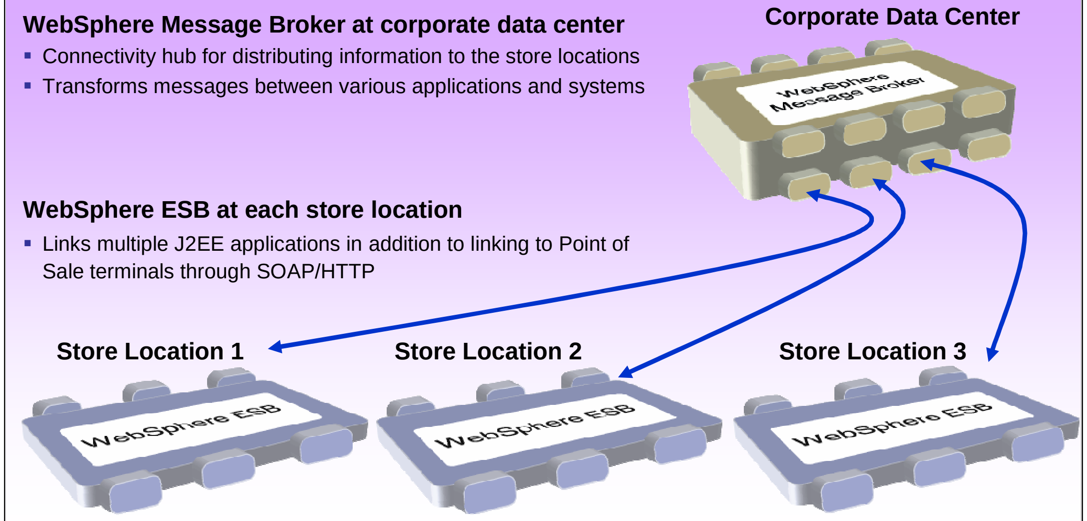


IBM Software Group

Scenario 3: WebSphere ESB and WebSphere Message Broker

WebSphere Message Broker at corporate data center
- Connectivity hub for distributing information to the store locations
- Transforms messages between various applications and systems

WebSphere ESB at each store location
- Links multiple J2EE applications in addition to linking to Point of Sale terminals through SOAP/HTTP



21

Enterprise Service Bus introduction

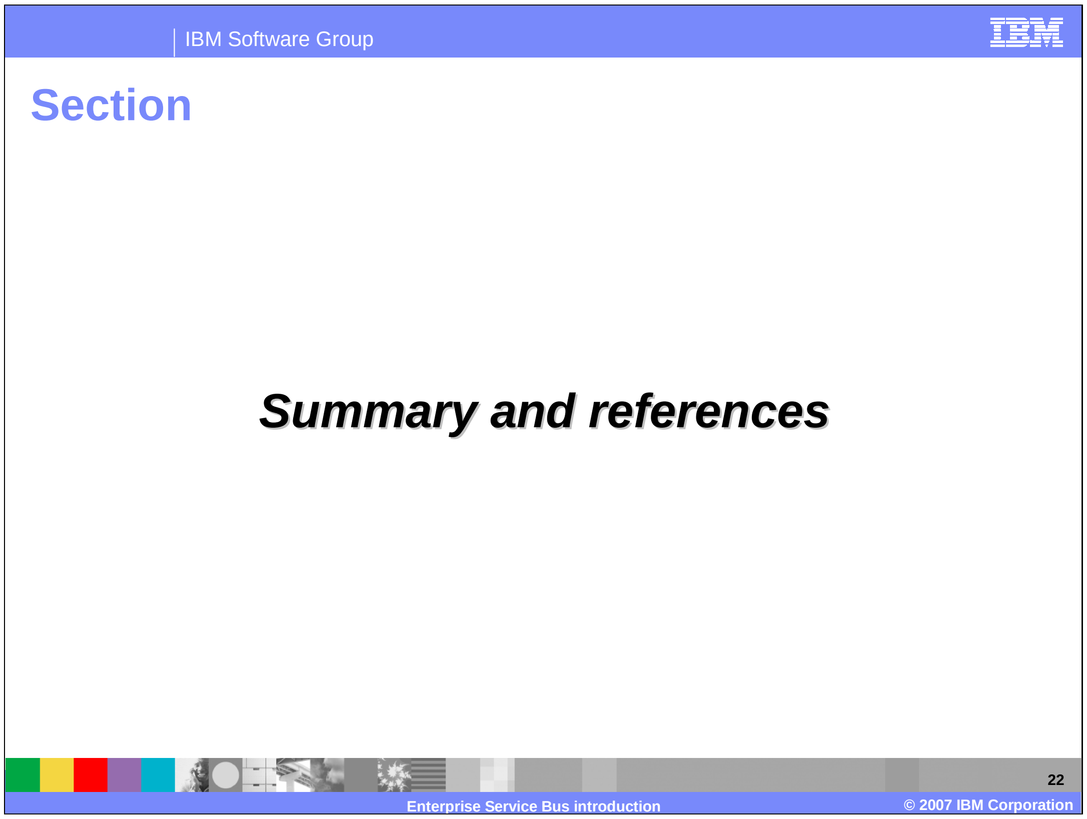
© 2007 IBM Corporation

Scenario 3 uses both WebSphere Enterprise Service Bus and WebSphere Message Broker.

In this scenario, a retail sales corporation utilizes WebSphere Enterprise Service Bus and WebSphere Message Broker together to implement an ESB that spans the entire business from the corporate data center to the individual retail outlets.

At each retail store location, WebSphere Enterprise Service Bus is used to link Point of Sale terminals through SOAP over HTTP, integrate with several local J2EE applications and to communicate with the corporate data center.

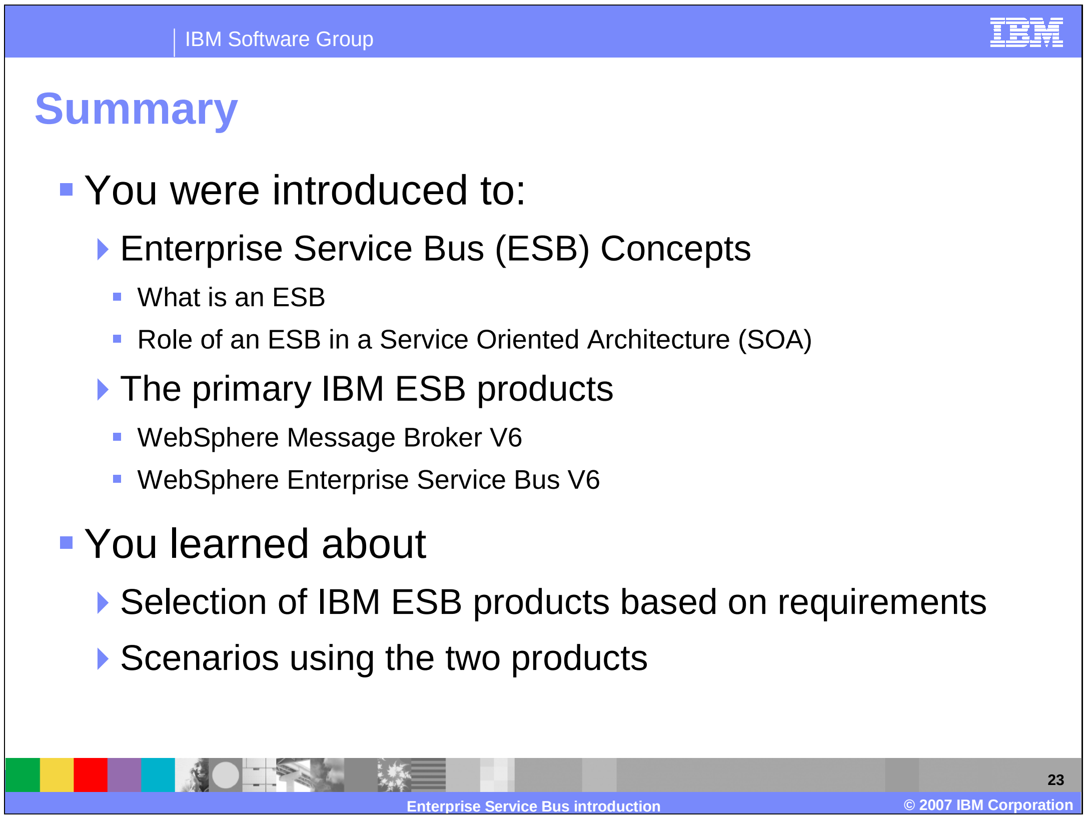
At the corporate data center, WebSphere Message Broker serves as the connectivity hub. It sends and receives information with the retail store locations and handles transforming messages between the various packaged applications and mainframe systems utilized by the corporation. It handles business to business interactions with other companies for things like wholesale ordering of products. All this allows the retail store locations to participate in the overall business process flows of the corporation while masking the complexities of the various systems that are integrated by the corporate data center with WebSphere Message Broker.



Section

Summary and references

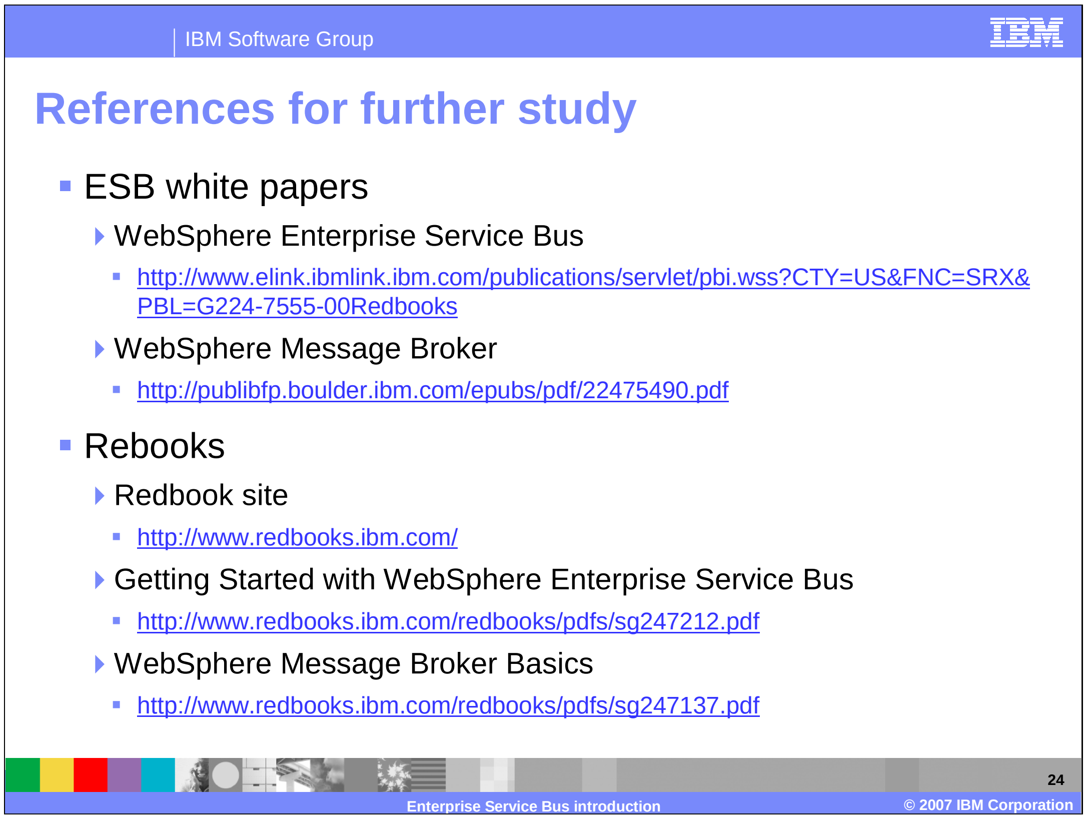
The next section covers the Summary and References.



Summary

- You were introduced to:
  - Enterprise Service Bus (ESB) Concepts
    - What is an ESB
    - Role of an ESB in a Service Oriented Architecture (SOA)
  - The primary IBM ESB products
    - WebSphere Message Broker V6
    - WebSphere Enterprise Service Bus V6
- You learned about
  - Selection of IBM ESB products based on requirements
  - Scenarios using the two products

In this presentation, you were introduced to the concepts of an Enterprise Service Bus which is also commonly referred to as an ESB. You learned what the function of an ESB is and the role that it plays in a service oriented architecture. IBM has two primary products which are classified as Enterprise Service Bus products, specifically the WebSphere Message Broker version 6 and the WebSphere Enterprise Service Bus version 6. You learned how the functionality delivered by these two products compares and were shown scenarios that make use of the products. With this background, you should now have a basic understanding of how to evaluate which of the products would be a more appropriate fit for a given set of requirements.


IBM Software Group

References for further study

- ESB white papers
  - WebSphere Enterprise Service Bus
    - http://www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=G224-7555-00Redbooks
  - WebSphere Message Broker
    - http://publibfp.boulder.ibm.com/epubs/pdf/22475490.pdf
- Rebooks
  - Redbook site
    - http://www.redbooks.ibm.com/
  - Getting Started with WebSphere Enterprise Service Bus
    - http://www.redbooks.ibm.com/redbooks/pdfs/sg247212.pdf
  - WebSphere Message Broker Basics
    - http://www.redbooks.ibm.com/redbooks/pdfs/sg247137.pdf

24

Enterprise Service Bus introduction

© 2007 IBM Corporation

On this slide you are given pointers to a few of the many resources that are available for further study.

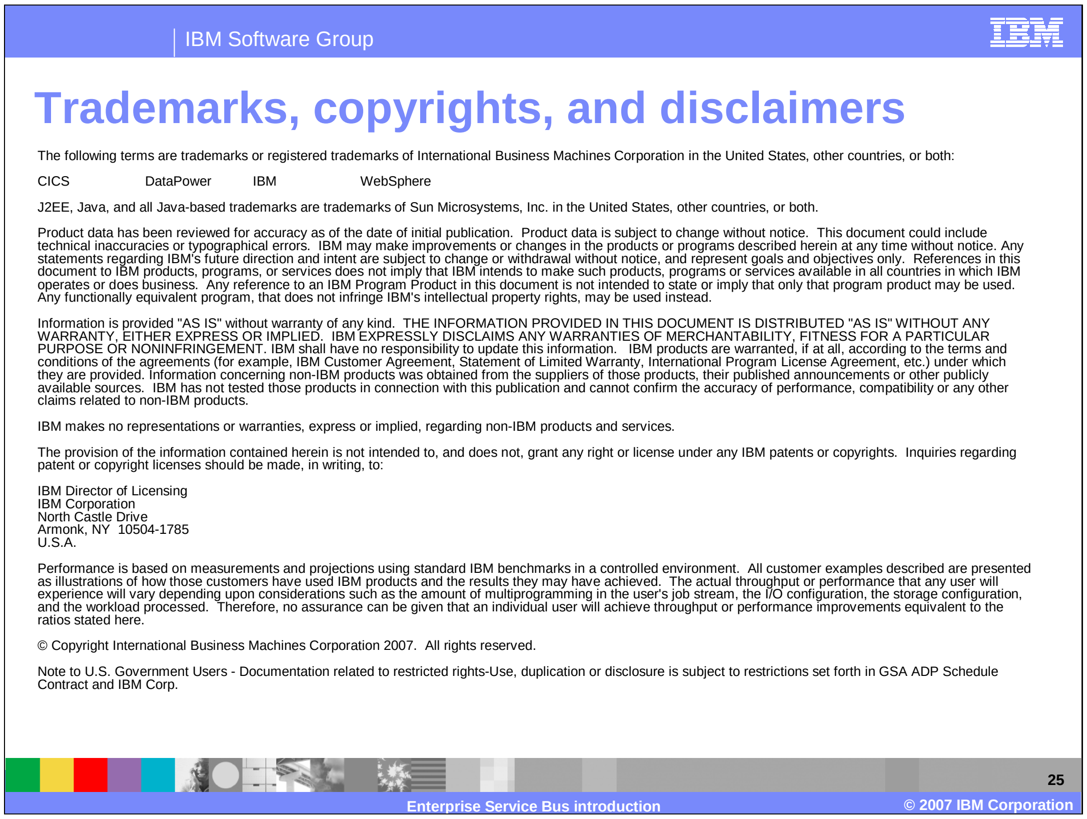
The first is a white paper that introduces the WebSphere Enterprise Service Bus and discusses how it enables your IT infrastructure to be more flexible. It provides an overview of Service Oriented Architecture and the place of an Enterprise Service Bus within a SOA. It introduces the WebSphere Integration Developer tools used to develop mediation applications and the WebSphere Enterprise Service Bus which provides the runtime support for the ESB. The white paper can be easily read and is only about 16 pages long.

The next white paper introduces WebSphere Message Broker and discusses the business value that is delivered through use of the product. It provides a discussion about meeting the challenges of universal connectivity and rich data transformation. It examines topologies, performance and complex event processing. This white paper is easily consumable and is also 16 pages long.

There are several Redbooks available that deal with Enterprise Service Bus topics and products. You can use the link to the Redbook site and from there search for titles that look to be of interest to you.

One of the Redbooks specifically listed here helps you get started with WebSphere Enterprise Service Bus. It provides an overview of the technology and then goes into details about development, testing and administration, and provides a complete set of examples. This Redbook is over 500 pages. It is worthwhile as a reference resource even if you cannot devote the time to reading the entire document.

The other Redbook shown here helps you get started with WebSphere Message Broker version 6, discussing installation, administration and the development and testing of a variety of approaches to mediations. It is an update to a previous version which was focused on WebSphere Message Broker version 5 and therefore has less content directly discussing ESB concepts. However, it is still very useful as a starting point to understand the WebSphere Message Broker runtime and development environments used to develop an ESB. This Redbook is about 350 pages long.



Trademarks, copyrights, and disclaimers