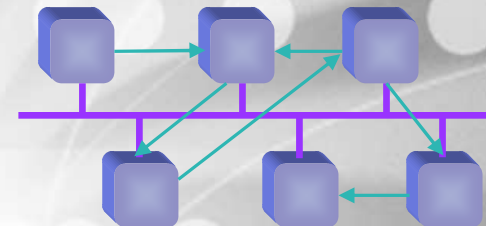




IBM Systems & Technology Group

SOA and System z The Perfect Match

Kristoffer Stav
IT Specialist, System z
krisstav@no.ibm.com



ON DEMAND BUSINESS™

© 2006 IBM Corporation

Agenda

- **Key concepts of SOA**
 - Objectives, terminologies, web services, steps to SOA, IBM SOA lifecycle
- **System z and SOA**
 - Why System z and z/OS as the "SOA hub"?
 - The WebSphere software product family

Service Oriented Architecture (SOA) helps customers increase the flexibility of their business processes, strengthen their underlying IT infrastructure and retain and reuse their existing assets. z/OS is THE platform of choice where the main enterprise assets reside. This session will define the basics of SOA and will highlight the solution available under z/OS.

What is Service Oriented Architecture (SOA)?

... a service?

A **repeatable business task** – e.g.,
check customer credit;
open new account

... service orientation?

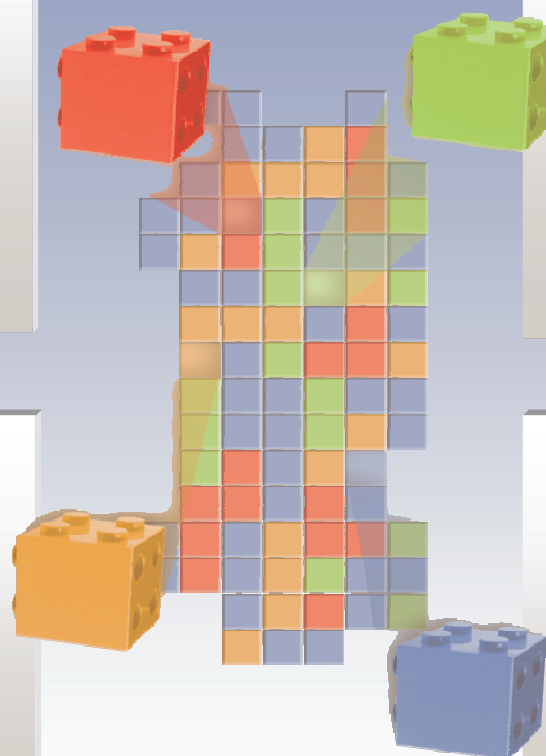
A way of integrating your
business as linked services
and the outcomes that
they bring

... service oriented architecture (SOA)?

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

... a composite application?

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



What is SOA?

- **Service oriented architecture is a composition model**
- **Service**
 - a business task / function that are invoked through well designed and standardized interfaces
 - Hides implementation details
 - For example: OS function, customer written business logic, an application, etc.
- **Interface neutrality**
 - Uniform, universal, platform independent, programming language independent

Service

- **Expose a well defined interface**
- **Hide their implementation details**
- **Are callable through open standard mechanisms**
- **Can be coarse grained, fine grained, or somewhere in the middle.**
 - A coarse grained service is one that exposes a very high level business function that, when invoked, calls many other internal services. (think of this as a main program that calls subroutines for example)
 - A fine grained service is one that implements a very specific function and only that function (think of this as a atomic subroutine with a specific purpose)
- **Exhibit either request/response or “fire and forget” behavior.**

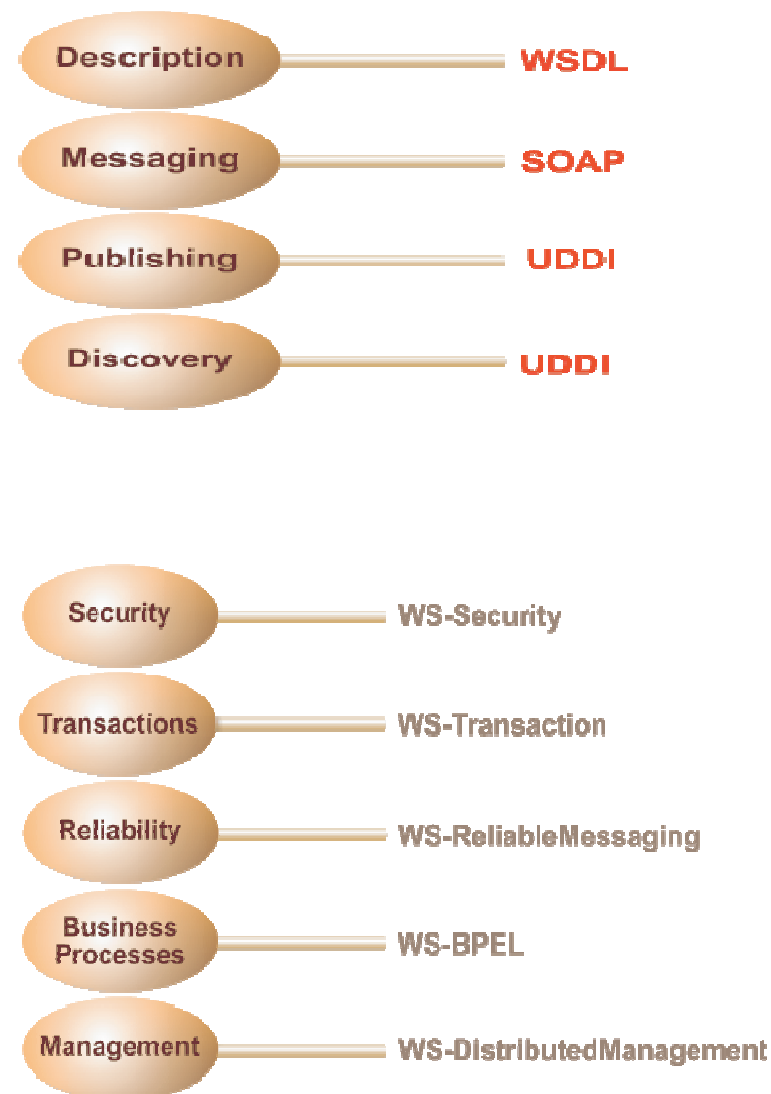
Why SOA?

- **Standards**
 - The standards and technology are finally in place, with broad industry support
 - The necessary software to get started is available today
- **Loosly coupled**
 - Services are loosely bounded and dynamically linked
 - Platform independence, language independence
 - Change internal workings of a service does not effect its relations to other service components – not a new idea.
- **Reusability and flexibility**
 - Integrate new functionality with existing assets in a cost effective manner
 - Extend lifetime of existing applications by making it easier to integrate with new functionality.
- **Reduce the gap between the "techies"/developers and users/requestors**
 - Make application development more business oriented, align development process to business needs
- **Tight link between business and technology**
 - Not pure technology focus, but driven by business goals
 - SOA unites business and IT (as opposed to IT alone defining the design)

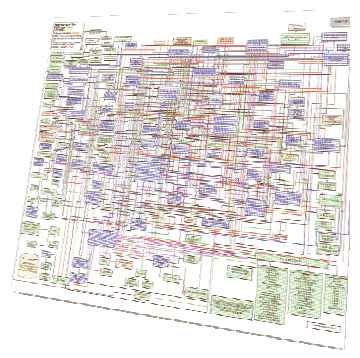


Web Services and SOA

- **SOA: a concept and a set of design patterns**
 - For componentization
 - Flexibility
- **WebServices: a set of standards which an entire industry is agreeing on**
 - Software components described via WSDL that are capable of being accessed via standard network protocols such as SOAP over HTTP
 - Message exchange is neutral - XML (SOAP)
 - Standards for defining service interfaces expressed in XML (WSDL) and exposed
 - Interoperability
- **Web Service is an example of a Service**
 - Standardization through neutral interfaces as regards to hardware, OS, and programming language
 - Advanced Web Services standards support security, transactionality, reliability, business process execution, and management
- **SOA does not require Web Services**
 - But a collection of Web Services may form an instance of SOA



SOA: the next step on the evolution of enterprise integration

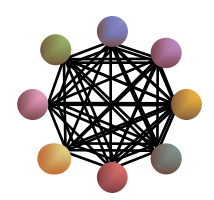


Direct Connectivity

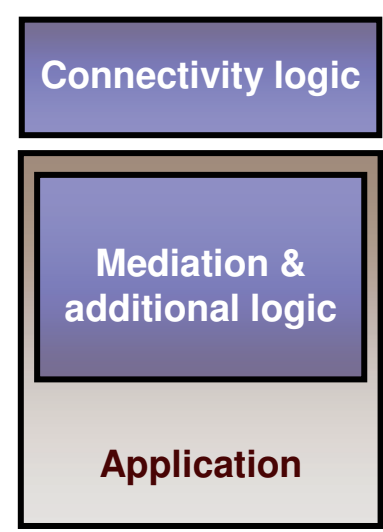


All connectivity, mediation & additional logic buried in the application

Point-to-Point connection between applications

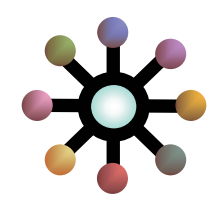


Message Queuing

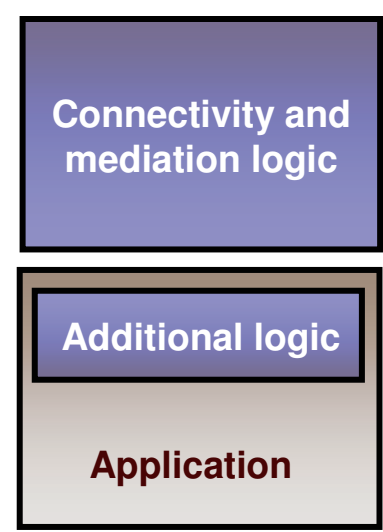


Abstracts the connectivity logic from the application

Applications via a centralized hub

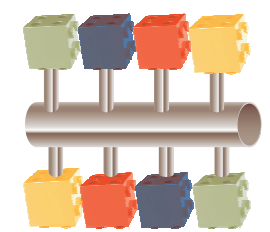


Message Brokering

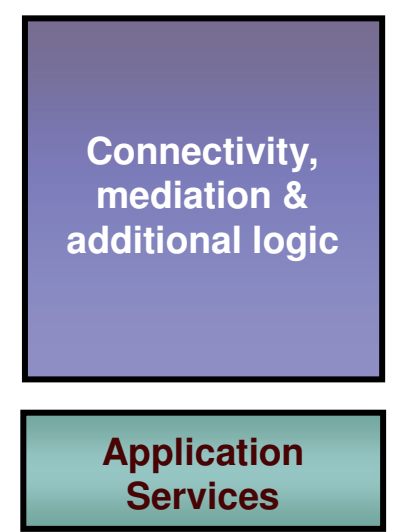


Abstracts the connectivity & mediation logic from the application

Integration and choreography of services through an Enterprise Service Bus



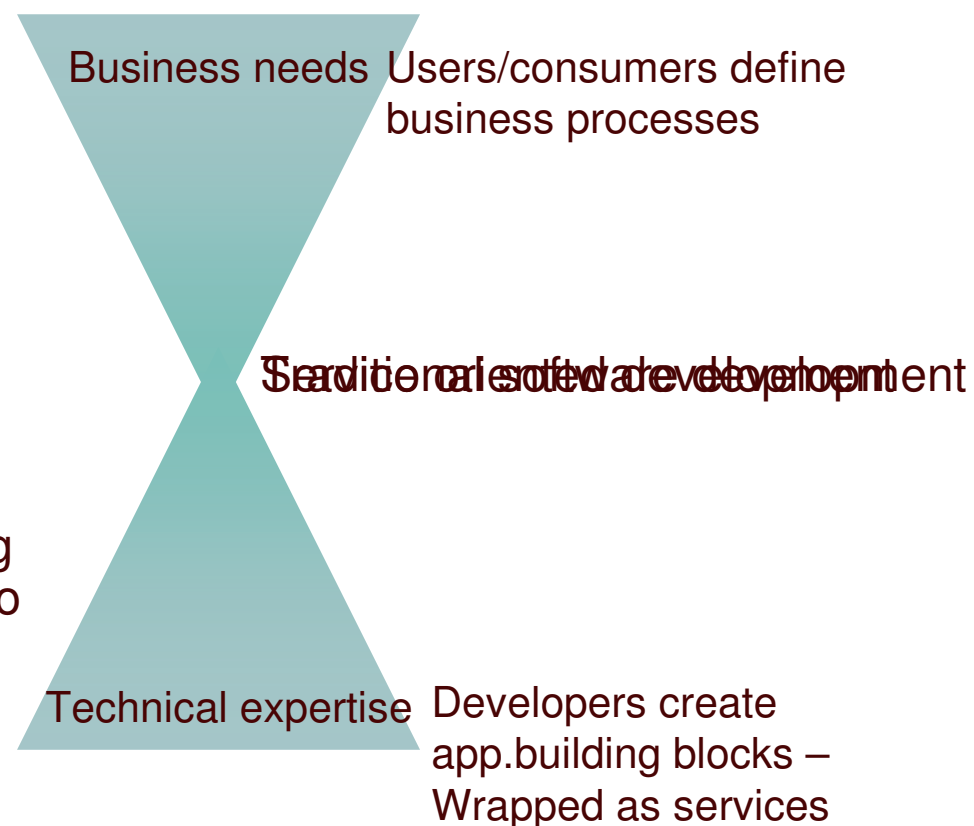
Service Orientation



Reduces application to its core business functions

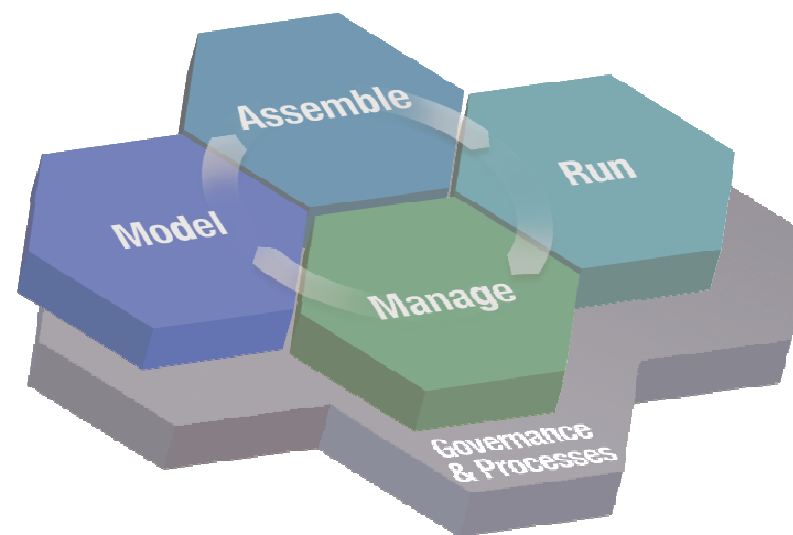
Technical expertise meets business needs

- Enables an unprecedented degree of business flexibility
- Enables faster creation of new business processes and composite applications in response to changes in the environment
- Increases operational efficiency by focusing attention on what is core and differentiating – not what is simply operational
- Improves constituent service by providing services people can use without having to worry about the underlying IT infrastructure (at the business level)



The SOA Lifecycle

- **Model**
 - Understand the business needs and business environments
 - Identify the components that make up business services
 - Design, model and simulate
- **Assemble**
 - Discover and extend
 - Construct and test
 - Compose
- **Deploy**
 - Integrate people
 - Integrate processes
 - Manage and integrate information
- **Manage**
 - Manage services
 - Manage identity and compliance
 - Monitor business metrics

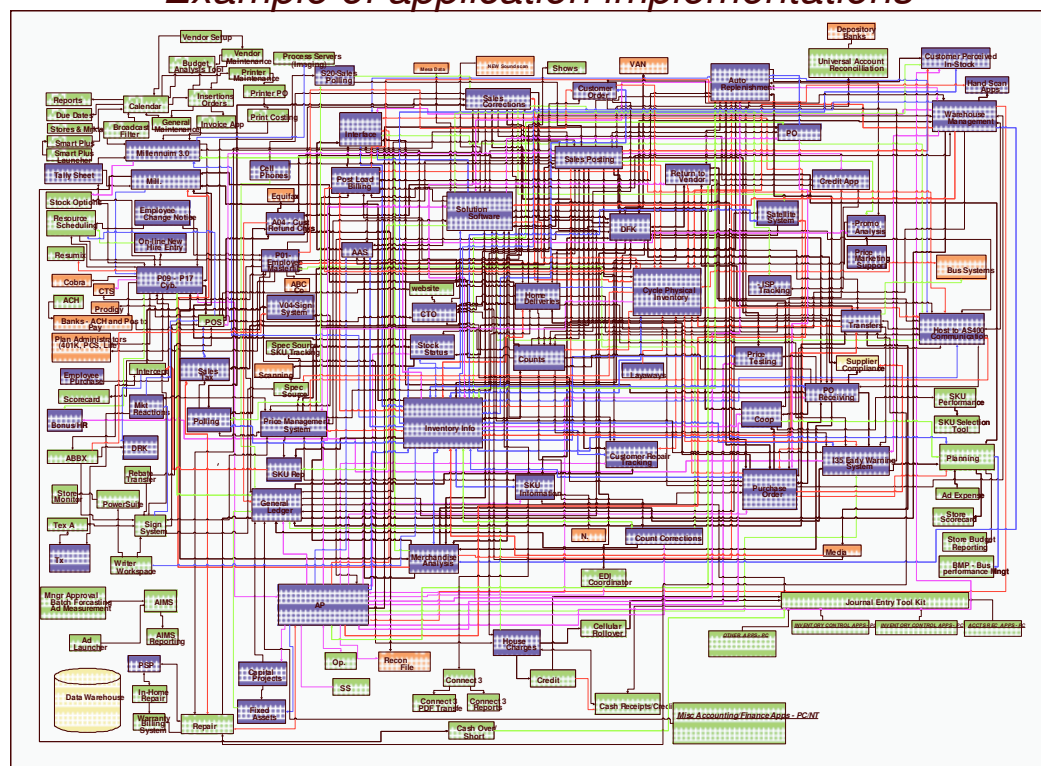


IBMs SOA Lifecycle

Typical application topology

- Point to point interfaces
- Monolithic design
- Complex infrastructure
- Not very flexible
- Maintenance is expensive and difficult
- Application silos

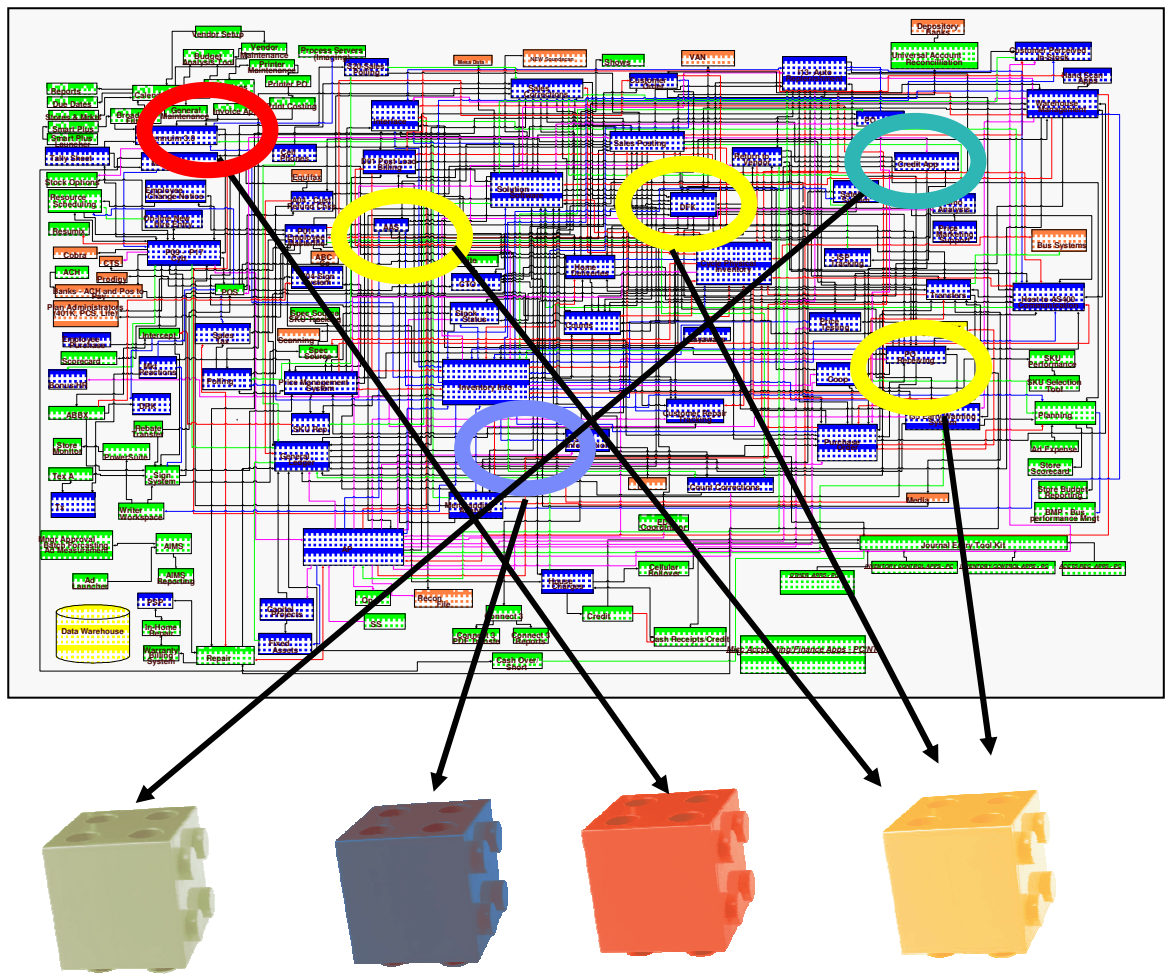
Example of application implementations




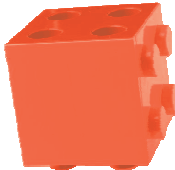
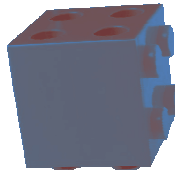
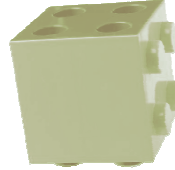
Reference: *Software Strategies, Enterprise Integration Challenge: Vendor Middleware-based Integration Solutions Offer Major Advantages Over Custom In-house Solutions, 2nd Edition, April 2005*

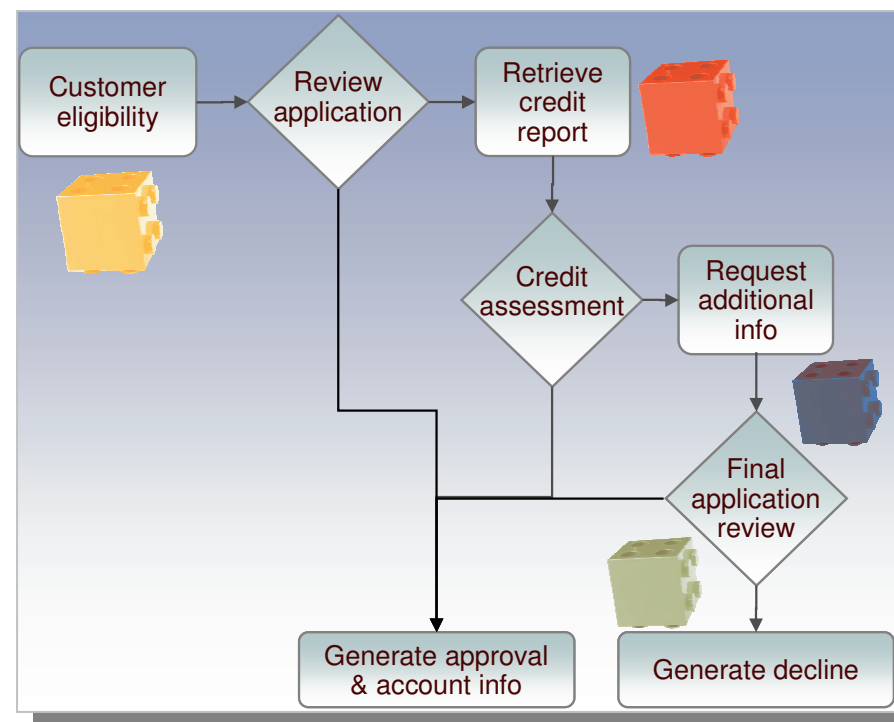
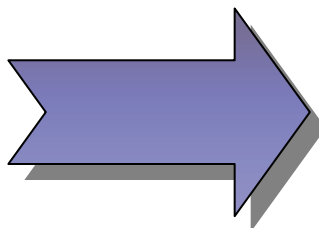
Create service components of existing application components that can be reused

- Step 1**
Identify the Business Service – the basic SOA building block
- Step 2**
Locate the service components
- Step 3**
Construct the interface
- Step 4**
Repeat



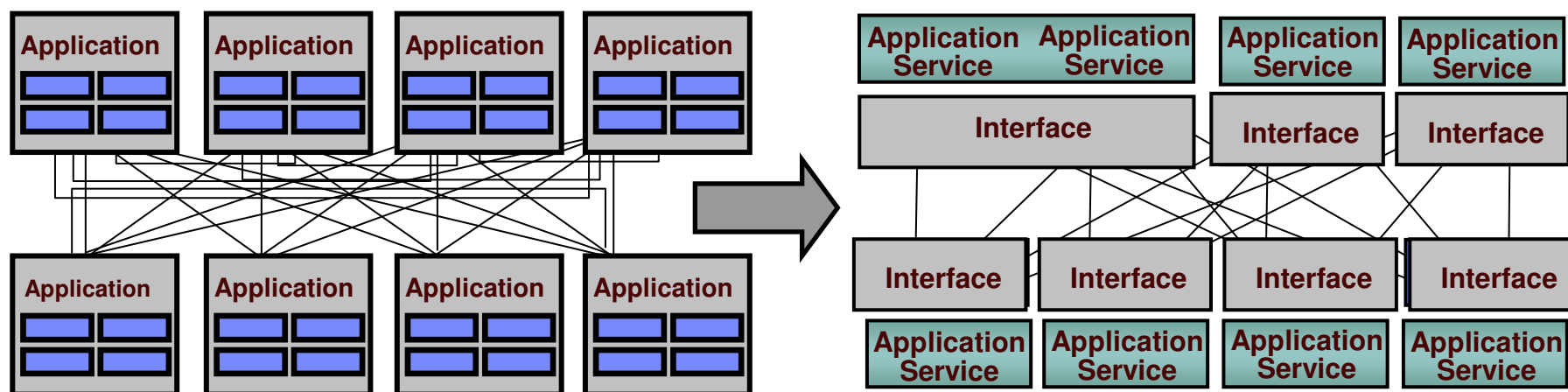
Integrate these services to create business processes

-  **Determine Customer Eligibility**
-  **Retrieve Credit Report**
-  **Request additional info**
-  **Generate decline**
- Etc....**



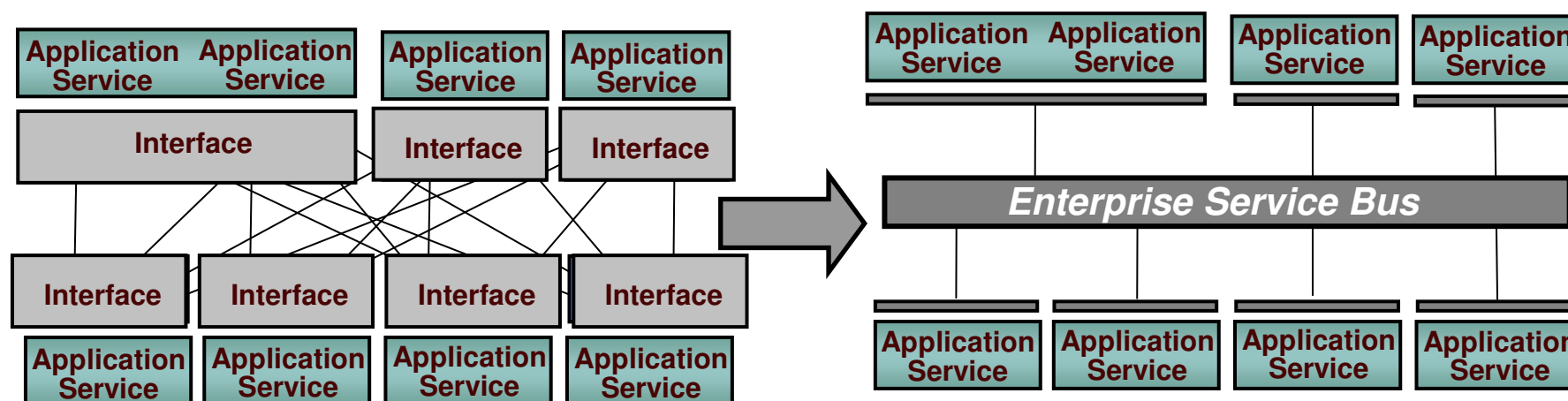
The application(s) is now rendered as services
Business Process is implemented by integrating services

From traditional applications to services

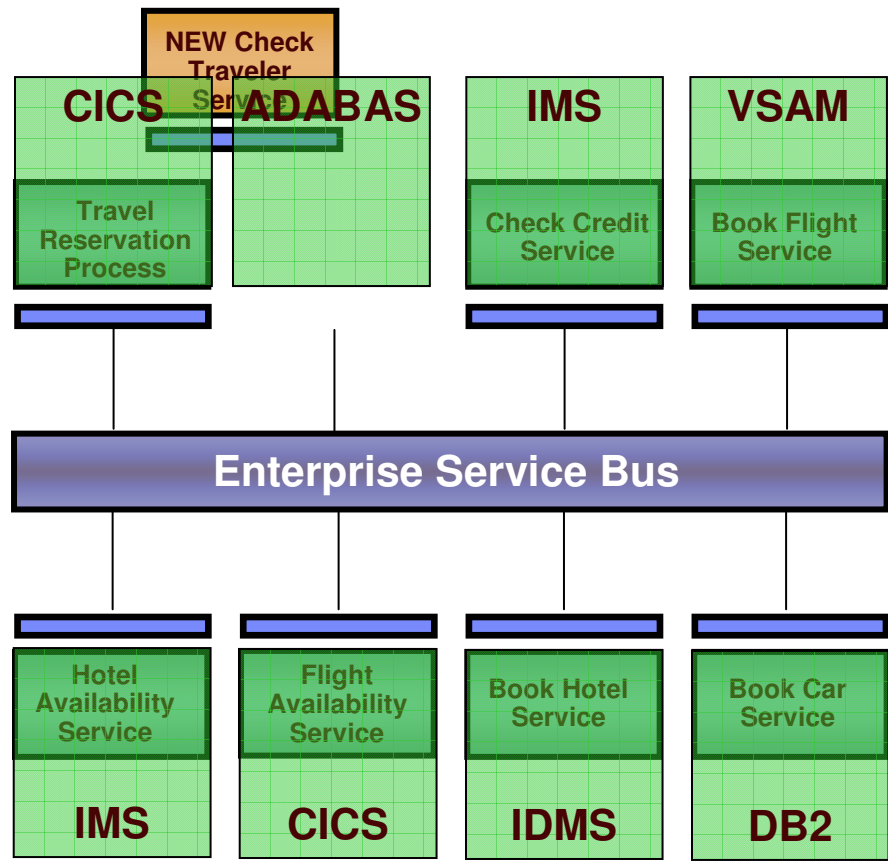


Interconnect the services via an Enterprise Service Bus

- Eliminate separate connection points



SOA lets you focus on core business, not IT



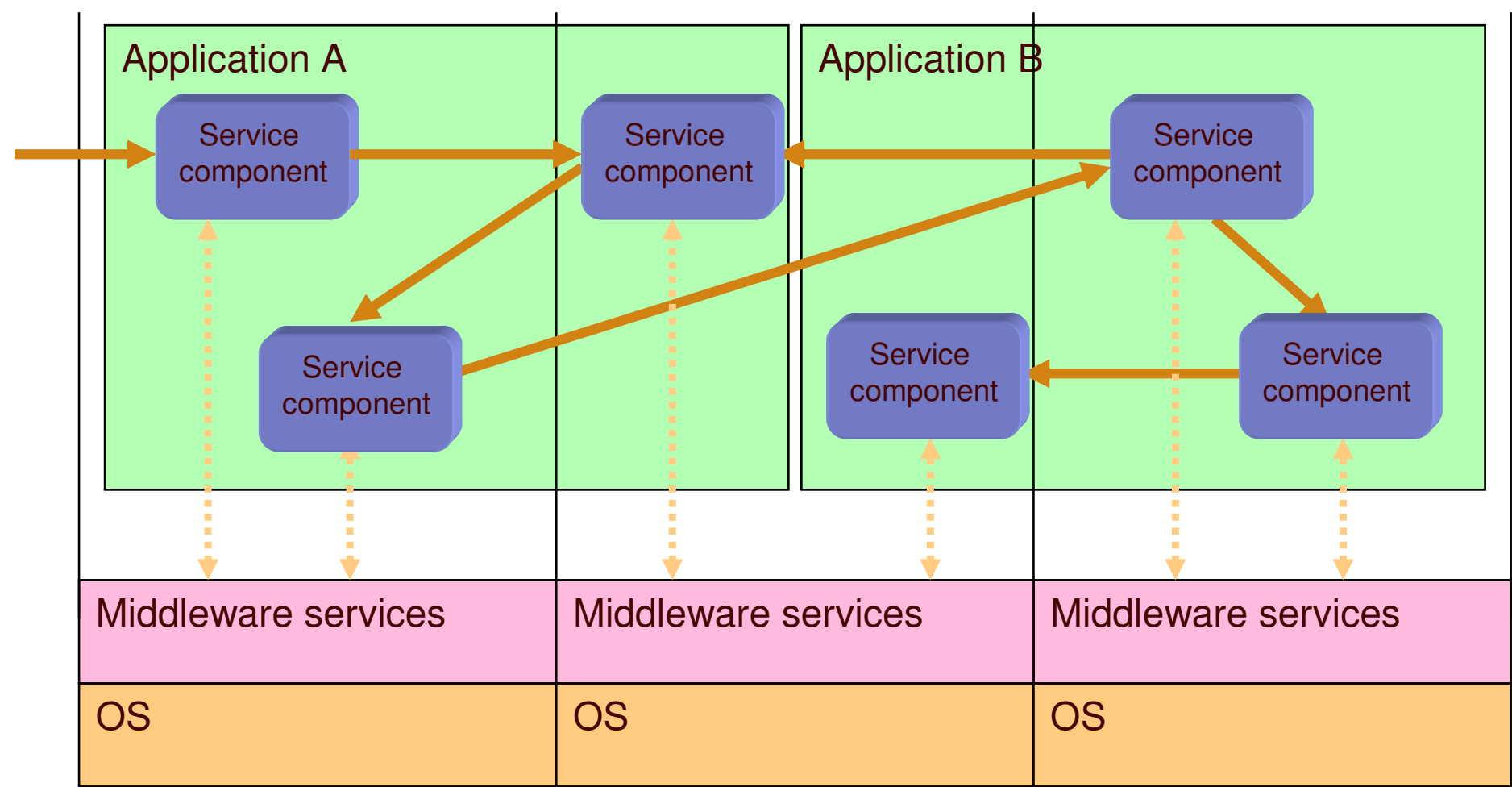
Add new services faster

Change services with minimal impact to existing services

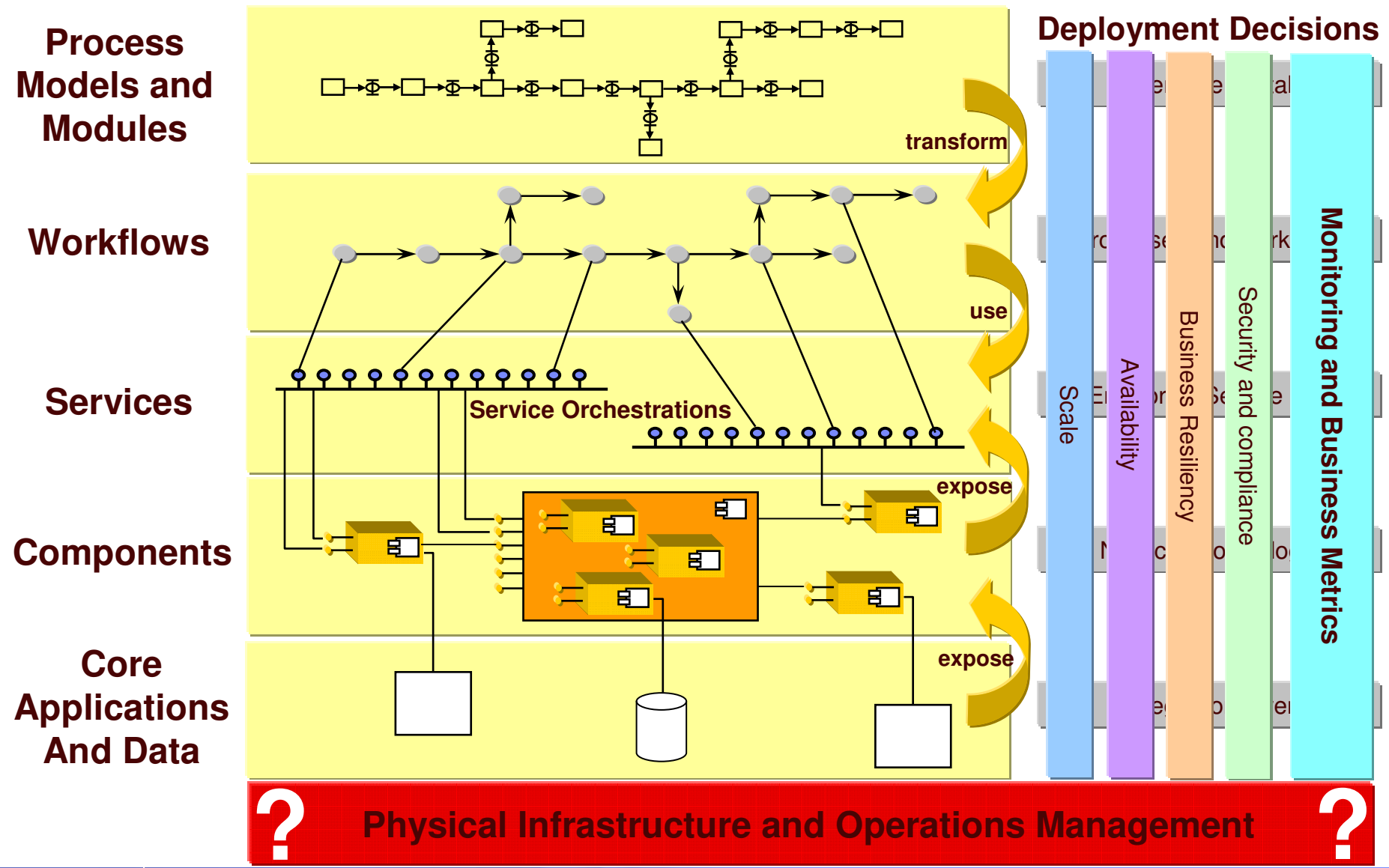
REUSE core z/OS resources in composite SOA service implementations!

NEW Flight Availability Service

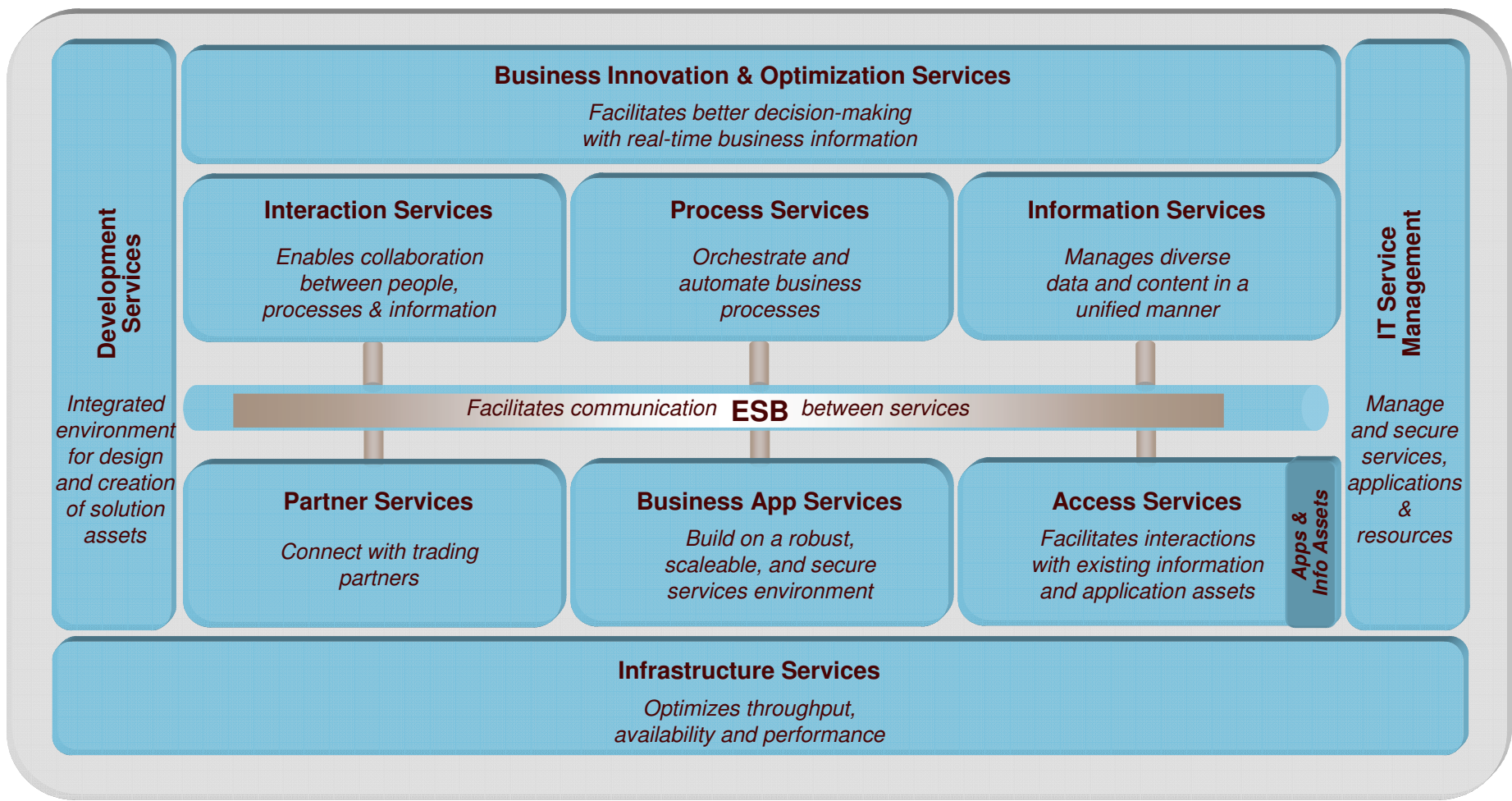
SOA architecture



Building and deploying SOA applications



The SOA Reference Architecture

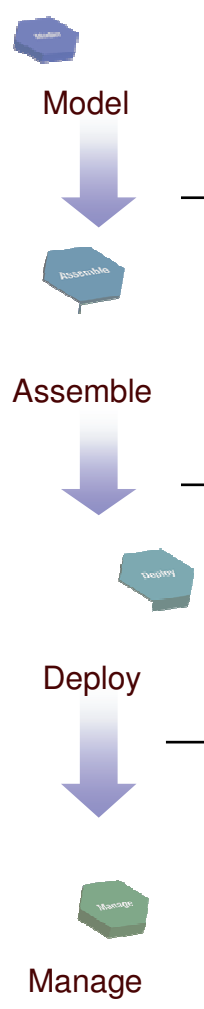


Leverage z/Middleware for maximum business flexibility.

Agenda

- **Key concepts of SOA**
 - Objectives, terminologies, web services, steps to SOA, IBM SOA lifecycle
- **System z and SOA**
 - Why System z and z/OS as the "SOA hub"?
 - The WebSphere software product family

Creating SOA composite applications with existing assets



Model a new business process that builds on your current capabilities

WebSphere Business Modeler



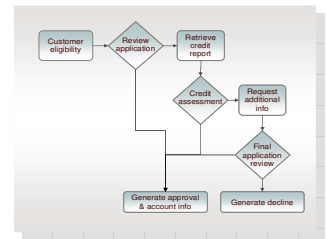
...and discover program units and business rules you can reuse in the new process.

WebSphere Studio Asset Analyzer



Wrap programs as services, creating composite appl'ns from core assets....

WebSphere Developer for zSeries, plus Service Flow Modeler



... and assemble the services across multiple platforms

WebSphere Integration Developer



Choreograph and deploy your new composite applications

WebSphere Process Server



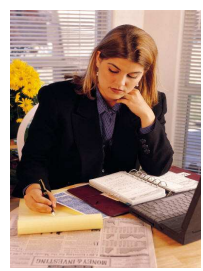
... using an advanced ESB to power your SOA

WebSphere Message Broker and/or WebSphere ESB



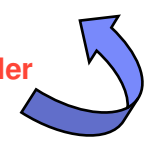
Monitor the processes across your SOA, and intervene if necessary

WebSphere Business Monitor

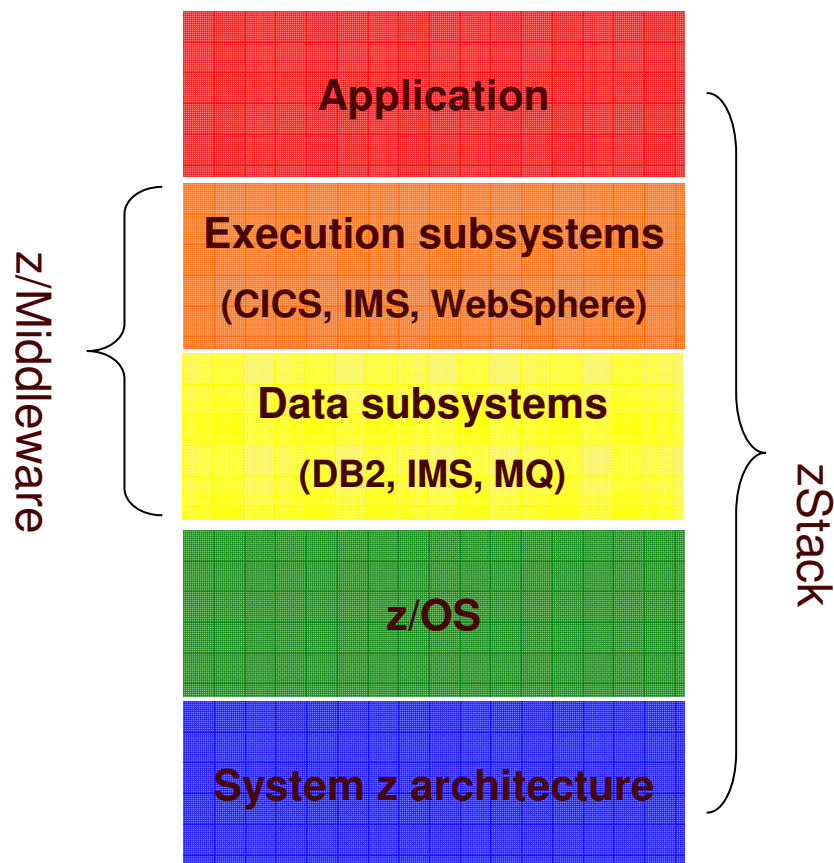


.... and export data for analysis and process improvement, back to

WebSphere Business Modeler



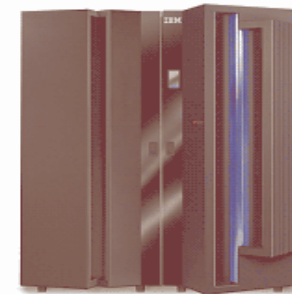
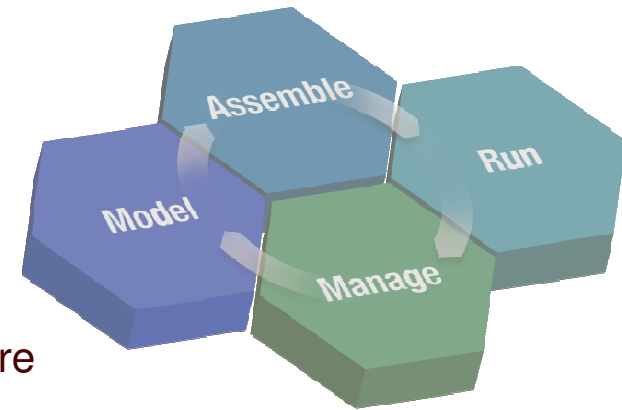
zStack Technologies Value



- **Not just a collection of technologies**
- **A \$100 billion dollar investment in an integrated stack ...**
- **Software and hardware *designed and optimized to work together* to achieve business objectives in demanding customer environments**
- **Deliver**
 - Rock-solid transaction processing
 - Never goes down
 - Unbreakable security
 - Ready to go
 - Operates at high levels of utilization
 - First class virtualization
 - Ready to run multiple workloads
 - Easy to scale up
 - Modern application development tools
 - Low Total Cost of Operation

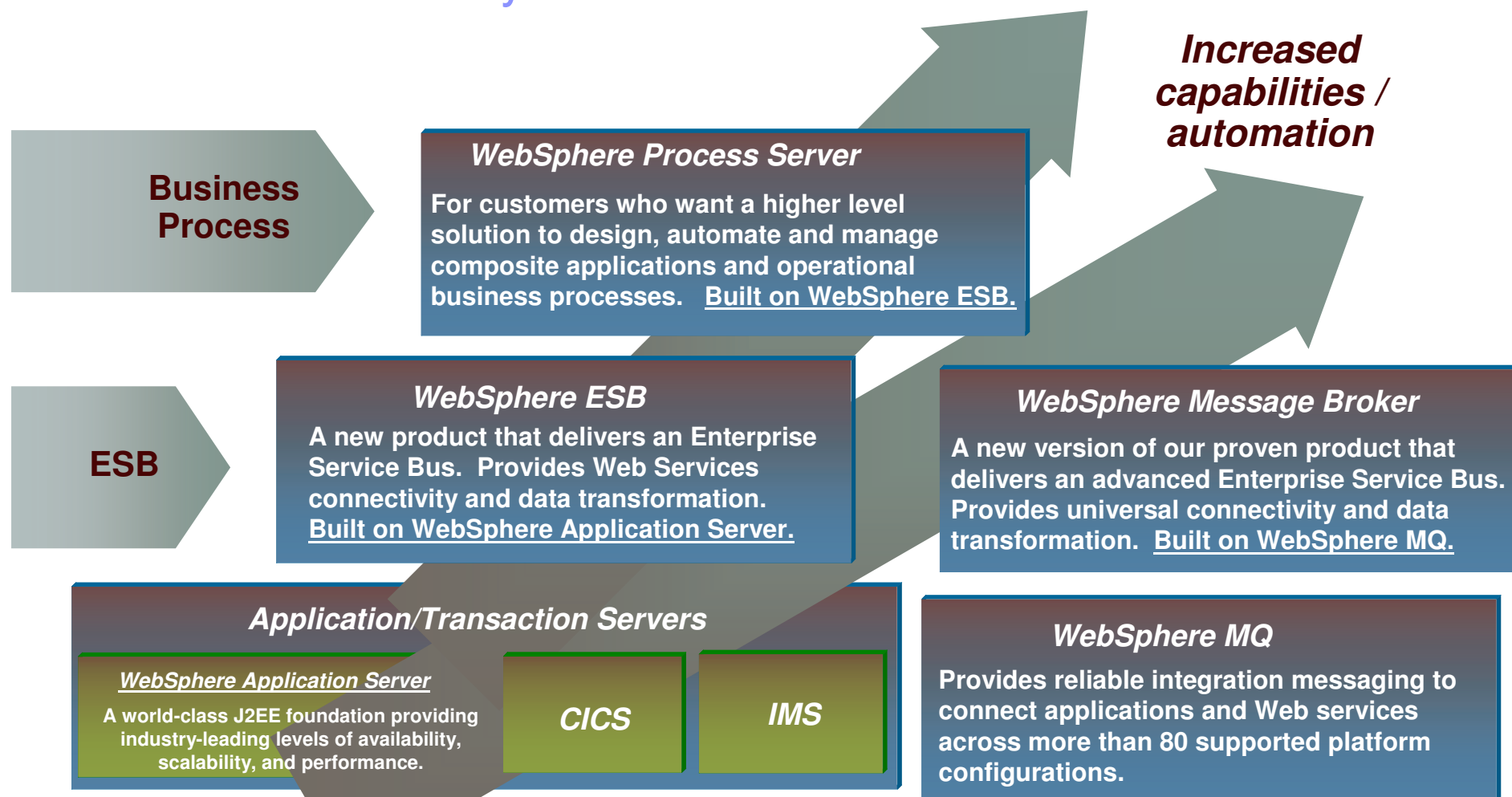
Why SOA on “z” ?

- **High availability for critical components**
 - Application Server
 - Used as server for new Java applications
 - Used as underlying component for other middleware
 - Enterprise Services Bus
 - Process Server
- **Highest security capabilities for critical components**
- **Centralized management**
- **Easier integration of core business assets**
 - “Unlock” existing assets using SOA
 - Applications
 - Optimized two-phase-commit flow
 - Data proximity
 - JDBC Type 2 driver versus JDBC Type 4 driver





SOA on z/OS – a complete solution from existing systems to the full SOA Lifecycle



WebSphere Application Server for z/OS V6

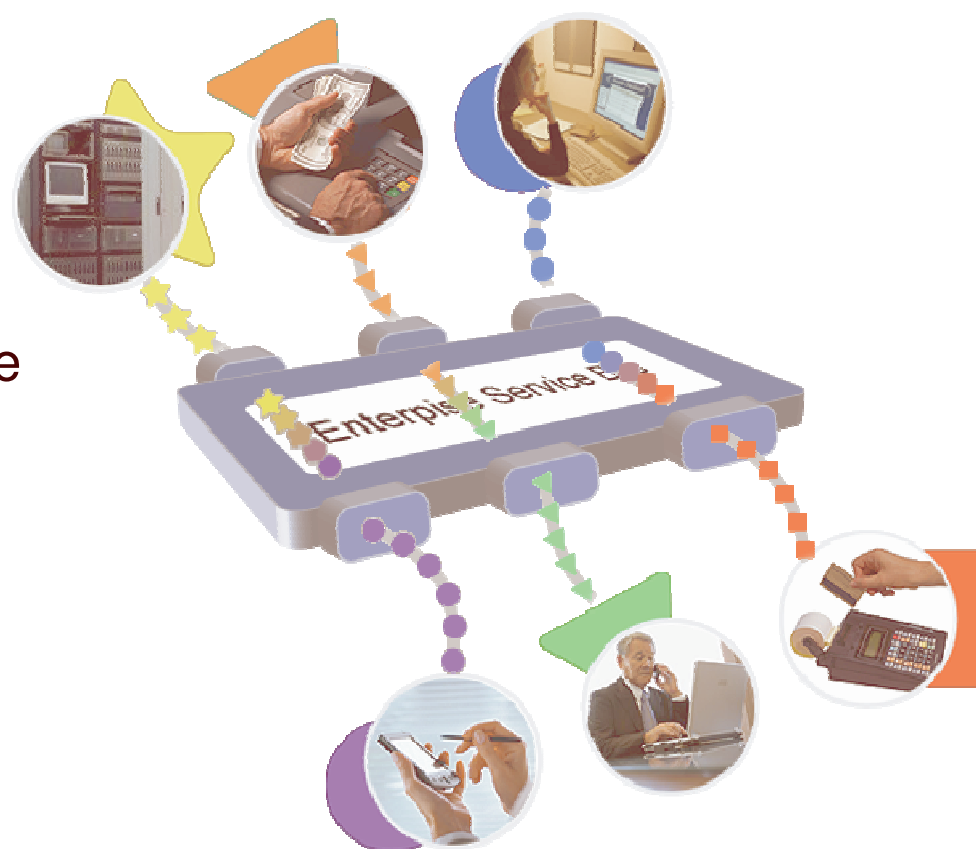
- **A Compliant J2EE Server**
 - Extend existing Java assets with support for Web Services standards and standards-based messaging
 - Certified J2EE 1.4
 - Web Services standard support (WSI, WSS 1.0, WS-Transaction, UDDI v3)
 - Help ensure 24x7 availability of business-critical applications with clustering and high availability
 - J2EE applications on different J2EE server can be ported to WAS z/OS
- **A Compatible WAS on the z/OS operating system**
 - Functionally equivalent to WAS Network Deployment (distributed)
 - Same WAS V6 common code
- **Development Tools**
 - Same ones than for WAS distributed - based on Eclipse
 - WebSphere Developer for zSeries (WDz) to unify COBOL, PL/I, Java development tools
- **WAS works natively with relational (DB2) data and hierarchical (IMS DB) data using the JDBC standard**
- **All are optimized and exploit zArchitecture for**
 - Performance, scalability, reliability and easy management

WAS for z/OS – Quality Of Services

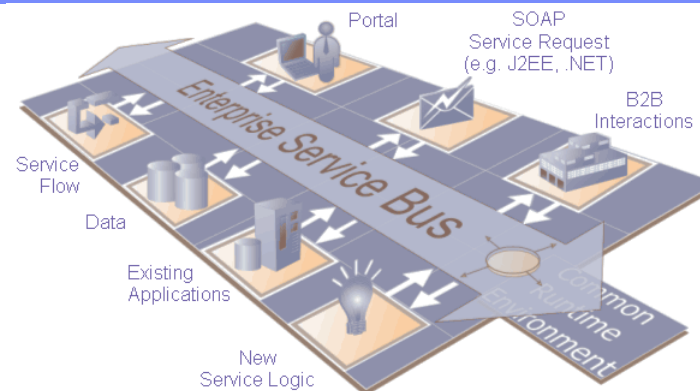
- **Workload Management**
 - WLM and Parallel Sysplex exploitation
- **High-availability**
 - Based on System z availability
 - Server clustering
- **Security**
 - Based on z/OS Security Server / RACF (SAF)
 - Security context passed between WAS applications and other z/OS applications
- **Transactional support**
 - Based on z/OS Resource Recovery Services (RRS)
- **Performance improvement**
 - Parallel Sysplex scalability
 - Use of JAVA specialized processors (zAAP) to improve TCO

The ESB is a core component of a Service Oriented Architecture

- ▶ that connects and integrates an organization's IT infrastructure, across many locations, using different transport services, and...
- ▶ that can route messages to the appropriate systems and transform messages formats, and...
- ▶ is standards based, and...
- ▶ does all this securely, reliably, at very high volumes in a manageable manner

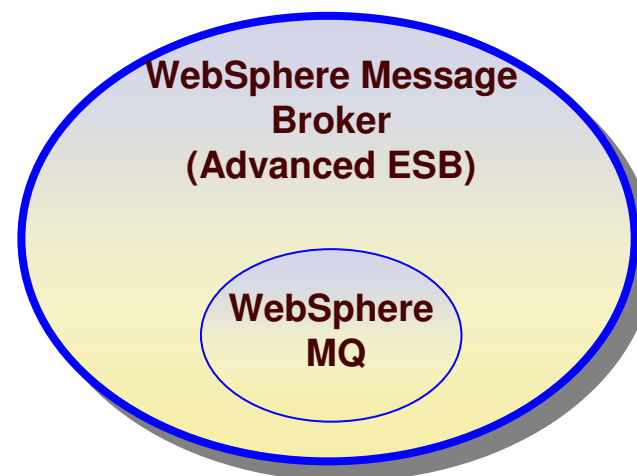
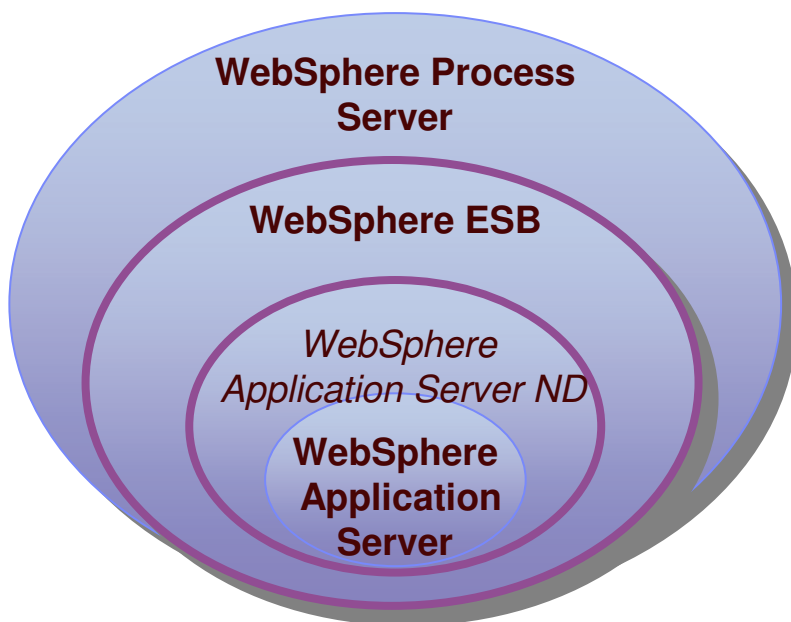


Enterprise Service Bus products



- **WebSphere Application Server V6.1 for z/OS**
- **WebSphere ESB for z/OS**
- **WebSphere Process Server V6 for z/OS**

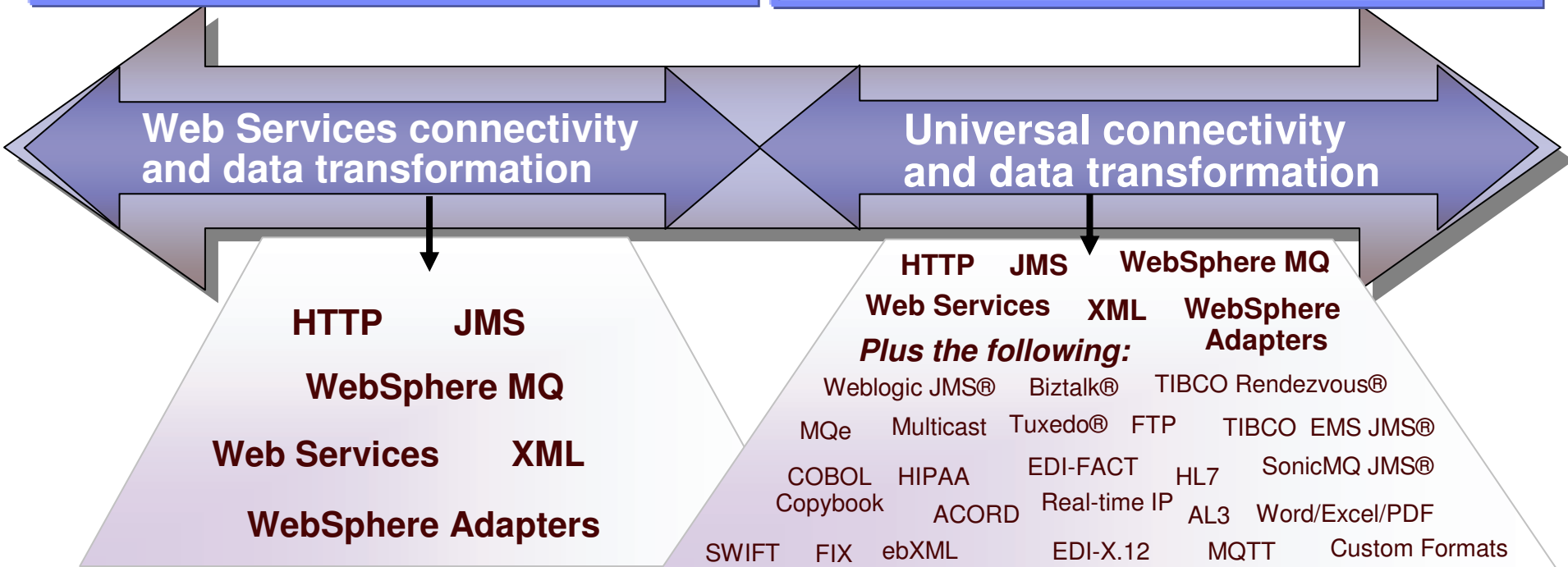
- **WebSphere MQ V6 for z/OS**
- **WebSphere Message Broker V6 for z/OS**



WebSphere ESB and WebSphere Message Broker

ESB:
WebSphere ESB

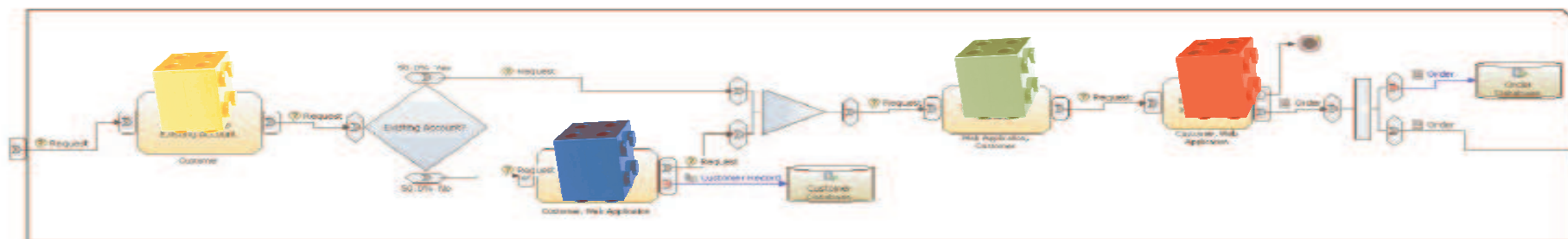
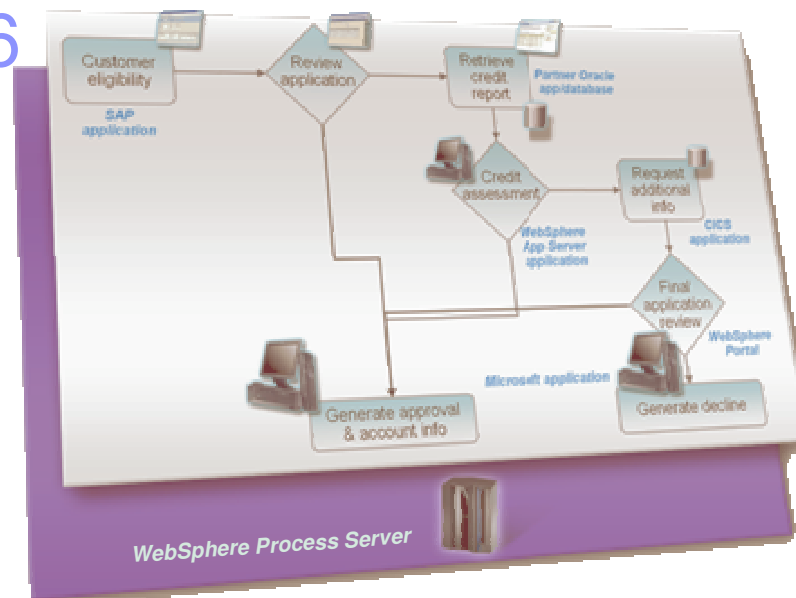
Advanced ESB:
WebSphere Message Broker



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

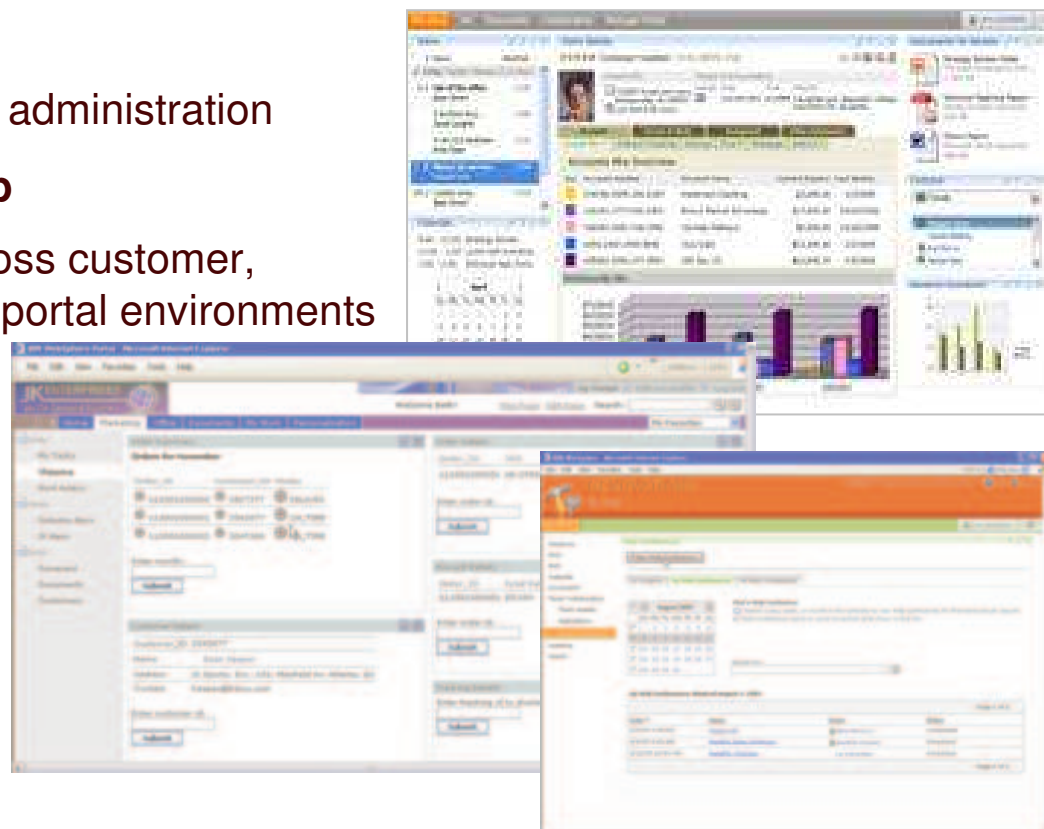
WebSphere Process Server V6

- **Based on WAS**
- **Used for deployment of composite applications**
- **A Single Process Server built on SOA**
 - Reliable, scalable, secure, open standards (WS-BPEL, SCA ...)
 - Single integrated runtime for all SOA based process automation
- **Support all aspects of process integration**
 - A single process integration platform
 - process flows, business rules, human steps, services, state machines
- **Rapidly change process behavior to keep pace with business requirements**
 - Reuse existing services that you already have and create new services for future use
 - Build process flows without knowing where the information is coming from (late binding of services)
 - business rules control the execution sequence of the process and can change dynamically



WebSphere Portal V5

- **Deliver Collaboration, Mobility, & Composite Applications**
- **Ease of use**
 - Now supporting WebSphere Process Server V6 for process-driven portal capabilities
 - Ease of management
 - Virtual portals and policy-driven administration
- **Reduced total cost of ownership**
 - Reuse of SOA components across customer, employee, and partner/supplier portal environments

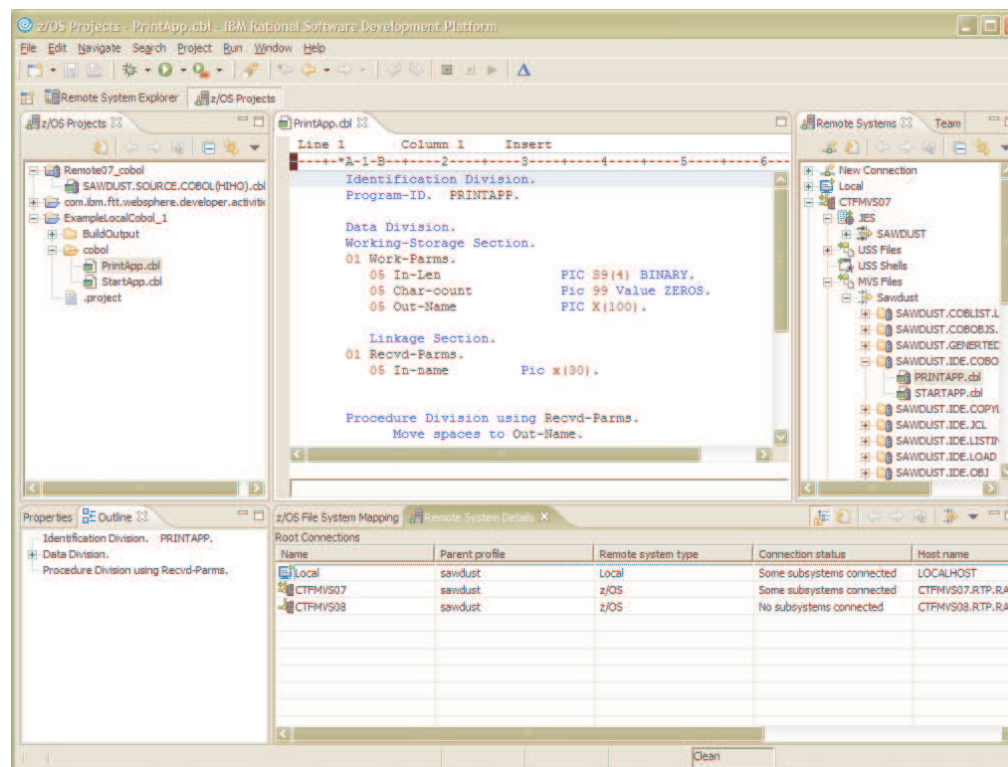


WebSphere Developer for System z

- **Bridges mainframe and client-server development**
 - For end-to-end integrated processes running across platforms
 - Shared developer repositories and tooling framework

- **For deployment to WebSphere, CICS, IMS, DB2 and batch**
 - Cross platform development
 - Workstation based development tool
 - Workstation based mainframe application analysis tool

- **Develop SOA systems that combine Web services, web applications and traditional applications**
 - COBOL, PL/I, EGL (4GL)
 - Java, J2EE, WSDL, Java Beans, XML adapters
 - Portlets and complex user interfaces
 - Service flow modeler for CICS and HATS



Business needs are driving a renaissance in mainframe interest and use

<p>The mainframe is obsolete</p>	<p>Enterprises are rediscovering mainframe values. IBM is seeing a resurgence of interest and use of the mainframe. The ability to efficiently deliver core services - without interruption – is critical to becoming an On Demand enterprise.</p> <p>IBM has invested \$bb of dollars to rejuvenate the platform. You will find most of the latest hardware & software technologies on the mainframe.</p> <p>The platform protects and enhances 40 years of investment in application assets. Our customers tell us they want this indefinitely.</p> <p>Competitors now wish their servers were more like the IBM mainframe.</p>
<p>The mainframe is too expensive</p>	<p>The mainframe delivers higher utilization, lower overheads and the lowest total cost-per-user of any platform.</p> <p>Mainframe delivers huge value through its leading security, availability and recoverability capabilities.</p> <p>You may be surprised at the affordability of new mainframe application projects</p>
<p>I can't get the skills</p>	<p>There is a resurgence of interest in the mainframe amongst younger IT professionals, driven by a recognition of career opportunities.</p> <p>IBM is sponsoring major initiatives in universities and colleges to facilitate the development of mainframe skills.</p>

References / sources

- Based on "SOA and z/OS – The perfect match" by Hélène Lyon, z/Middleware Technical Specialist
- IBM WebSphere Application Server for z/OS Version 6: A performance report, by Mike Cox, et al.
- WebSphere and System z9, Powering the On Demand Business, presentation by David McCorkle, solutions architect, WebSphere for z/OS.
- <http://www-03.ibm.com/systems/z/soa/>
- Enterprise Service Bus - Leadership in the Marketplace, presentation by Russ Newcombe in IBM et al.