



WebSphere Software

# SOA and the Enterprise Software Platform: transforming your enterprise applications

TPF Users Group - 2006

Jim Rhyne  
IBM DE & Enterprise Software Platform Architect

**ON** DEMAND BUSINESS™

© 2006 IBM Corporation

# Agenda

- The Enterprise Application Environment and SOA
- SOA Enabling Existing Applications
- Component Architectures and Integration
- Questions and (maybe) Answers

# Solving Top Business Challenges

*Creating or Improving IT Solutions by Leveraging the SOA Foundation*

## Productivity



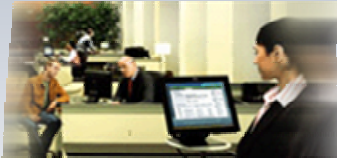
*Easier change management*

## Agility and compliance



*Single source of information*

## Collaboration



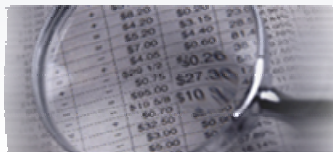
*role-based access for all*

## Customer experience



*Tasks provided by others*

## Responsiveness and service



*Reduce transaction time*

## New opportunities



*Expand access to apps and systems*

# What is SOA?

IBM views SOA as a holistic relationship between the business and the IT organization. SOA encompasses the tools and methodologies for capturing business design, and using that design information to help *improve the business*. It also encompasses the tools, programming model and techniques for implementing the business design in information systems. It encompasses the middleware infrastructure for hosting that implementation. SOA encompasses the management of that implementation to ensure *availability to the business*, and to ensure *efficient use of resources* in the execution of that implementation. It encompasses the establishment of who has authority and the processes that are used to *control changes* in the business design and its implementation in the information system. ***And ultimately, SOA accelerates the time-to-value for these benefits***

<http://www-128.ibm.com/developerworks/webservices/library/ws-soa-whitepaper/>

# The IBM SOA Foundation

*Provides What You Need to Get Started with SOA*

*IBM SOA Foundation: Integrated, open set of software, best practice, and patterns*

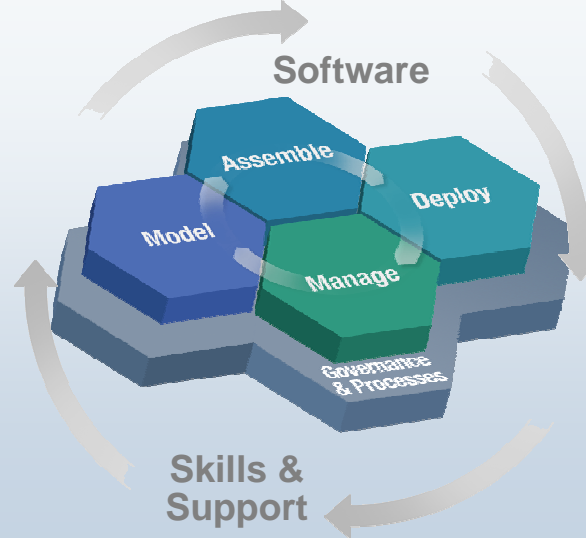
Supports complete lifecycle with a **modular** approach

**Extends value** of your existing investments, regardless of vendor

**Scalable**; start small and grow as fast as the business requires

Extensive business and IT standards support; facilitating greater **interoperability & portability**

## IBM SOA Foundation



Leveraging existing IT Infrastructure



CICS



IMS



Custom Apps.

## Your Existing Applications Assets are the Foundation of an SOA Strategy

*Many of the IT assets required to enable an on-demand business already exist and have been supporting the business for years or even decades. Enabling these IT assets to participate in integrated business processes is key to improving responsiveness.*

DH Brown Associates, Inc.

*Application Transformation: Leveraging Existing IT Assets to Build Competitive Advantage (September 2004)*

# What about “before SOA”?

- **Significant business intelligence exists in core systems**

- ✓ "200 Billion lines of COBOL code in existence" *eWeek*
- ✓ "5 Billion lines of COBOL code added yearly" *Bill Ulrich, TSG Inc.*
- ✓ "2 Million COBOL developers" *Gartner*
- ✓ "Majority of customer data still on mainframes" *Computerworld*
- ✓ "Replacement costs \$20 Trillion" *eWeek*

- **Rewriting - is it an option.....**

- ✓ Who will do it? (who has the business knowledge?)
- ✓ How much will it cost?
- ✓ Risk?
- ✓ How long will it take? (lose strategic benefit)

## Rewriting all existing applications and moving them to new platforms is not a viable option

Existing applications are among the most valuable assets a company owns

It is 5X less expensive to reuse existing applications than to write new applications from scratch -- Software Productivity Research (SPR)

Reusing proven, time-tested applications results in significantly lower risk and faster time to market

Host applications are probably better suited for exposure as part of an SOA than many applications based on more modern 4GL object-oriented languages -- Forrester Research

We will help you identify an incremental, affordable migration path to extend and integrate your existing applications



# How does SOA technology accelerate time to value in applications?

- It decreases complexity by providing a uniform way of linking services and a uniform framework for finding and integrating them.
- It replaces static linkage of services with dynamic linkage and reduces the resistance to change, allowing IT to track changes in business processes
- It provides reuse and encourages consistency
- It simplifies operations by providing a uniform way to monitor and manage services
- It simplifies service implementations by handling resource and other management tasks in service containers

# Enterprise Environments

- Lots of pre-SOA software
- Quality of Service is very important
  - ✓ Performance
  - ✓ Resource utilization
  - ✓ Security
  - ✓ Robustness and Reliability
- High degree of interdependence among software elements
- Continuity

# IBM Middleware Appliance Strategy...

Appliances make sophisticated software consume-able...

## Appliance Definition

- ✓ Purpose Built Hardware Product
  - Delivered as a “box”
- ✓ Combines SW/HW
- ✓ Provides High Value Functionality
- ✓ “Drops In”  
and Virtually Maintains itself
- ✓ Delivered to and operated  
by the masses

## Benefits

- Consumable
  - ✓ Easier deployment and  
ongoing operations
- Simpler
- Cheaper
  - ✓ Primary focus on operating cost,  
not acquisition cost
- More secure



*...DataPower fits this mold...*

**Appliance Strategy and Strong SOA Requirements Lead to...**

## The Datapower attraction...

- Consumable
  - ✓ Appliance form-factors
  - ✓ Comprehensive SOA functionality
  - ✓ XML-centric programming model
  - ✓ Hardened / Tamperproof
- Wire Speed
  - ✓ Leverages XML compilation technology
  - ✓ Accelerated via hardware (XG4)
  - ✓ Mitigates XML/security performance penalties
- New business model
  - ✓ Creates new customer opportunities
  - ✓ Integrates well with other IBM offerings
  - ✓ Paves the way for additional appliances

XI50 Integration Device



XS40 XML Security Gateway



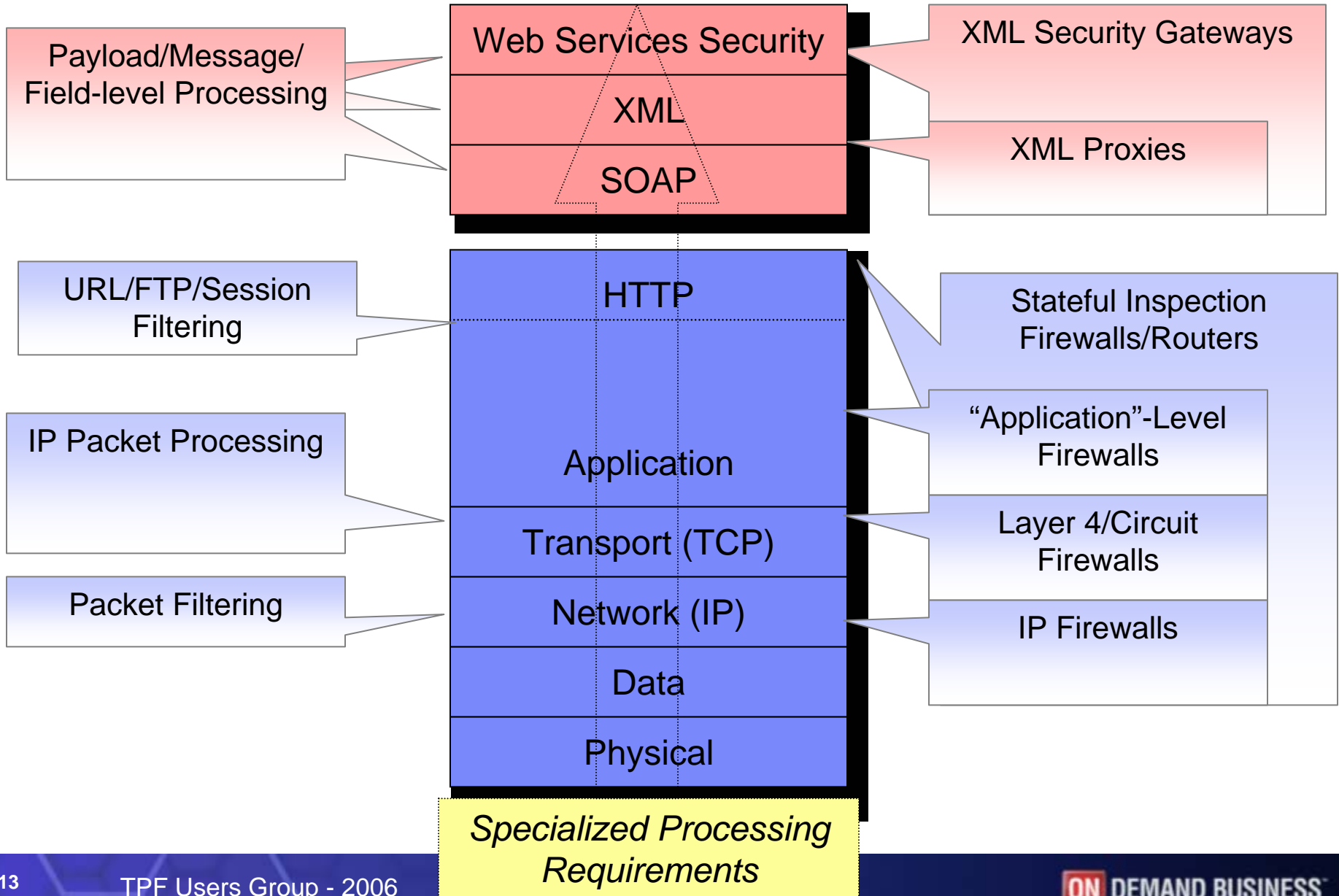
XA35 XML Accelerator



### Functionality

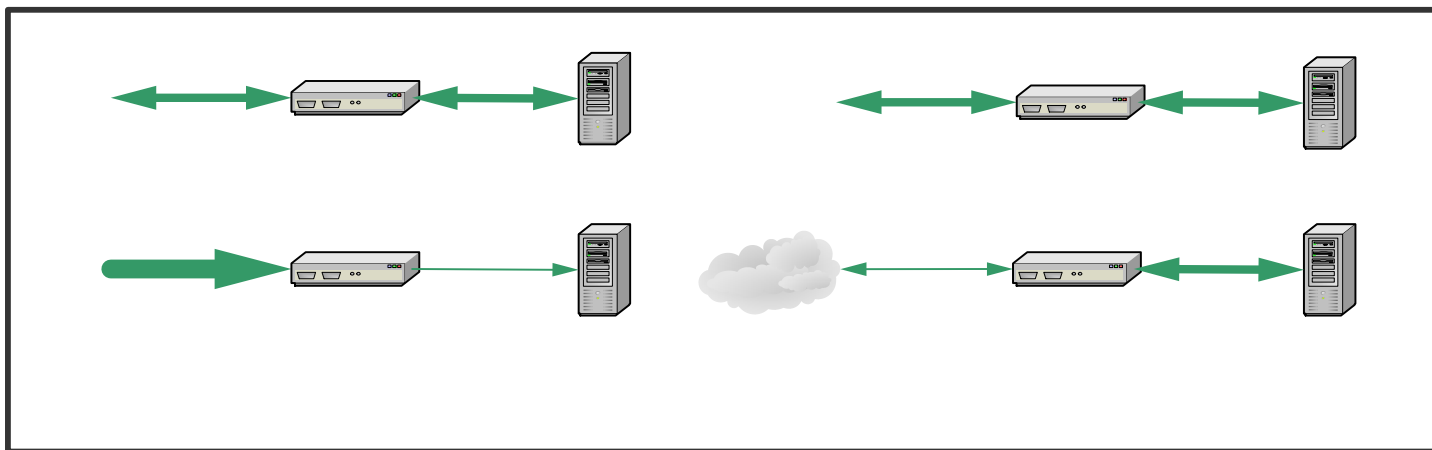
### Network Security Stack

### Infrastructure



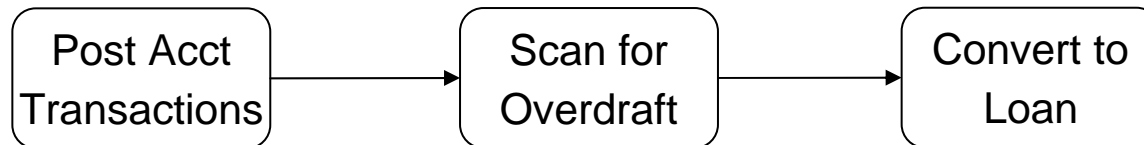
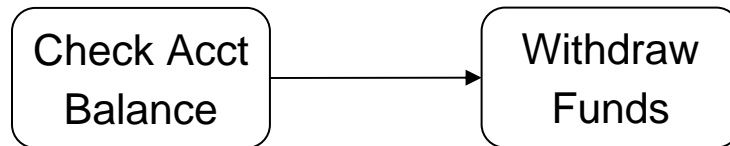
## Datapower as an XML Accelerator

- Offload XML Processing to dedicated Network Hardware
- Schema Validation
- XML Transformation, XSLT
- Compression
- Format Conversion
- XML Parsing / Object Marshalling Acceleration



## SOA Application Component Structure

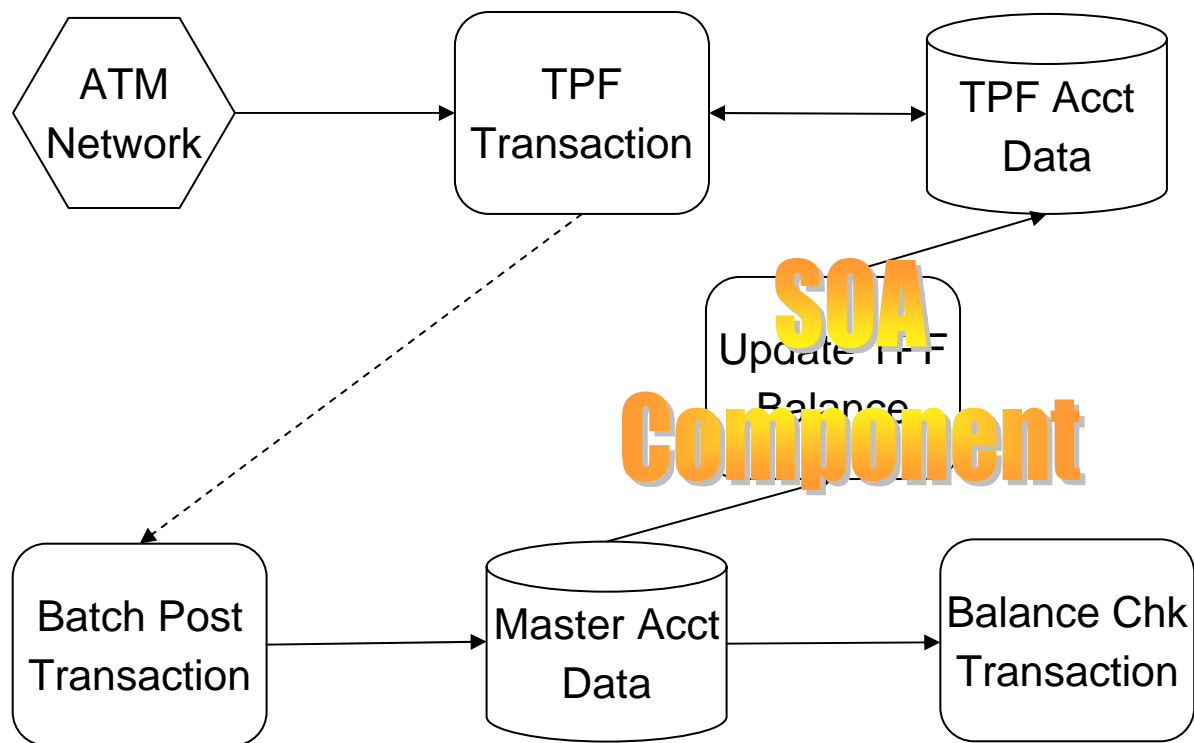
# Business Scenarios



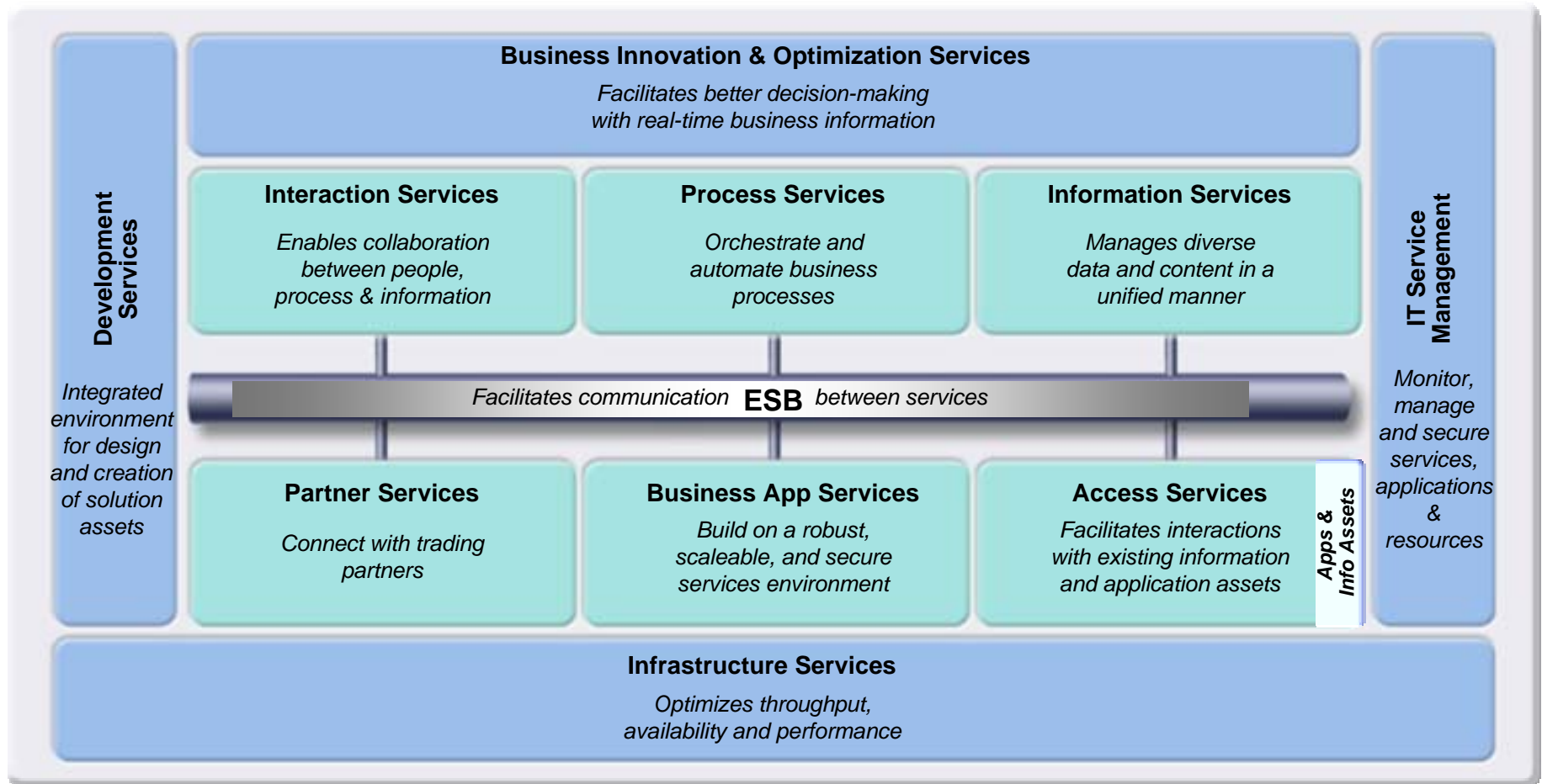
- These scenarios occur at different points in time, hence the balance seen in the first scenario is not the balance seen in the second. Customer may inadvertently overdraw account.
- Business opportunity comes from being able to offer consistent view of balance.



# Technology view



# SOA Reference Architecture



# Logical SOA Service Types

Enduser  
Service

- Enduser service is responsible for presentation and interaction and includes the user

Workflow  
Service

- Workflow service controls the sequencing and coordination of tasks involving one or more endusers

Coordination  
Service

- Coordination service is responsible for coordinating interactions with multiple services. Typically multithreaded and asynchronous and may require compensation

Process  
Service

- Process service contains synchronous logic (state machine, procedure, rules) controlling other services that fail or complete

Mediation  
Service

- Mediation service transforms data in linkage between other services

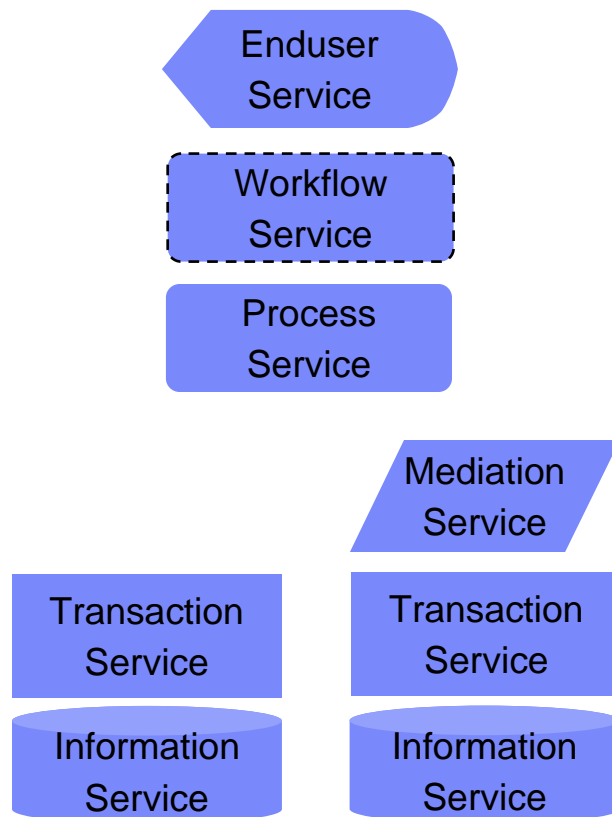
Transaction  
Service

- Transaction service is single threaded logic that completes or fails. Typically does not wait

Information  
Service

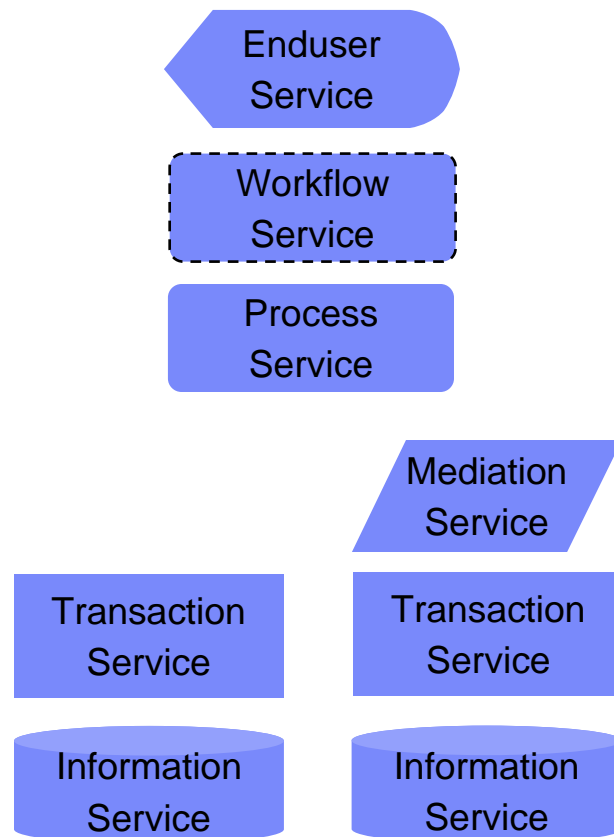
- Information service manages data (CRUD operations) and may wait or cause delays before producing results

# An Old Application



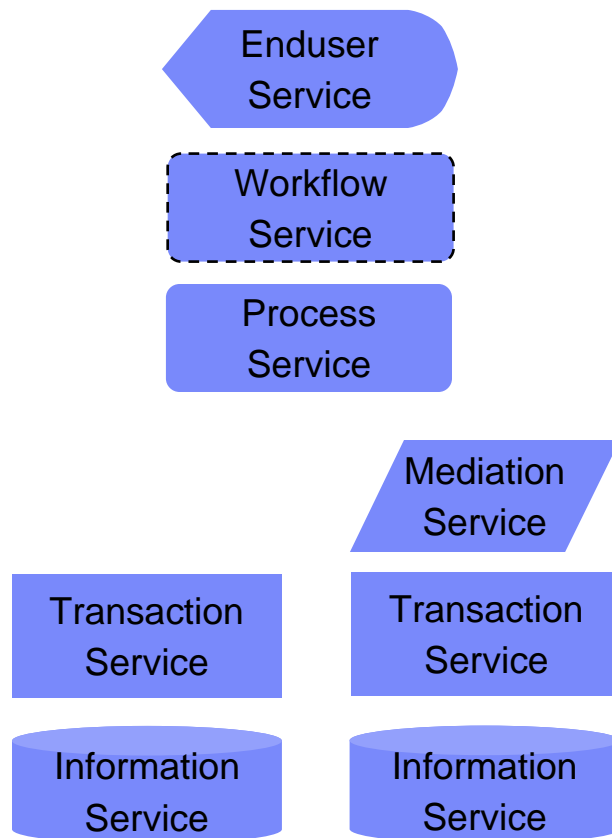
- The application assembly is statically bound within the application (the red rectangle)
- A subassembly is statically bound within the lime rectangle
- The application developer wrote some mediation code to adapt the subassembly interface. Typically this involves copying and reshaping data.
- The enduser service is controlled by a workflow process, and both are statically bound within the application
- **Making any change requires opening and redeveloping the application**

## An Extreme Example of SOA (WS Binding)



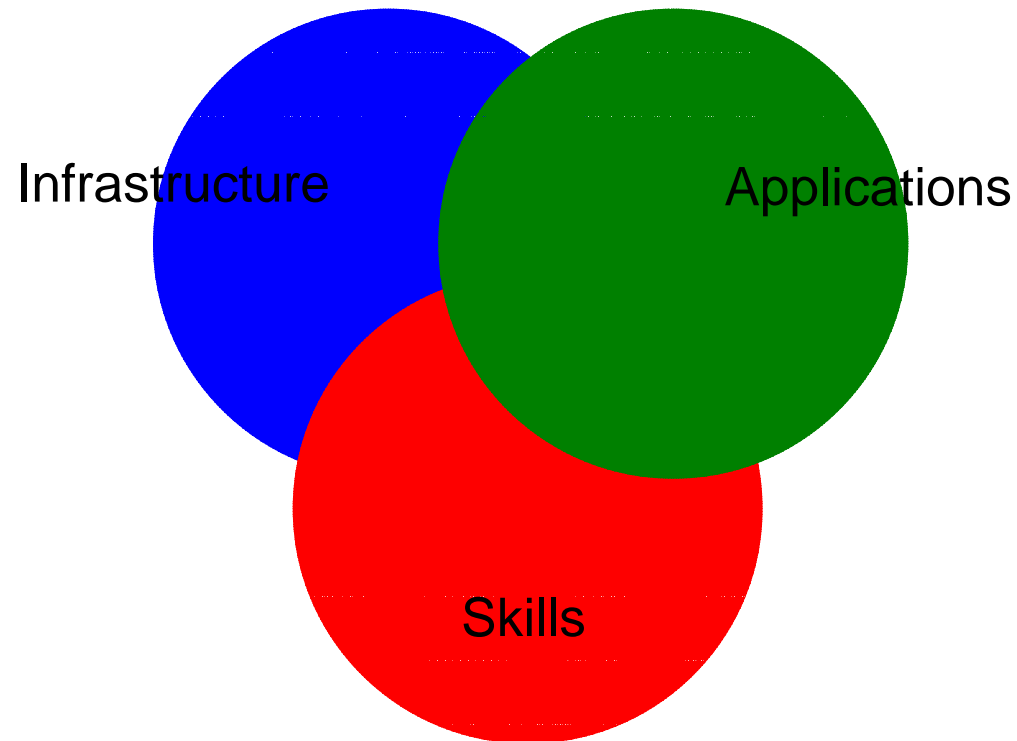
- The use of WS Binding everywhere will decrease performance and increase resource usage
- The use of WS Binding everywhere provides maximum flexibility to change the application, as only the component needing to be redeveloped must be opened.

# A Good Example of SOA



- The process service and its associated components have high performing but less malleable bindings
- Additional dynamic bindings may be made to the process service, increasing its value through reuse
- Workflow and enduser services are separated because they:
  - Tend to change more rapidly than other elements
  - Are rate limited by human performance

# Three Aspects of SOA Enablement



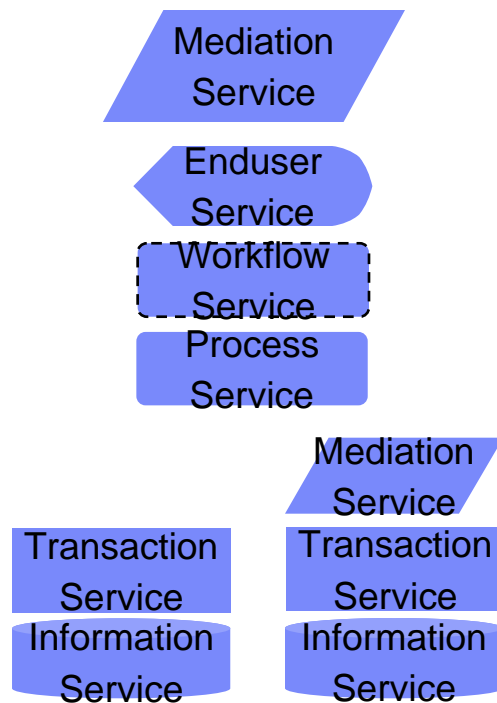
## SOA Enabling Existing Applications



## Why SOA enable existing applications

- “The irony is that ***host applications are probably better suited for exposure as part of an SOA than many applications based on more modern 4GL object-oriented languages***, said Phil Murphy, a principal analyst with consultancy Forrester Research, in an interview last year. “When folks wrote screen-based transactions many months ago, they wrote it at a business function viewpoint: I add a customer, I add an order for that customer, I check backlogs for that customer, etc. So in many respects, those CICS screens of 15 years ago are better suited to service orientation than a lot of the newer, distributed code that’s been written over the last several years, because of their affinity with a business function,” he argued, adding: “What did the object-oriented guys do? They took those screens and they broke them down into a thousand different objects.”

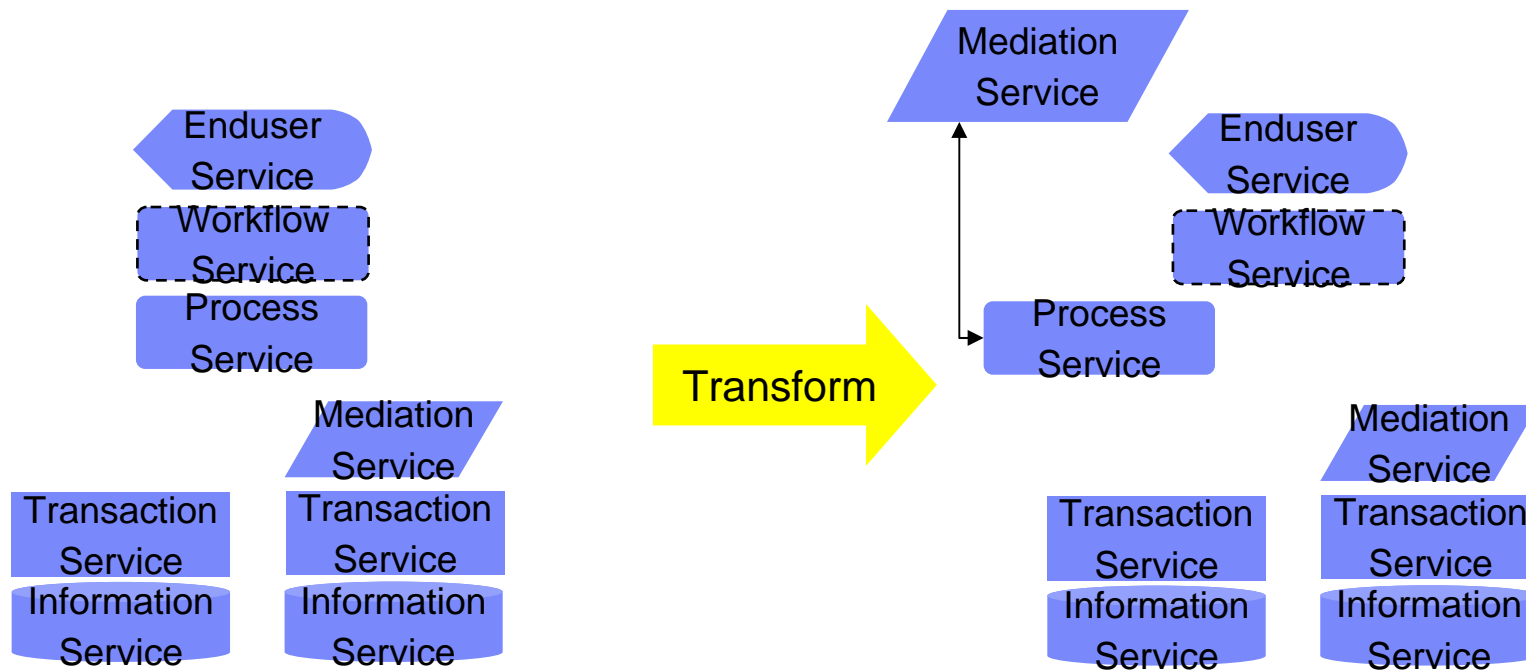
# Wrapping



- No change to the application – but limited reuse and may be less efficient
- Only available for Web Services invocation of existing application
- Mediation can be supplied in various forms
  - CICS Service Flow Feature and LINK3270 Bridge
  - Host Access Transformation Services (HATS)
  - Websphere Application Server via JCA connector
  - Websphere Application Server via JDBC

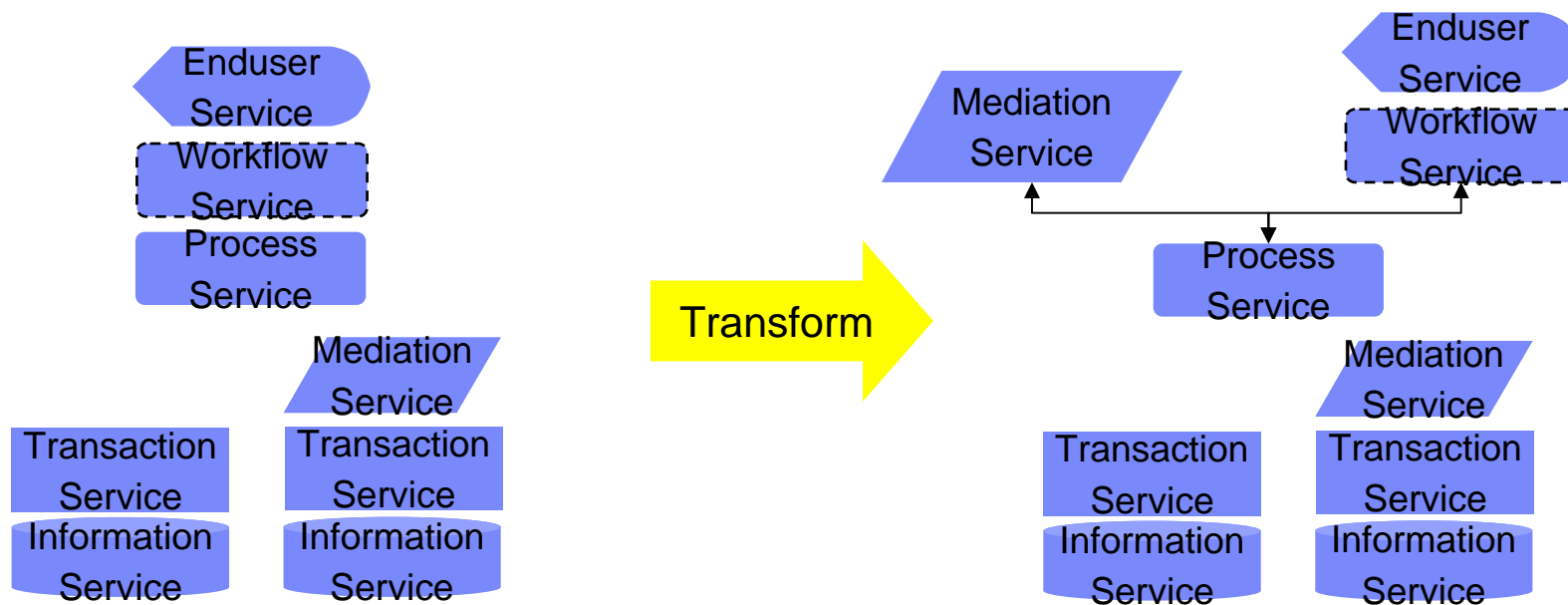
# Refacing – Application Transformation Technique 1

- The application is reengineered to expose internal process services. A mediation service is attached to these internal process services to make them accessible by Web Services. This technique requires the least change to the application, but may not give optimal performance.



## Refactoring – Application Transformation Technique 2

- The application is reengineered to expose internal process services and to separate the workflow and enduser services. This technique requires more change to the application but may provide better performance and resource usage.

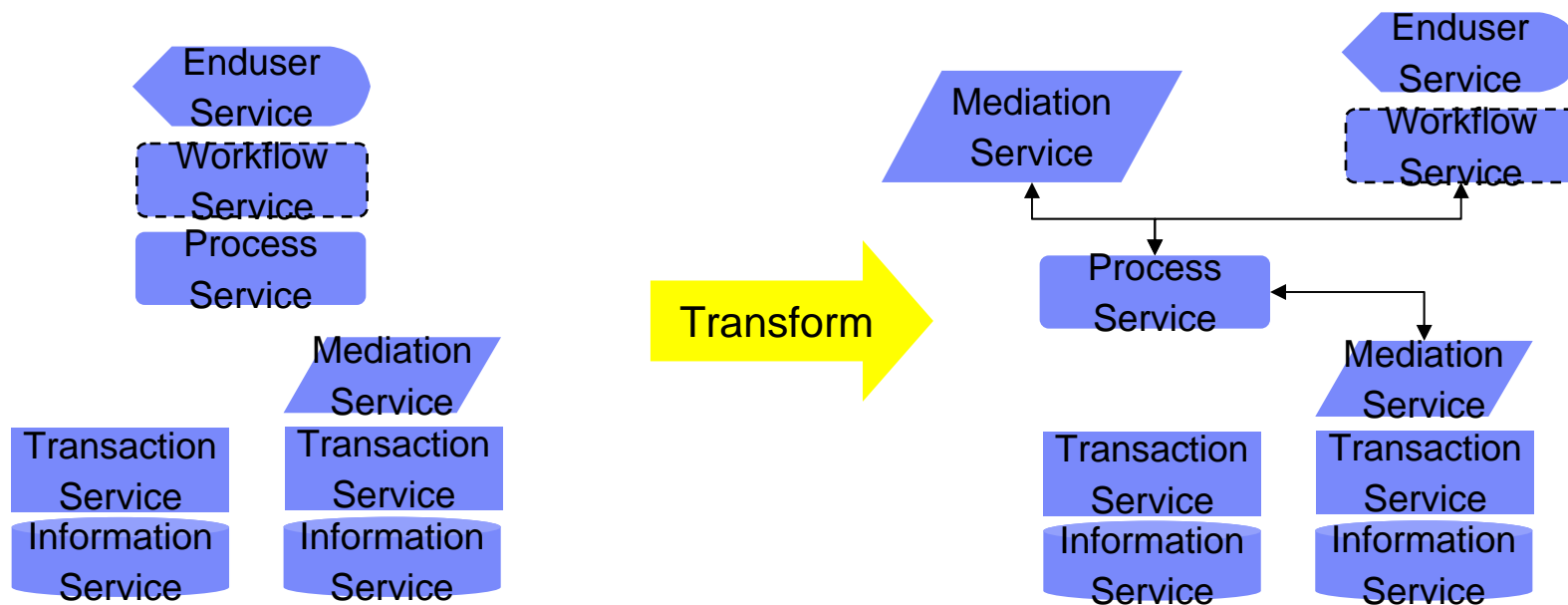


## Stateful and stateless services

- Older applications have “session affinity” – each request from “user A” must be routed to “application instance A”
  - ✓ Application instances are large resource bundles and hard to move from one computing platform to another to deal with load changes
- Modern applications are stateless – each request carries a “session token” that allows any application instance to retrieve the session state
  - ✓ Workload routing can be used to balance requests and resources
- When planning a refactoring project using technique 2, consider introducing stateless operation to the application

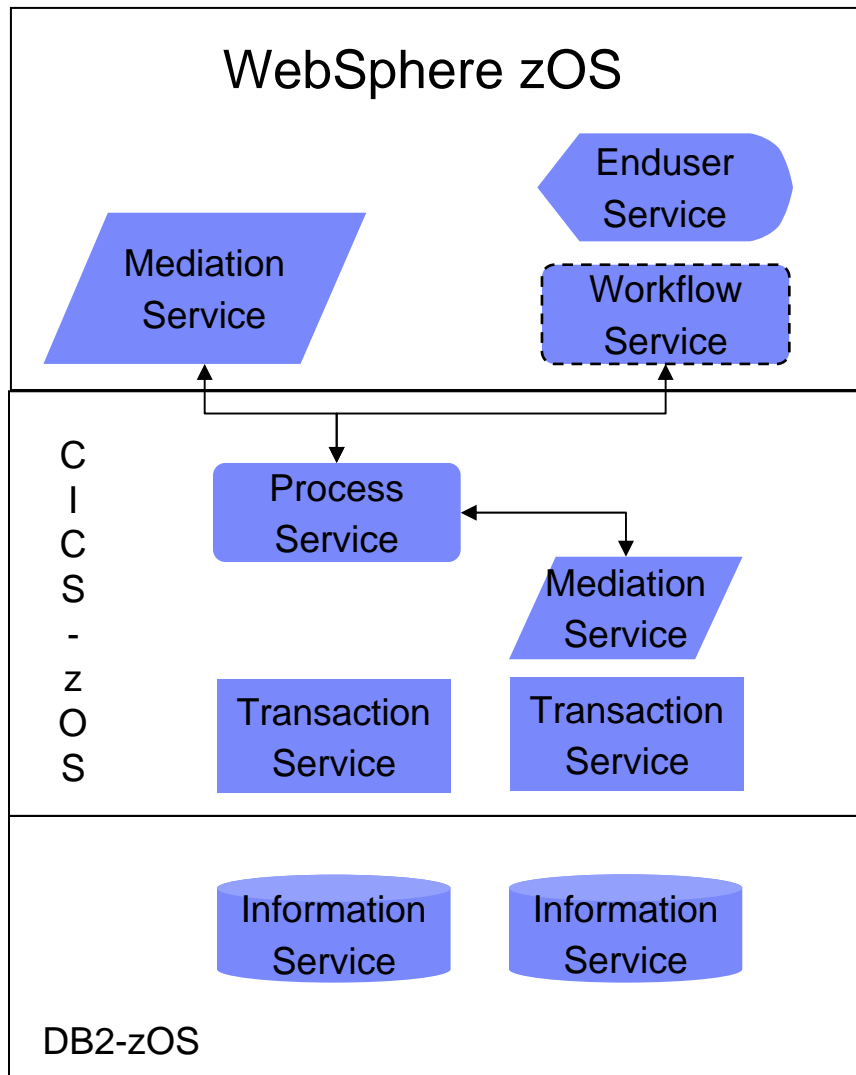
# Componentization

- The application is reengineered to expose the process service and to detach the internal component as a separate service. This would be done when there is need to reuse the internal component. The separation may introduce additional execution overhead (compared with Refactoring Technique 2)



# Application Architecture

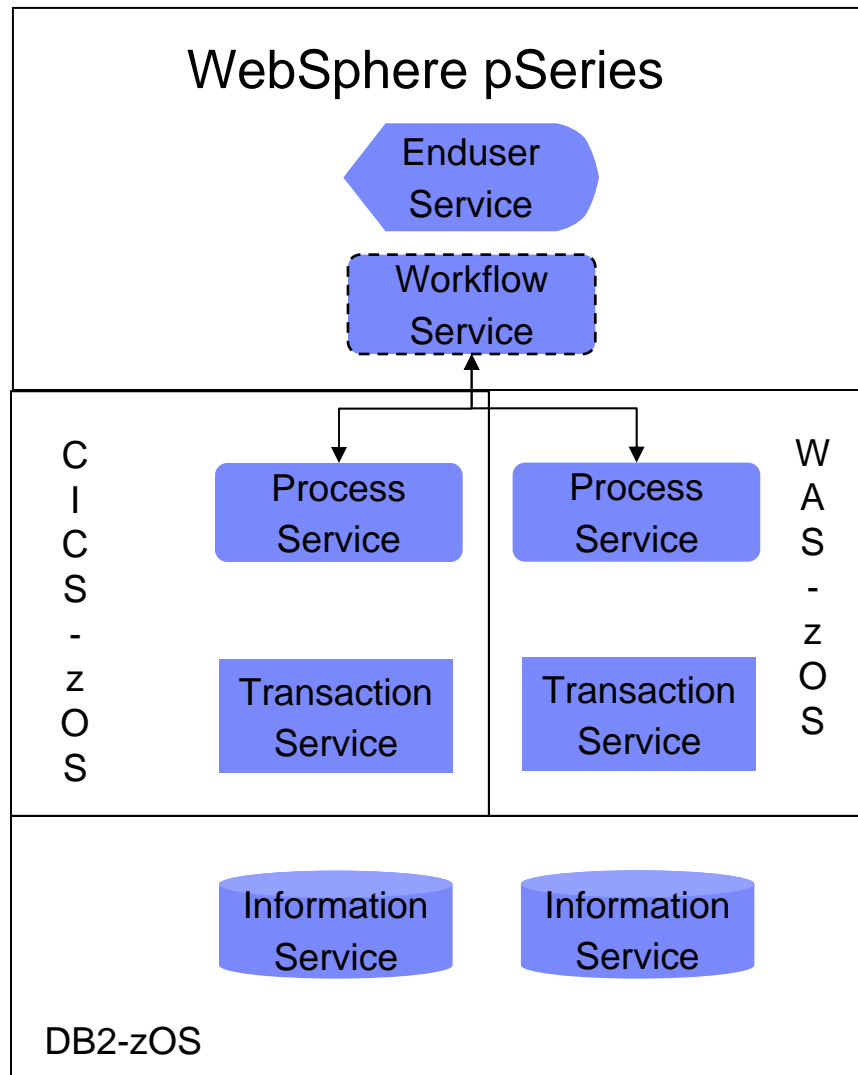
# Application/Data and UI/Application Coupling



- The Transaction Service and the Information Service are designated as closely coupled and are deployed in the same environment:
  - High transaction rates are expected
  - Large data structures are retrieved and updated
  - Multiple data operations are performed in the transaction service
- The Enduser Workflow Service and the Process Service are designated as moderately coupled and are deployed on the same computing platform (zOS)
  - Multiple interactions occur between the Workflow and the Process Service
  - Many simultaneous endusers are anticipated

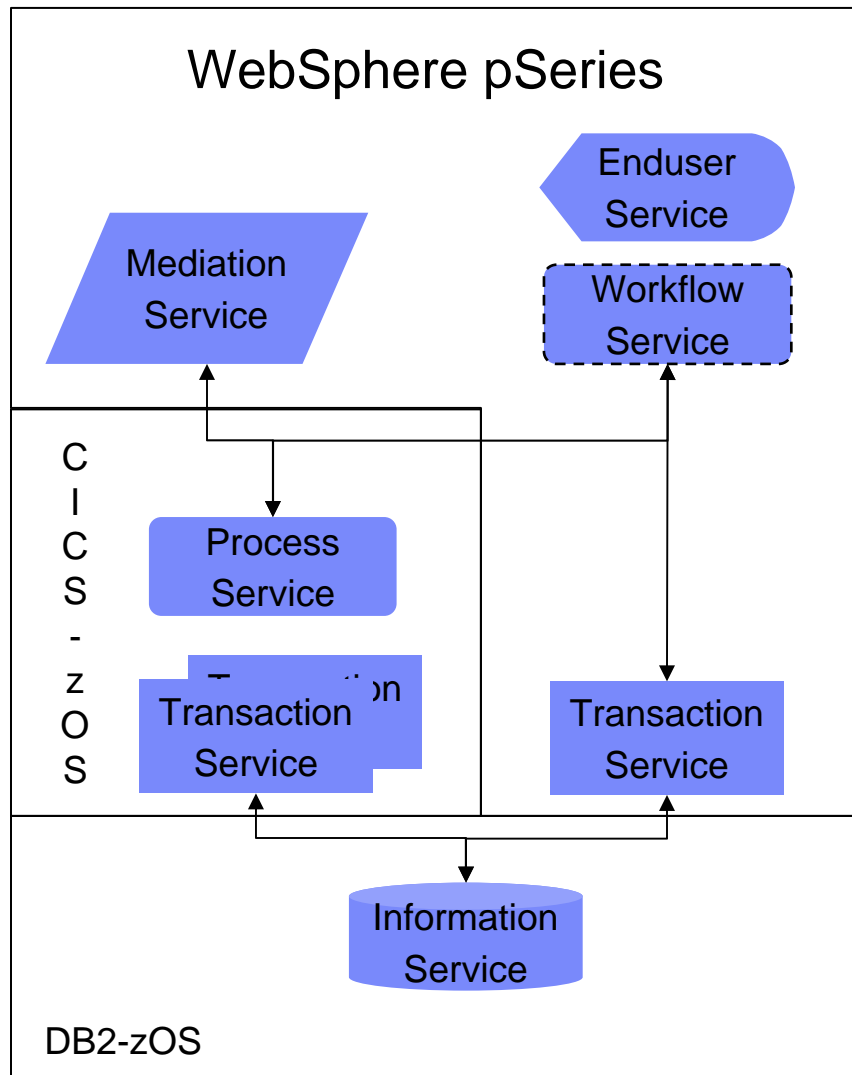


# Coupling continued 1



- A Process Service and a Transaction Service are deployed in CICS because they previously existed there.
- A Process Service and a Transaction Service are created in J2EE and deployed in WebSphere on zOS because of:
  - Anticipated high transaction rates
  - Strong coupling to data in DB2 zOS
- Record data is held in DB2 on zOS because of substantial amounts of batch processing done coincidentally with online work.

## Coupling continued 2



- The Information Service is shared among multiple transaction services. The most intense uses of the information service come from the CICS-based Transaction Services. That coupling establishes the proximity between CICS and DB2 deployed services.
- Another transaction service accesses the same information service. The coupling is designated as distant because:
  - Moderate transaction rates are expected
  - A single access is needed to the Information Service
  - The data is retrieved only and the data size is low

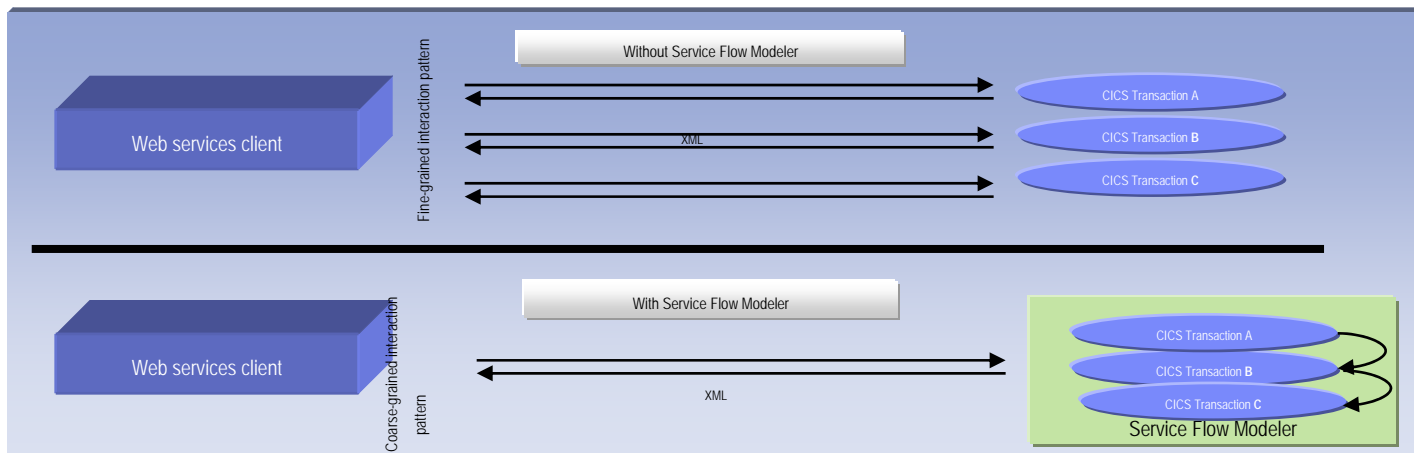
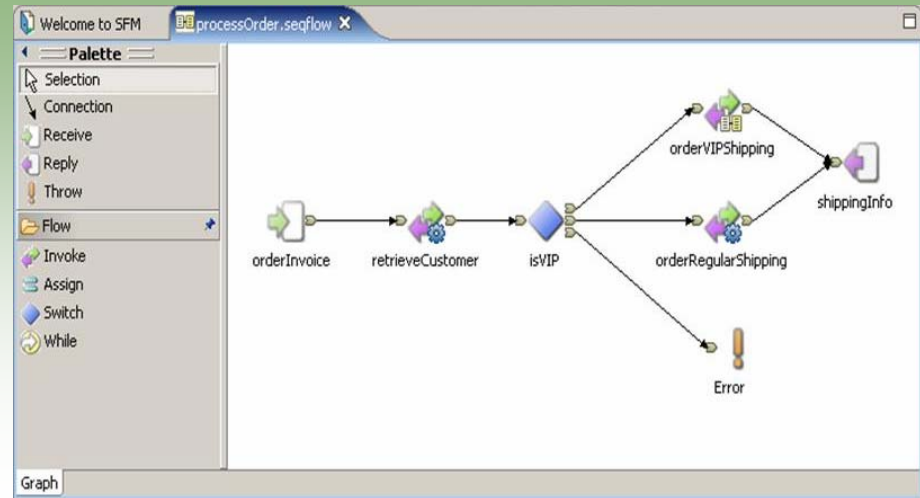
# Process Implementations

- Process execution has high performance requirements and low to moderate rate of change
  - ✓ Implement in Java, deploy in Websphere or batch
  - ✓ Implement in COBOL, PL/I or C for highest performance
- Process execution has moderate performance requirements and high rate of change
  - ✓ Implement in BPEL, deploy in Websphere Process Server
  - ✓ Implement in CICS Service Flow Modeler, deploy in CICS

# What is Service Flow Modeler

## *New Feature!* Service Flow Modeler in WebSphere Developer for zSeries

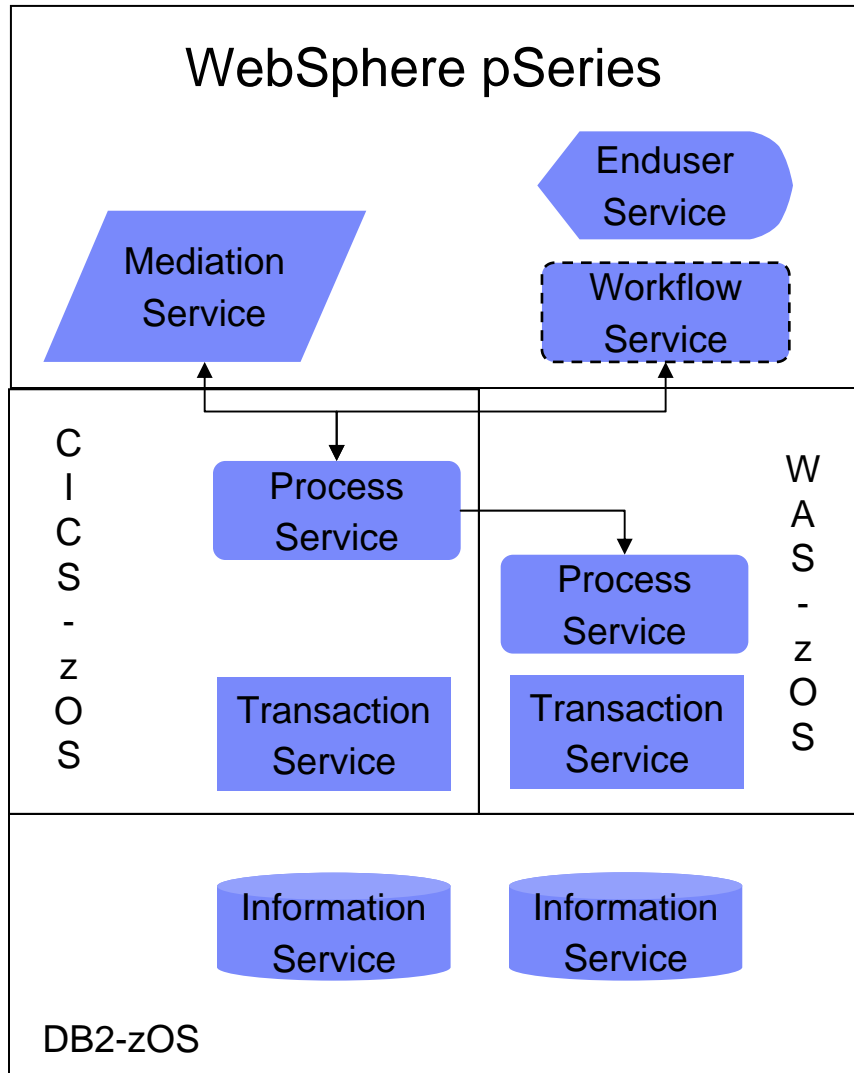
- **Builds Web services from existing CICS applications**
  - ✓ Aggregates multiple CICS transactions into high-level business processes through visual modeling
  - ✓ Supports CICS BMS (terminal-based) applications & CICS commarea applications
  - ✓ Highly optimized CICS runtime supporting Web services and XML interfaces



# Asynchronous Linkage

- Asynchronous linkage offers better scaling, but is more complex to program
  - ✓ Work can continue without waiting for invoked service response
  - ✓ Better use of multiple processors
- Model Driven Design offers patterns for asynchronous linkage
  - ✓ Code generation reduces development complexity
  - ✓ Model analysis reduces design errors

# Asynchronous Example



- The Process Service in CICS invokes another Process Service in WAS zOS using an asynchronous linkage (e.g. MQ)
  - The WAS zOS Process Service does not return a result that is needed by the CICS Process Service (for example, a transaction audit record is written to a database)
  - Reliable messaging is often used to guarantee that the asynchronous process executes

# A Vision of the Future in Application Development



Thank You!