WebSphere in an SOA Environment

Sofia, 15th September, 2004

Anthony B Chapman
IBM SWG Business Integration Sales Manager
IBM Central and Eastern Europe / Middle East / Africa, Inc. (CEMA)

Business Integration Capabilities from IBM

Model business functions and processes

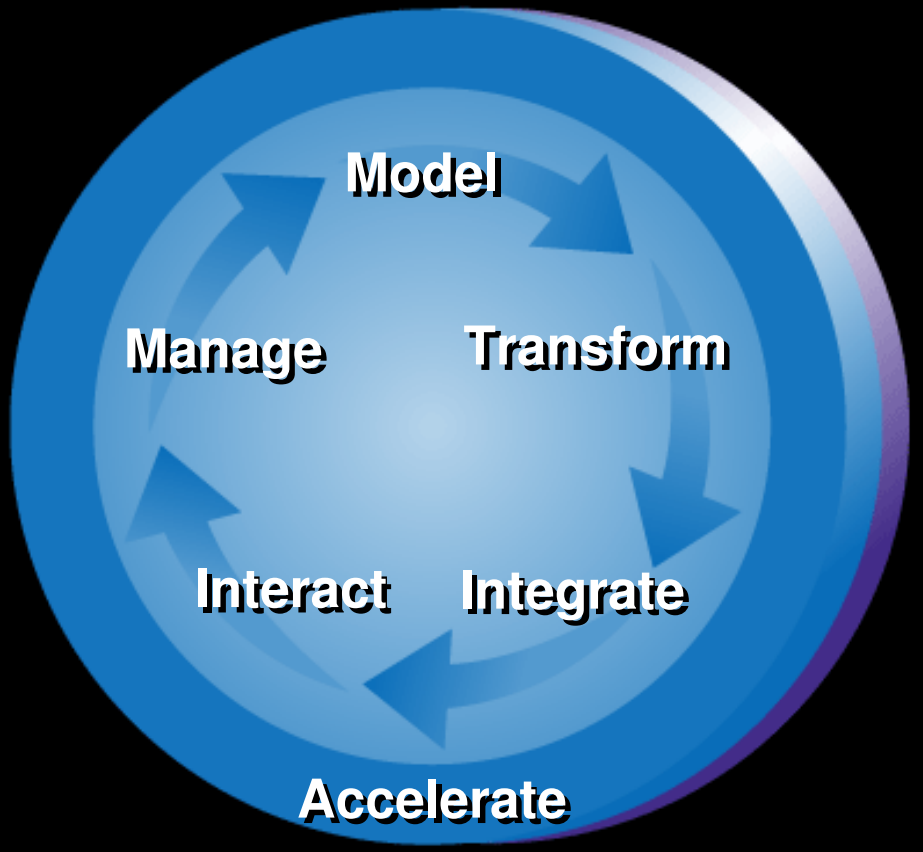
Transform applications, processes and data

Integrate islands of applications, processes and information

Interact with resources anytime, anywhere with any device

Manage performance against business objectives

Accelerate the implementation of intelligent processes



Business Integration Qualities from IBM

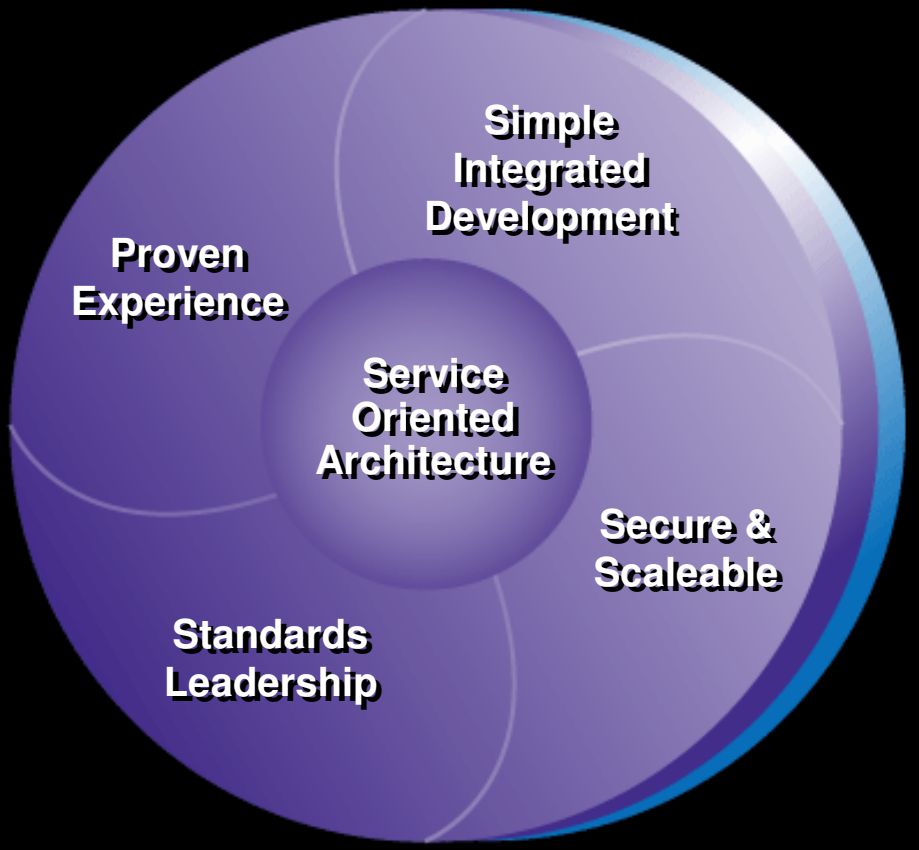
Simple, integrated development

Secure and scalable deployment

Standards leadership

Proven experience

Service Oriented Architecture (SOA) enablement

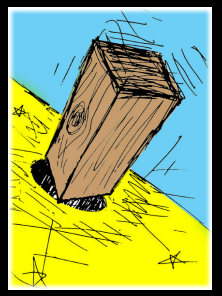


The Problem: Integrating Heterogeneous Systems

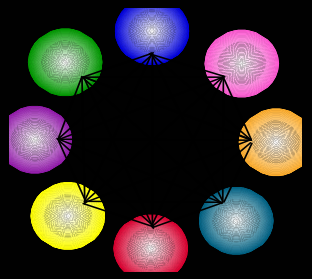
- IT systems were originally built to facilitate manual work
  - computers were introduced to speed the work, but lack of interoperability between systems has made them become part of the problem
  - the trend today is increasing automation, and integration on an larger scale
- As business continues to grow and become more complex, we need to deal quickly and efficiently with challenges like:
  - new business opportunities
  - new customer requirements
  - competitive threats
  - reduced resources to cope with change
- Companies are struggling to become more responsive and efficient in order to win business from competitors.
  - We need end-to-end integration – internally, as well as between partners
  - Integration needs to be interoperable, flexible, and independent of the technology choices for each component in the integrated system

Barriers to Interoperability

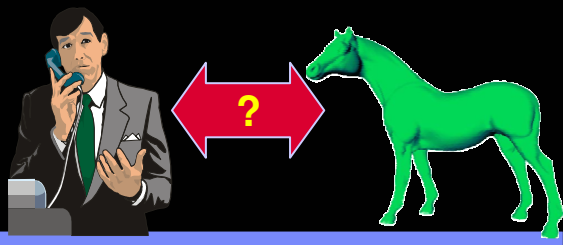
- Our legacy of heterogeneous application systems gives us these challenges in integration:
  - different programming languages, operating systems or hardware platforms
  - different software vendors, in-house code
  - APIs and file formats that change because of evolving business requirements



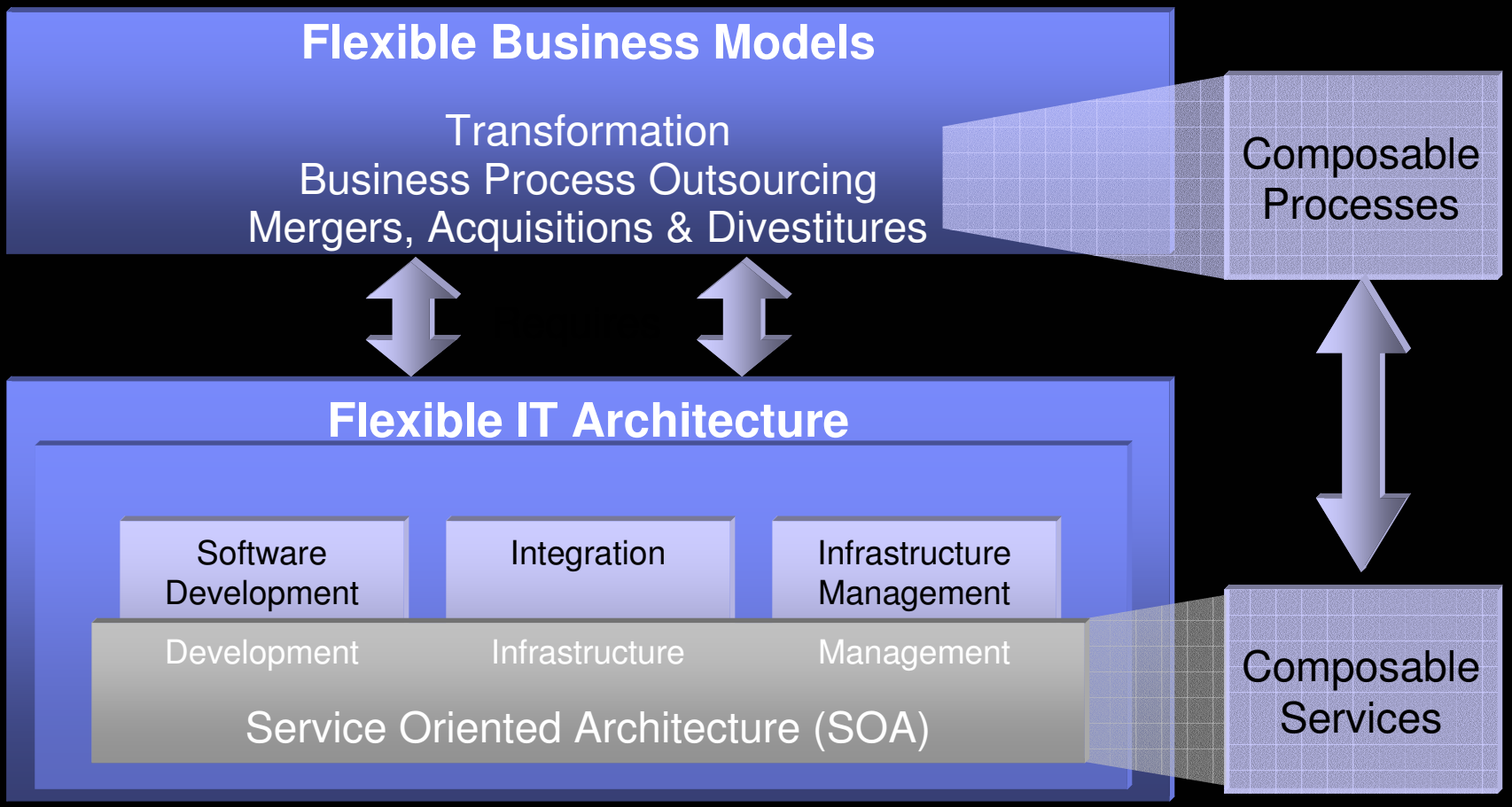
- Historically, integration solutions have been ad-hoc
  - programming language bridges, binary conversions lead to high maintenance costs and brittle infrastructure
  - multiple technologies, specialized for each integration connection, lead to increasing complexity
  - the expensive work of integrating two systems can't be leveraged for integrating a third system



- Proprietary EAI solutions offer some relief internally but integration with a partner with different technology is "a horse of a different colour"



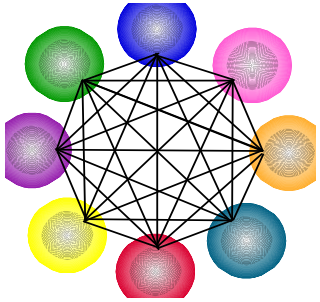
Flexible Business Models require a flexible IT architecture



Service Oriented Architecture

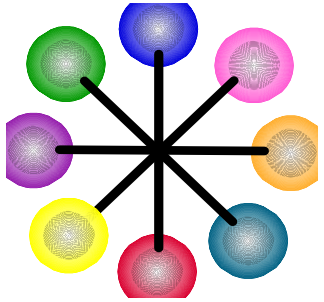
- A Service Oriented Architecture enables flexible connectivity of applications or resources by
  - Representing every application or resource as a ***Service*** with a ***Standardized interface that is shared***
  - Enabling them to ***Exchange Structured Information*** (messages, documents, 'business objects')
  - Mediating the message exchange through a ***Service Integration Framework***
  - Providing ***On-ramps*** to the bus for legacy application environments
- This allows ***quick combination*** of new and existing applications to address changing business needs
- The SOA infrastructure is also used to facilitate the management of business performance and ***quality of service***
- SOA Design Principles:
  - Design functionality as accessible and ***reusable services***
  - Expose service functionality through programmatic ***visual interfaces***
  - Maintain an ***abstraction layer*** between service interfaces and implementations
  - Generate communication code ***automatically*** from service descriptions
  - Bind services using ***standard protocols***

SOA vs Traditional Integration



Traditional integration requires technology-aware bridges between components. It is complex and expensive to implement and to maintain.

Program bridges are typically based on APIs and File Formats, but these change, leading to instability in the integrated system.



In SOA, each component uses the same way of talking to other components, based on platform-neutral standards.

Anything that understands Web services can talk to other Web services, regardless of underlying implementation.

This approach greatly simplifies and strengthens integration efforts

Characteristics of a Service

- Services can be invoked through defined communication protocols that stress interoperability and location transparency.

Component appearance: The Service **appears as a self-contained function** from the perspective of the Service Requester, even if it is not actually self-contained.

Loosely-coupled: it uses a **well-defined** interface designed to expose business functions and data, but also to **hide underlying implementation** details from service requesters, by focusing only on message content, not APIs or file formats

Well-defined: it uses a **common definition of services** that is independent of any particular technology, but can be used by all technologies

Stateless: services may not remember the last thing they were asked to do, nor care what the next is. Services are not dependent on the context or state of other services.

Any dependencies between services should be defined in terms of common business process, function and data models, not implementation artifacts

What are **Web** Services?

•"Web services are **software components described via WSDL** that are capable of being accessed via **standard** network protocols such as SOAP over HTTP.“

- Web Services technologies were developed by IBM, Microsoft, and others and offered to bodies like W3C and OASIS for standardization
- Web Services technologies are being adopted by vendors and customers as a common way of interacting between or within businesses
- IBM leads the industry in driving standardization, providing middleware and development tools to create new Web services applications, as well as repackage existing applications as services

▪Web Services standards makes sense for integration between organizations, where applications are controlled by different organizations.

What's the difference between SOA and Web Services?

- SOA is the architectural style of building applications based on Services, and deals with the arrangement of Services in application systems.

- Web Services is a particular set of technologies – SOAP, WSDL, and others – that can be used to create a Service Oriented Architecture.

- SOA can be implemented using Web Services technologies, but it does not require the use of Web Services technologies
  - SOA is an architectural style, but does not require any particular technologies
  - Web Services is the best technology we have today for SOA

- Any Web Services deployment might be considered SOA
  - Some argue that a Web Services implementation that is not stateless, or fine-grained services, is not SOA. I think it is SOA because of its basis on Services, but using Services does not necessarily make it good SOA.
  - Good SOA depends on using various Best Practices

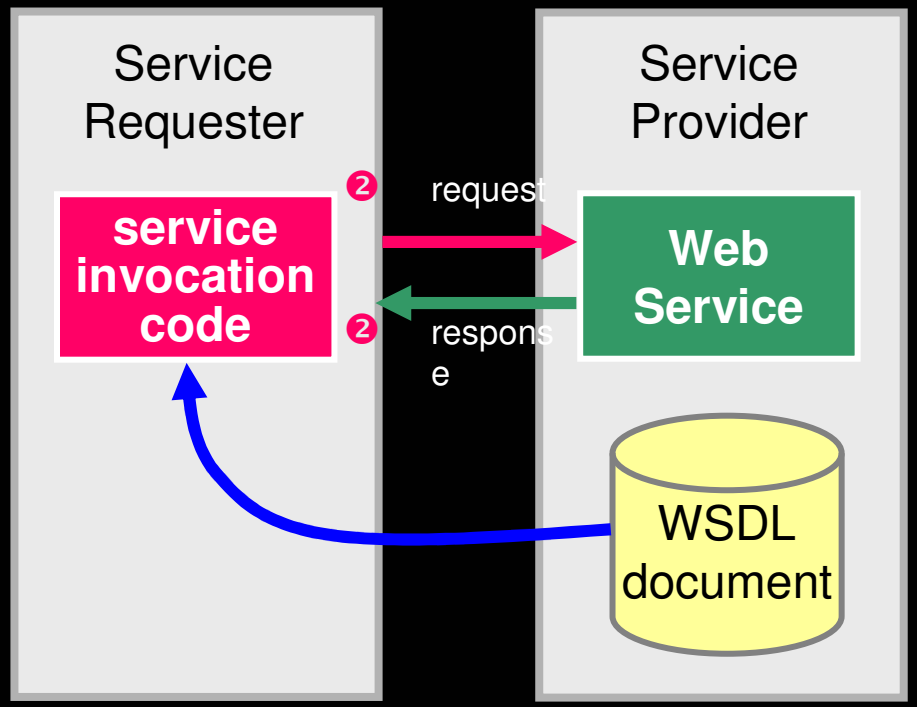
Web Services: The Basics

- The SOAP* specification defines the use of XML content in a particular form for Service Invocation.
  - HTTP is commonly used as a protocol for exchanging SOAP messages, but SOAP can be used with any protocol.
- The WSDL (Web Services Description Language) document tells you (or your dev tools) what you need to use the service
  - the requirements of the request and response messages
  - where and how to send the message
- Most of today's Web Services applications depend on request and response messages for communication

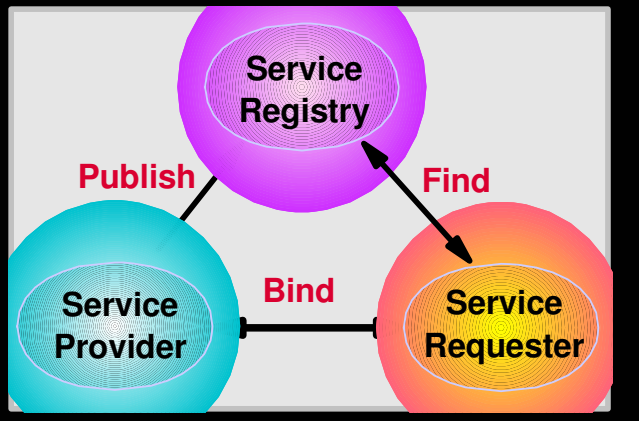
*** Note:** The early SOAP specifications claimed that SOAP stood for "Simple Object Access Protocol," but the authors of the SOAP 1.2 Specification explain that SOAP no longer stands for anything in particular: it's just a word.

How Web Services work: a very simple view

1. A developer gets the WSDL description for the desired service from the Service Provider, and uses it to create a Service Requester application.
2. The Service Requester runs, creates a request message (in SOAP/XML), and sends it to the Service Provider for processing.
3. The Service Provider processes the request, and answers with a “response message.”
4. The Service Requester processes the data from the response message.

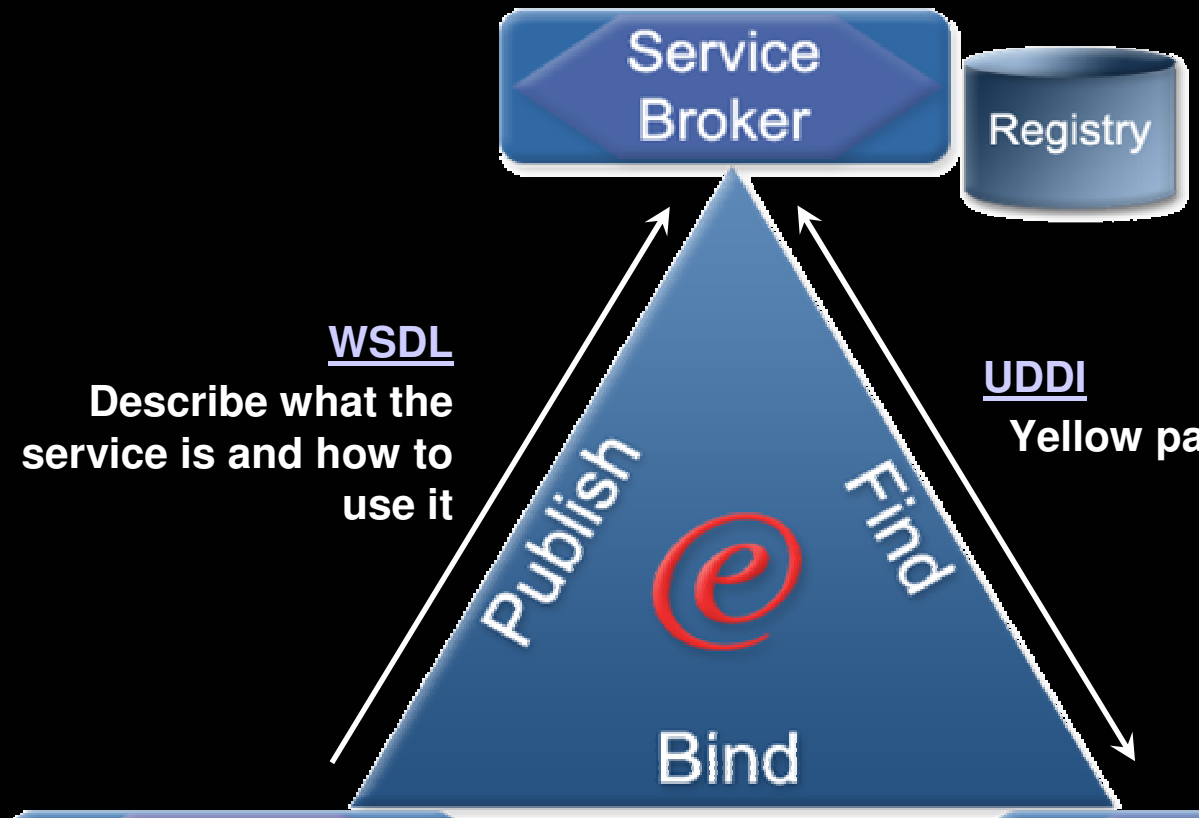


Service Discovery: UDDI



- Universal Description, Discovery, Integration
  - A Service Registry or Directory standard
  - Providers can publish service availability
  - Requesters can finds services
  - SOAP messages are used for finding and publishing entries
  - UDDI is a standard for a Web services registry
    (see UDDI.org and OASIS-open.org)
- Service discovery may be done at design time or run time
  - UDDI Service entries contain URL of its WSDL
  - WebSphere Studio provides tools to work with UDDI
- A UDDI registry may be private to an enterprise, or made publicly available
- WebSphere App Server 5 Network Deployment Edition includes a UDDI registry that you can deploy.

Web Services Technologies



- A searchable repository of service descriptions
- Service Providers publish their services
- Service requesters find services
- Service broker matches providers with requesters

WSDL

Describe what the service is and how to use it

UDDI

Yellow pages for web services

SOAP

Connect the service

Service Provider

- Provides application interface as a Web Service
- Publishes interface to Broker

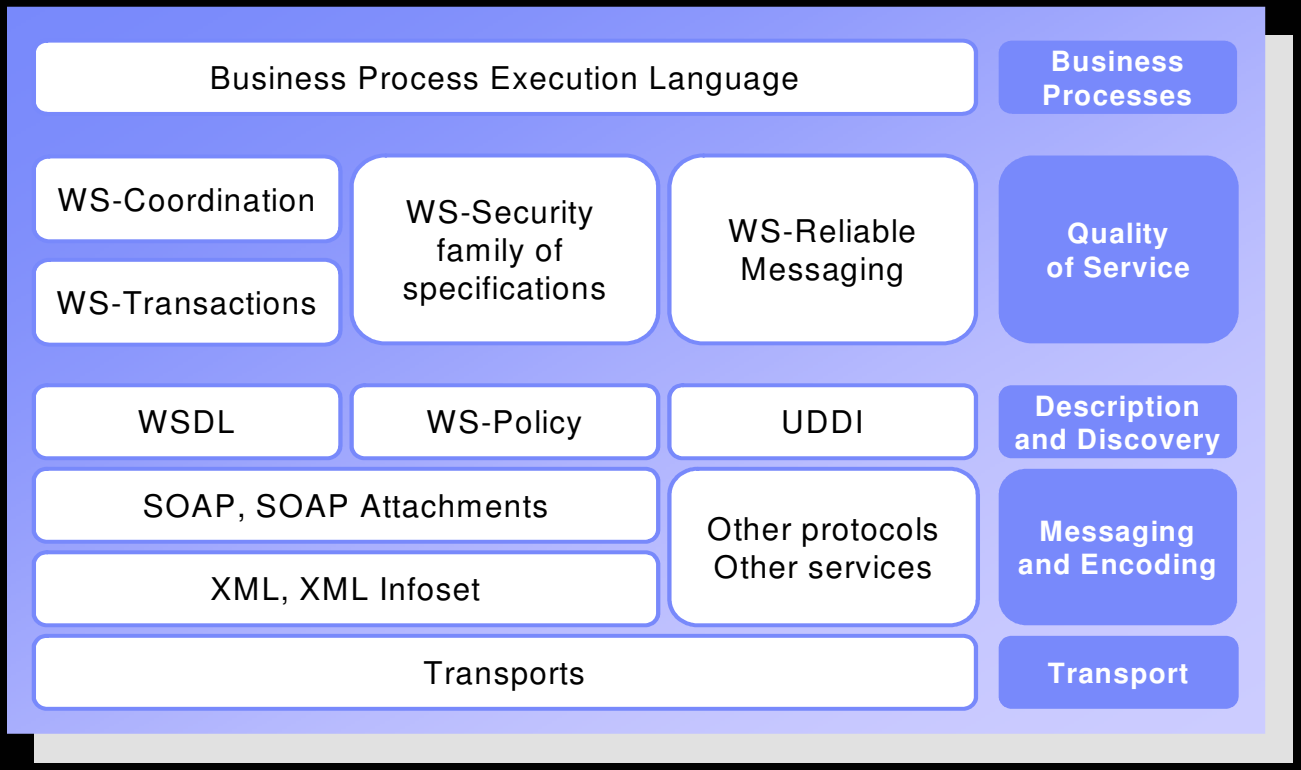
Service Requester

- Finds Web Service to use
- Binds to Web Service at requester
- Obtains output as response

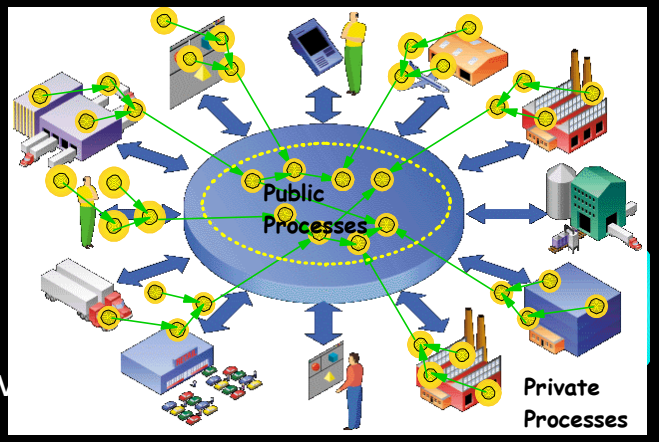
The Web Services "Stack" of Specifications

SOAP, WSDL, and other emerging standards from W3C and OASIS provide interoperability, platform-neutral security, reliability, business process description, transactionality, and more.

The specifications can be used as required to implement requirements of the service.



Service Choreography

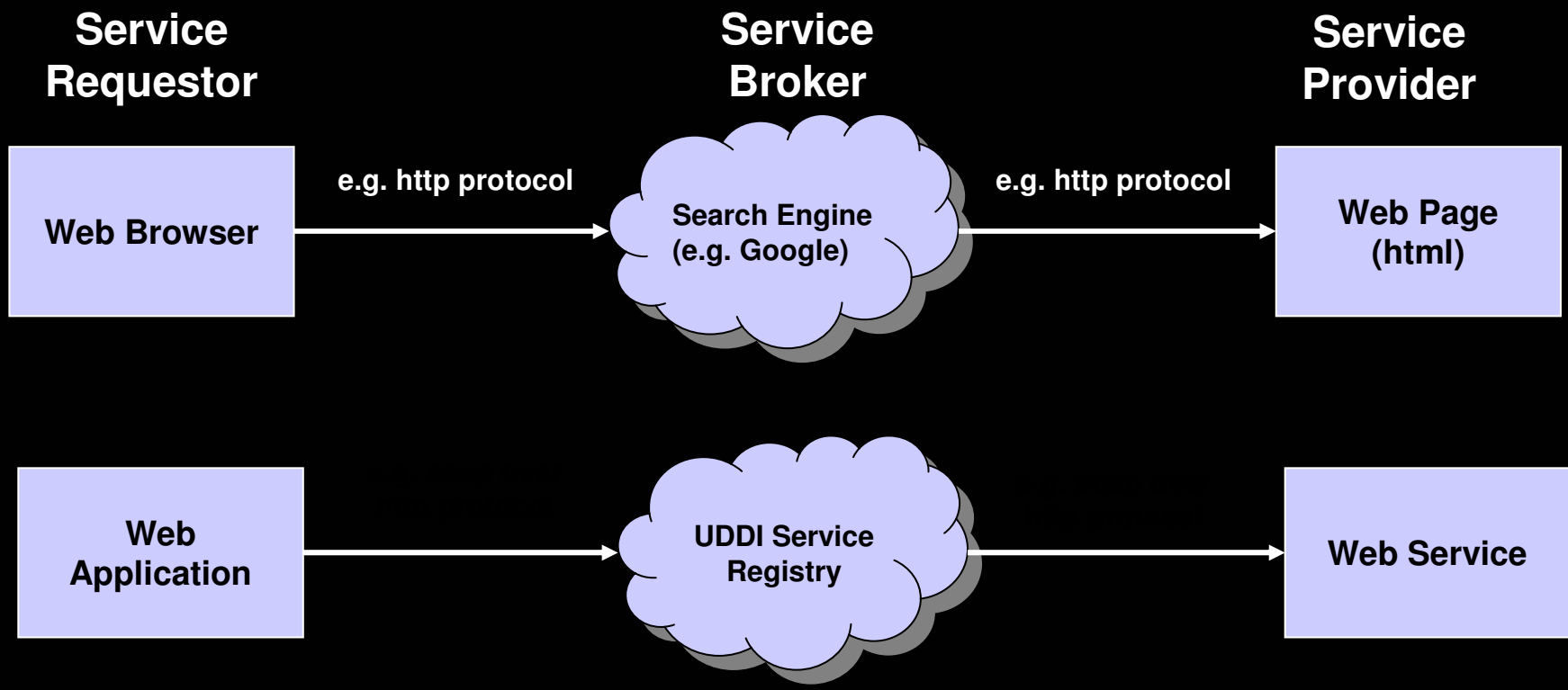


- Business Processes are a set of activities that are carried out in a particular sequence to support a business activ
- Services can be choreographed to implement a Business Process as a new Service
  - A step or activity in a business process can invoke a Service
  - The resulting business process itself becomes a Service like any others – it can be described in WSDL, invoked with SOAP, etc
  - We can compose new Services out of a Business Process driving, using BPEL, Java, or any other language
- BPEL (Business Process Execution Language) is a specification to describe a portable XML representation of such business processes
  - OASIS-open.org has formed a Technical Committee to create a standard based on BPEL, and the work is in progress.
  - Products from IBM, Microsoft and others currently support the proposed BPEL spec
- BPEL provides:
  - Processes that combine applications and people
  - Transactionality, fault handling, compensation
  - Manipulation of process data

Web Services - Analogy

"..in short, a Web Service is to an application what a Web page is to a person.."

- Andy McCright, Web Services Journal, Preview Issue June 2001, pg.4



Why the interest in Web Services?

Trends driving the use of Web Services:

- Integration of Java and .Net
- Inter-enterprise BPM
- Back-end integration
- Software components

Just as companies were cautious when the internet first started, we expect a similar progression with Web Services.

1. Internet technologies were first used within the company (intranet),
2. Then with known partners (extranet),
3. And then with the public (internet).

Web Services will follow this same progression.

Web Services Examples

Retail –

- ✓Sells products on the internet
- ✓Wants to integrate a package tracking system on their retail web site

Travel –

- ✓Sells tickets, hotel reservations, car rentals, etc.
- ✓Wants to integrate a weather function for the destination

Financial –

- ✓Sells Stocks, Bonds, etc.
- ✓Wants to integrate a news feed for companies of interest

Benefits of a Service Oriented Architecture

Cost savings through the reuse of services

*"Building toward a common standardized **SOA** gives us huge benefits for reusability across the company," says Rick Wiseman, CTO for Galileo, a travel-booking service that is building applications based on the concept. "As new product requirements come about, you don't have to start from a whiteboard anymore. You can assemble products from existing components."*

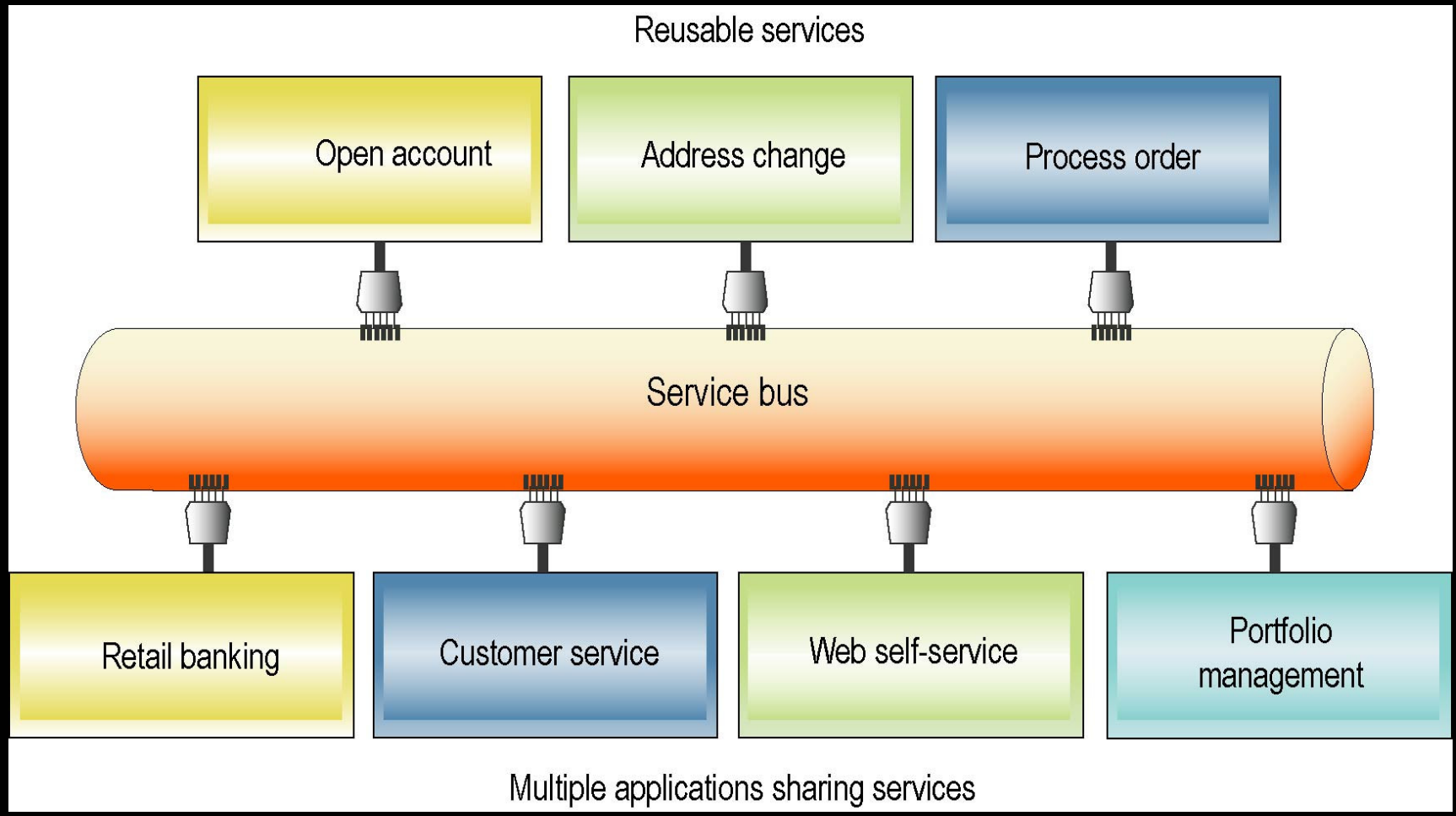
More quickly respond to customer demands

*The **Service-Oriented Architecture** promises to succeed where others have failed in building a network-friendly, reusable app architecture. Network World, 09/29/03.*

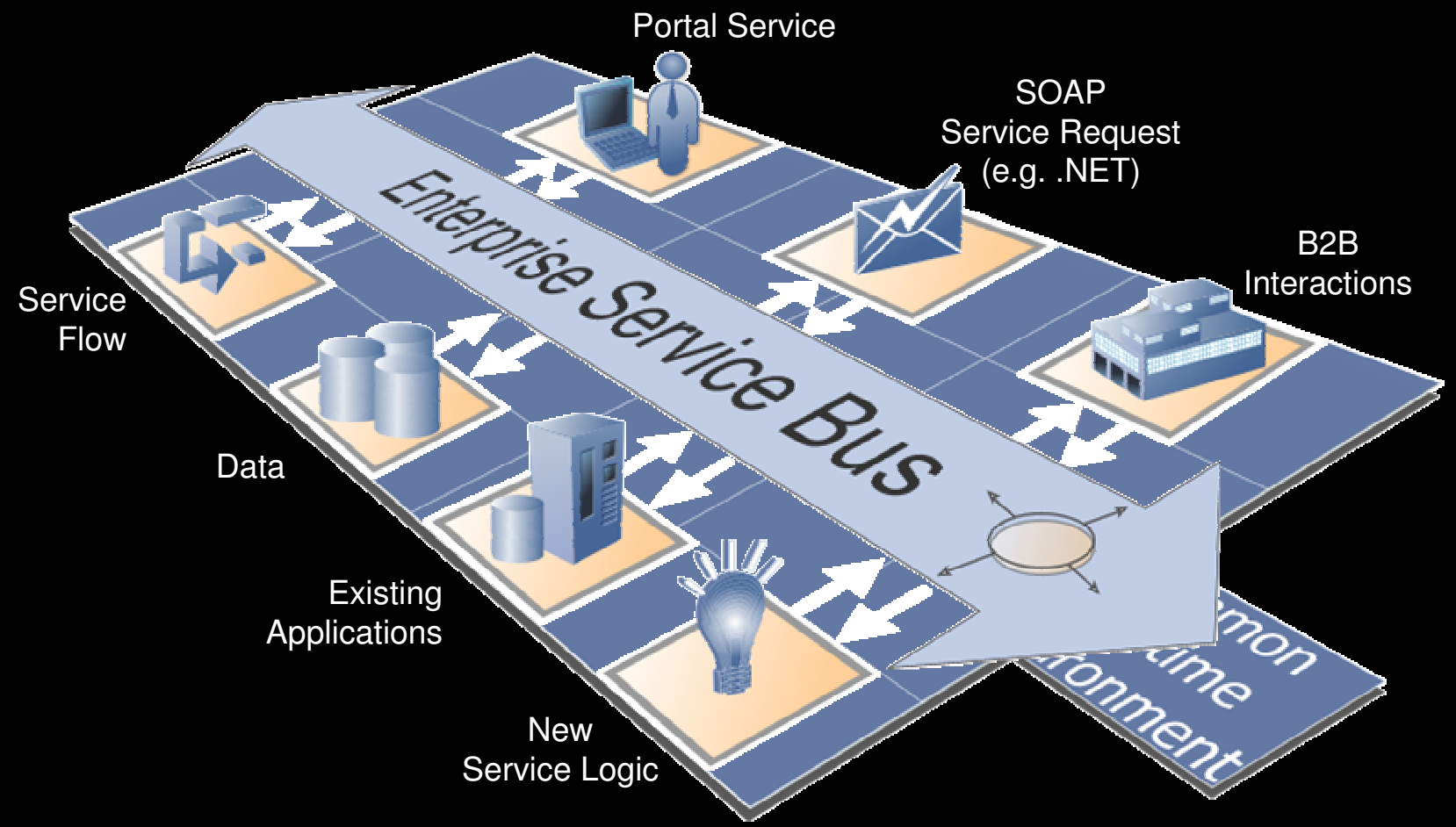
Innovation through new capabilities

*"The reality is this might be the right time for us to be thinking about loosely coupled distributed computing, in part because of the movement to do it **based on standards as opposed to proprietary integration technologies**," says Ron Schmelzer, another ZapThink senior analyst. "*

SOA Concepts



Service Oriented Architecture



Enterprise Service Bus
A Best Practice Design Pattern

Mediation Services

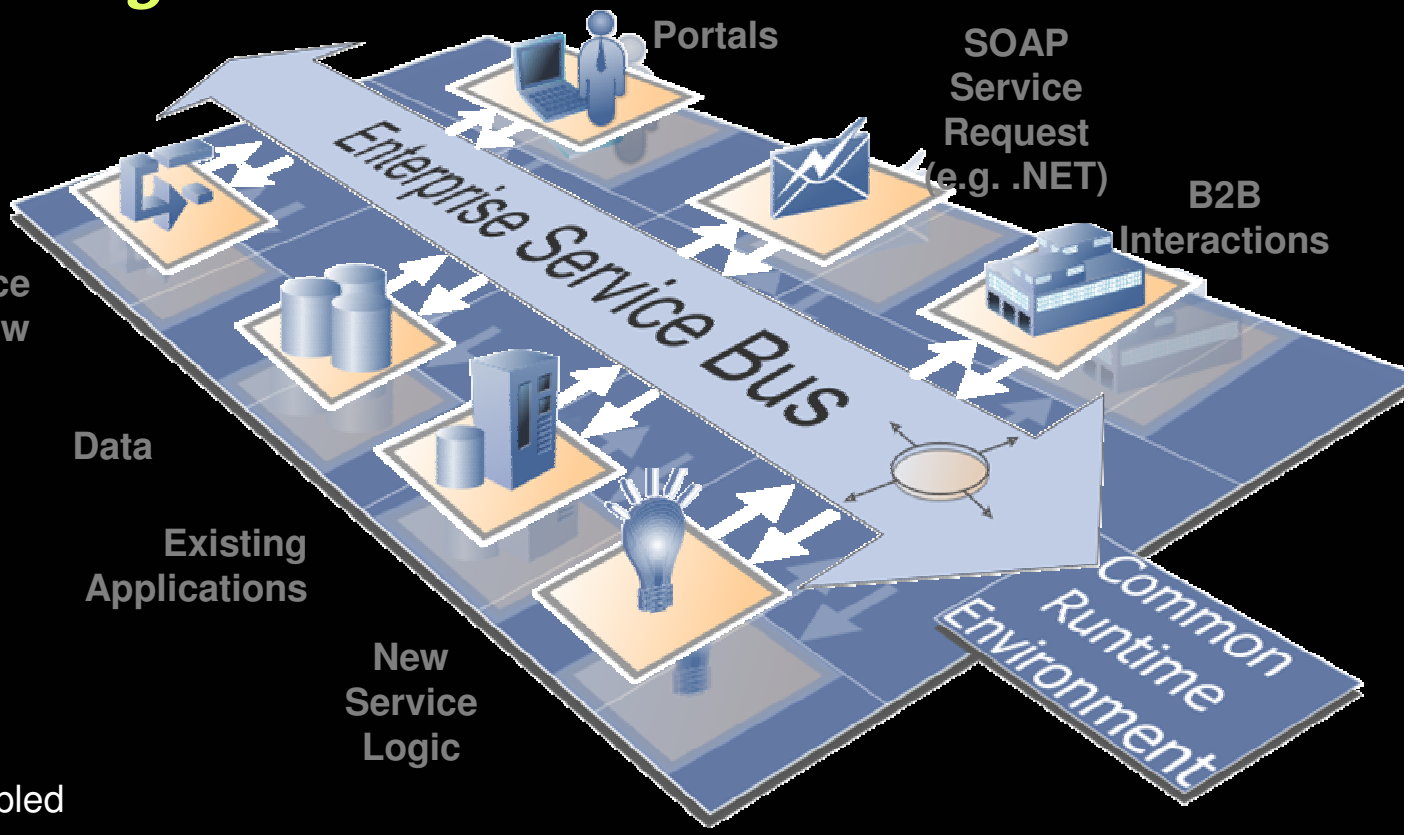
- Routing
- Transformation

Event Services

- Publish and Subscribe

Transport Services

- Synchronous/Asynchronous
- Persistent/Non-persistent
- Loosely-coupled/Tightly-coupled



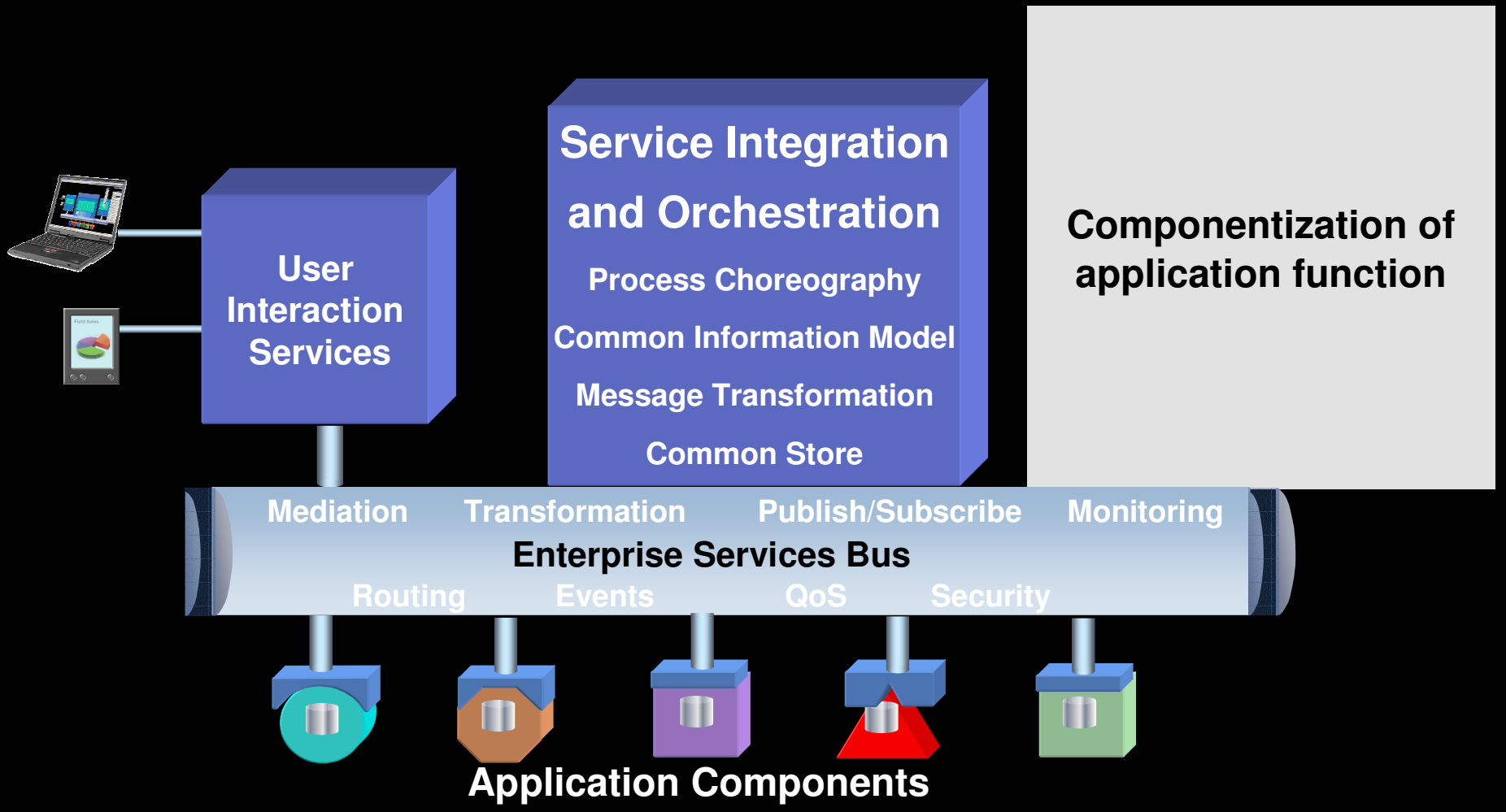
… with unsurpassed technical characteristics …

- Multiple standards – J2EE, WebSphere MQ, Web Services …

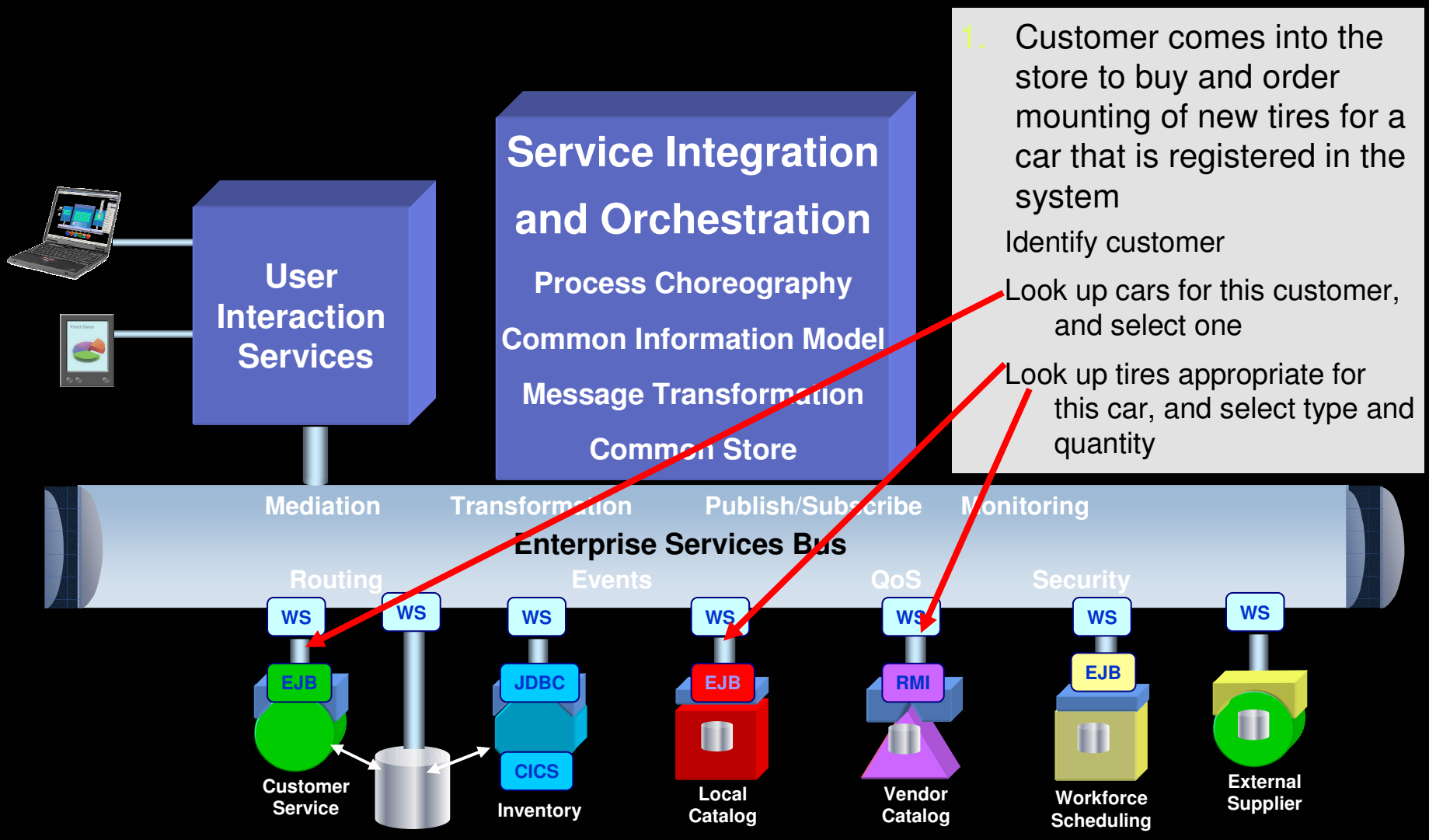
- Scale to match performance and throughput needs

- Security conscious

Conceptual ESB Structure



Process Overview: Auto parts sales and service

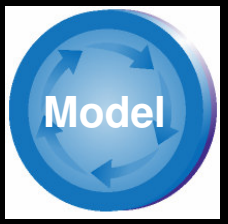


The Path to SOA – BI Capabilities

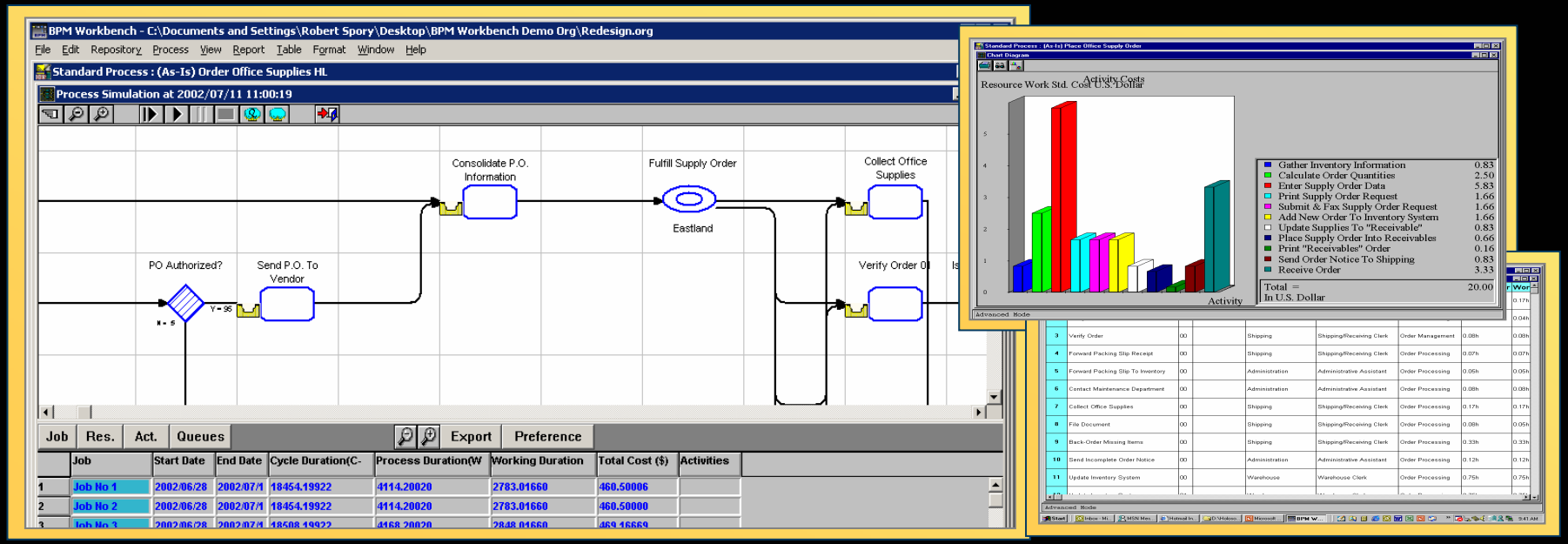
- Model
- Transform
- Integrate
- Interact
- Manage

Accelerate

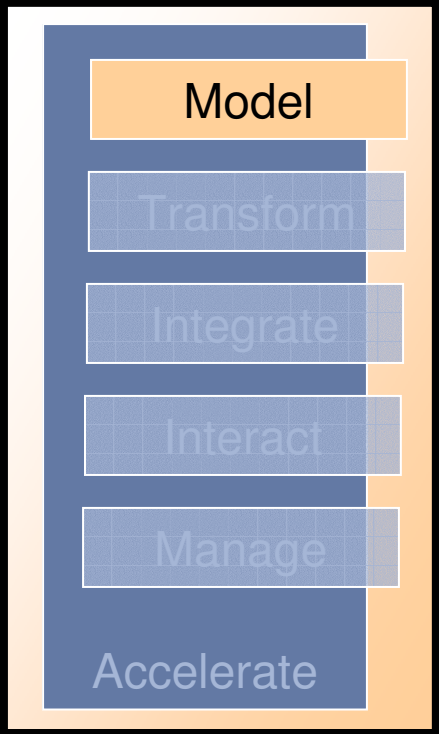
Model Business Functions and Processes



- Graphically design processes and quickly redesign across people, partners and applications
- "What-if" simulation of operations to optimize and project business benefits
- Fast start to deployment—generates code from model



The path to SOA



WebSphere Business Integration Modeler

WebSphere Studio Family

Development Studio

- **User:**
  - iSeries Developer
- **Function:**
  - Leverage and extend iSeries date, code and skills
- **Target Runtime:**
  - WAS on iSeries

Integration Edition v5.1

- **User:**
  - Advanced J2EE Developer
- **Function:**
  - SOA
  - Business Process Choreography tools:
    - BPEL and FDML process editors
    - Services wizards
  - J2C Connectors Tools
  - Connectors for IMS, CICS-ECI/EPI, IMS, HOD
- **Target Runtime:**
  - WBI SF
- **Built on WSAD v5.1.1**
  - JSF is Beta only

Application Developer v5.1.2

Site Developer v5.1.2

- **User:**
  - Web, Java, XML, Web Services Developer
- **Function:**
  - HTML, JSP, Servlet, JSF, Struts Tools
  - EGL Tools for Java
  - Web Services Tools
  - XML Tools
  - Java Visual Editor
- **Target Runtime:**
  - WAS, WebLogic

- **User:**
  - J2EE Developer
- **Function:**
  - EJB Tools
  - Performance and Trace tools
  - Component Test tools
  - Relational DB tools
  - ClearCase LT
- **Target Runtime:**
  - WAS, WebLogic

Enterprise Developer v5.1.1

- **User:**
  - Enterprise Developer
- **Function:**
  - EGL Tools for COBOL (v5.1.2)
  - Legacy z/OS support
  - COBOL, PL/I, ASM Tools
  - COBOL XML Tools
  - J2C Connectors Tools
  - Connectors for IMS, CICS-ECI/EPI, IMS, HOD
- **Target Runtime:**
  - WAS z/OS
- **Built on WSAD v5.1.1**
  - JSF is Beta only

WebSphere Studio Workbench

- IBM's commercially supported version of Eclipse Workbench

WebSphere. software

Eclipse Workbench

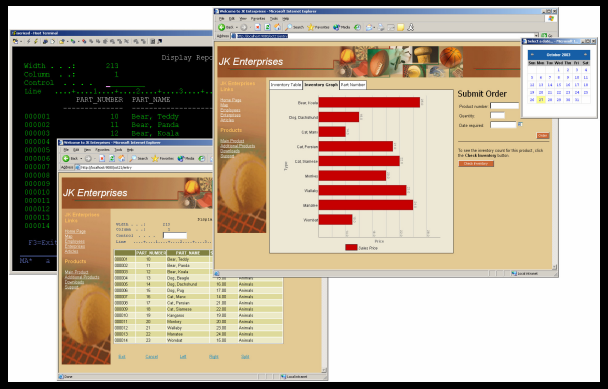
- Open Source Universal Tool Platform Initially Developed: core Java IDE, core VCM API/CVS Plug-in, etc...

eclipse

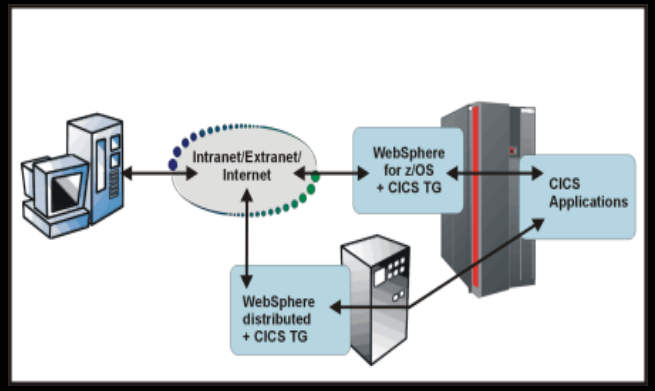
Transform Applications, Processes and Data

- Create new business value from existing IT systems
- Integrate traditional zSeries and iSeries applications and new Java applications into an efficient mixed workload environment
- Leverage existing enterprise skills and improve developer productivity
- Transform business-critical legacy processes into reusable, shareable business components

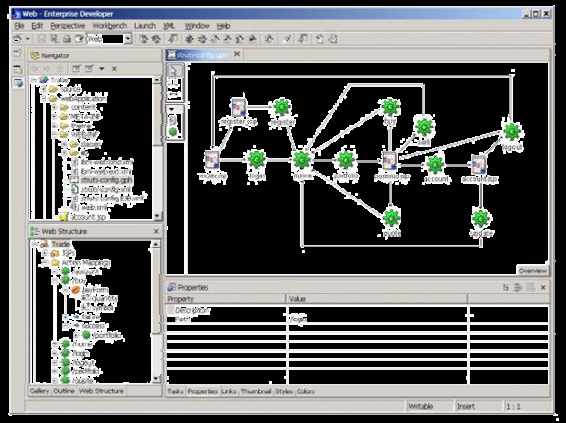
Transformed user interface and workflow for quick return on investment



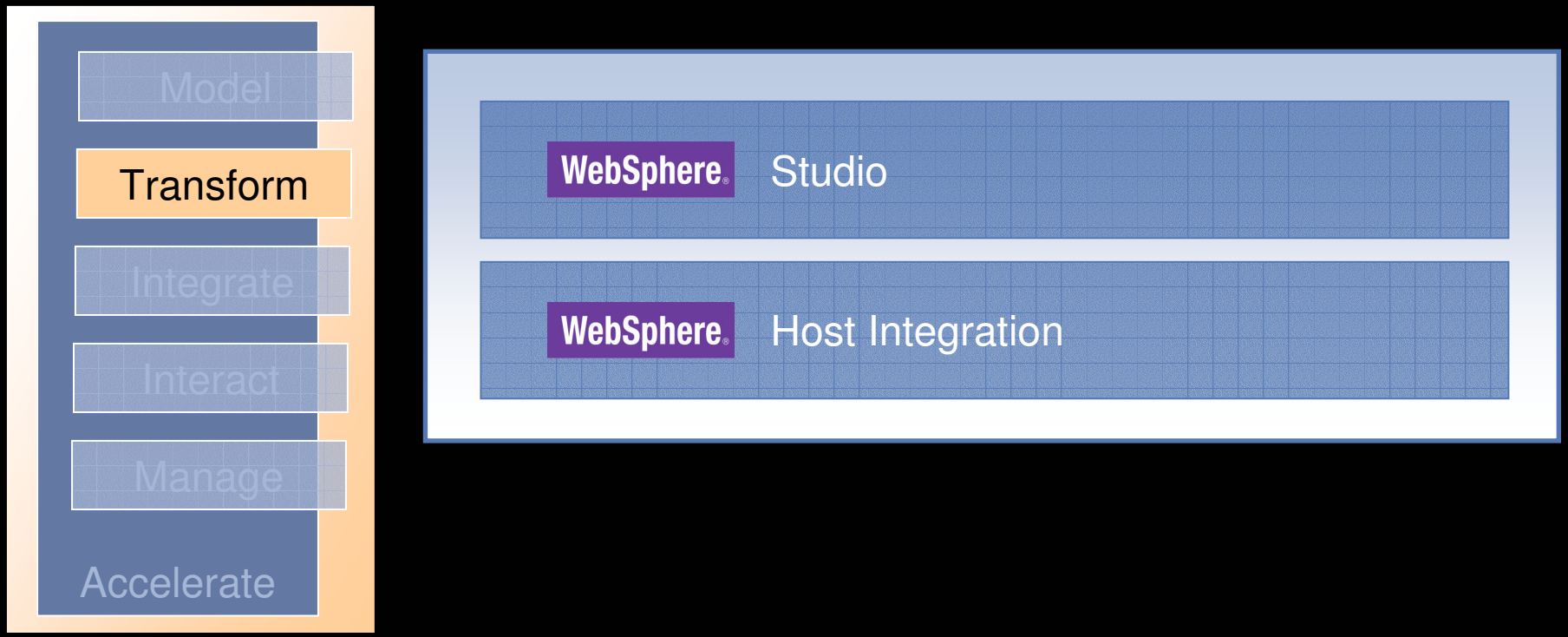
Transformed business processes using Web services and Java connectors



Transformed applications and data with tools for discovery, development and deployment of legacy assets



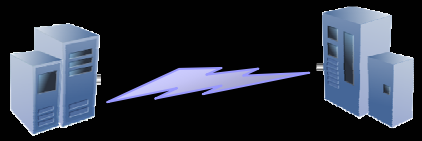
The path to SOA



Integrate Islands of Applications, Processes and Information

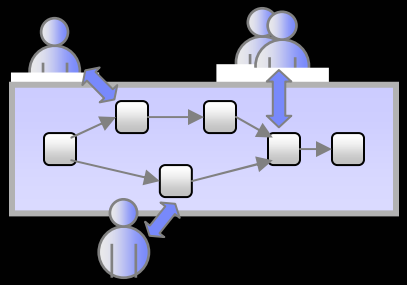
Adopt the integration methods that complement your existing infrastructure

Flexibility to grow as your business requirements grow

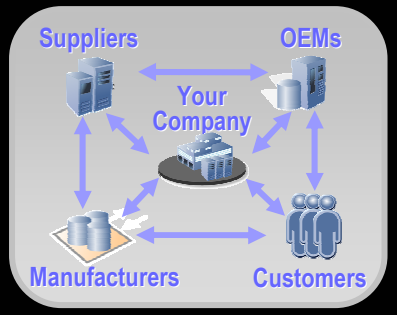


Broker reliable high speed communications across platforms and applications

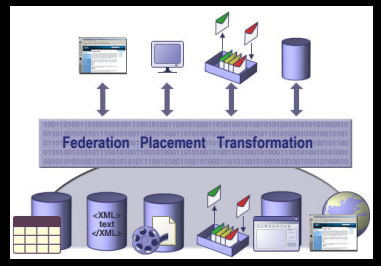
Deliver new Web services based composite applications that extend and integrate your existing IT assets



Automate processes that involve systems and people

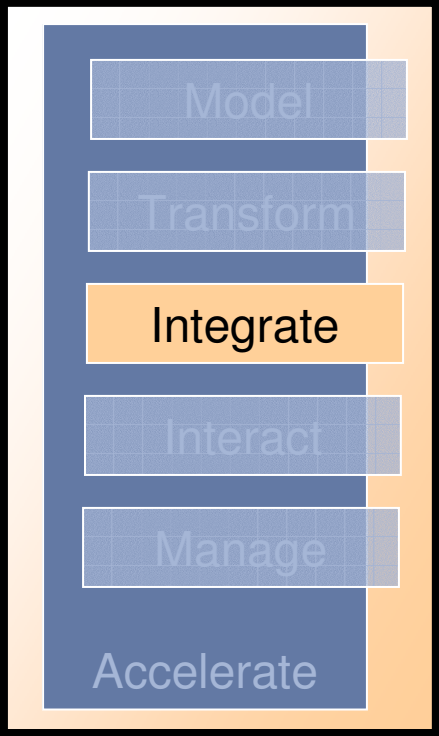


Extend business processes to trading partners



Provide integrated, real-time access to diverse data as if it were a single database

The path to SOA



WebSphere Application Server
- Built Applications
- Runtime environment

WebSphere Business Integration Server
- Business process
- FDL workflow
- MQ Messaging

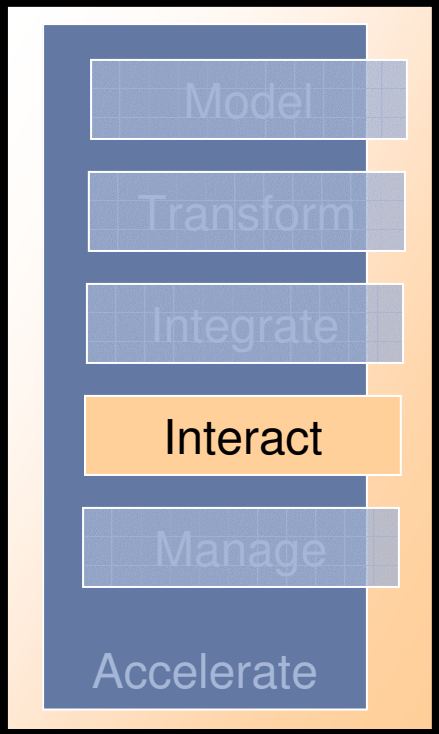
WebSphere Business Integration Server Foundation
- Business process and application choreography
- BPEL workflow
- MQ and JMS messaging

DB2 Information Integrator
- Federation
- Replication
- Warehouse

Interact with Resources Anytime and Anywhere

- Provide a secure personalized single point of interaction to people, data, applications and processes accessed by devices anywhere and anytime
- Customizable and flexible web portals based on business priorities
- Integrated collaboration and human interaction capabilities, easily managed and extended beyond organizational boundaries

The path to SOA



WebSphere Portal

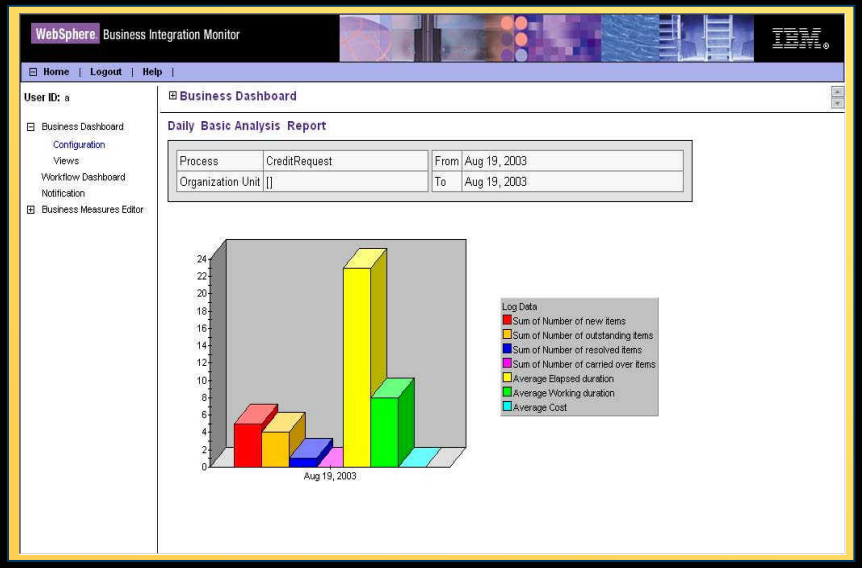
WebSphere Voice

WebSphere Everyplace

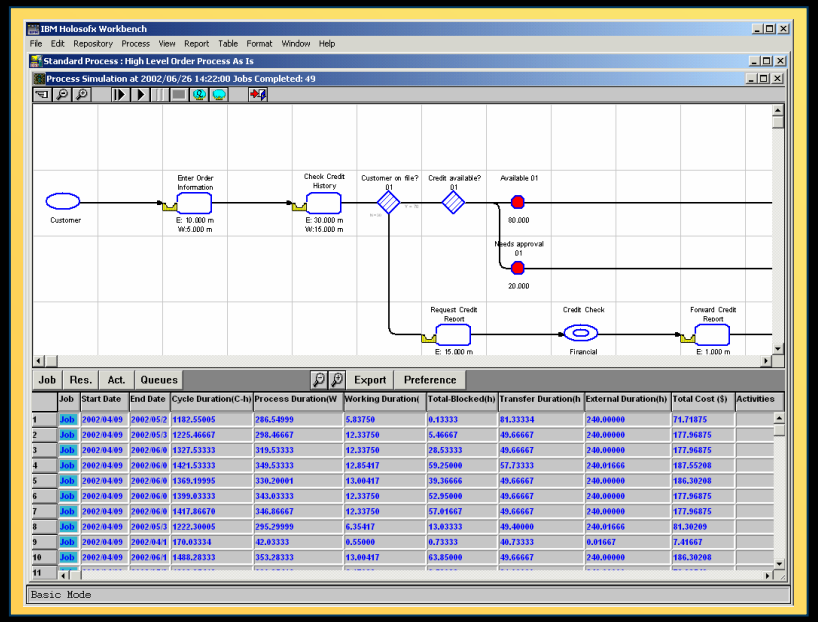
Lotus Workplace

Manage Performance Against Business Objectives

- Measure business performance against key objectives; Provide framework for improvement
- Dynamically change business processes to meet shifting business conditions and meet objectives
- Optimize resource reallocation

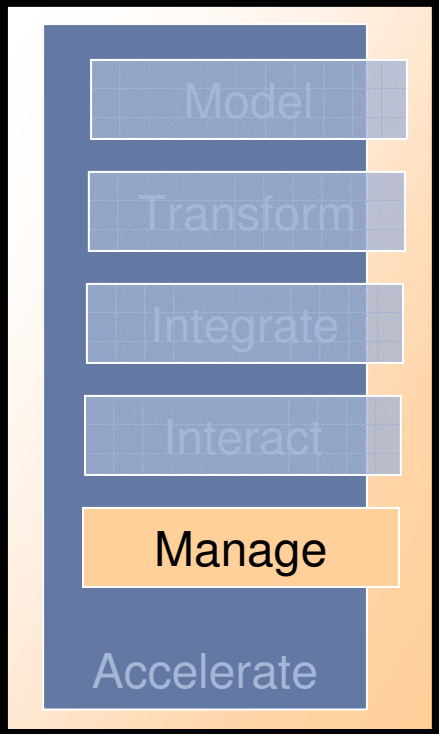


Actual Process Data



Process Simulation Data

The path to SOA



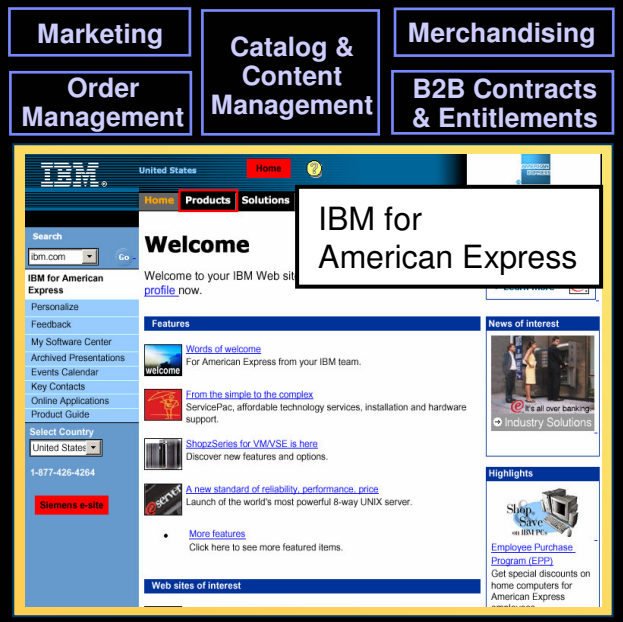
WebSphere Business Integration Monitor

Tivoli Business Services Management

DB2 UDB & Content Manager

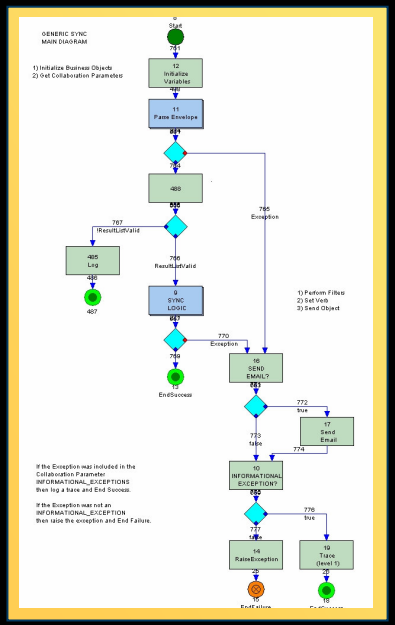
Accelerate Implementation of Intelligent Processes

- Online commerce and business transactions ‘out-of-the-box’
- Leverage industry best practices and industry-specific processes
- Deploy pre-built/pre-tested intelligent and extendable business processes
- Utilize pre-built adapters to back-end applications for faster integration



Personalized Experience = Customer Loyalty

Hundreds of pre-built processes based on industry best practices



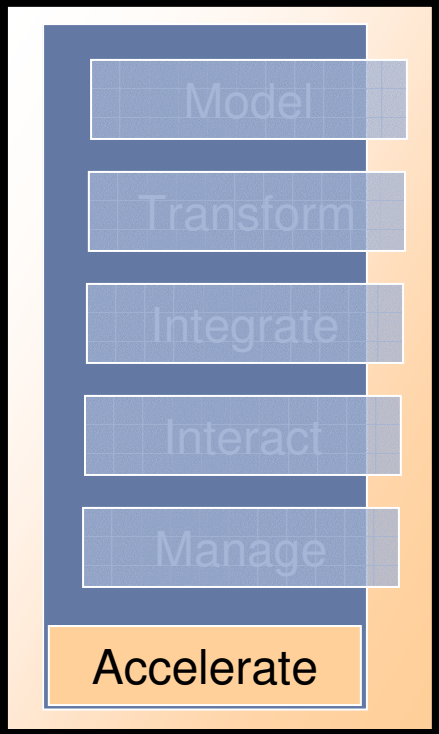
Accelerates implementation

Extended via process templates to meet Industry requirements

Standards Based

Easily adaptable to technologies or business need

The path to SOA



WebSphere

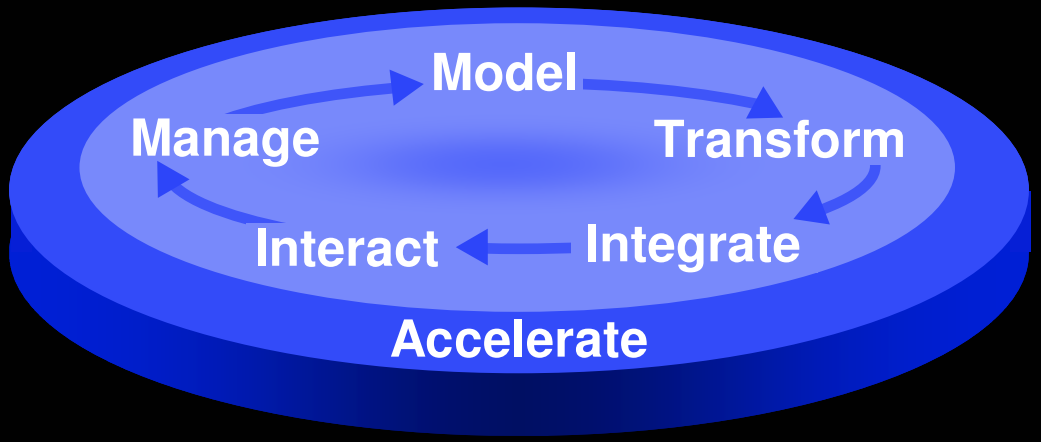
WebSphere:

- Pre-Built Portlets
- Process Templates
- Adapters
- Commerce Accelerators

DB2

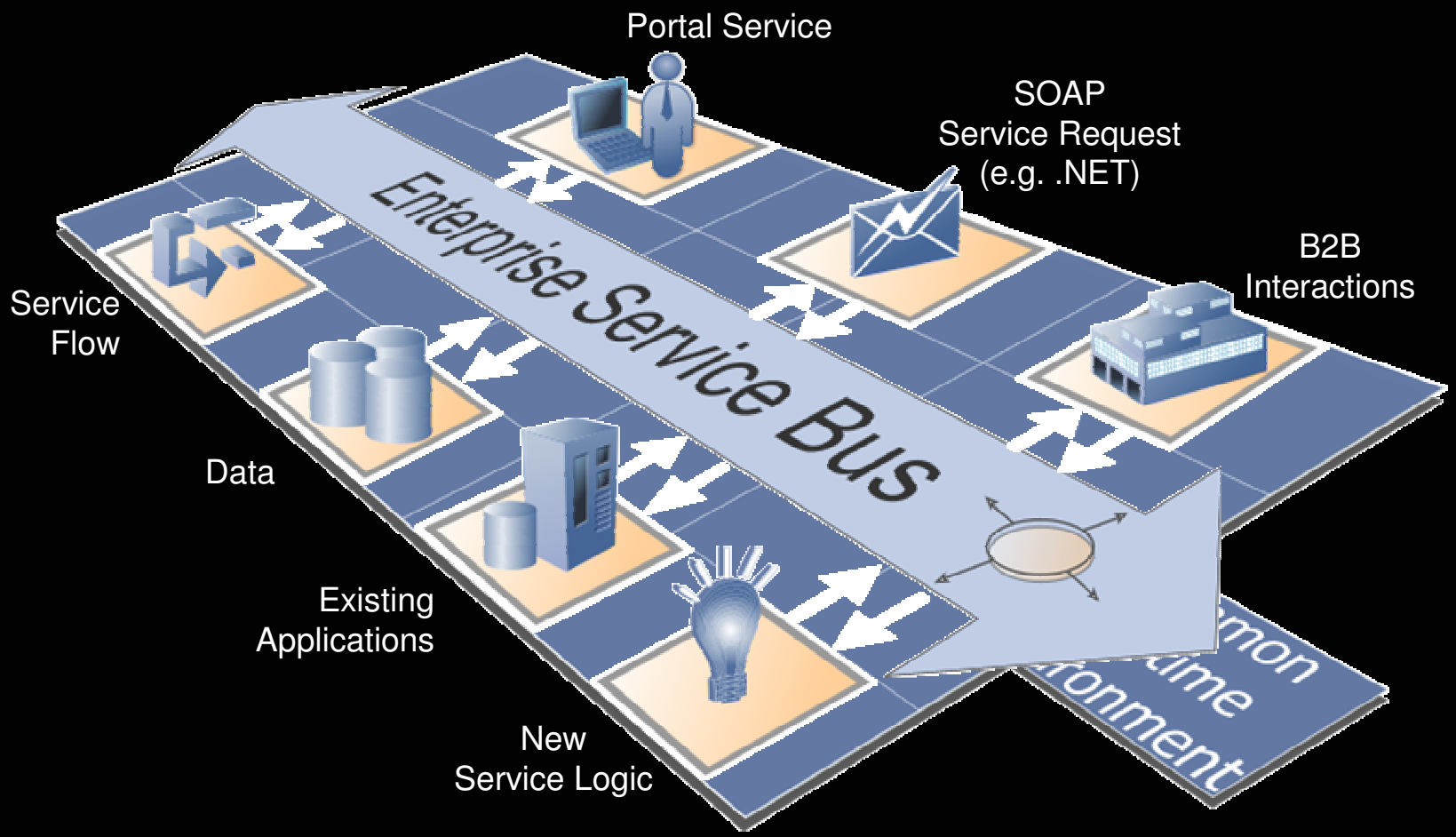
Information Integrator

IBM Business Integration Capabilities

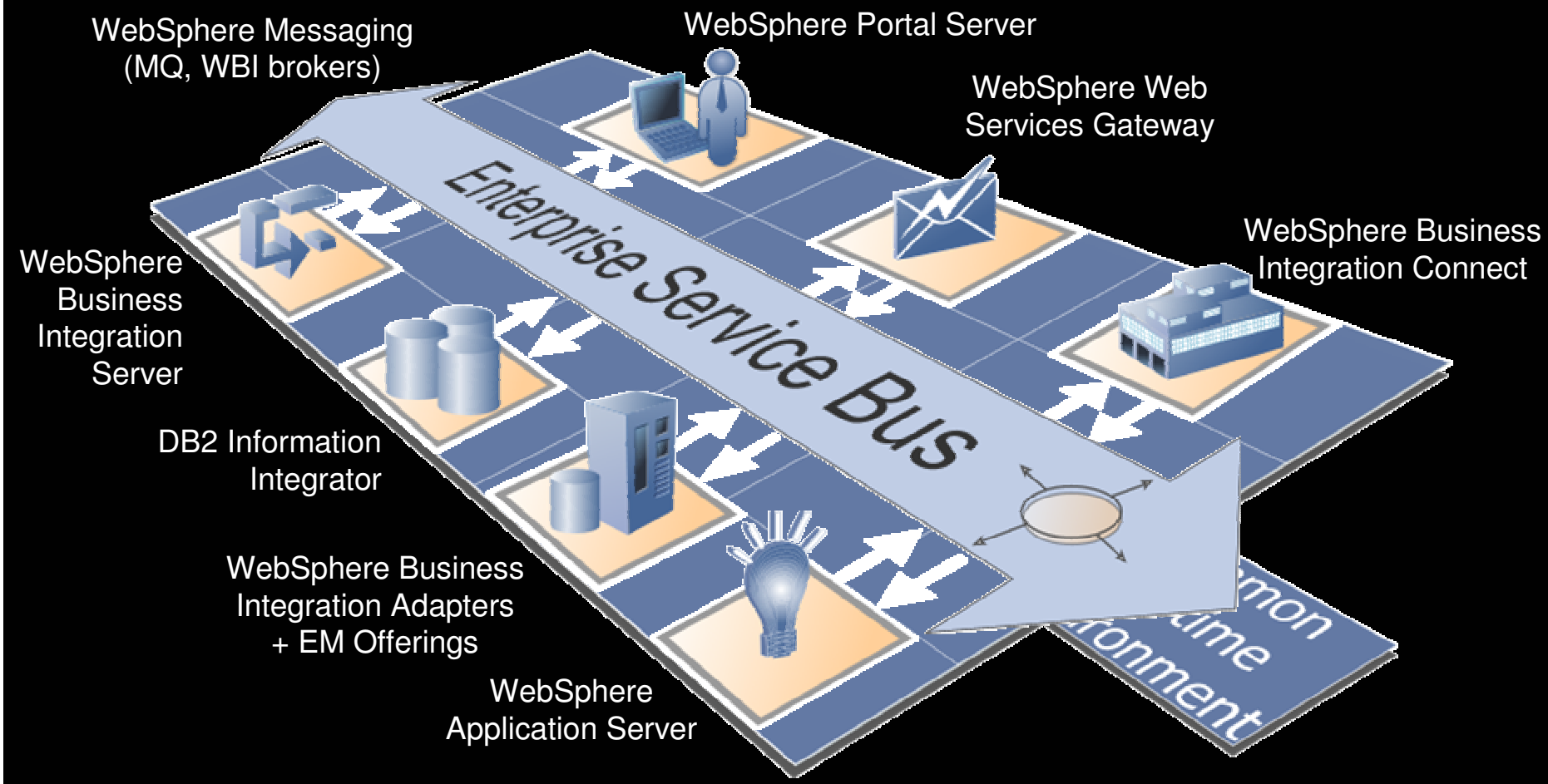


- **Model** business functions and processes
- **Transform** applications, processes and data
- **Integrate** islands of applications, processes and information
- **Interact** with resources anytime, anywhere with any device
- **Manage** performance against business objectives
- **Accelerate** the implementation of intelligent processes

Service Oriented Architecture



Service Oriented Architecture and WebSphere



IBM Business Integration Qualities

- **Simple Integrated Development**
  - Common tools platform
  - Re-use and unification of assets
- **Secure & Scalable Deployment**
  - Common and flexible deployment environment
  - Flexible management & security infrastructure
- **Proven Experience**
  - Risk mitigation
  - Improved time to value
  - Augmentation with best practices
- **Standards Leadership**
  - Interoperability
  - Investment protection
  - Portability - Freedom of choice