Session: EM02

Service Enable Your New and Existing IBM System z Applications

Russell Bonner
Consulting IT Specialist, Rational Software
bonnerr@uk.ibm.com

IBM Rational Software Development Conference 2008

WHERE TEAMS ARE R-HEROES

IBM

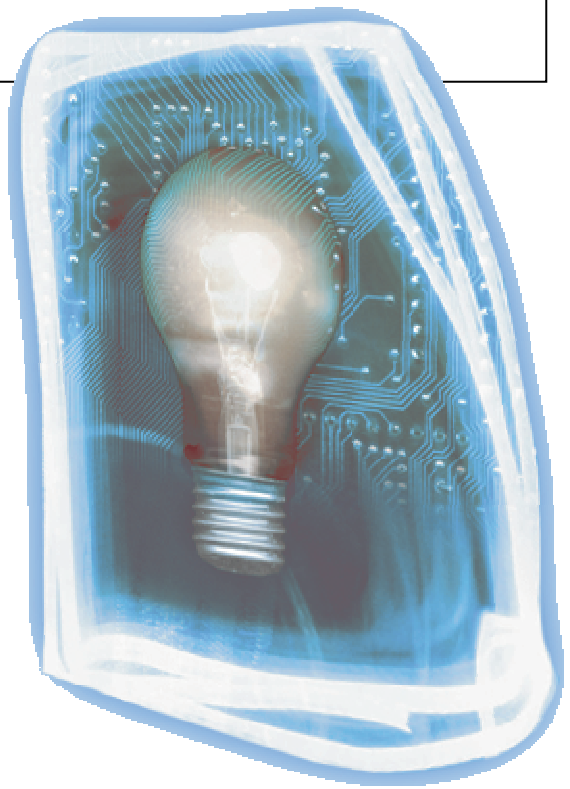
Rational. software

Agenda

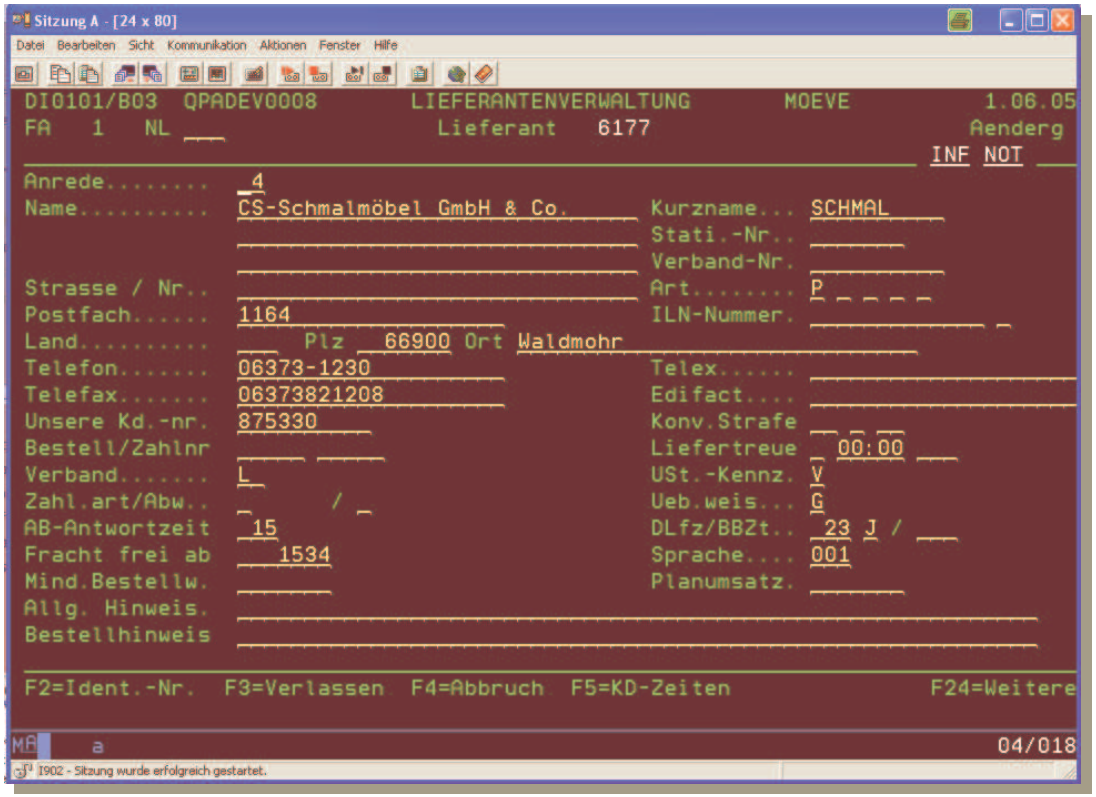
Why service orient your System z Applications?

Approaches to service orientation

- Wrapping
- Re-engineering
- Redeveloping



Who has seen this screen?



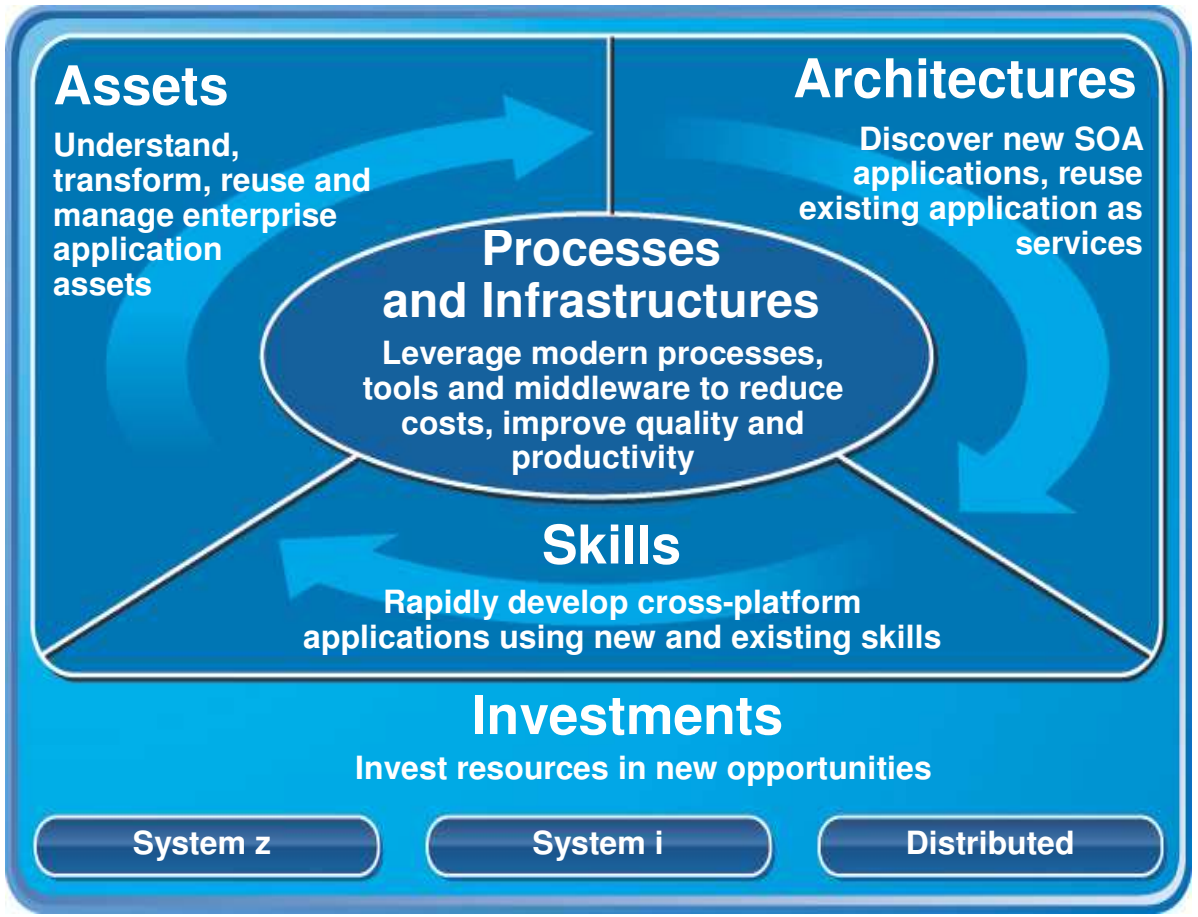
Pros

- Works today!
- Existing staff is skilled and efficient
- Minimal work to maintain

Cons

- Screen space is limited
- Text only
- Not easy to expose to new audiences (e.g., online check-in)
- High training costs for inexperienced users

Enterprise Modernization – Solution Overview



- ✓ *Leverage value in existing assets*
- ✓ *Drive innovation with SOA and web technology advancements*
- ✓ *Leverage existing and new staff on multi-platform projects*
- ✓ *Improve quality and flexibility with consolidated team infrastructure*
- ✓ *Reduce maintenance costs*

http://www-306.ibm.com/software/info/developer/solutions/em/

Modernize Architectures

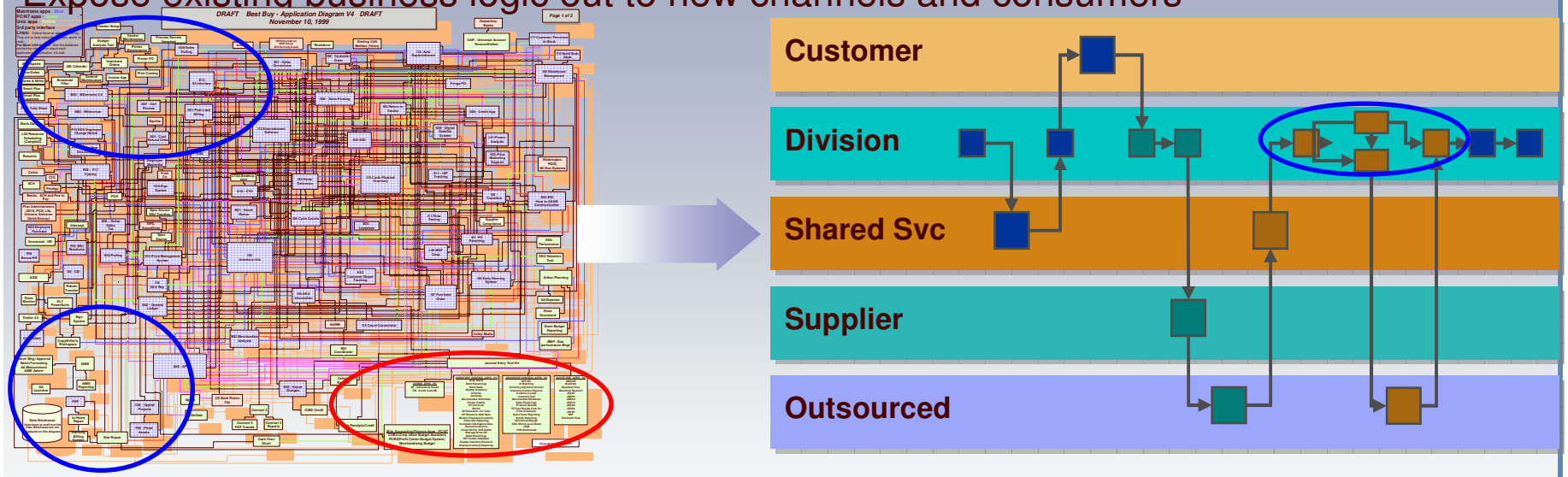
Flexible architectures to enable business agility

Easily create services from existing code, including CICS, IMS, and terminal applications

Define new services for all deployment platforms from initial design to implementation

Separate service flow from service implementation to attain optimal flexibility

Expose existing business logic out to new channels and consumers

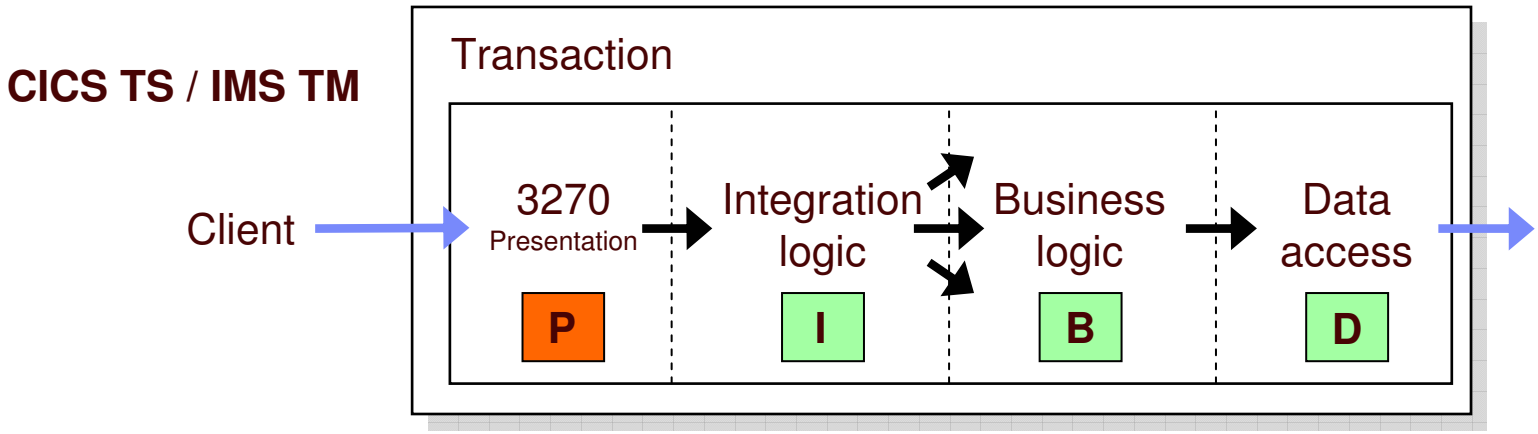


"SOAs cost 20% less to implement and saves 50% more with each reuse than traditional component-based development… the level of reuse in SOA development averages 2.5 times more than non-SOA development"
Jeffrey Poulin, PH.D. and Alan Himler, MBA, 2006, "The ROI of SOA – Based on Traditional Component Reuse"

Enterprise services

- Services provide standardized access to assets for different software applications residing on disparate platforms
- Service definitions provide abstract interfaces which allow for loose coupling between business components – implementation can vary without affecting consumers
- You can reuse applications exposed as Services in a variety of service-oriented architecture frameworks, such as a process choreographer or an enterprise service bus.

Modern “System z” architecture



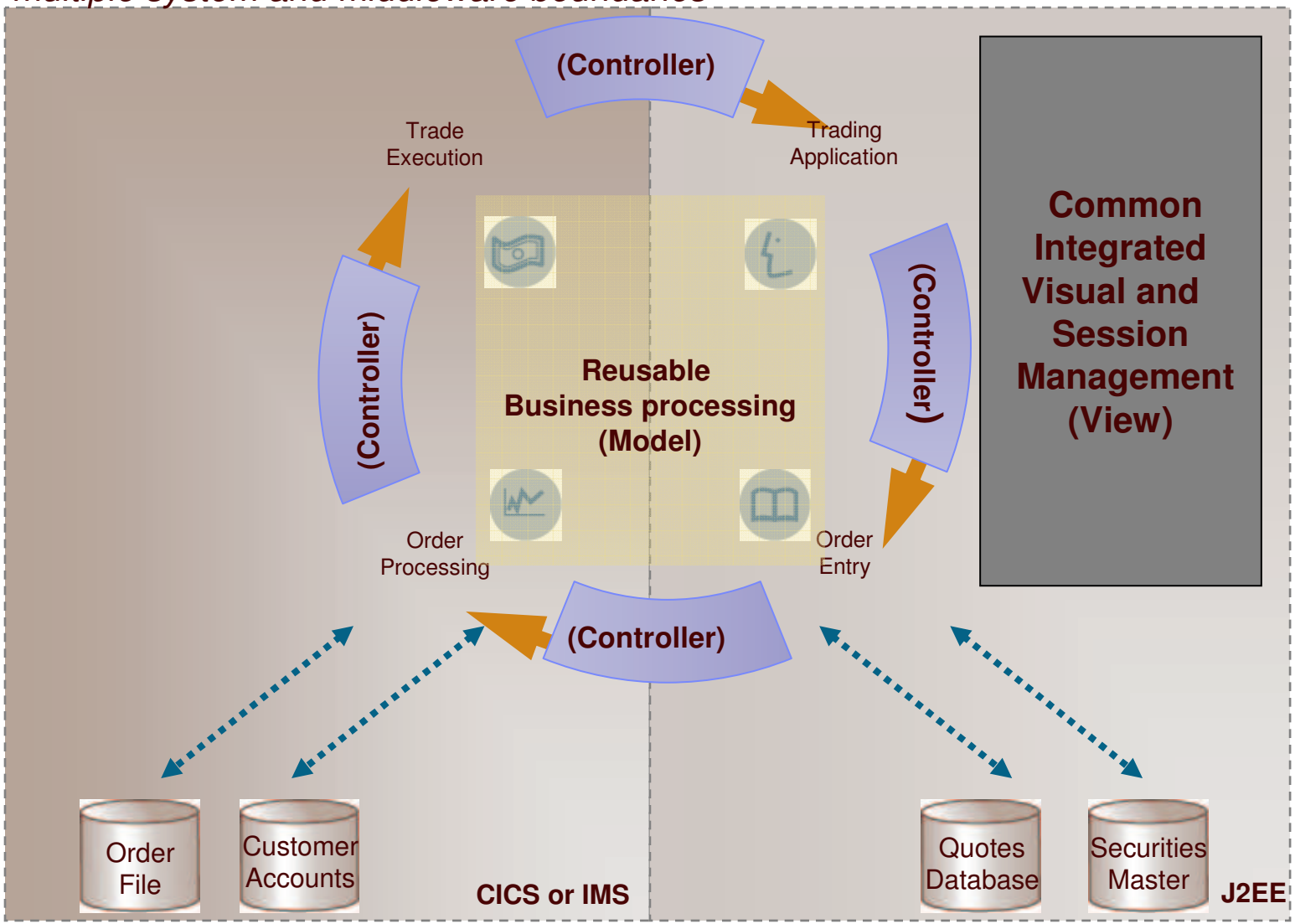
Best practice in System z application design is to separate key elements of the application, in particular:

Presentation logic	3270, HTML, XML
Integration or aggregation logic	Menu, router, tooling
Business logic	COBOL, PL/I, Reusable component
Data access logic	VSAM, DB2, IMS, …

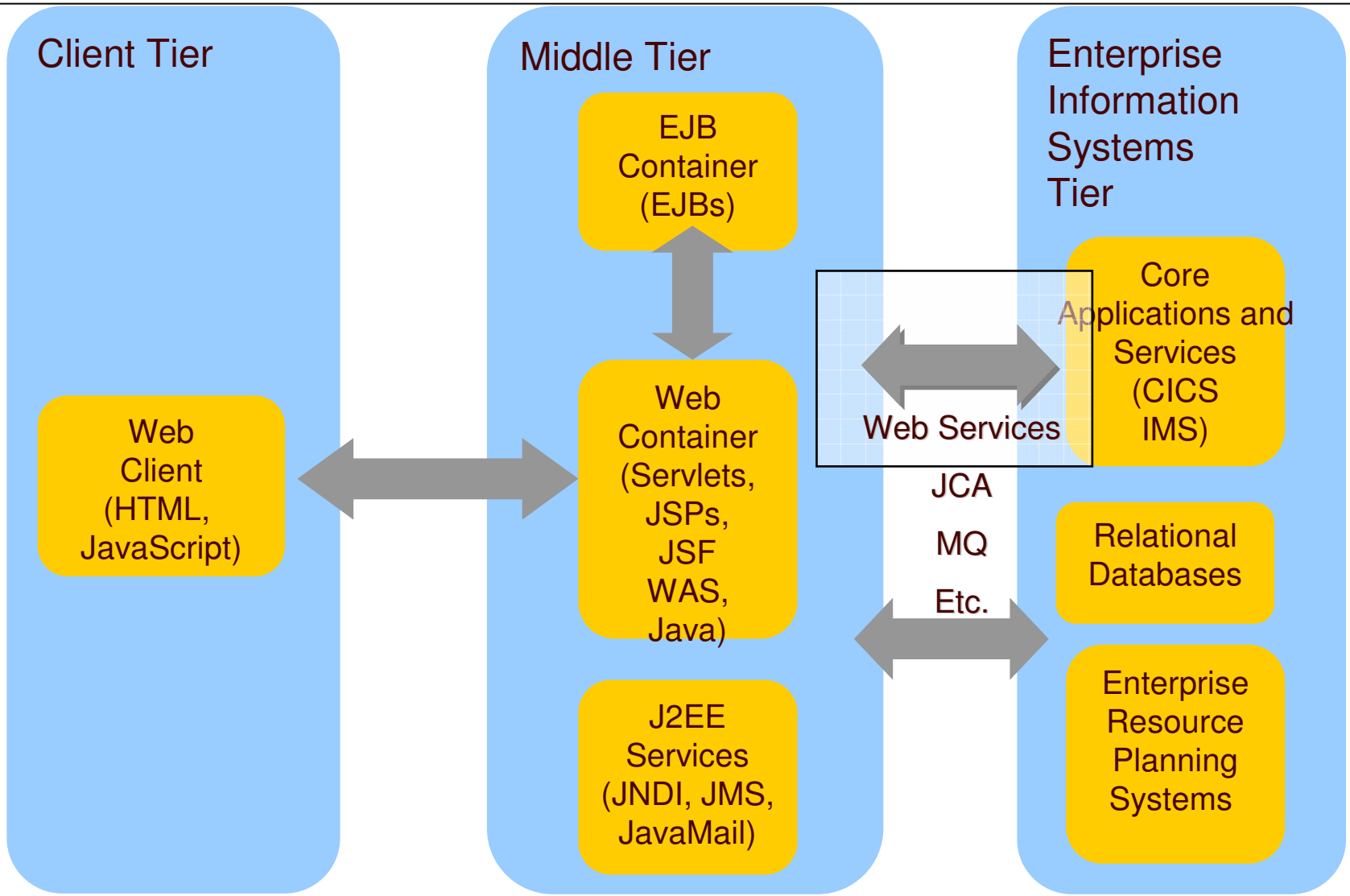
Provides a framework for reuse and facilitates separation of concerns, clear interfaces, ownership, and optimisation

Composite Workload Application Components

Spans multiple system and middleware boundaries



Modern Multi-tier architecture



Web Services

Architecture for

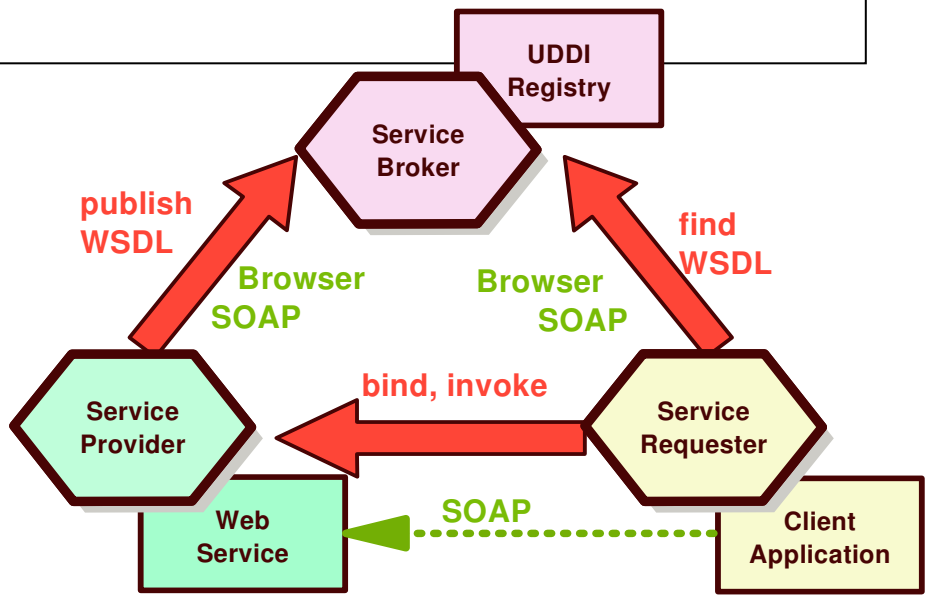
- Application to application
  - Communication
  - Interoperation

Definition:

- Web Services are **software components described via WSDL** that are capable of being accessed via **standard** network protocols such as SOAP over HTTP

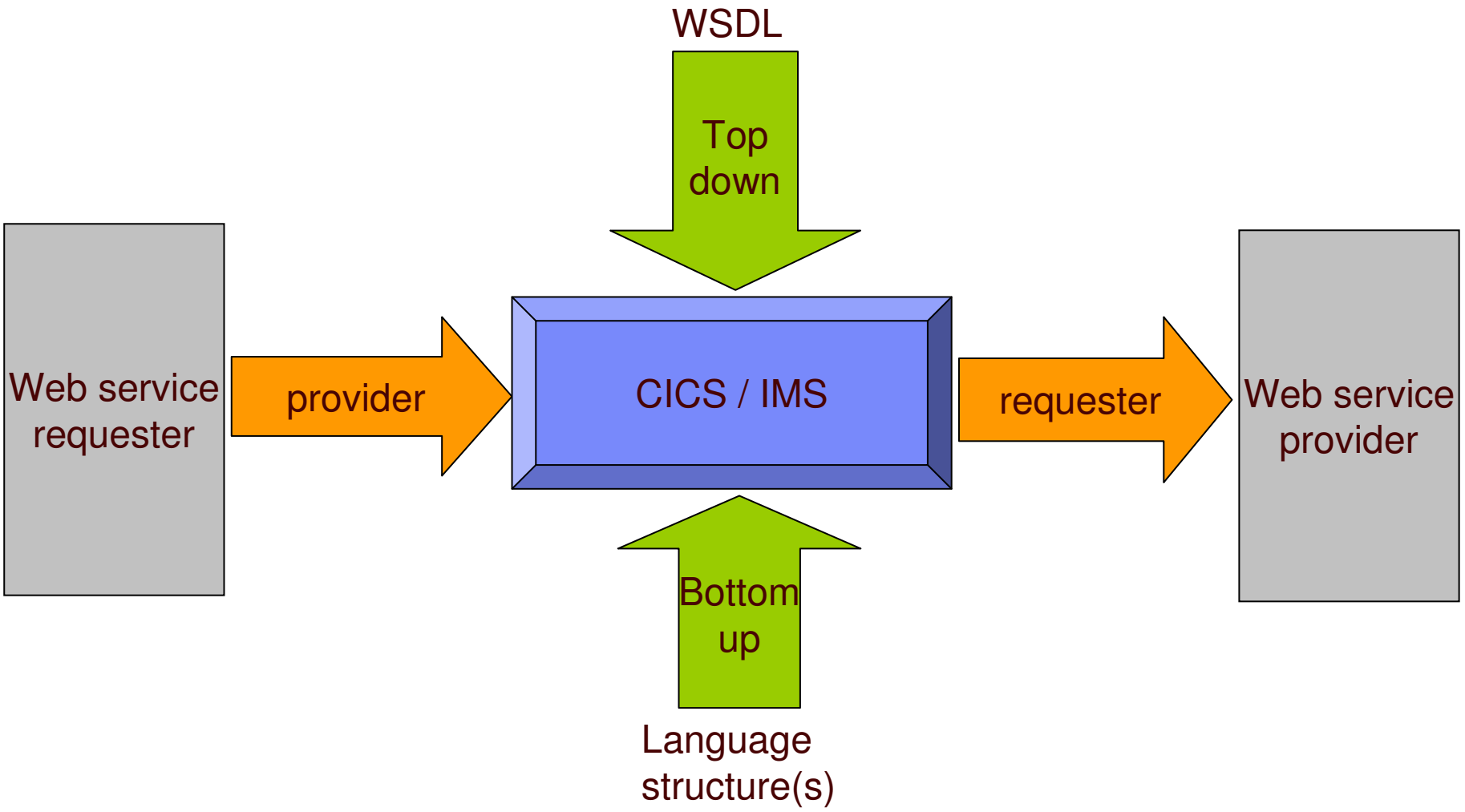
WS-I.org (Web Services Interoperablity Organization)

- Ensure interoperability



The entire industry is agreeing on one set of standards !!

Web Services Enablement Styles



Agenda

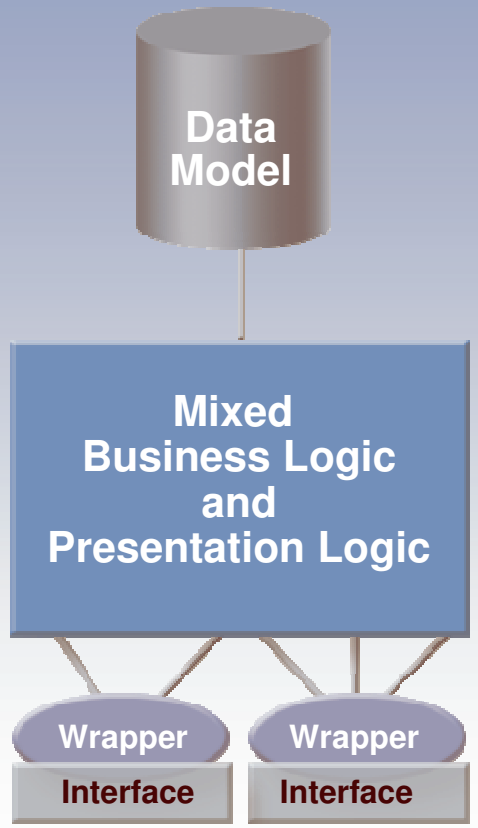
Approaches to SOA

- Wrapping
- Re-Engineering
- Redeveloping

Three ways of re-using services from existing applications

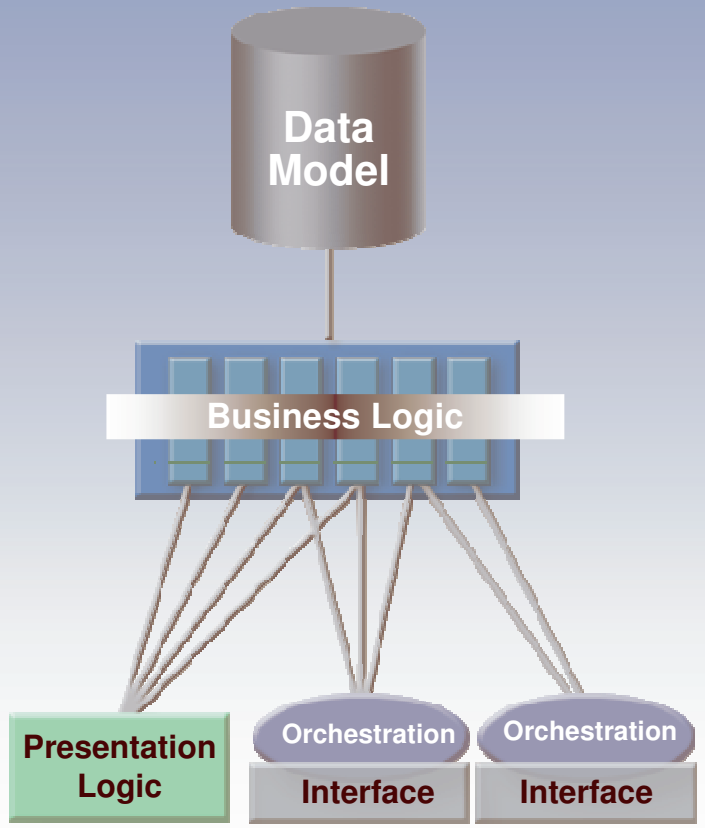
Wrapping

Drive existing screen-based applicatons to package "pseudo-services"



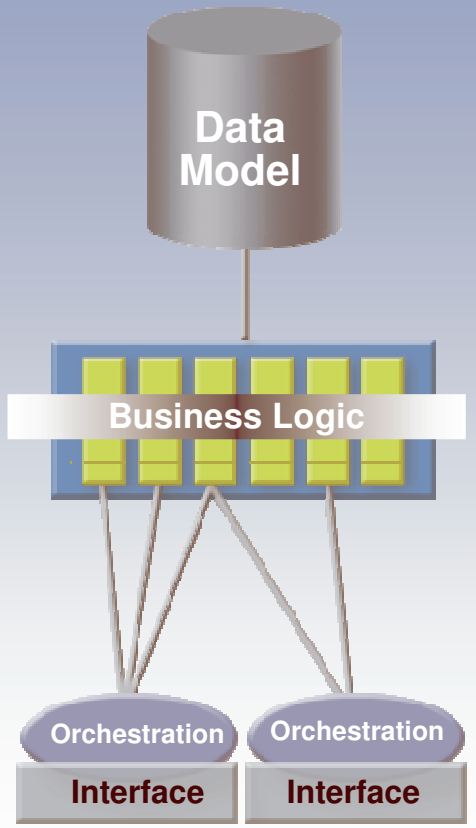
Re-engineering

Business logic is modularized and separated from presentation



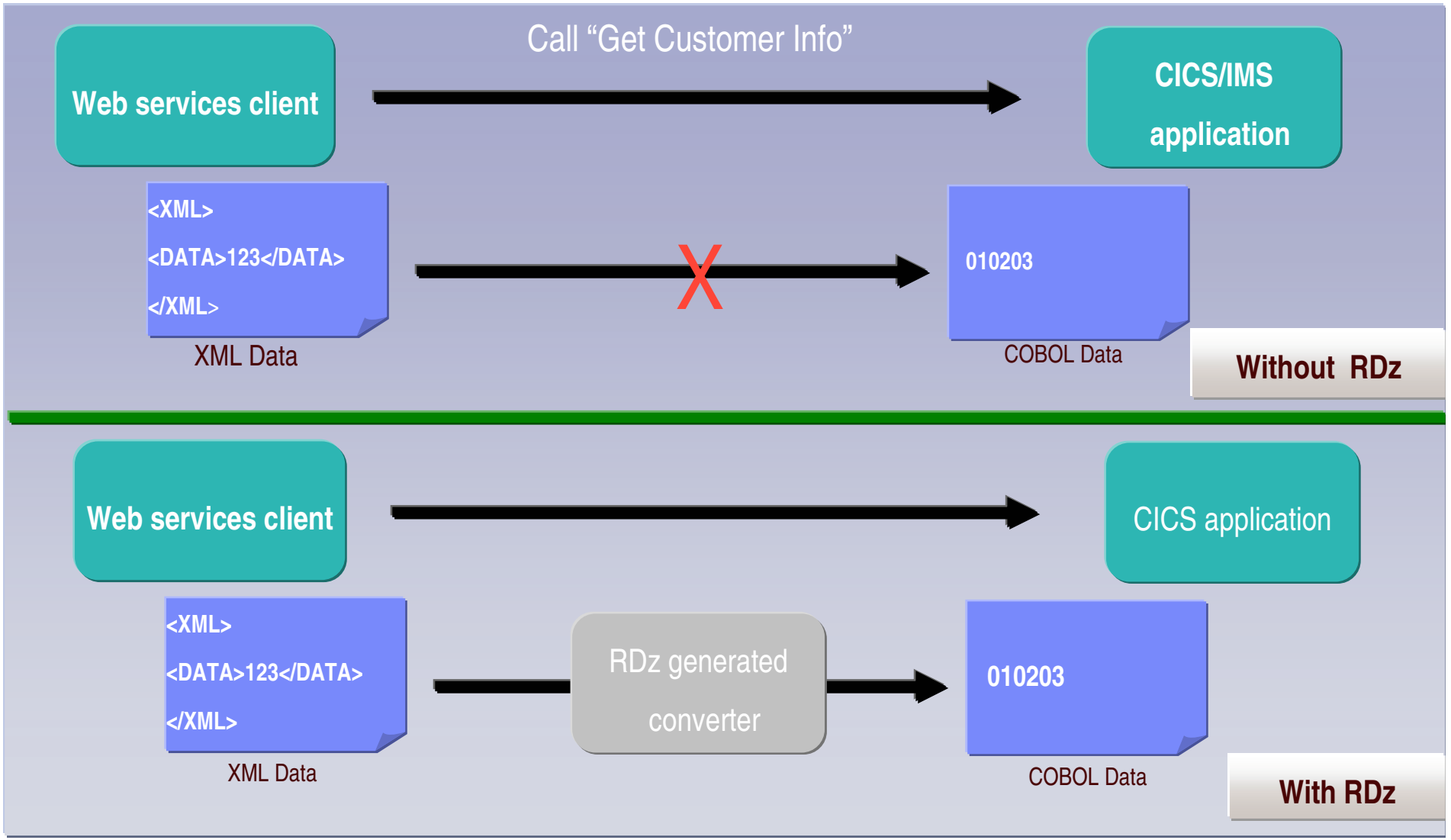
Redeveloping

Business logic of services is redesigned from scratch



Source: Gartner

We need interfaces to talk "XML" ….



Agenda

- Approaches to SOA
  - Wrapping
  - Re-Engineering
  - Redeveloping

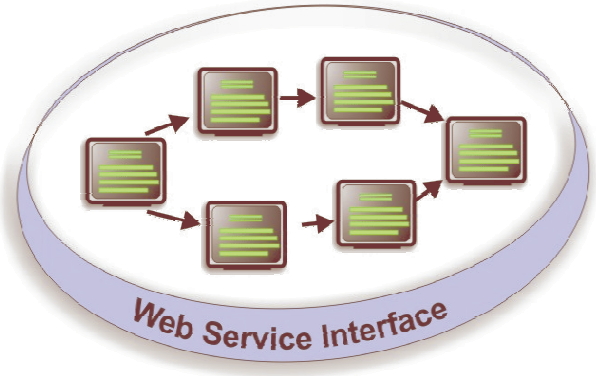
Host Access Transformation Services (HATS)

What can you do with HATS?

- Quickly and easily create Web, portal or rich client applications that provide an easy-to-use GUI for your green-screen applications
  - Low skills requirement
  - Highly customizable
  - Iterative development process
  - Transformation “on the fly”
- Extend terminal application tasks as Web services

Benefits

- Extend host application to new users
- Improve the navigation of your host application
- Reuse your existing assets in a Service Oriented Architecture
- Avoid rewriting or reengineering host applications



Host Access Transformation Services Values

Rich Client	Browser	Portal	Mobile	Web Services
				
Integration at the desktop with other Eclipse-based applications Client side processing Rich set of user interface widgets Supports Lotus Expeditor deployment	Zero footprint Pure HTML Access through your favorite browser, including Internet Explorer and Firefox.	Integration at the glass Cooperative portlet support JSR 168 compliant	Access host applications from mobile devices	Build self-service transactions Expose host business processes as Web Services Provide controlled access to vital host applications and host data.

Quick ROI : Host applications can be quickly deployed with a new user interface

Low cost : No need to rewrite application

Low risk : Leverage open, proven platforms and technologies

Increase productivity and reduce training costs

Improving work flow from multiple applications

Provides integration of host business processes and data with other Web, portal, and rich client applications

The Value of HATS SOA

HATS is an important part of the IBM SOA strategy!

IBM SOA Foundation helps to **extend** the value of the applications and business processes that currently run your business, **not replace** them.

HATS does not require application reengineering or rewriting

Rewriting can consume significant amounts of time and resources

Rewriting requires access to and understanding of application source code

Rewriting inherently introduces new bugs that must be found and fixed

Simply put - rewriting is not always a realistic option!

HATS is unique

HATS is the only IBM product that Web service-enables 3270, including CICS and IMS, 5250, and VT terminal applications – great for customers with a mix of platforms.

HATS generates standard Web Service interfaces

HATS services can be integrated into an existing SOA.

Agenda

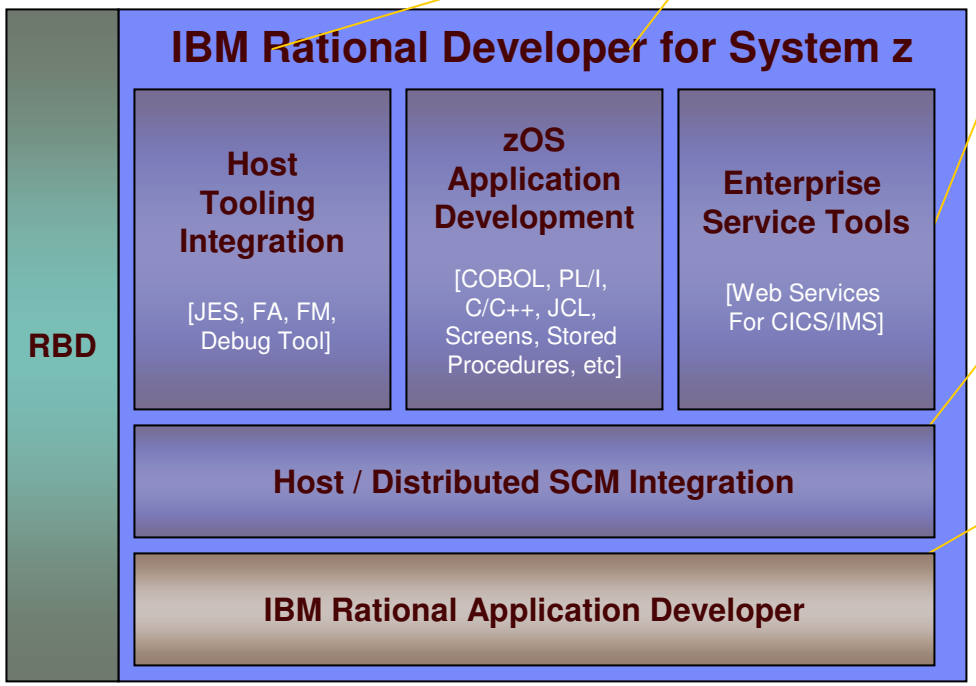
Approaches to SOA

- Wrapping
- Re-Engineering
- Redeveloping

IBM Rational Developer for System z V7.1

Develop EGL using RBD
- Quick and easy development of modern enterprise applications for business programmers
- Simplify and speed up creation of Web applications and services without having to learn Java or J2EE

JES and PD Tools
•Debug zOS applications from the workstation as they execute live in the remote runtime
•Read/Write/Update VSAM datasets via integration with IBM File Manager
•Access IBM Fault analyzer reports for analyzing ABENDS and associating back to source code
•Interact with the Job Entry Subsystem (JES) to submit jobs, monitor jobs, and review job output



Traditional Development
Development Environment
- Connect to z/OS systems
- Work with z/OS resources like COBOL, PL/I , C, C++, JCL, assembler, etc.
- Perform dataset management actions like allocating datasets and migrating datasets
- Perform typical edit, compile, and debug tasks on remote z/OS resources from the workstation
- Create, build, and catalog DB2 stored procedures on zOS
- Compile and test programs locally to ensure correctness

Screen design
- Visually create, modify, build, and deploy BMS maps sets or MFS/IMS maps remotely or on the local workstation

Code Generation
- Generate CRUD DB2 program code from UML, which can also be easily integrated into web service applications

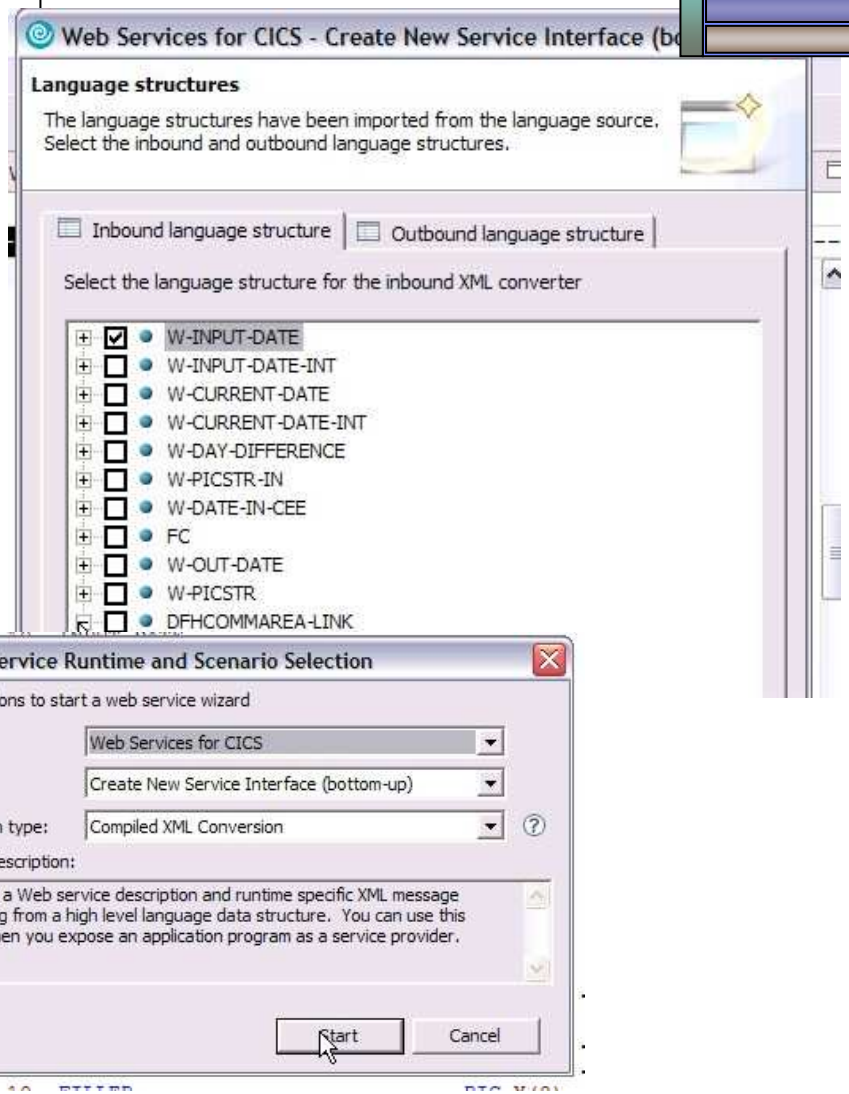
zOS Web Service and Flow Creation
- Implements SOA and Web Services
- SOA access to CICS V3.X and IMS V9/V10 COBOL and PL/I applications
- Bottom-up/Top-down/meet-in-the-middle COBOL or PL/I to XML mapping support
- Integrated XML converters, XML schemas, and WSDL generation
- Service Flow Modeler to build/deploy service flows out of your existing Commarea, Channel, MQ, and Terminal CICS applications.

SCM Support
- Access to host SCMs such as SCLM
- Framework for writing/deploying custom SCM integration code
- Support for storing zOS resources in distributed SCMs such as ClearCase

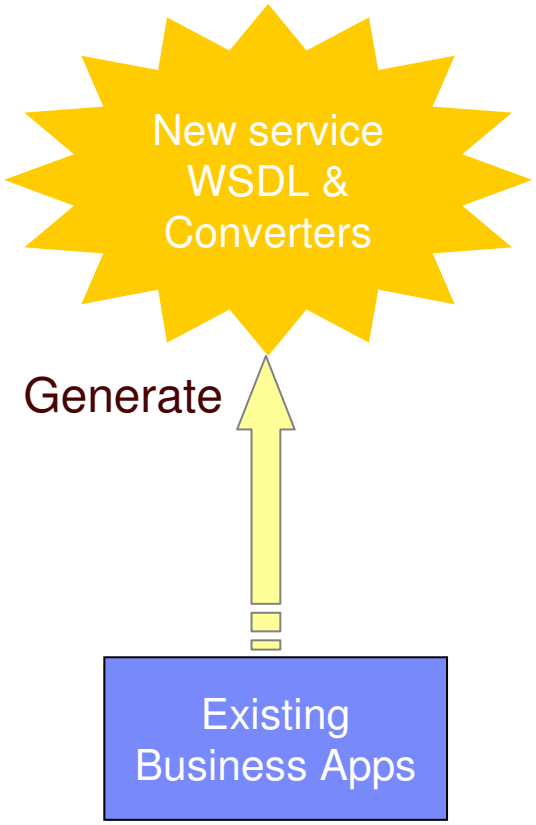
Web and JEE Development
- Create Web Pages / JSF / Struts
- JEE/Java Development
- JCA Connectors
- Distributed debugger
- Web Services and Test environment

Create web services for zOS runtimes

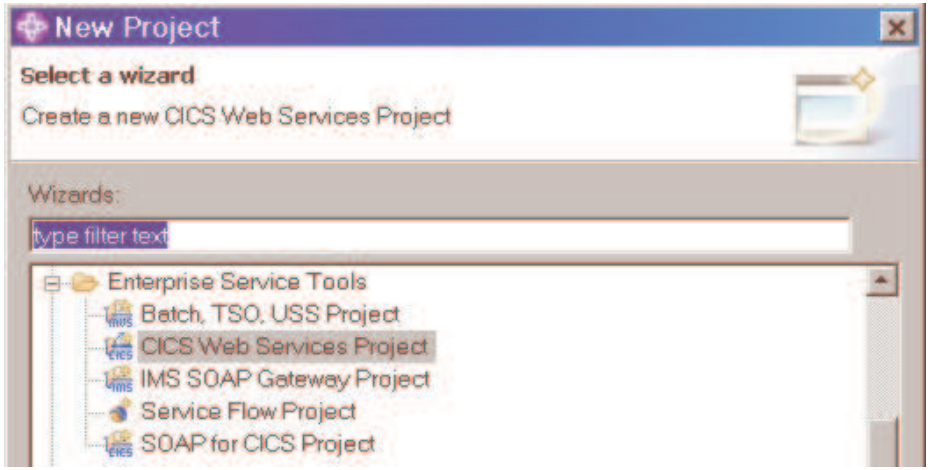
- Build, Deploy, and Test Web services from existing applications
- Create source code skeletons from web service definitions
- Map web service definitions to existing application modules
- Supports traditional languages
  - COBOL
  - PL/I
- Supports zOS specific runtimes
  - CICS
  - IMS
  - Batch

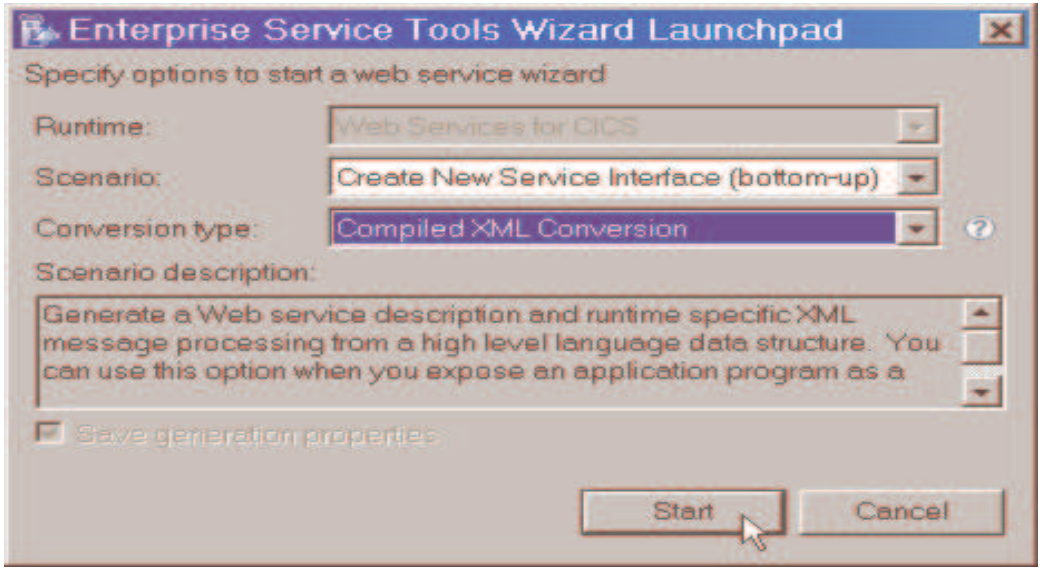


XML Services for the Enterprise (XSE)
Example Bottom-up



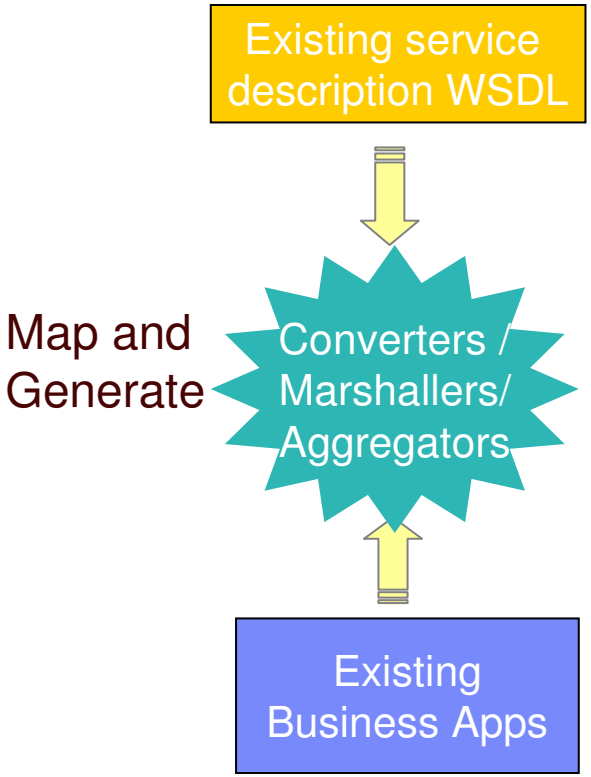
Bottom-up

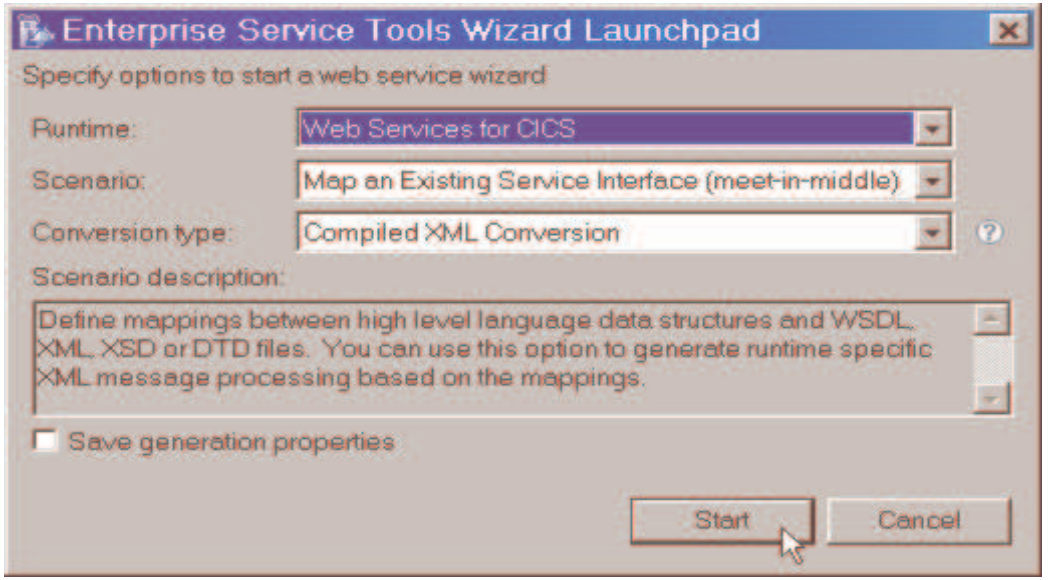




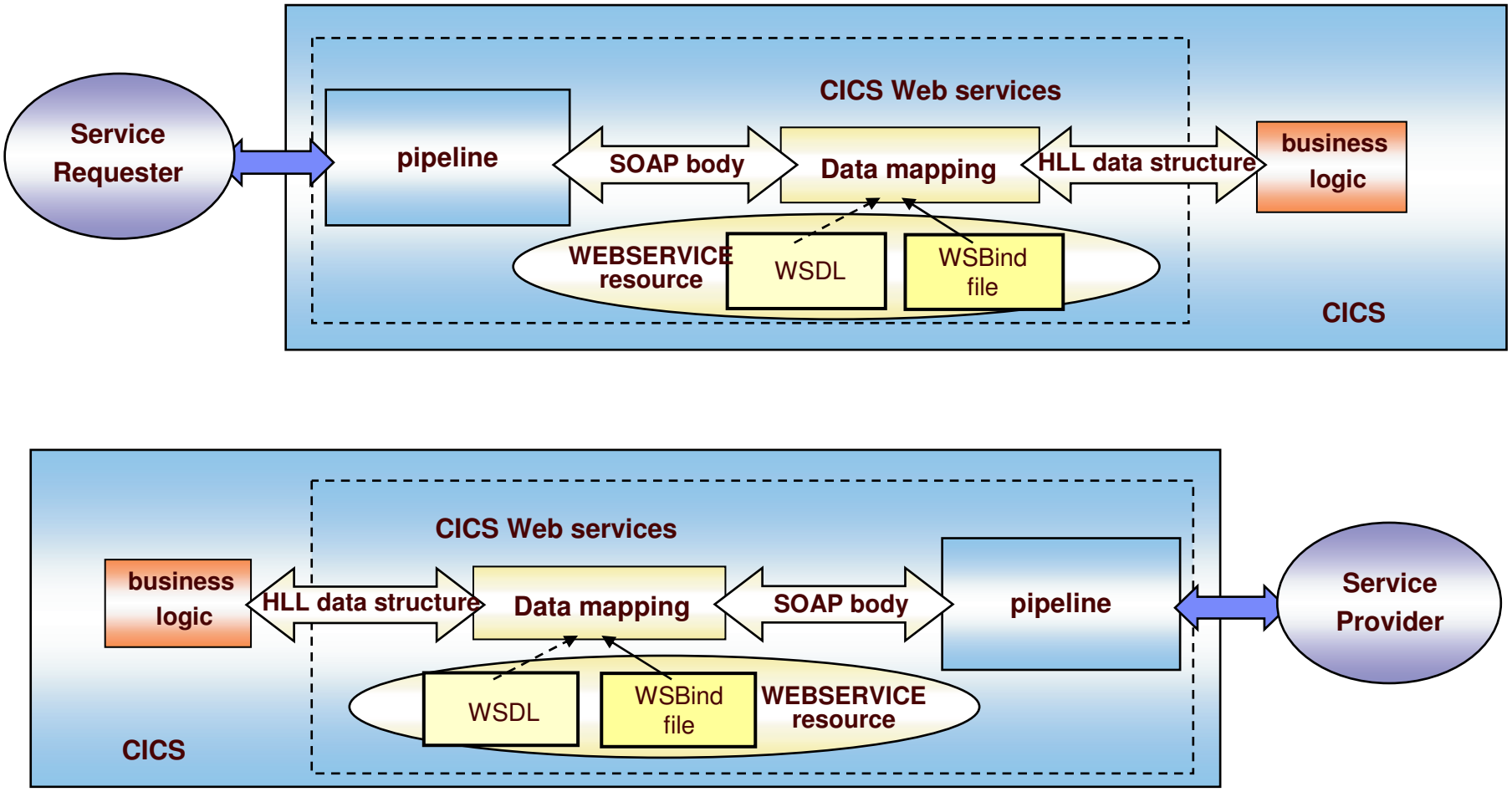
XML Services for the Enterprise (XSE): Example Meet in the middle

Meet in the middle

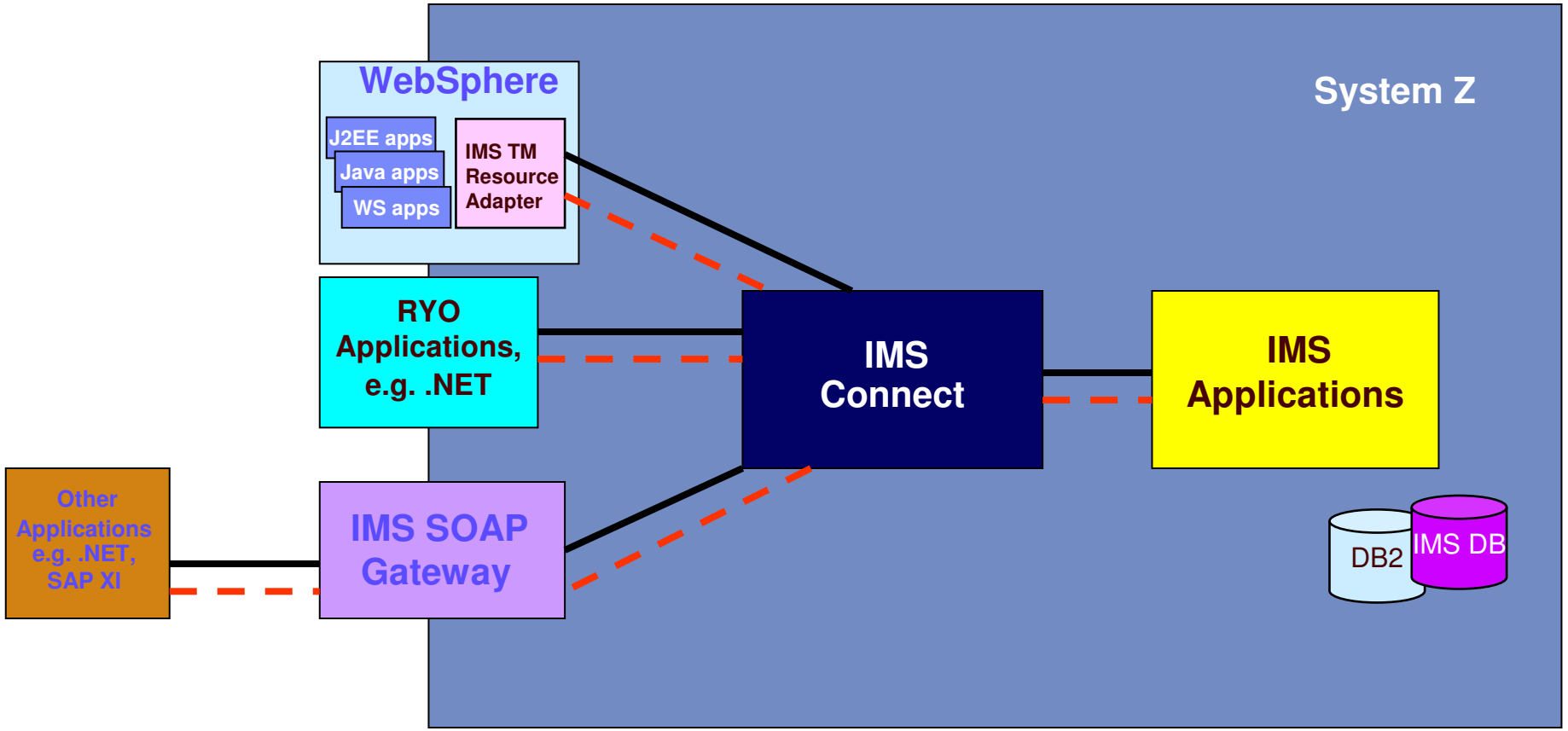




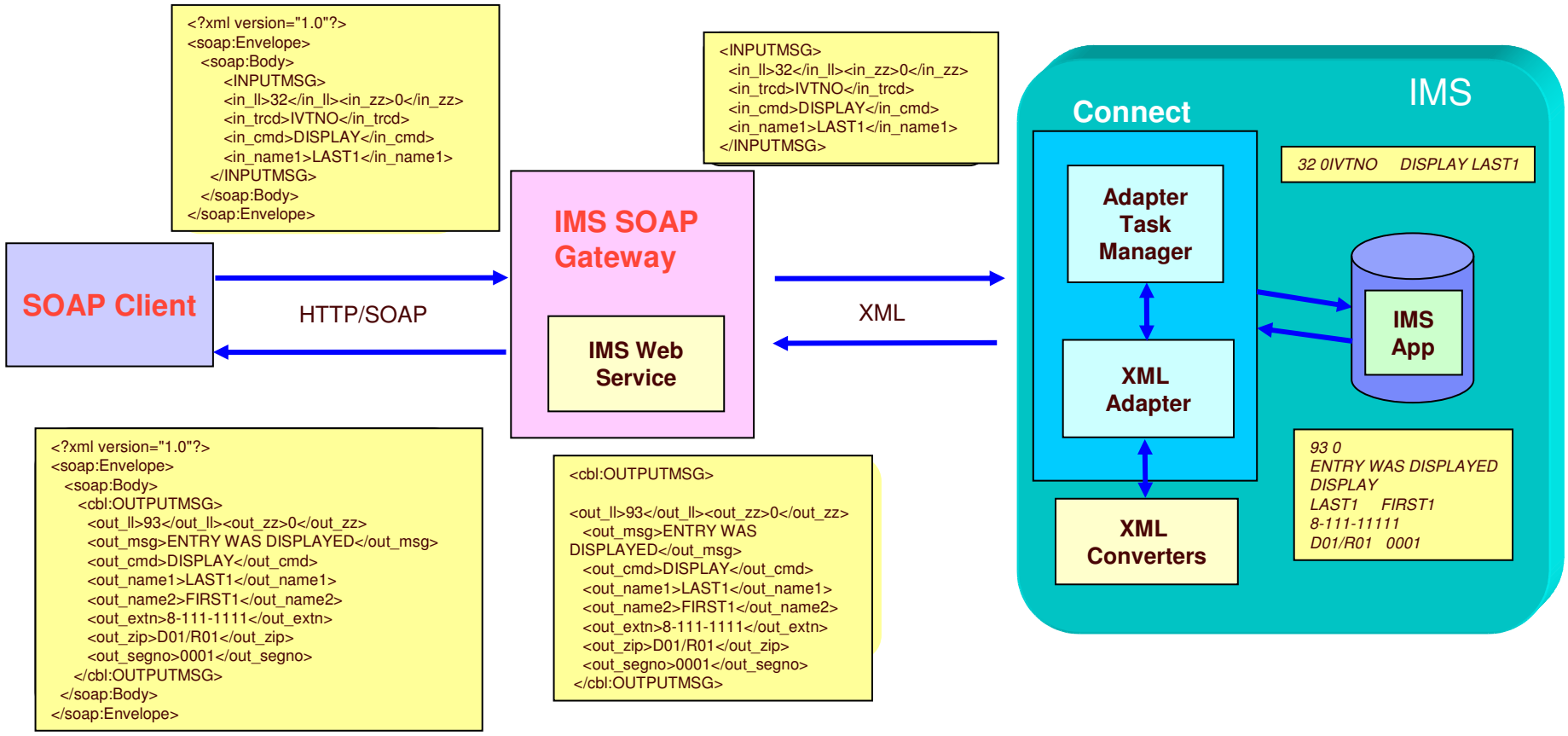
CICS usage of the WSBind file



IMS SOA Solutions – Service Provider and Service Consumer



Scenario – SOAP Client Invokes IMS COBOL Application as a Web Service

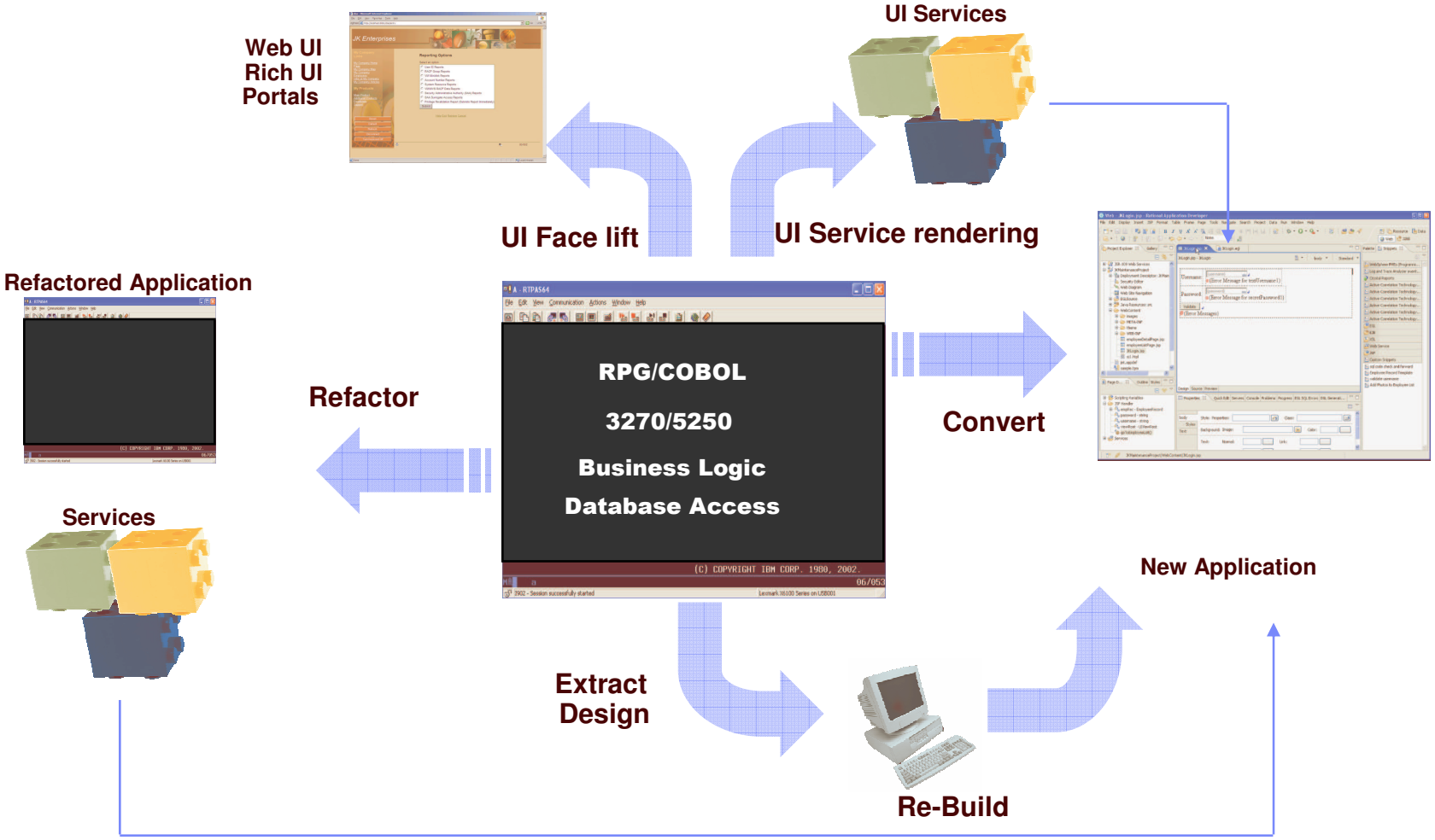


Agenda

Approaches to SOA

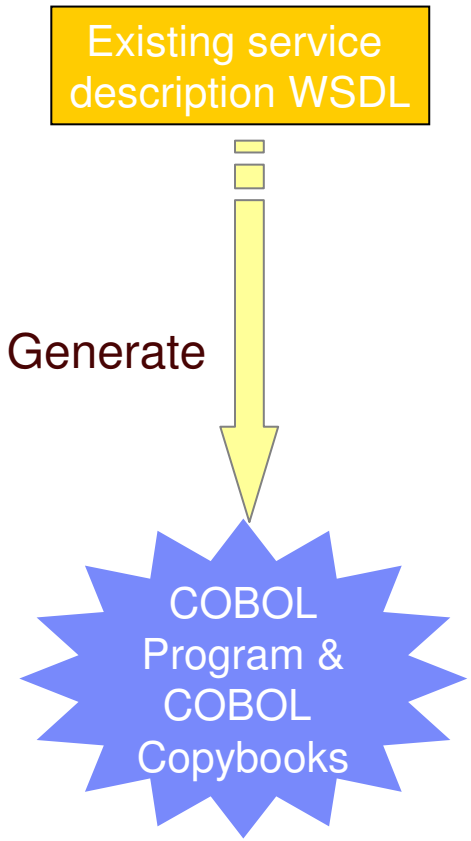
- Wrapping
- Re-Engineering
- Redeveloping

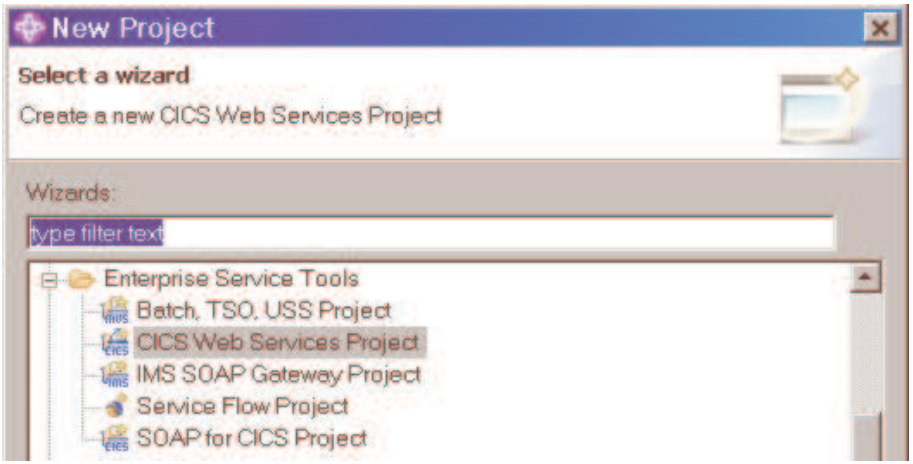
Application Modernization Options

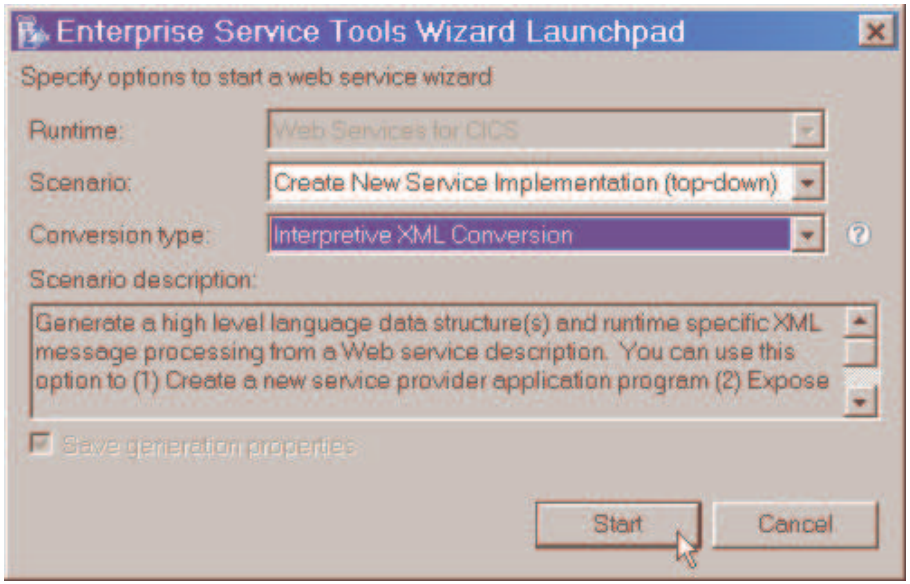


XML Services for the Enterprise (XSE): Example Top-down

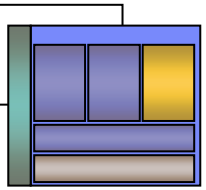
Top-down





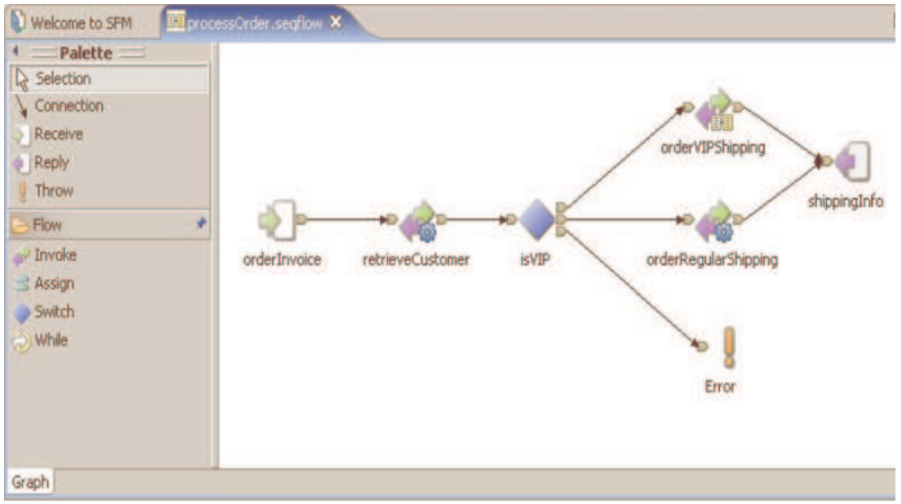


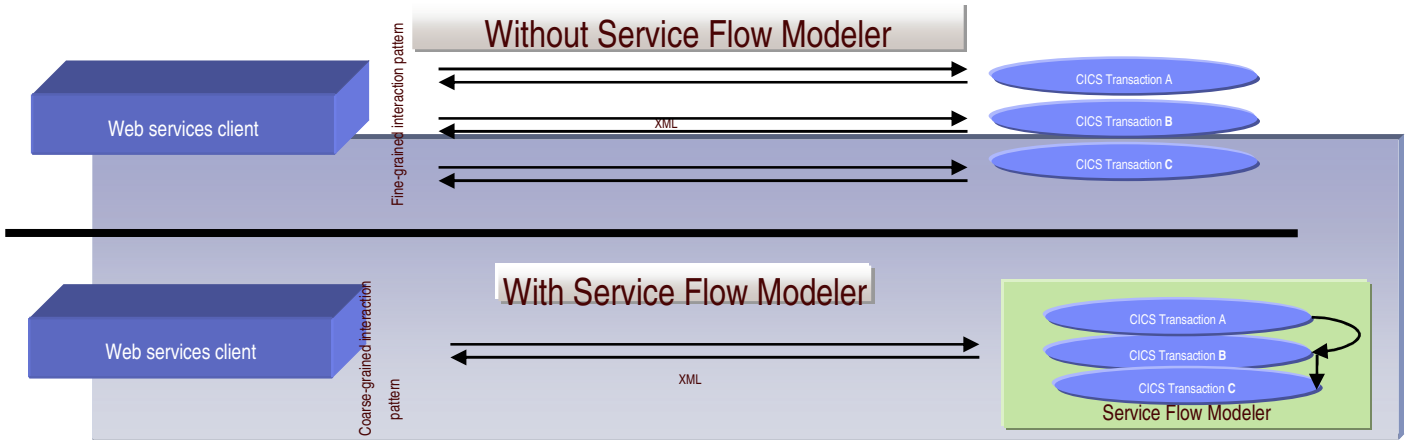
Orchestrate CICS services and screens



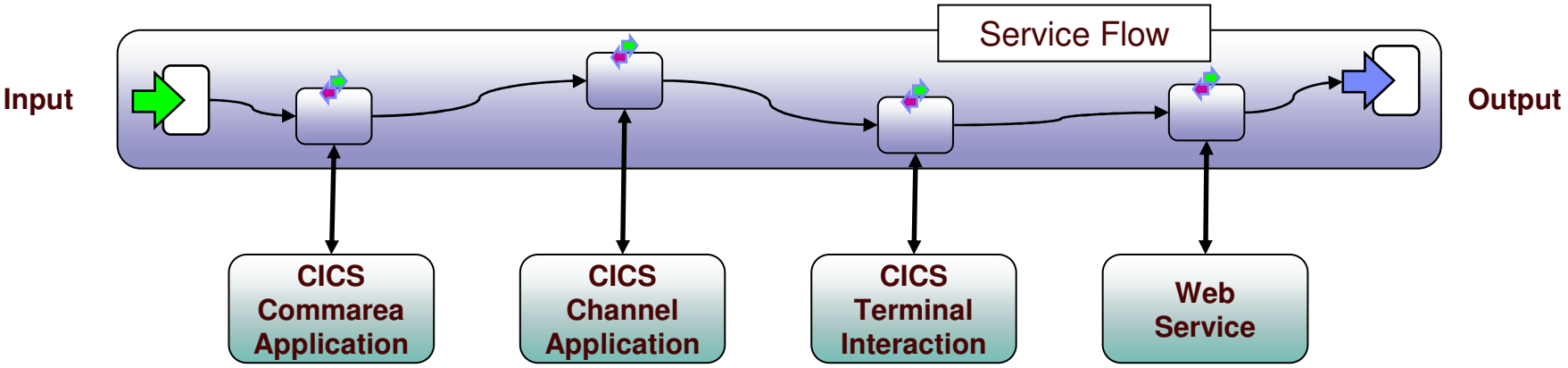
Model, Deploy, and Test Service Flows using Service Flow Modeler

- Aggregates multiple CICS transactions into high-level business processes through visual modeling
- Supports CICS BMS (terminal-based) applications & CICS commarea/container/channel applications
- Highly optimized CICS runtime supporting Web services and XML interfaces





What is a Service Flow?



Business Level

CICS Service Flow Feature provides the capability to aggregate existing CICS applications into composed business services which may be integrated into an SOA environment

- Aggregate multiple calls to CICS applications into one business level service call
- Automate the interaction with 3270 terminal based applications and expose as a business level service

Technical Level

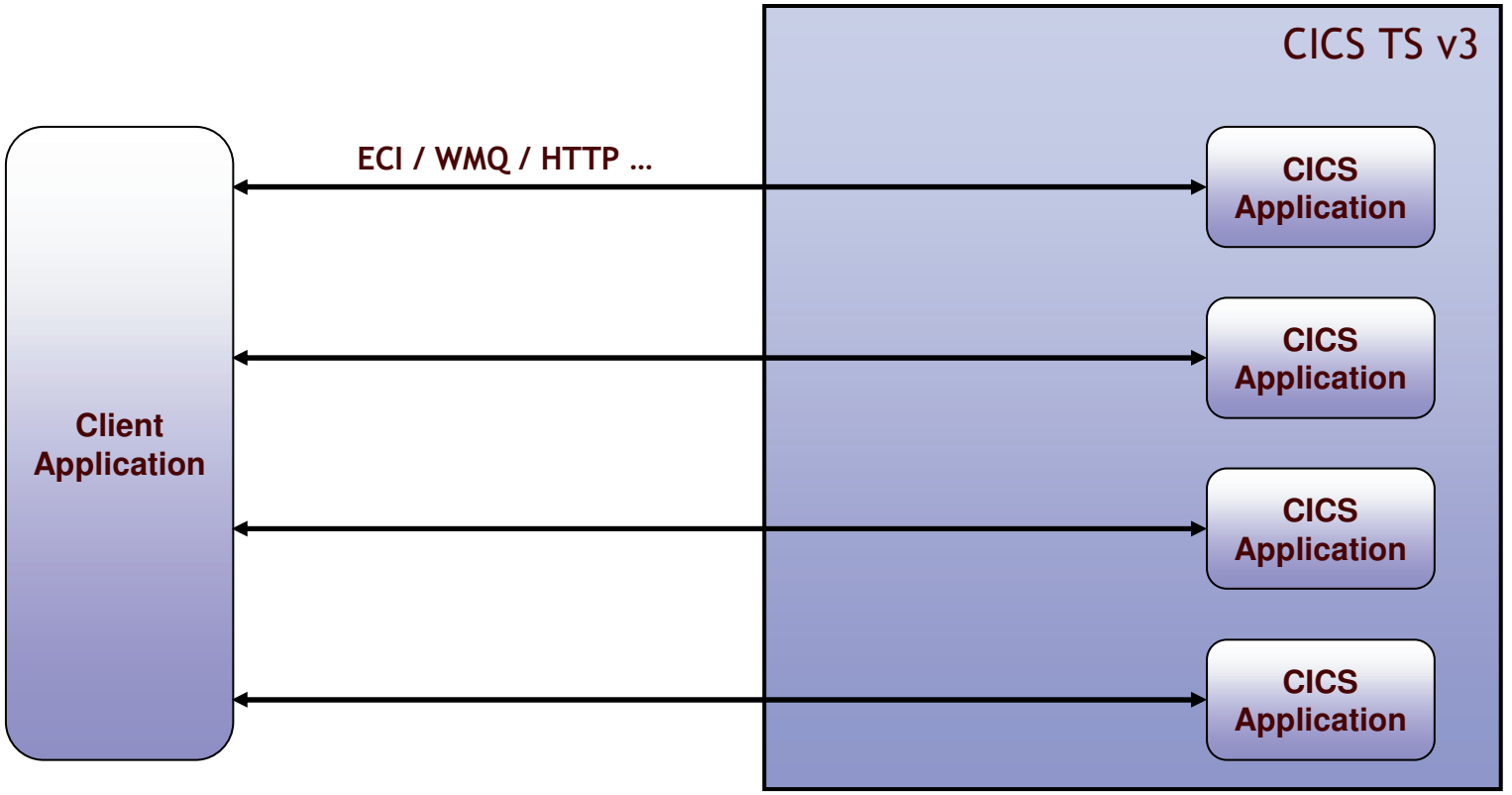
A service flow is a non-interruptible micro-flow that is constructed from a collection of nodes that represent the invocation of CICS resources

The flow describes the navigation of the nodes and allows data mapping between the nodes

A single request may cause the execution of many CICS resources

Allows for the development of coarse grained services from fine grained resources

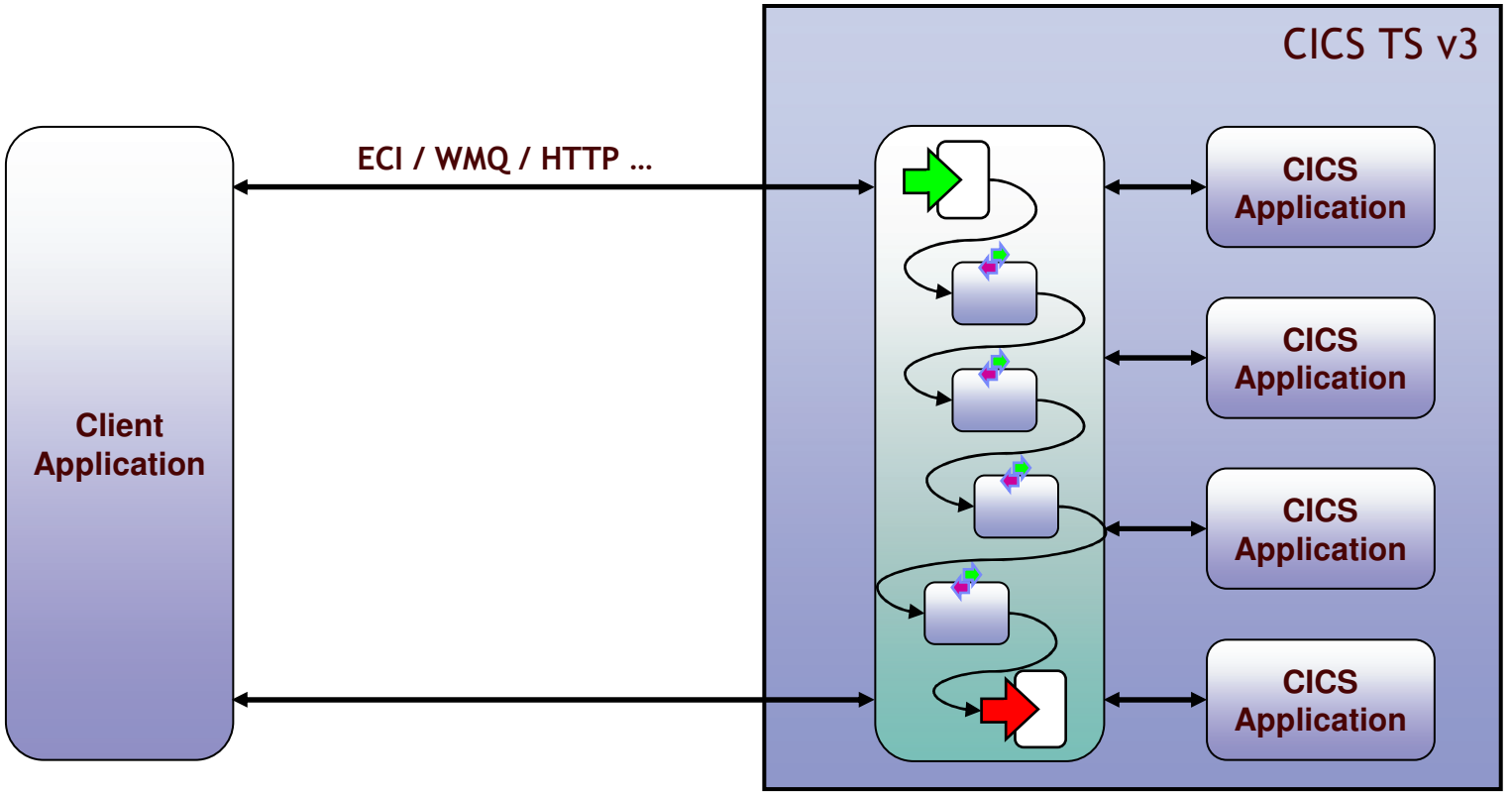
Traditional Access



Multiple requests from client application

- Expensive
- Low potential for reuse

Aggregated Access



Single request from client

- Potentially reusable component
- More efficient

How are Service Flows related to SOA?

SOA gives :

The flexibility to treat elements of business processes and the underlying IT infrastructure as standardized components (services) that can be reused and combined to address changing business priorities

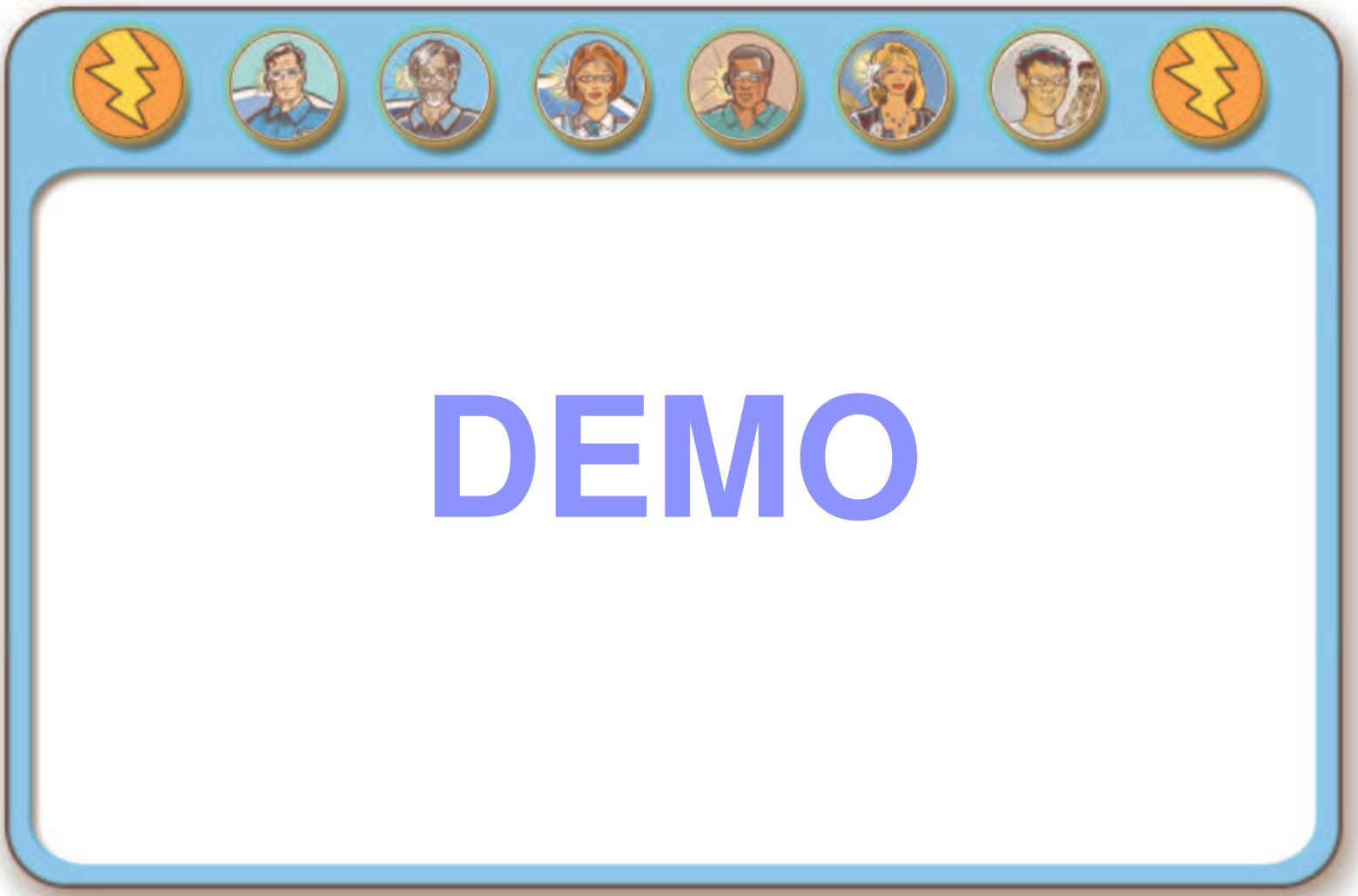
Services are the "building blocks" for business processes

Business processes can be offered as new services

Business processes can be composed from services

A Service flow is the aggregation of CICS assets into a new business service

That business service can be consumed by a business process


DEMO

QUESTIONS

THANK YOU

Learn more at:

IBM Rational software

IBM Rational Software Delivery Platform

Process and portfolio management

Change and release management

Quality management

Architecture management

Rational trial downloads

Leading Innovation Web site

developerWorks Rational

IBM Rational TV

IBM Rational Business Partners