



SOA Concepts and Web Services with z/VSE

Wilhelm Mild
IT Architect
IBM Germany

12-Apr-10

© 2010 IBM Corporation

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

* AS/400®, e business (logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Innovation that Matters in Today's IT

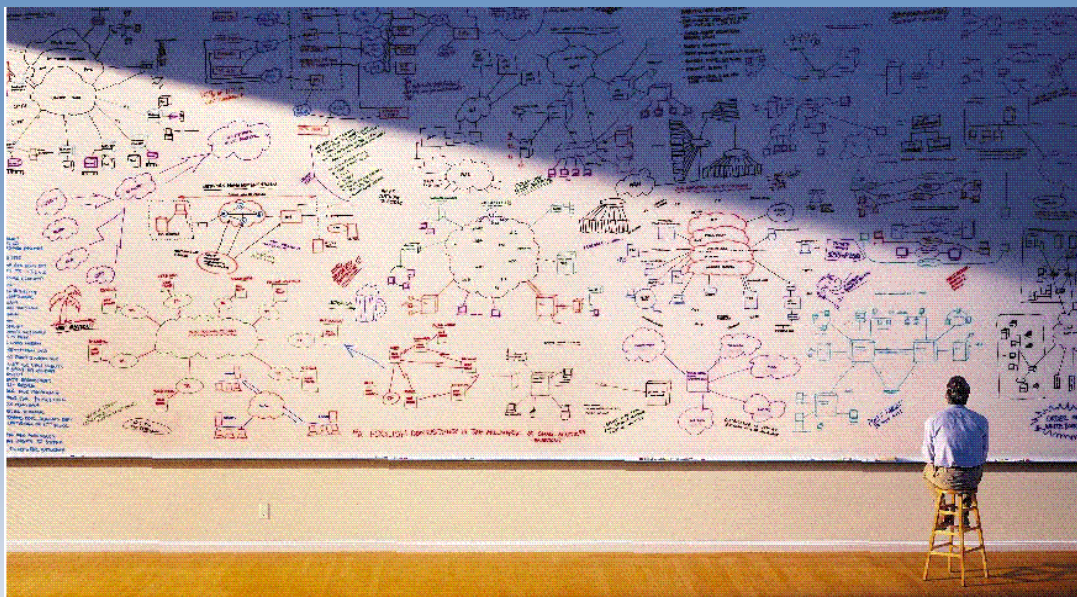
Top Innovation Priorities:

- Extend the ability to collaborate inside & outside
- Innovate business models & processes
- Leverage information for business optimization



*Innovation is all about change.
SOA makes it easier to change.*

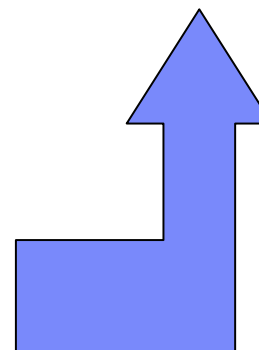
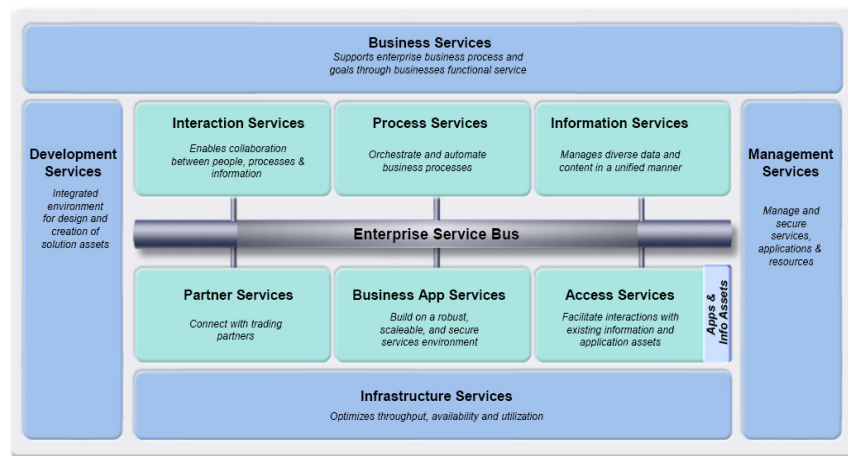
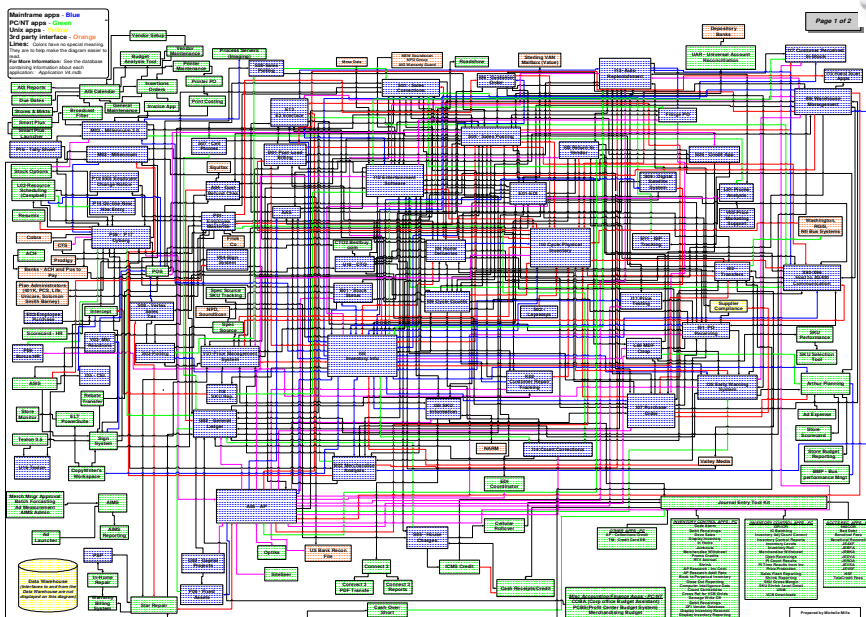
Why are interfaces so expensive to build and maintain?



- **Application interface logic is intertwined with business logic.**
- **The more tightly integrated the interface the more difficult the application is to change.**
- **The more interfaces that exist within a set of programs, the more complex the application becomes -- interface logic may, in many cases, exceed business logic.**
- **In such circumstances, re-use becomes difficult and impractical.**

Think different: business driven

- Complexity
- Lots of Interfaces
- Hidden Interfaces
- Proprietary Connections



Result → Much better Business process View

Think different with SOA:

With SOA to a Business Process driven IT

Traditional Thinking

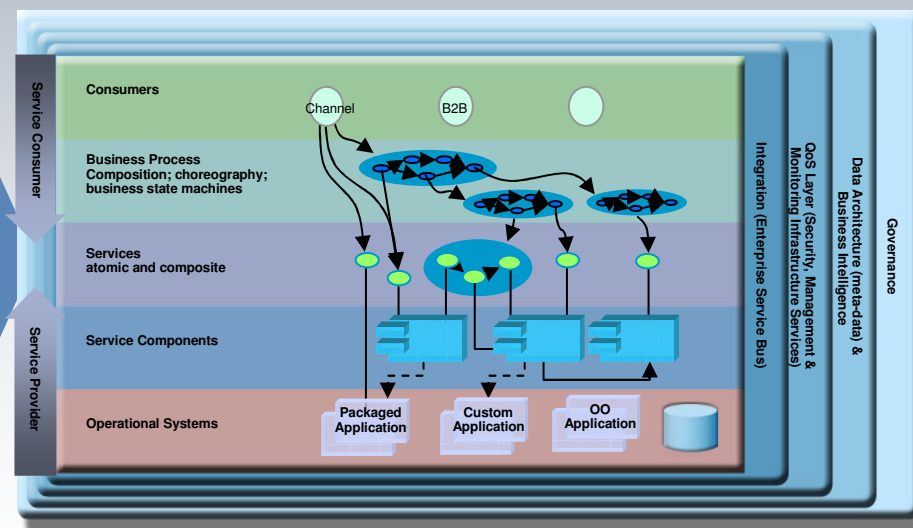
IT manages IT **assets** that support the business – implementation thinking



Silos, static

Business Thinking

IT manages **Services** and **Components** which **reflect** the Business processes – functional thinking



Flexibility, dynamic, virtualised

What is Service-Oriented Architecture (SOA)?

Business Definition

A SOA is a **framework** that provides:

- **flexibility** to treat elements of business processes and the underlying IT infrastructure as secure, standardized components

- **services**, that can be **reused** and **combined** to address changing business priorities



Technical Definition

A SOA is an **IT architecture** in which applications are built as **components and services** – that are **loosely coupled** and **well defined** to support **interoperability**, and to improve **flexibility** and **reuse**

With a Service oriented Architecture, (SOA) it is very efficient to align the business processes to changed Conditions and Requirements.

Similar to a Soccer game

Player = Service ●
Game field = SOA Infrastructure



SOA = flexible composition of the individuals to a team, to win a match dependent of the conditions and requirements.



4-3-3



4-4-2

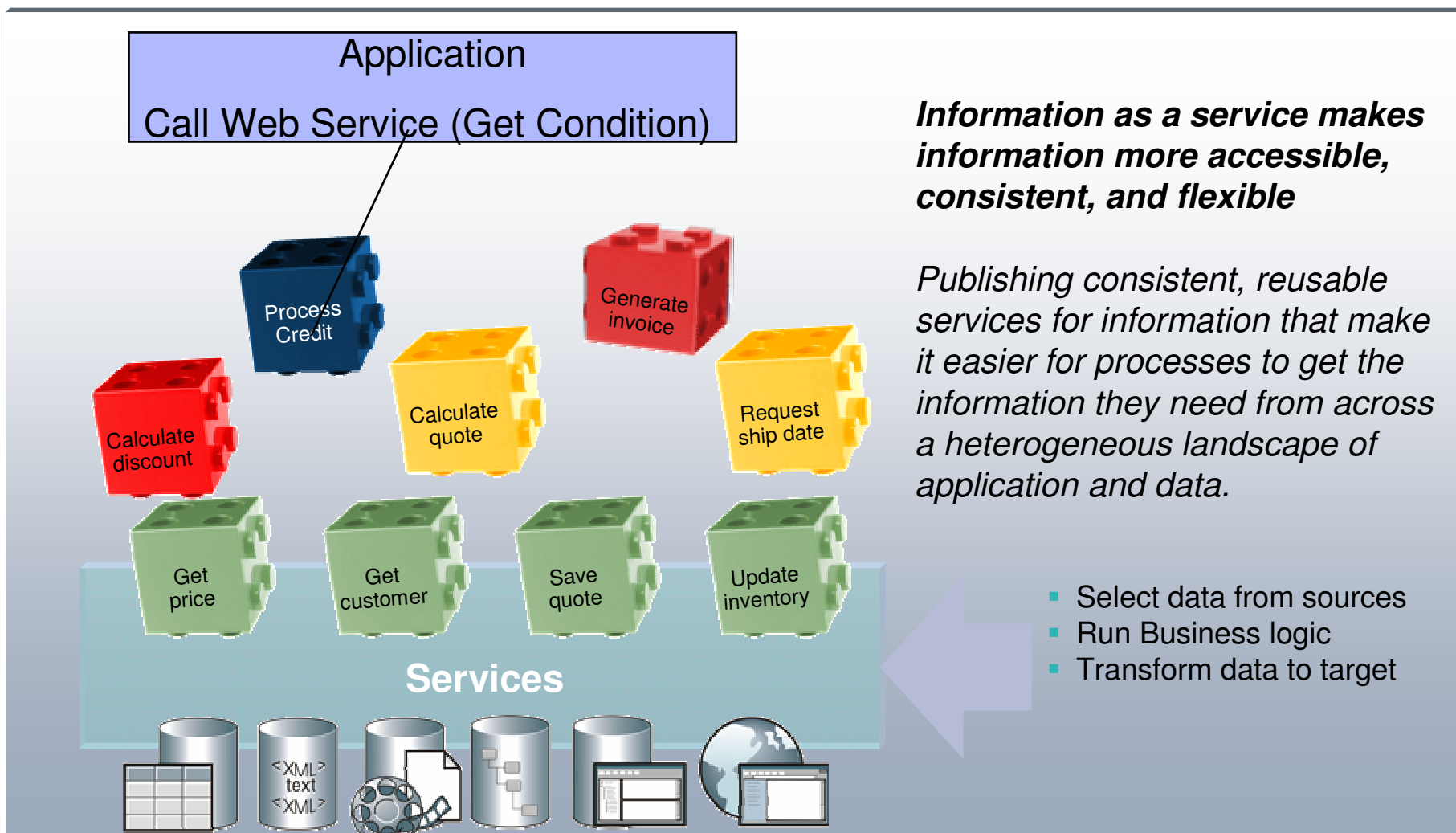


3-5-2

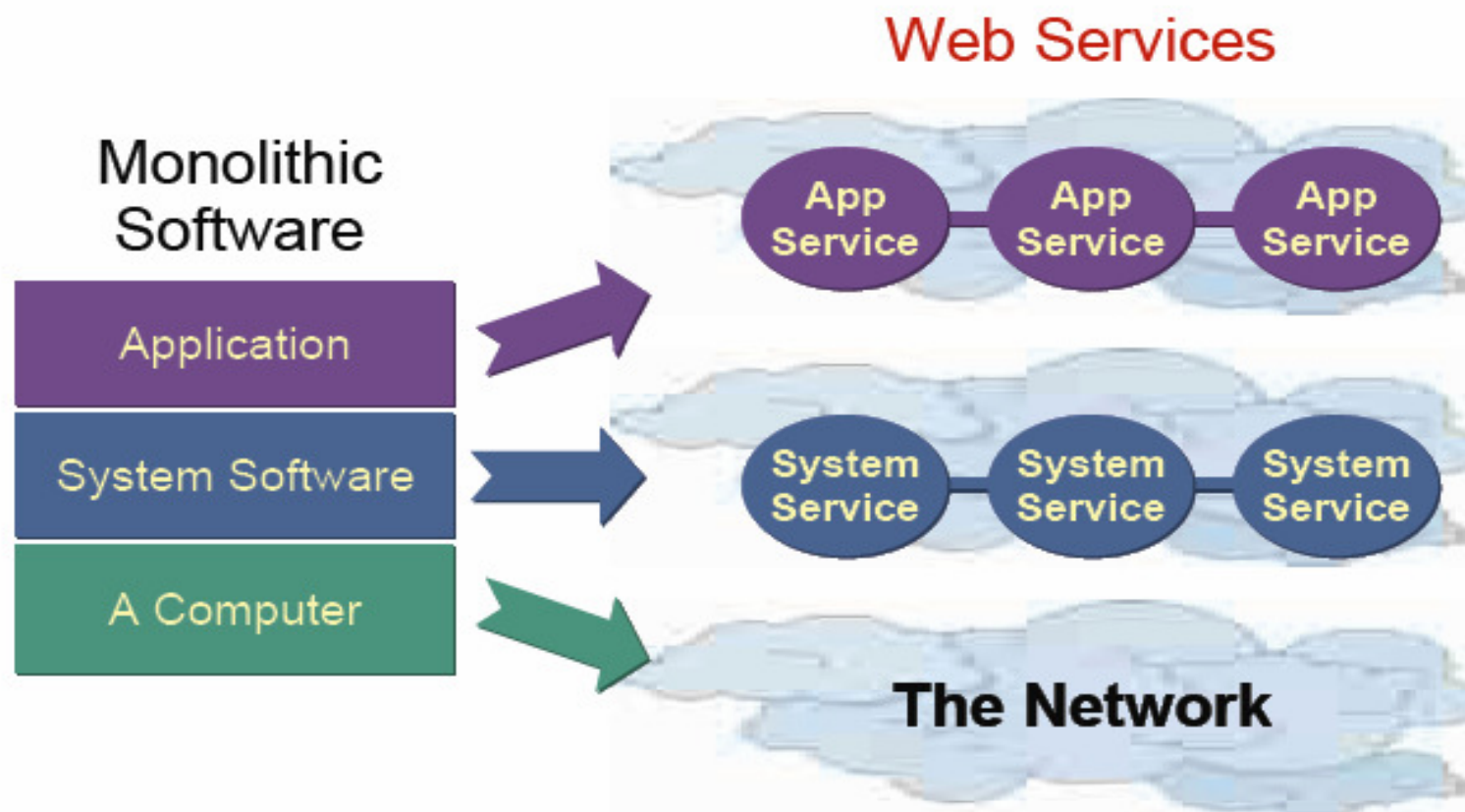


3-6-1

Integrating Logic in an SOA

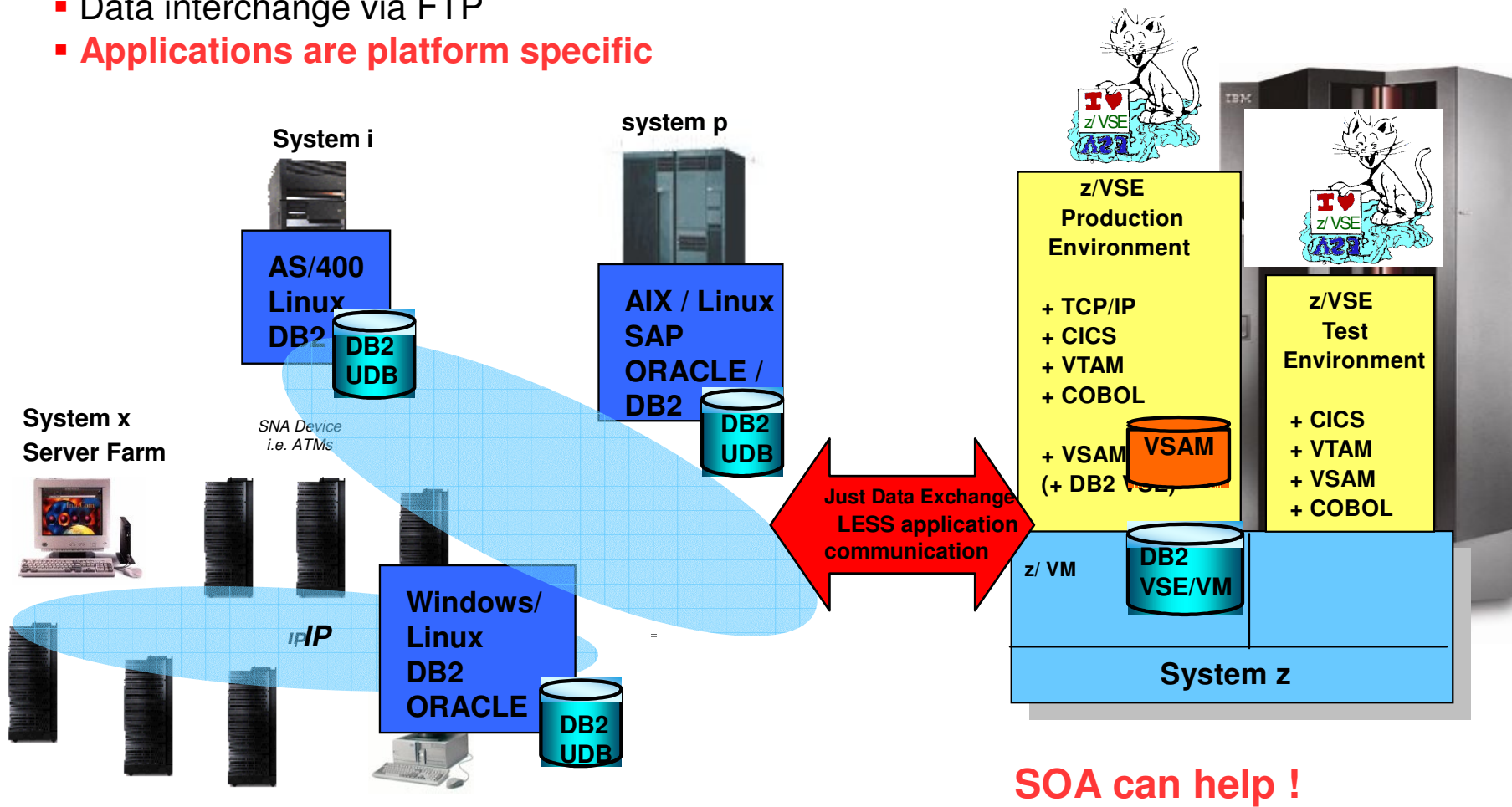


Application Dis-Integration using Web Services



Typical VSE Customer Environment Data interchange

- VSAM data on VSE (some DB2 environments)
- Relational databases on distributed platforms (DB2, Oracle)
- Data interchange via FTP
- **Applications are platform specific**



SOA can help !

The Challenge: Bringing People, Process, and Information Together for True SOA Value



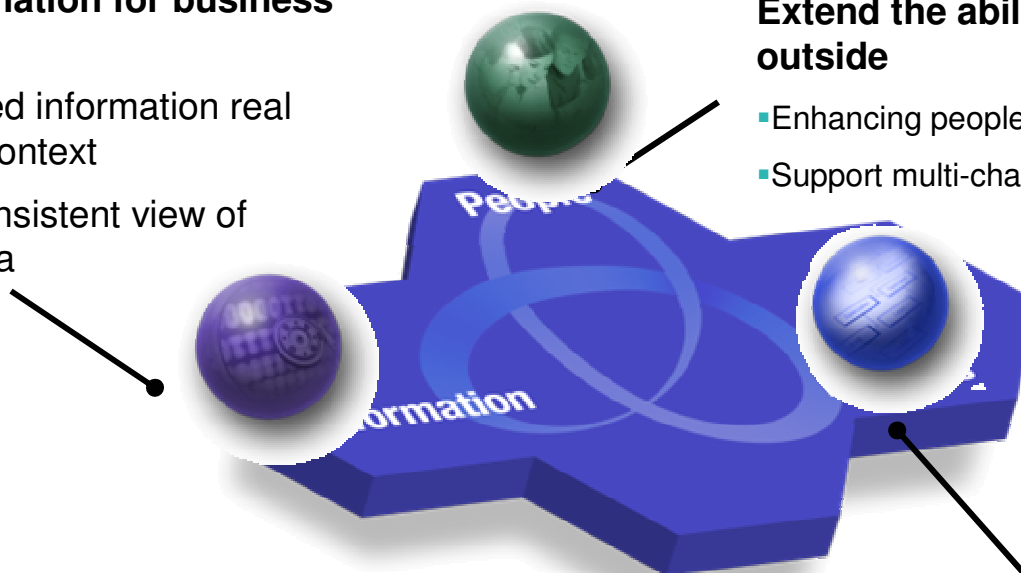
Business Centric SOA Starts with Your Most Critical Business Pain and Enables You to Build for Flexibility

Leverage information for business optimization

- Deliver trusted information real time and in context
- Provide a consistent view of business data

Extend the ability to collaborate inside & outside

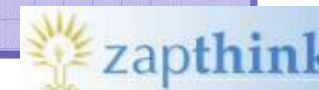
- Enhancing people to people collaboration
- Support multi-channel delivery



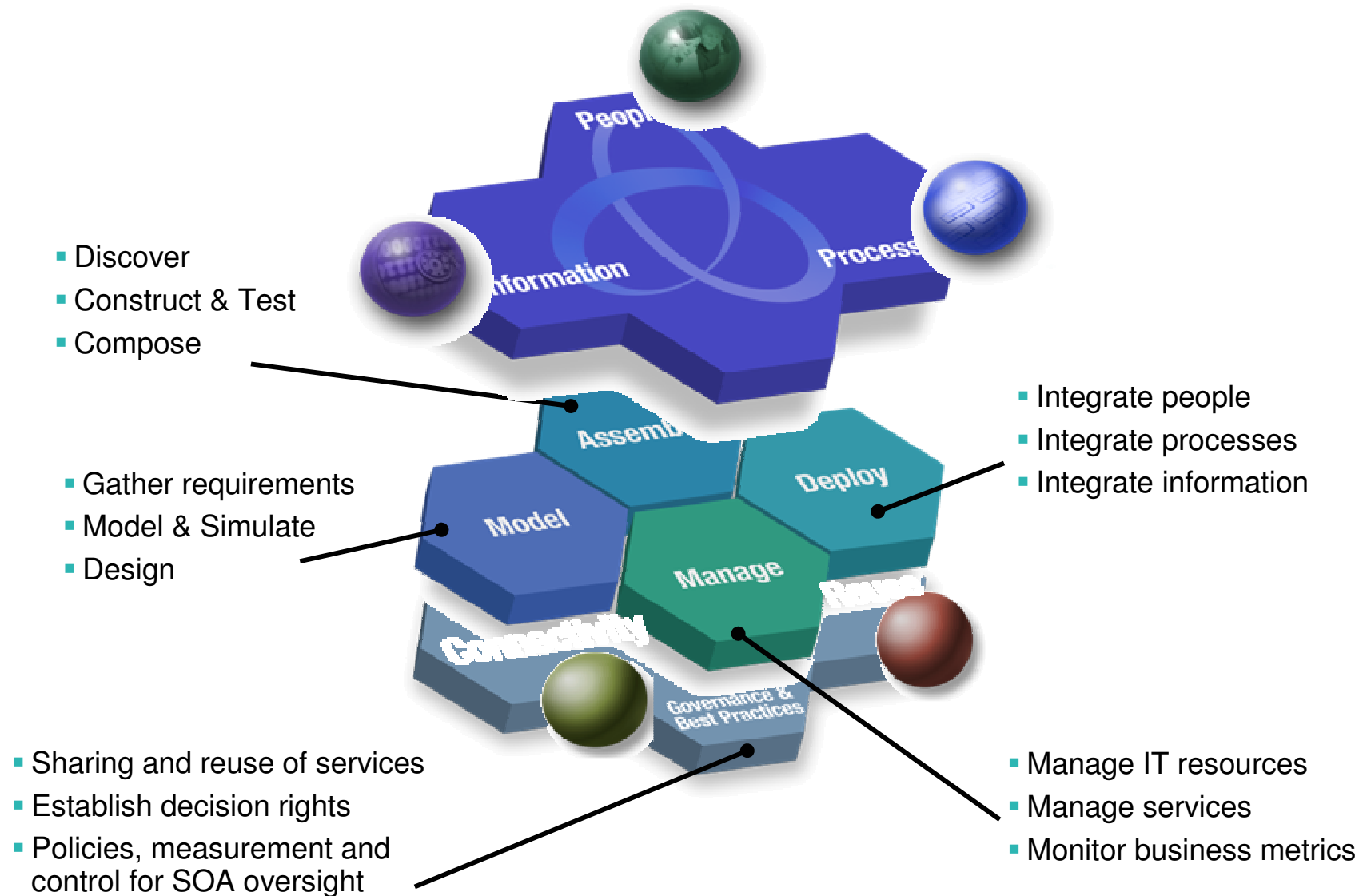
Business model & process innovation

- Seamless coordination between automated and people/information driven processes
- Increase organizational effectiveness

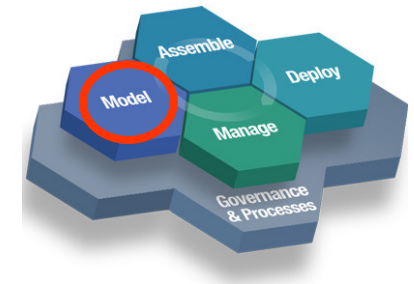
"Technology is certainly key to a successful SOA integration, [but] it is the coalescence of the people, process and information [with] business insight and emerging best practices for overall organizational change through SOA."



And SOA Lifecycle Is The Key to Successful Projects



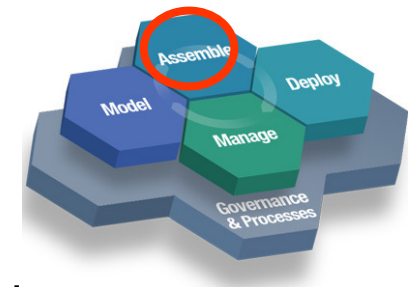
SOA lifecycle - Model



Focus on Business Requirements:

- The first step in a SOA project has little to do with technology and everything to do with your **business**.
- Service orientation, treats the activities performed by your business as services
 - the first step is to establish **business activities** or processes
 - Documenting your business architecture can be used not only to plan your SOA, but also to optimize your actual business processes.
- The level to which you model your business process is going to depend on the depth of your anticipated implementation.

SOA lifecycle - Assemble

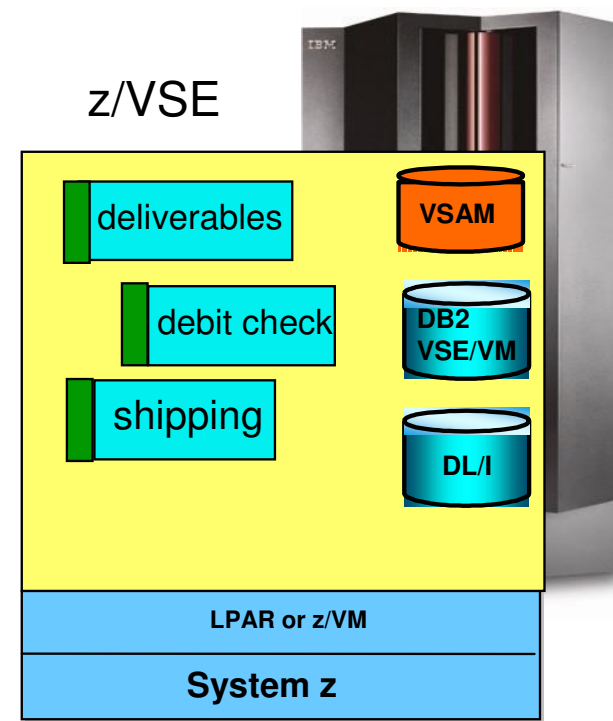
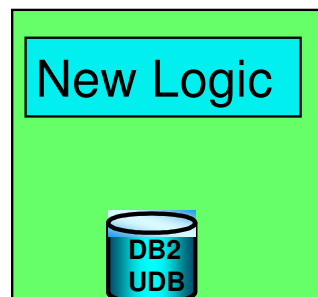
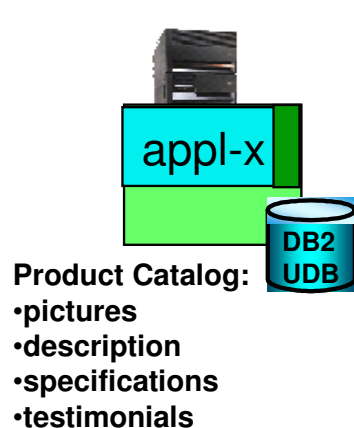


Focus on defining service components:

- Once business processes have been modeled and optimized, developers can build them by **building new services** and/or **reusing existing services**, then assembling them to form composite applications.
- In the Model step, you determined what kinds of services you needed and what kinds of data they would access.
 - Some of the software you needed to implement those services, or access that data, already exists in some form.
- The Assemble step is about finding functionality that already exists and **service-enabling** it.
- It is also about creating new services that provide functionality and access to data sources that you need to meet the range of business processes to be addressed by your SOA

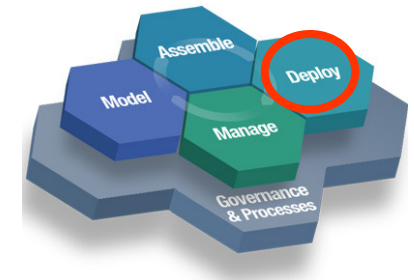
SOA – the way to New applications and processes

- Applications look the same for all users
 - Core applications can be enhanced (independent of their language, COBOL, ASM, PL/I)
 - New business logic can be built
- Increases business for the Company**



Integration of Processes

SOA lifecycle - Deploy

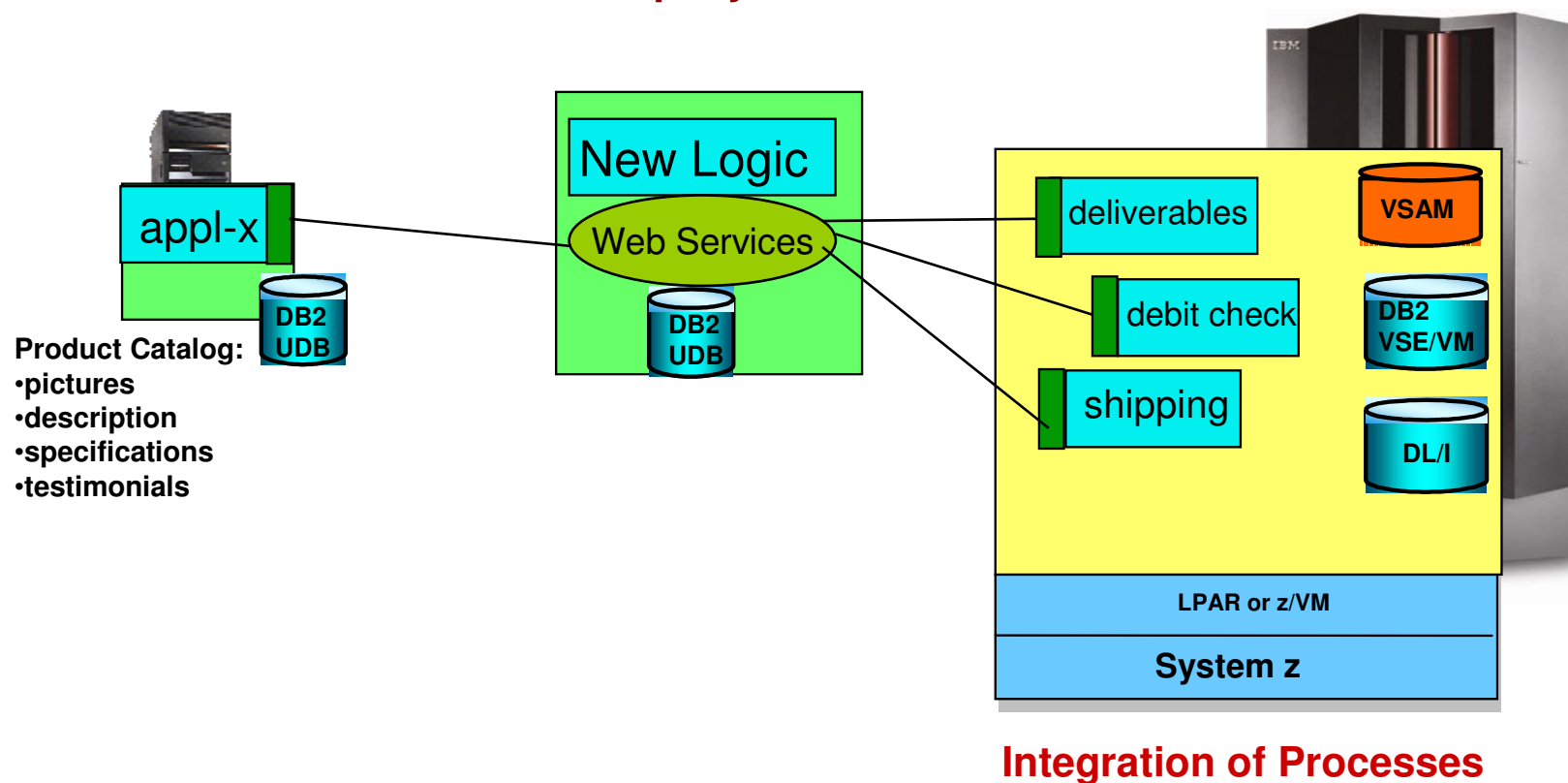


Focus on enabling service components:

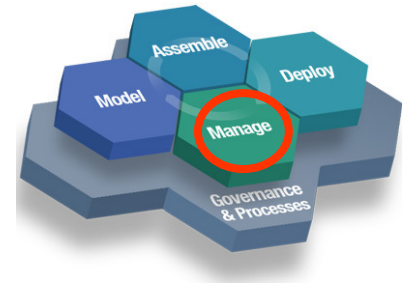
- the assets that make up your SOA are deployed
 - into a secure and integrated environment.
- This environment avails itself of specialized services that integrate people, processes, and information within your business.
 - This level of integration helps ensure that all the key elements of your company are connected and working together.
- The deployment needs to meet the **performance and availability** needs of your business
 - provide the **flexibility** to incorporate new services
 - decommission of old services - without major impact to the system as a whole.

SOA – the way to New applications and processes

- Applications look the same for all users
- Core applications can be enhanced (independent of their language, COBOL, ASM, PL/I)
- New business logic can be built
Increases business for the Company



SOA lifecycle - Manage

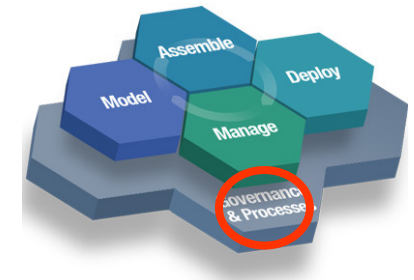


Focus on managing business process as service:

The system is in place and everything is running smoothly.

- You can just let everything run by itself now, right? Wrong.
- Once deployed, your system needs to be **managed and monitored**, both from an IT and a business perspective.
- In this step, you **monitor and optimize** the system, finding and correcting inefficiencies and problems.
 - You will need to deal with issues such as **quality of service**, **security**, and general system **administration**.
- The completion of this step is the start of a new Model step.
 - The data gathered during the Manage step is used to revisit the entire SOA lifecycle and start again.

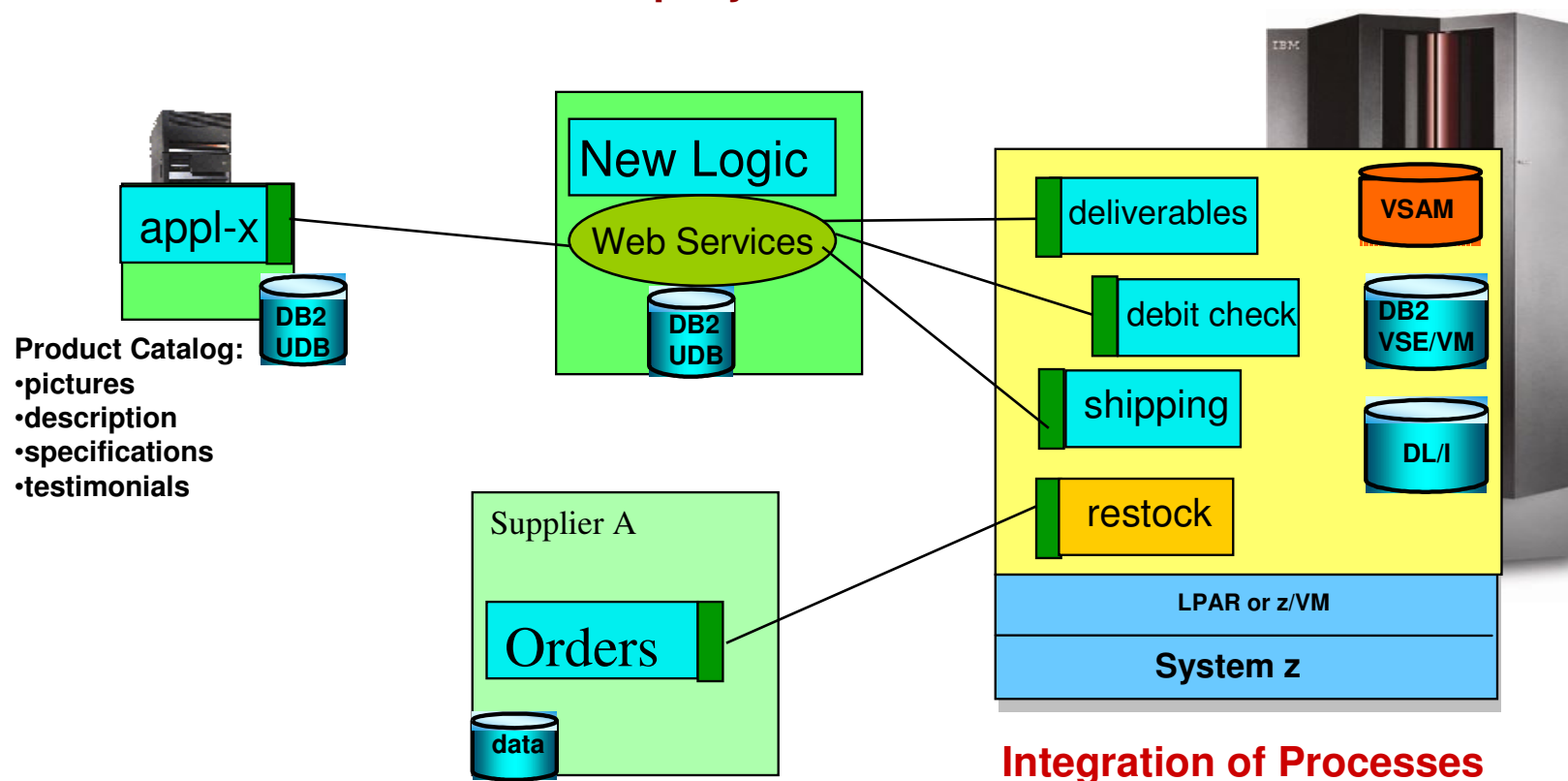
SOA lifecycle - Governance



- An SOA is a decentralized system
 - it can include services from different parts of the organization
 - services from different platforms
 - services that come from outside the organization.
- Without proper governance, the system can easily spin out of control.
- Governance underpins all the lifecycle stages, providing guidance and oversight for the entire SOA system.
- It provides both **direction and control**, shielding both service providers and consumers from encountering the unexpected.

SOA – the way to New applications and processes

- Applications look the same for all users
- Core applications can be enhanced (independent of their language, COBOL, ASM, PL/I)
- New business logic can be built
Increases business for the Company



SOA adoption phases:

- It's not likely that you will implement a complete business transformation around SOA all at once.
- In fact, an **all-or-nothing approach increases risk of failure**.
- Instead, you'll work your way iteratively through the stages of adoption, starting by developing a few services for a **pilot project**
 - and then increasingly revamping your IT systems as services that work within an SOA.
- We'll look at the following SOA adoption phases:
 - Build services
 - Integrate
 - Transform IT
 - Transform the business

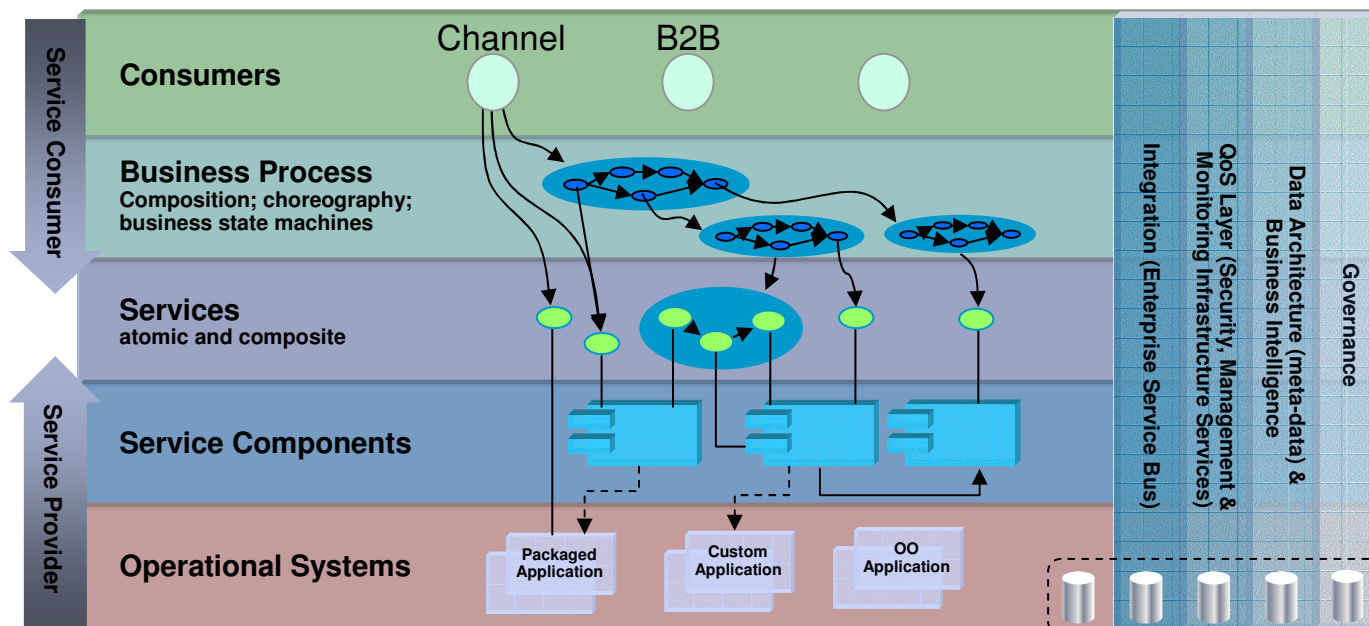
Build services: As needed services with ad hoc linkage

- In the first phase of SOA adoption, companies typically build SOA services opportunistically.
- In other words, as they need to **solve specific problems**, they choose a service-oriented approach rather than a traditional one.
- In this phase, service building will be more focused on solving a specific problem than on transforming the enterprise.
- The IT department will build some new services, perhaps transform some existing applications into a set of Web-based services.
- Linkage between them will be provided on an **as-needed basis**, not according to a thorough architecture.

Integrate: Systematic standardized service interfaces

- Once you've discovered the advantages, convenience, and maintainability of a **loosely coupled** architecture, the next step is to start putting that flexibility to work creating new, composite applications by combining services.
- For example, an employee status service can be combined with a manager approval service to create a vacation request service.
- This process can happen from the top down, that is focusing on the end result and looking for component pieces.
- Or it can take place from the bottom up, focusing on the pieces and seeing what can be built from them.
- Linkage between them is **preplanned and well-defined**.

SOA Solution Layers



Service Consumers

- Portal, B2B, Standalone, .Net

Business Process Layer

- BPEL Processes

Service Definition Layer

- WSDL, XML Schema, WS-Policy

Service Component Layer

- Service Platform based service facades: J2EE, .Net, SCA etc.

Operational Systems Layer

- ISVs, Middleware, Custom Apps, Platforms, Network



The SOA Architecture - Standards

- **Web Services**
 - defined Services

- **XML** (eXtended Markup Language)
 - platform independent data representation

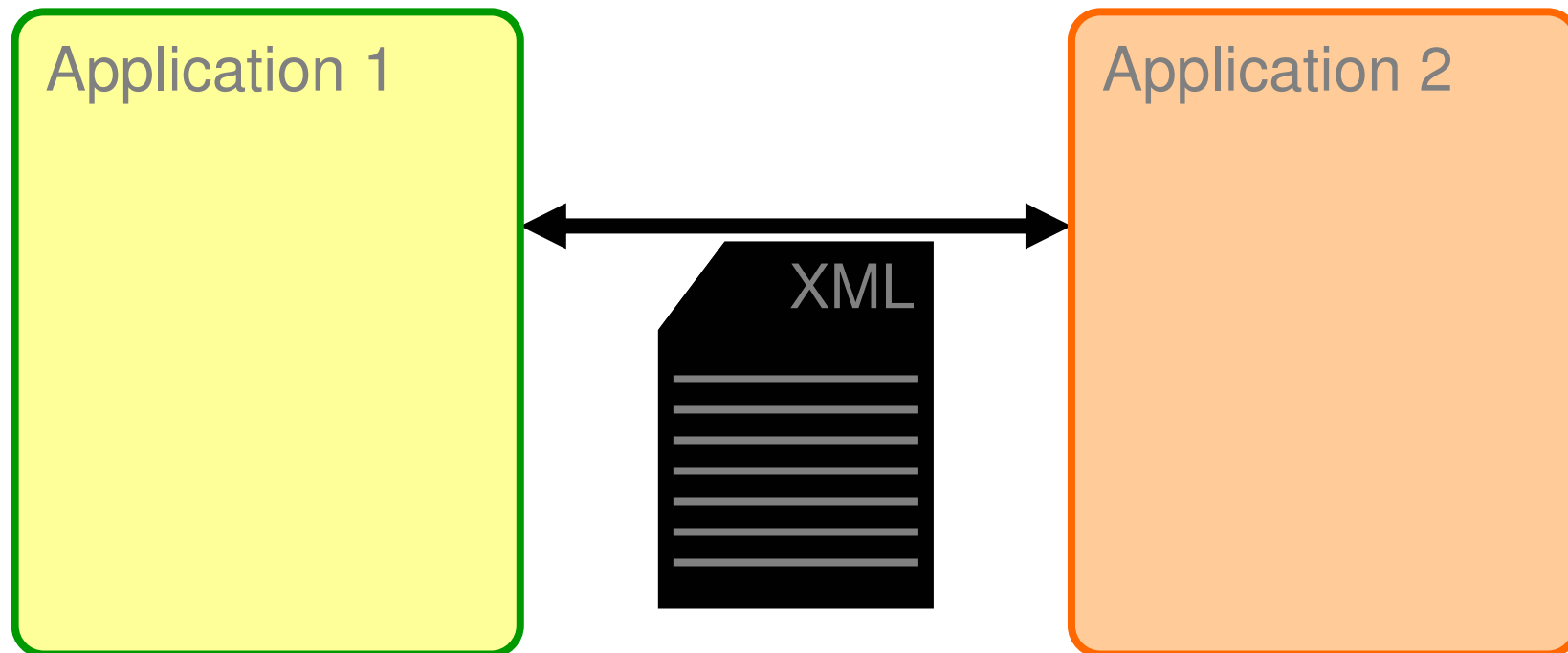
- **SOAP** (Simple Object Access Protokol)
 - protocol for Web Services

- **UDDI** (Universal Description Discovery Integration)
 - catalog to register and find Web Services

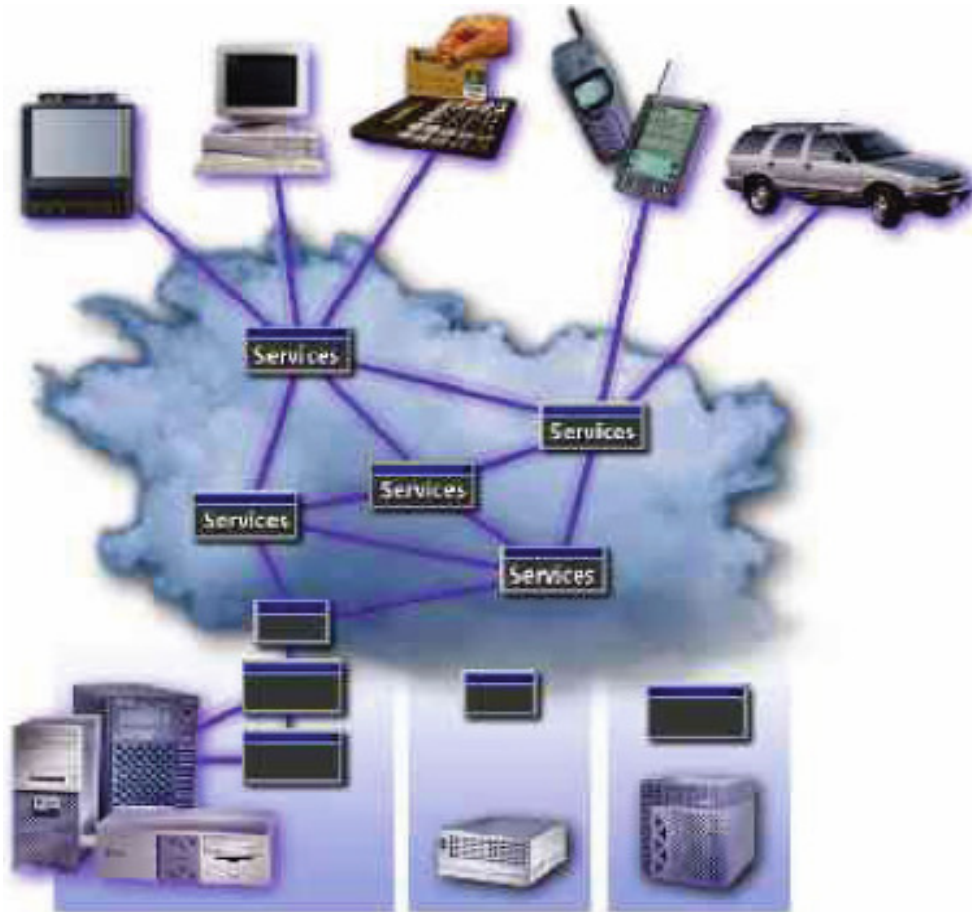
- **WSDL** (Web Services Description Language)
 - language in which the Web Services describes

- **Enterprise Service Bus**
 - The Plug for the Services

What are Web Services? Applications !



Why should we choose Web Services ?

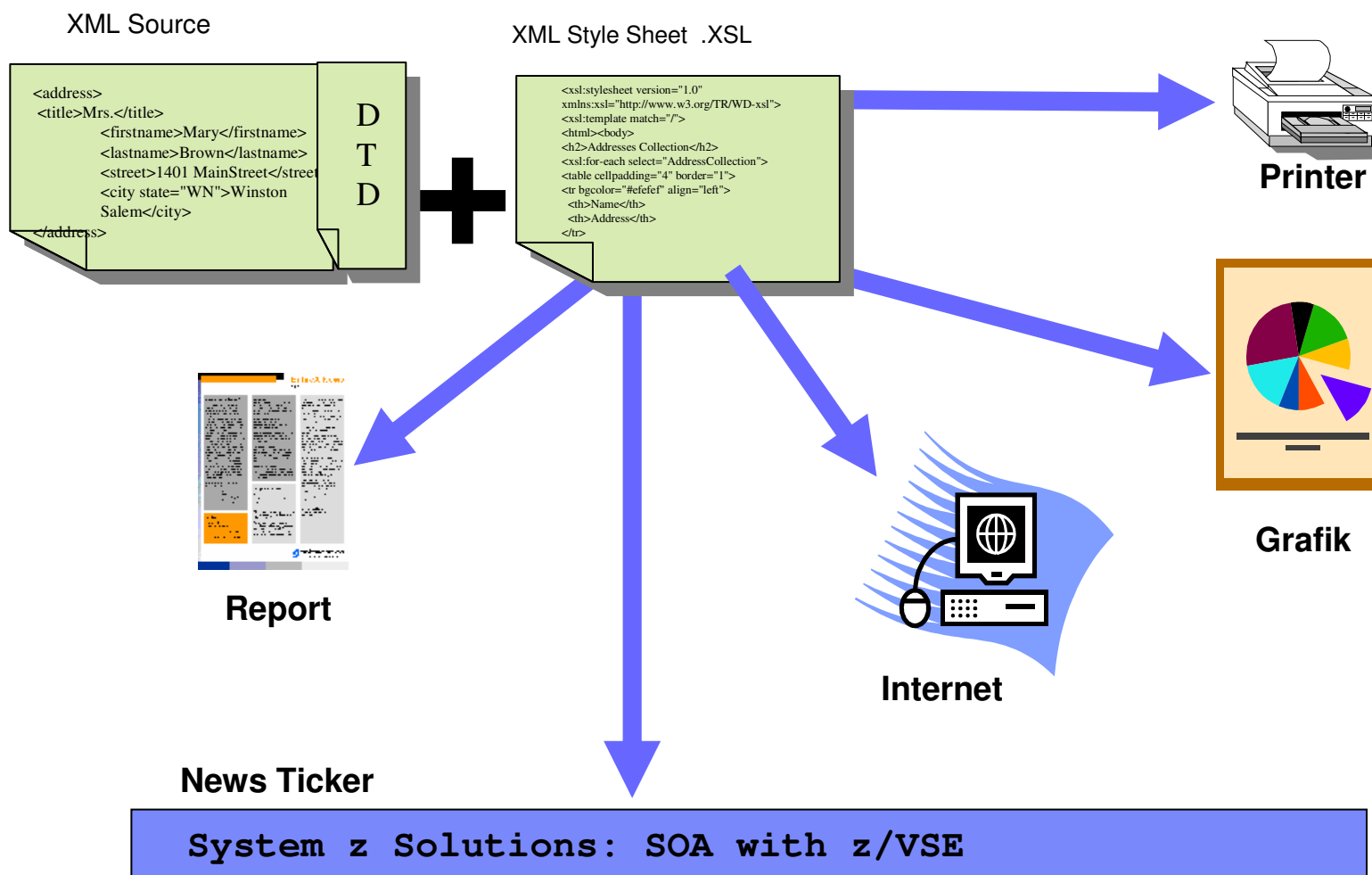


Web Services:

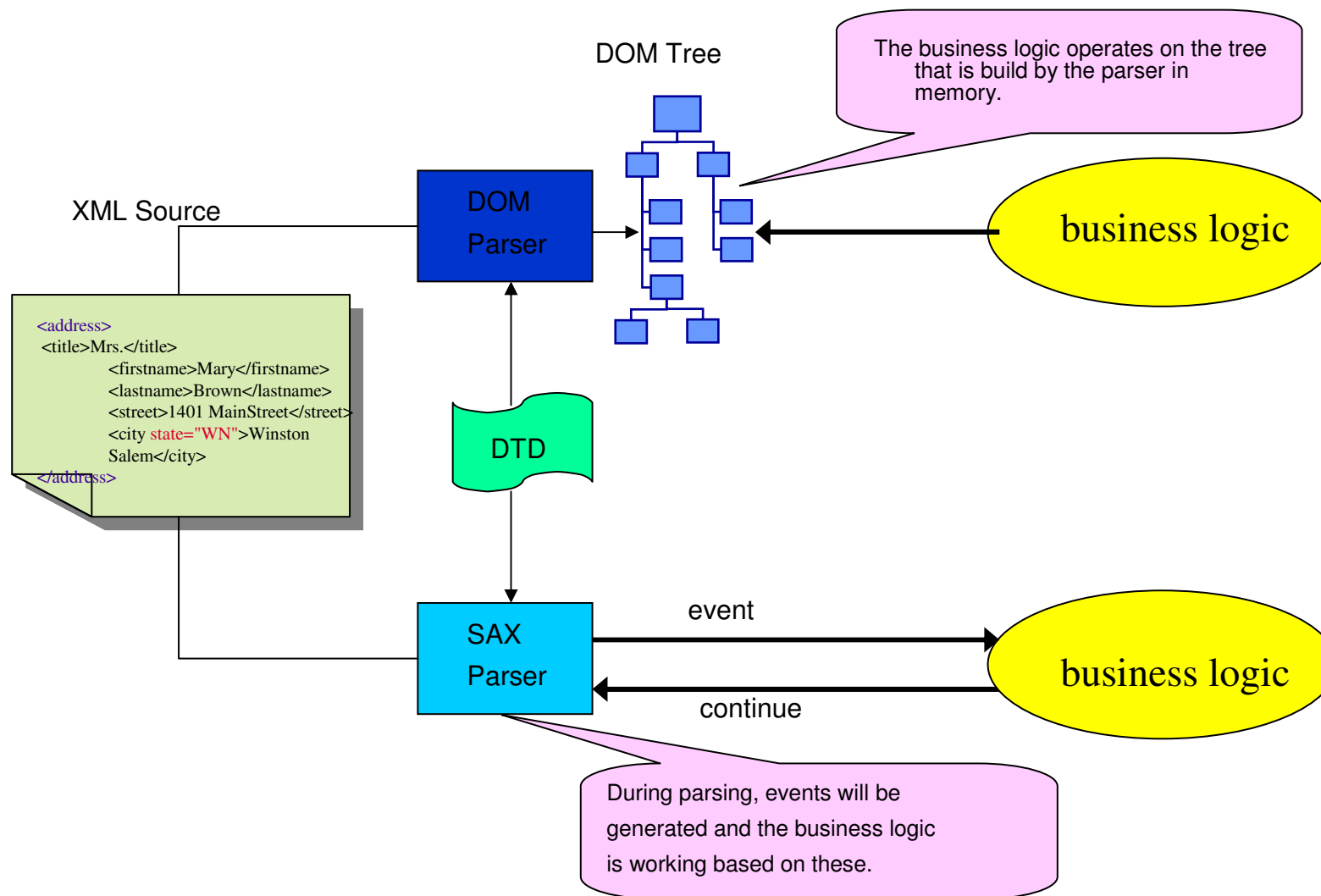
- Are platform neutral
- Are accessible in a standard way
- Are accessible in an **interoperable** way
- Use simple and ubiquitous plumbing
- Are relatively cheap
- Simplify **enterprise integration**

What is XML ?

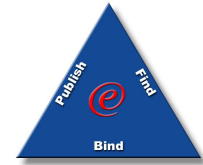
THE platform independent data representation !


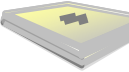


XML parser Technology (DOM,SAX)



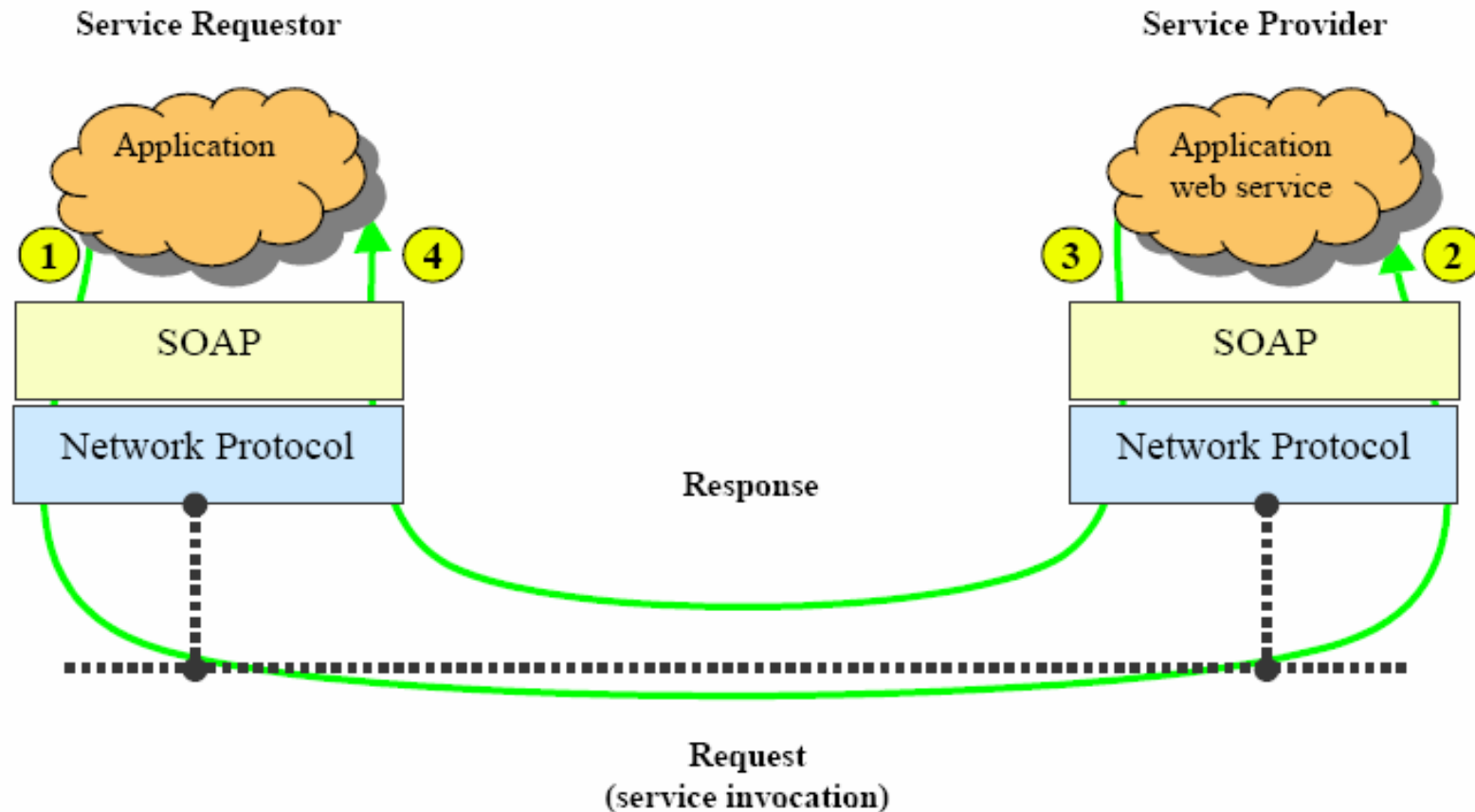
What is SOAP?



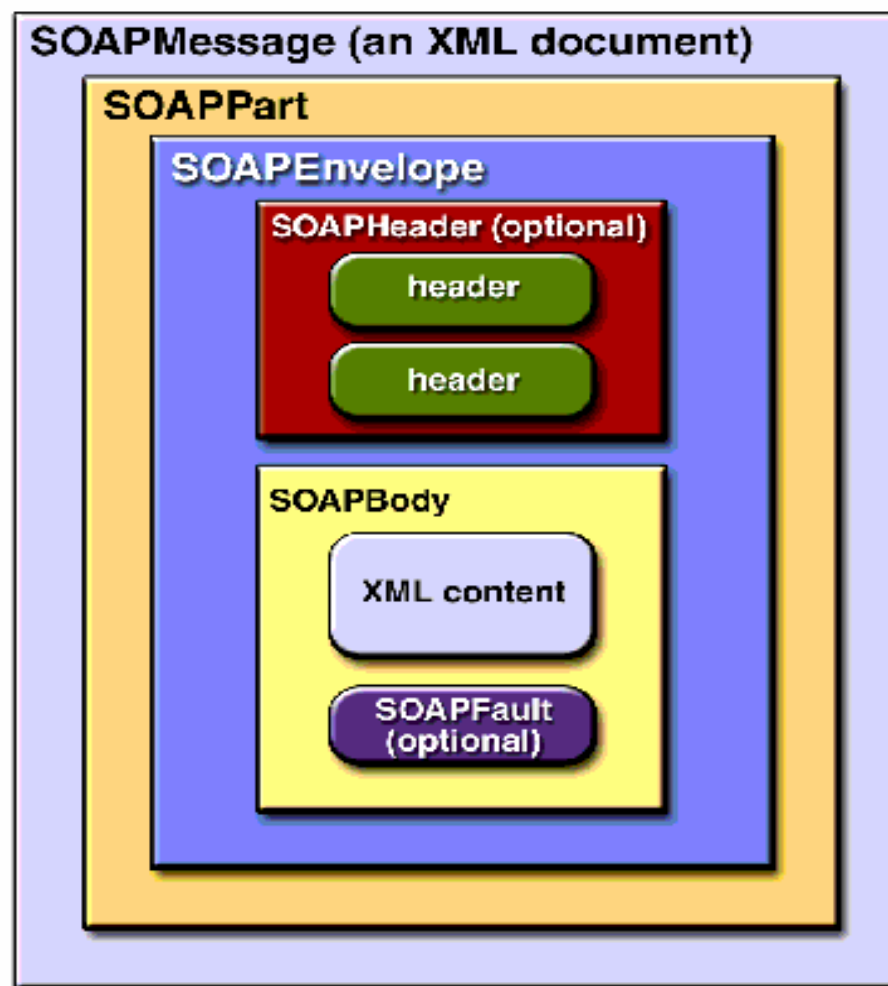
- **Simple Object Access Protocol**
- **SOAP is an XML based protocol for communication between two remote applications:**
 - is based on RPC messaging
 - is language independent (de-couples interface from implementation)
 - represents remote procedure calls and responses
- **A SOAP message consists of:**
 -  **envelope**
 - wraps the message itself
 - defines rules for decoding the message
 -  **message**
 - request
 - method to invoke on a remote object and parameters
 - response
 - result of running the method and exceptions

What is Simple Object Access Protocol (SOAP)?

Application communication protocol with XML !

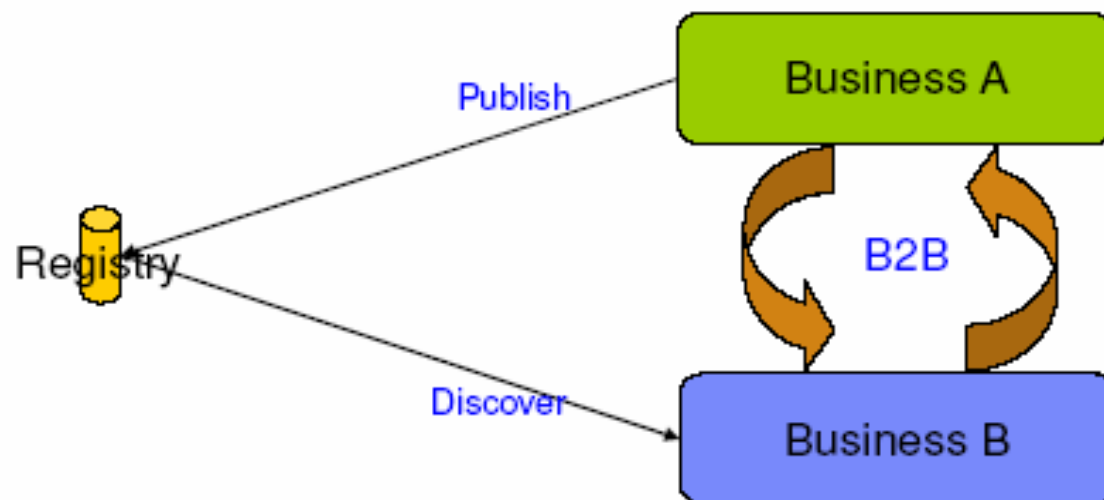


SOAP without Attachments



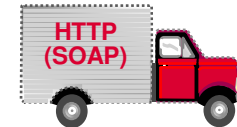
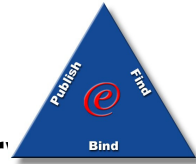
What is a Registry?

- An infrastructure that enables the publishing and discovery of Web Services
- Facilitates business-to-business (B2B) interactions



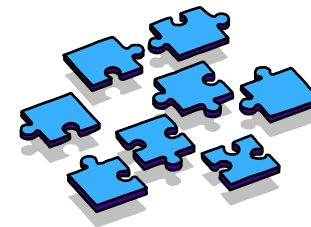
What is WSDL?

- **Web Services Description Language**
- **WSDL is an XML based vocabulary for defining a Web Service**
 - interfaces
 - operation types (i.e. one-way, request-response, notification)
 - messages defining a Web Service interface
 - definition of data types (XML Schema)
 - access protocol (i.e. SOAP over HTTP)
 - contact endpoints (i.e. Web Service URL and URNs¹)



(1 URNs are location independent pointers to a file, or to different representations of the same content. In most ways they can be used like URLs)

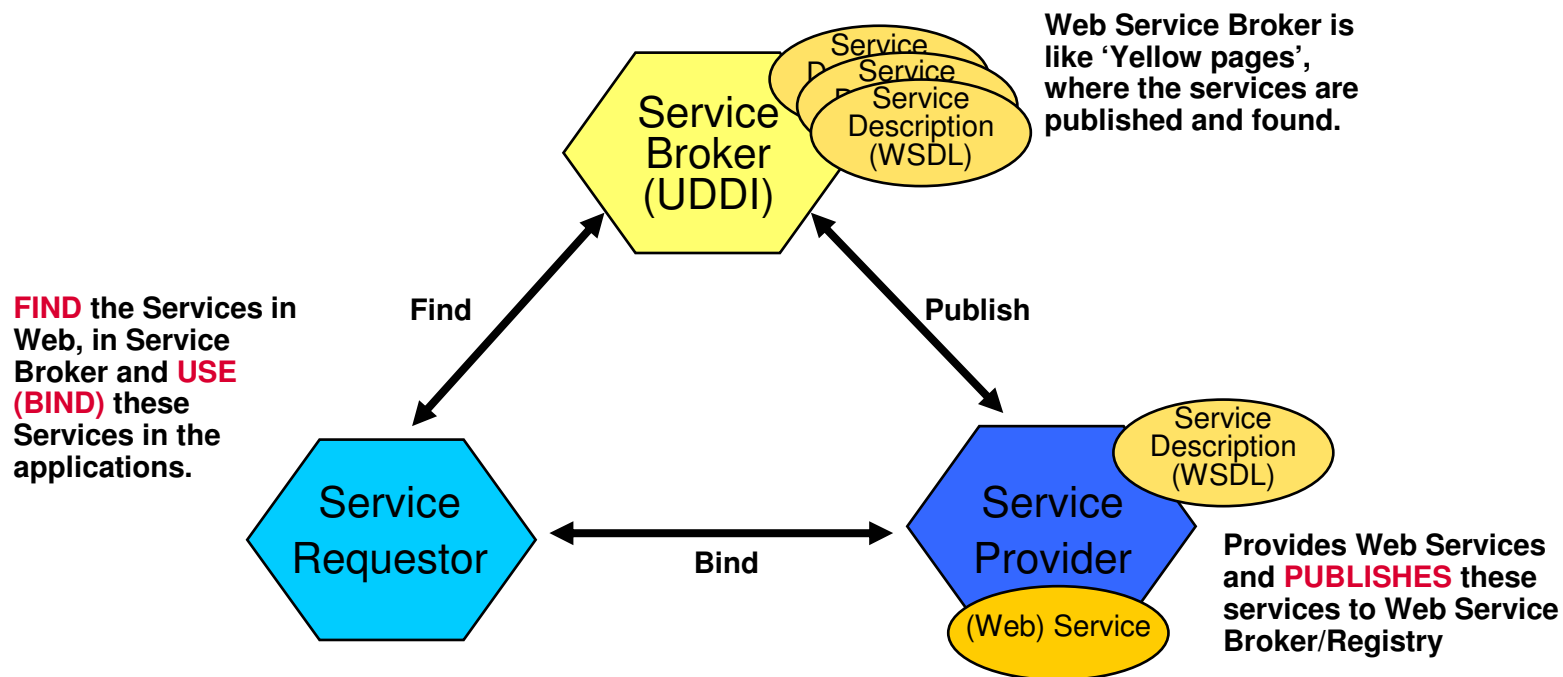
- **A Web Service URL returning WSDL makes Web Services self-describing**
- **Similar in purpose to IDL (Interface Definition Language)**
 - From a WSDL file, wizards can generate:
 - proxy classes for calling Web Service
 - skeleton classes to implement a Web Service



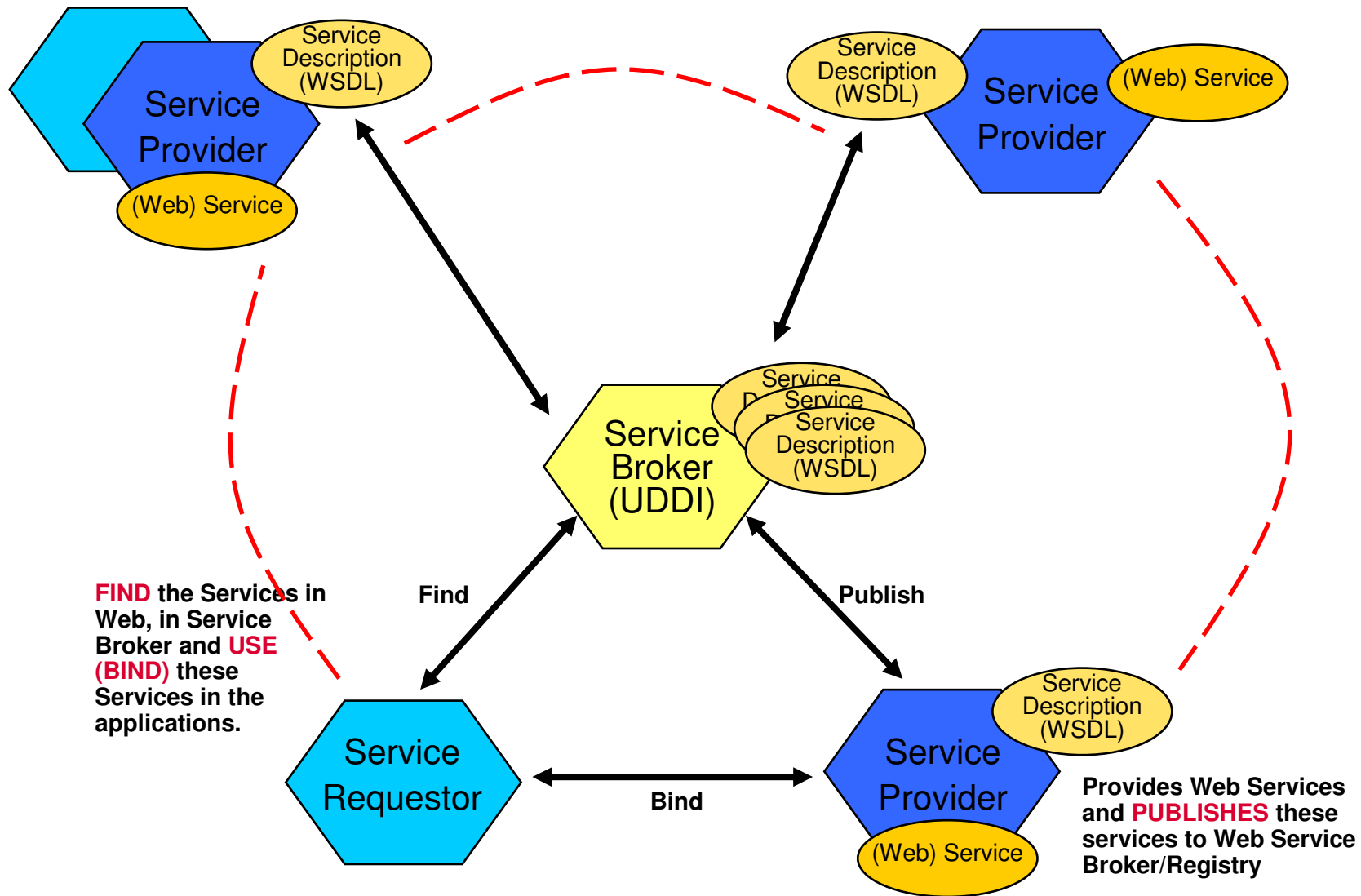
How Web Services work

„WebServices are self-contained, modular applications that can be described, published, located, and invoked over a network, generally, the World Wide Web.“ IBM

“A WebService is programmable application logic, accessible using standard Internet protocols“ Microsoft



How Web Services work



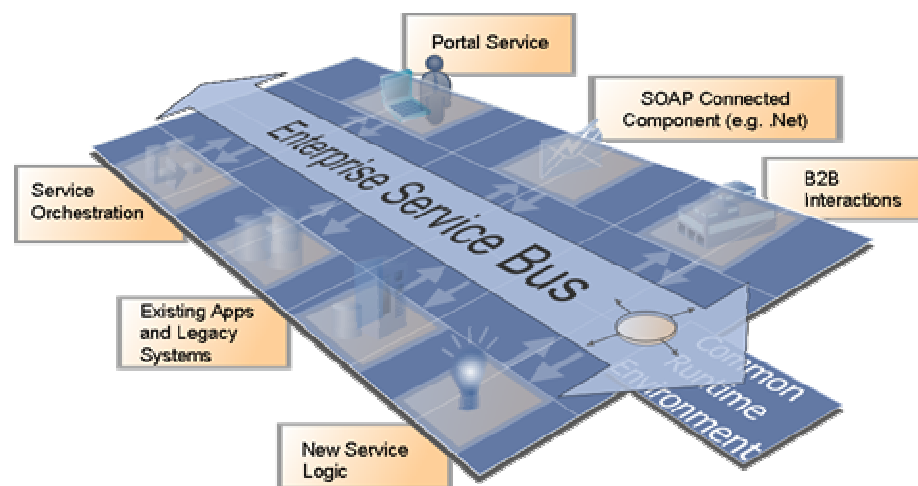
What is an Enterprise Service Bus?

An Enterprise Service Bus (ESB) is a flexible Infrastructure for services and application integration

An ESB reduces the number, size and complexity of your interfaces in a SOA solution.

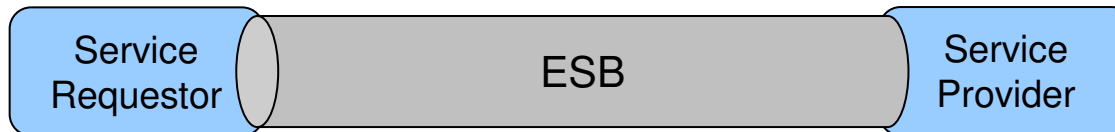
An ESB realizes following tasks between requestor and service

- **ROUTING** of messages between Services
- **CONVERTING** the transport protocol between requestor and service
- **TRANSFORMING** message formats between requestor and service
- **HANDLING** of business events between different types of services





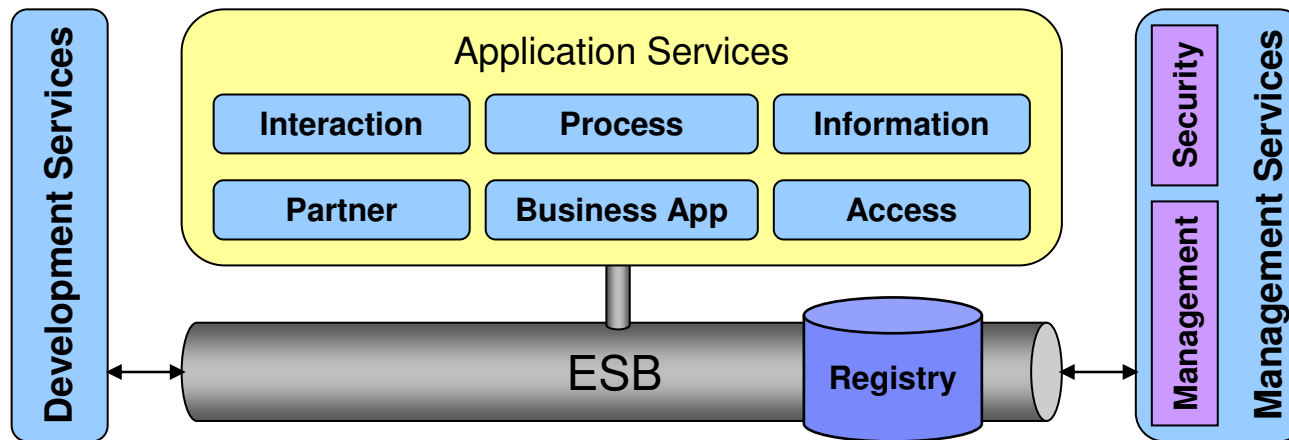
Core Principles of the ESB Architectural Pattern



- ESB inter-connects requestor and provider
 - Interactions are *decoupled*
 - Supports key SOA principle – *separation of concerns*
- ESB provides **Service Virtualization** of
 - *Identity* via routing
 - *Protocol* via conversion
 - *Interface* via transformation
- ESB also enables **Aspect Oriented Connectivity**
 - Security
 - Management
 - Logging
 - Auditing



An ESB-centric view of the Logical Model



- Outside ESB
 - Business Logic (Application Services)
 - ESB **does** contain integration logic or connectivity logic
 - Criteria: semantics versus syntax; aspects
- Loosely coupled to ESB
 - Security and Management
 - Policy Decision Point outside the ESB
 - ESB can be Policy Enforcement Point
- Tightly coupled to ESB
 - Service Registry
 - Registry a Policy Decision Point for ESB
 - ESB a Policy Enforcement Point for Registry
 - But, Registry has a broader scope in SOA
- Tooling required for ESB
 - Development
 - Administration
 - Configures ESB via Service Registry

More details at: <http://www.ibm.com/developerworks/library/ar-esbpat1/>

Why should VSE customers consider SOA?

- **SOA is modern (hype) and strategic**
 - It is mentioned in all IT journals and newspapers

- **Easy integration of existing VSE programs and processes**
 - Reducing the interface complexity
 - Reuse of existing application logic as services
 - Use of standard protocols (XML, SOAP, HTTP)

- **integration is platform independent**
 - independent of application programming language
 - independent of the data involved

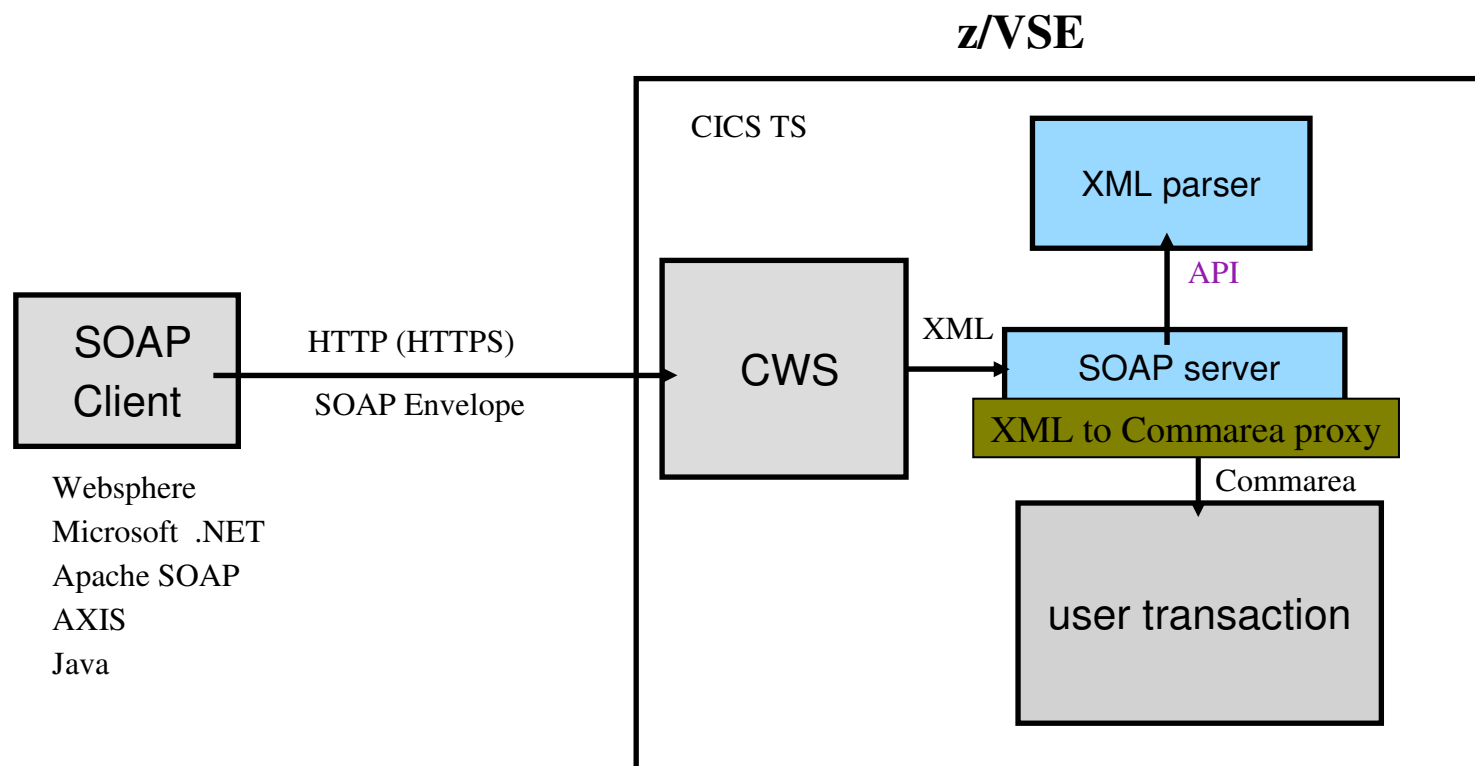
- **Integration of VSE into a Microsoft .Net environment**
 - without the use of Java
 - the most incompatible environments can be integrated

- **SOA enables the extension of VSE applications**
 - to other platforms and architectures
 - to partners and open world

VSE as SOAP server

Web Services (SOAP)

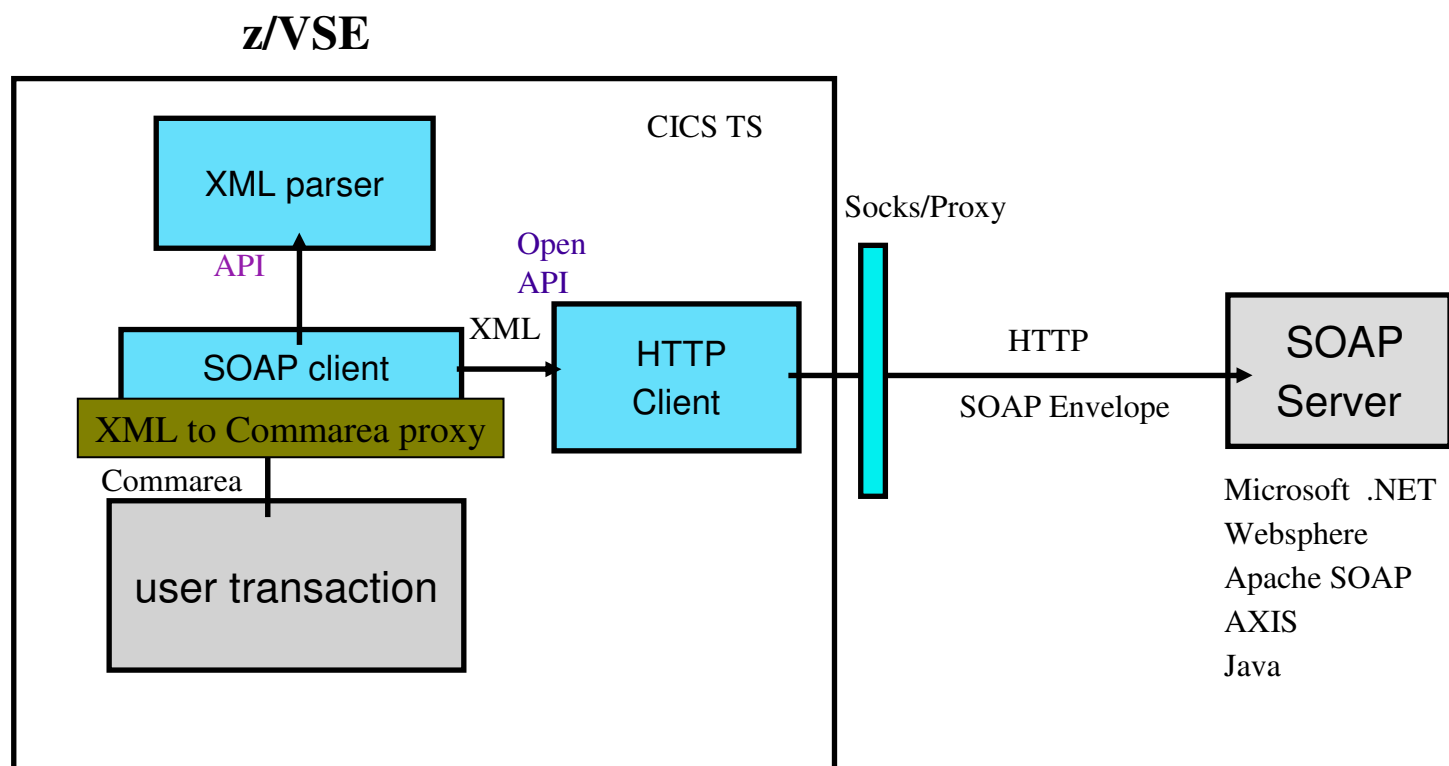
SOAP - Simple Object Access Protocol
(platform independent remote procedure call)



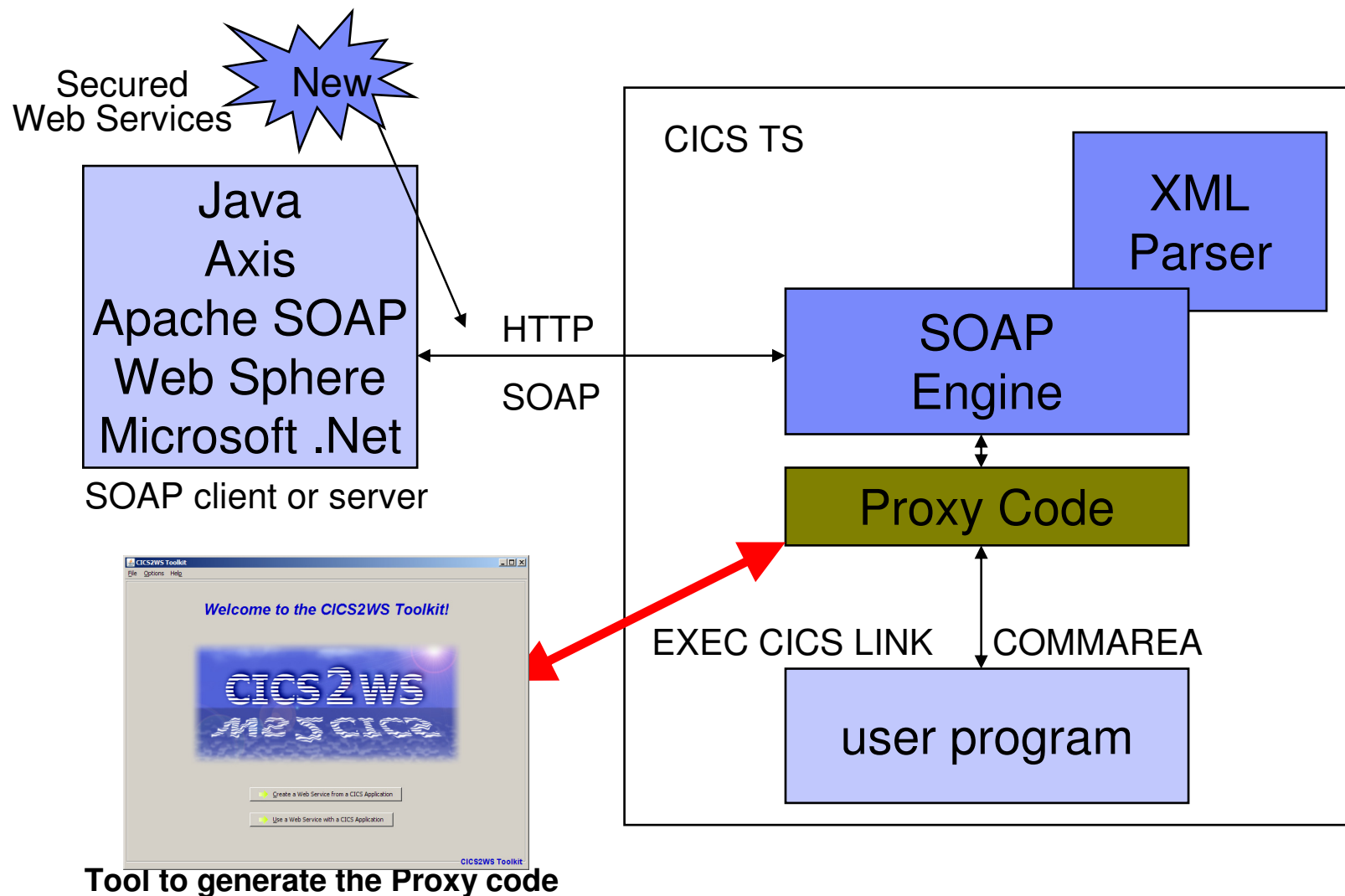
VSE as SOAP client

Web Services (SOAP)

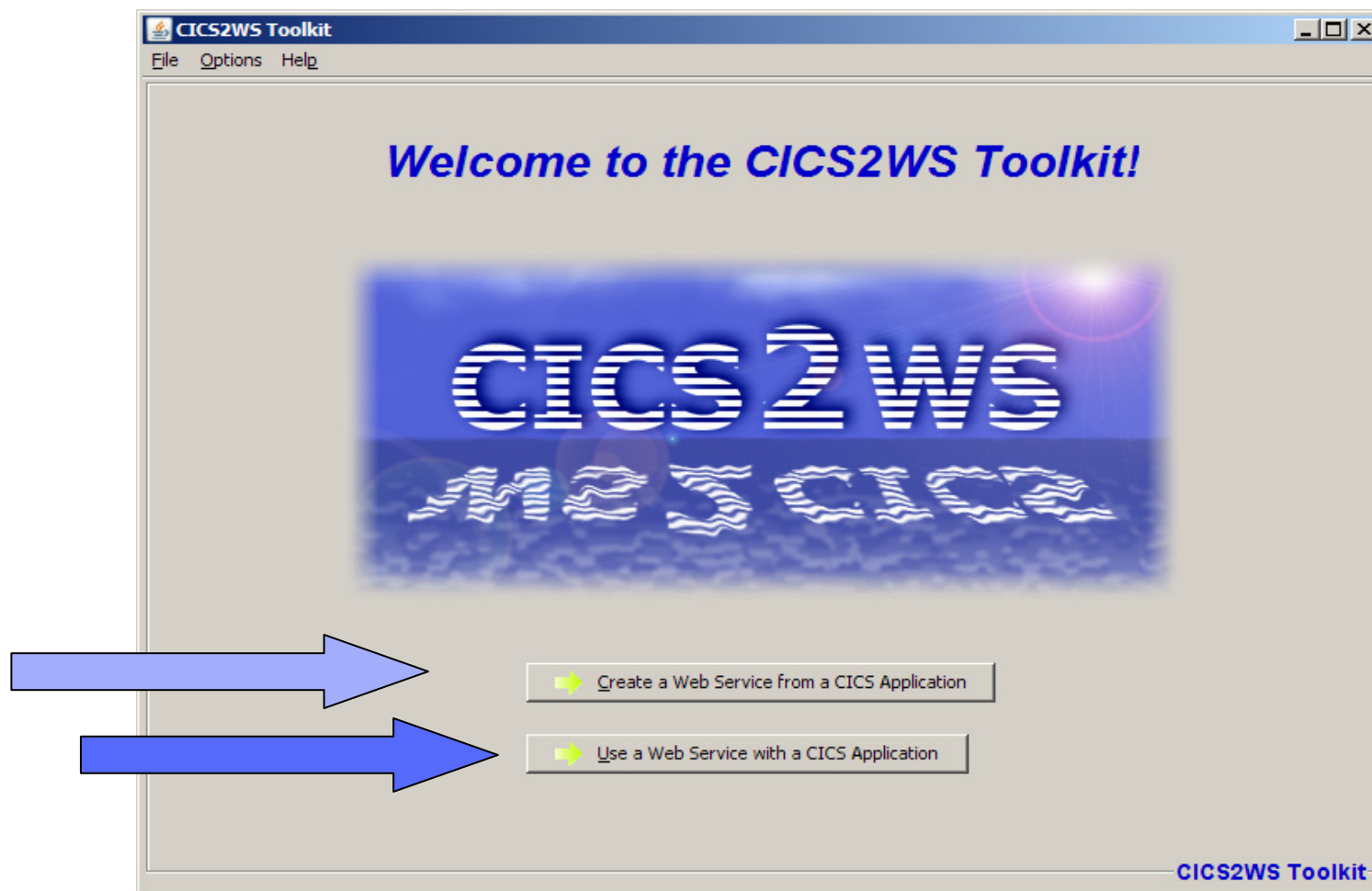
SOAP - Simple Object Access Protocol
(platform independent remote procedure call)



Web Services in and with VSE

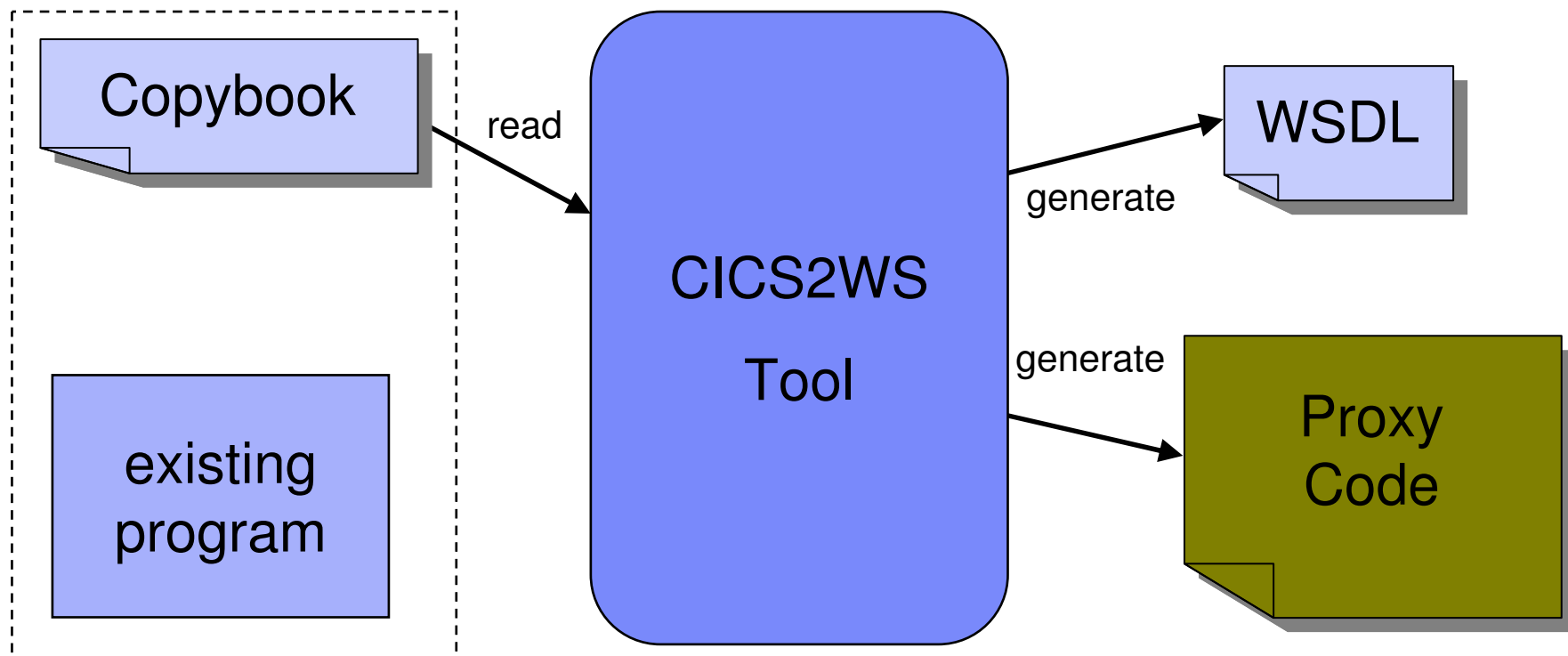


CICS to Web Services Tool

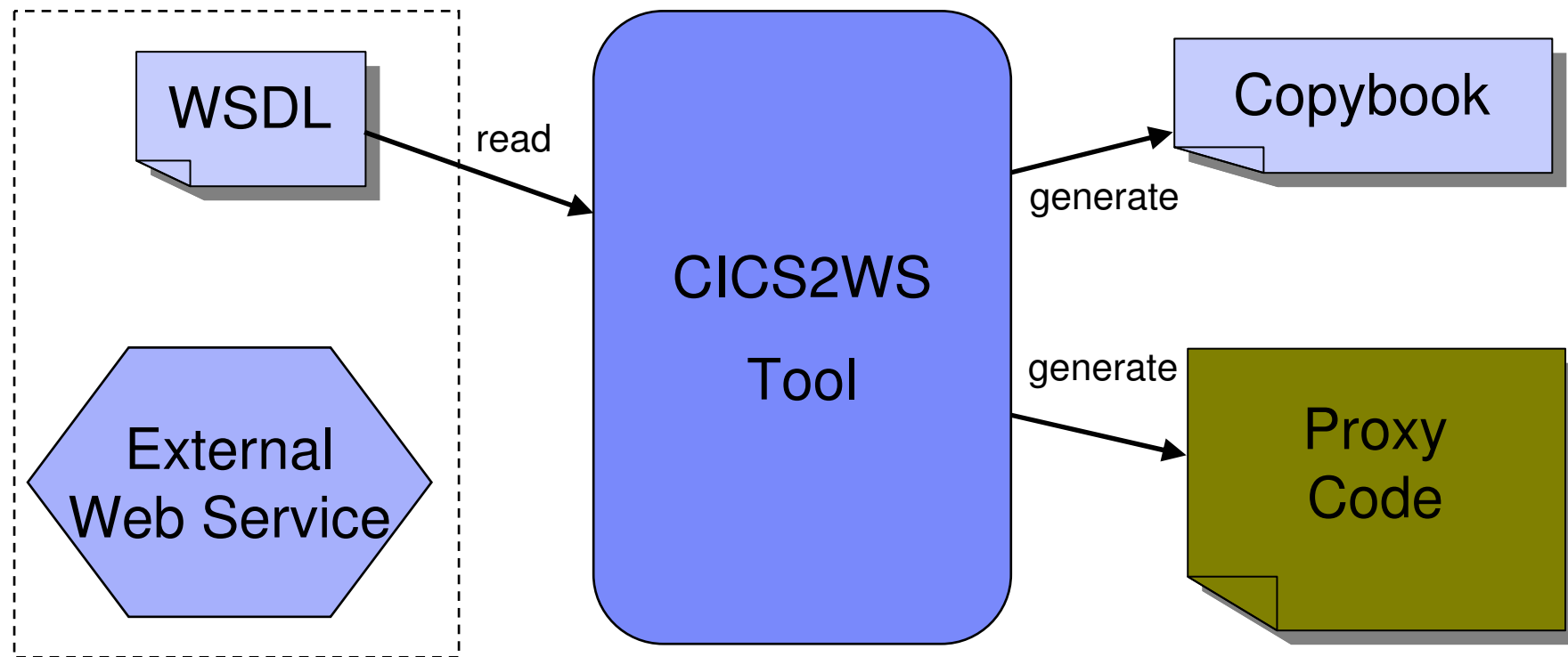


<http://www-03.ibm.com/systems/z/os/zvse/downloads/>

VSE as a SOAP server (service provider)

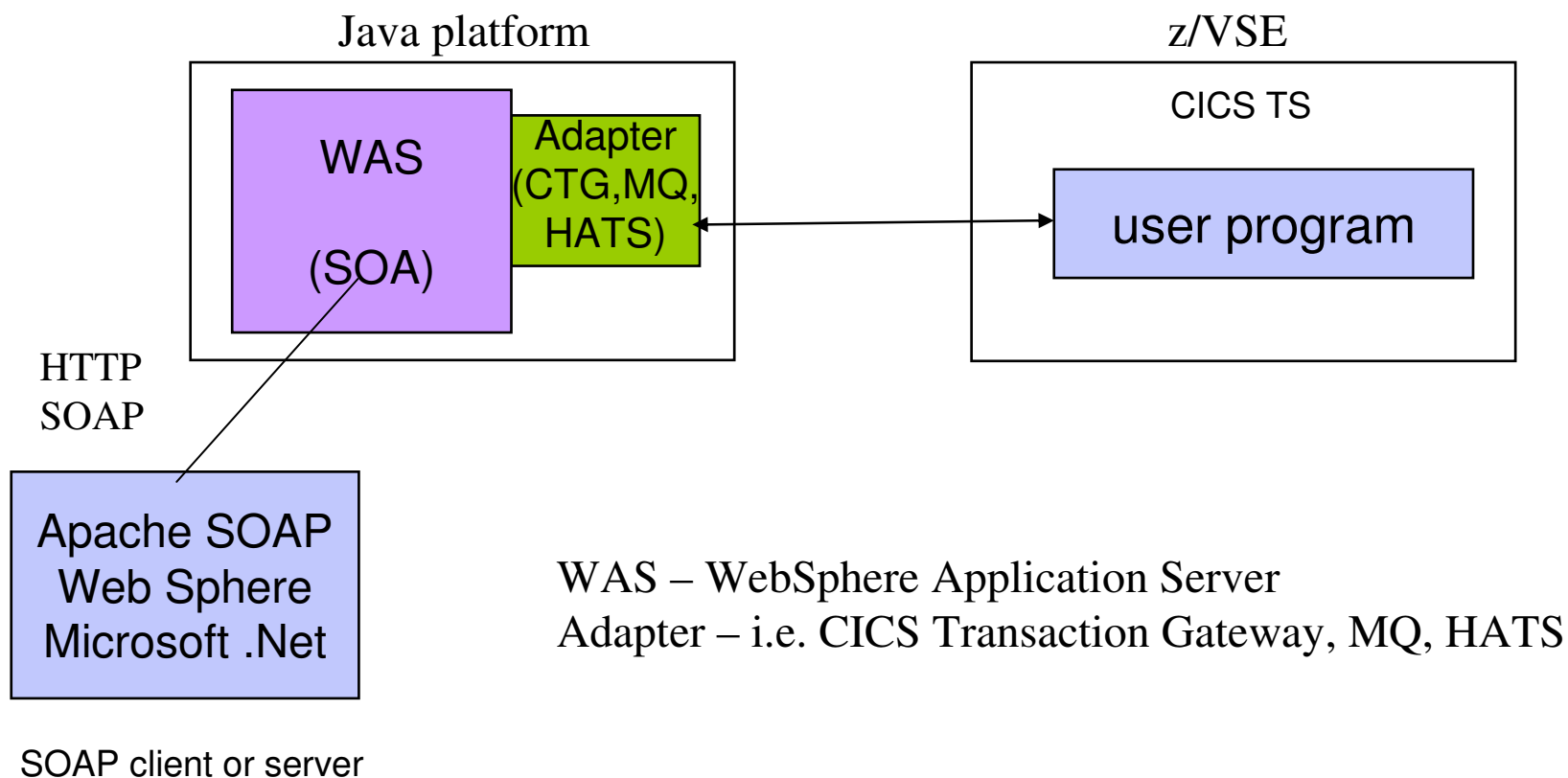


VSE as a SOAP client (service requestor)

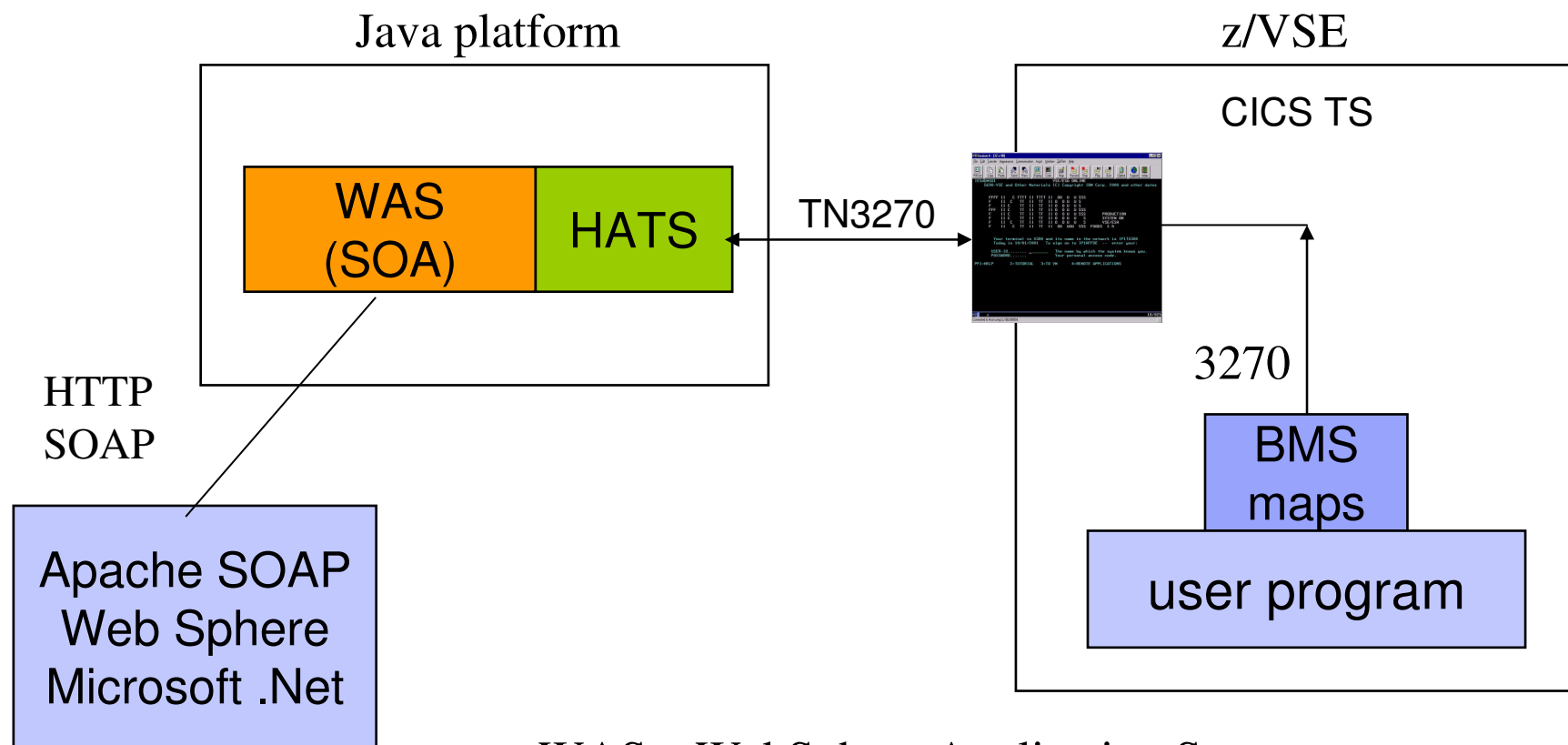


Web Services with **Middle tier** and z/VSE

The Web Service is implemented on the middle tier – WebSphere Connector technology is used to access VSE logic



Web Services with 3270 applications



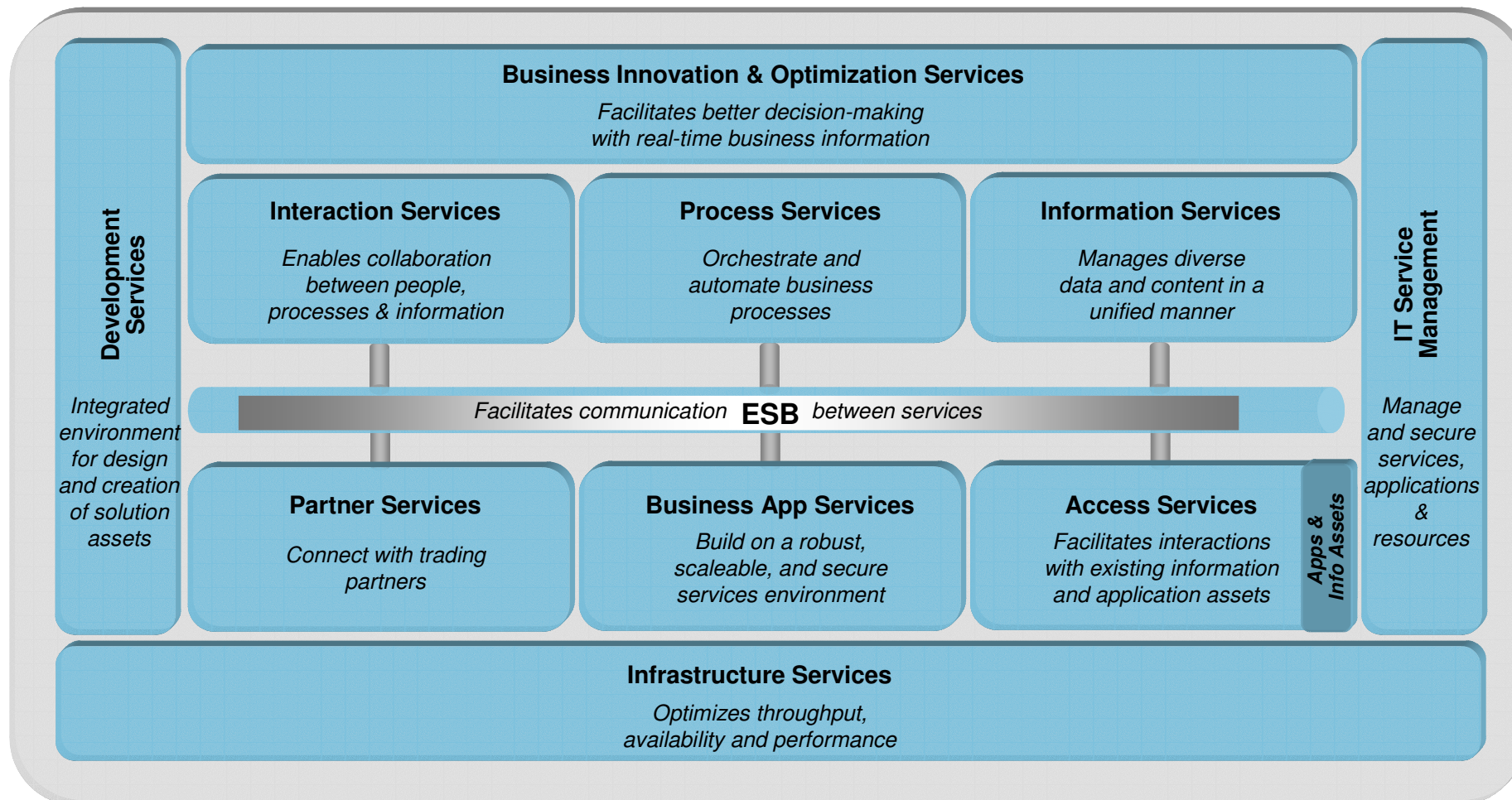
SOAP client or server

WAS – WebSphere Application Server

HATS – Host Access Transformation Server

The SOA Reference Architecture

Defining an Infrastructure for SOA



- Note: This is IBM's one-and-only SOA Reference Architecture. There is no "special" SOA architecture for System z. This is it – for all platforms.

The Enterprise Service Bus

WebSphere Enterprise Service Bus (WESB)

- **Native WebSphere application that provides ESB functionality**
- **Supports SCA and other J2EE standards**
- **Required to support WebSphere Process Server**

WebSphere Message Broker for Linux on System z

- **Native Linux-on-z run-time**
- **High-performance (written in c/c++) implementation**
- **Eclipse tooling for building message flows**
- **Mature product with MQ backbone**
- **Special “nodes” to directly integrate with CICS**

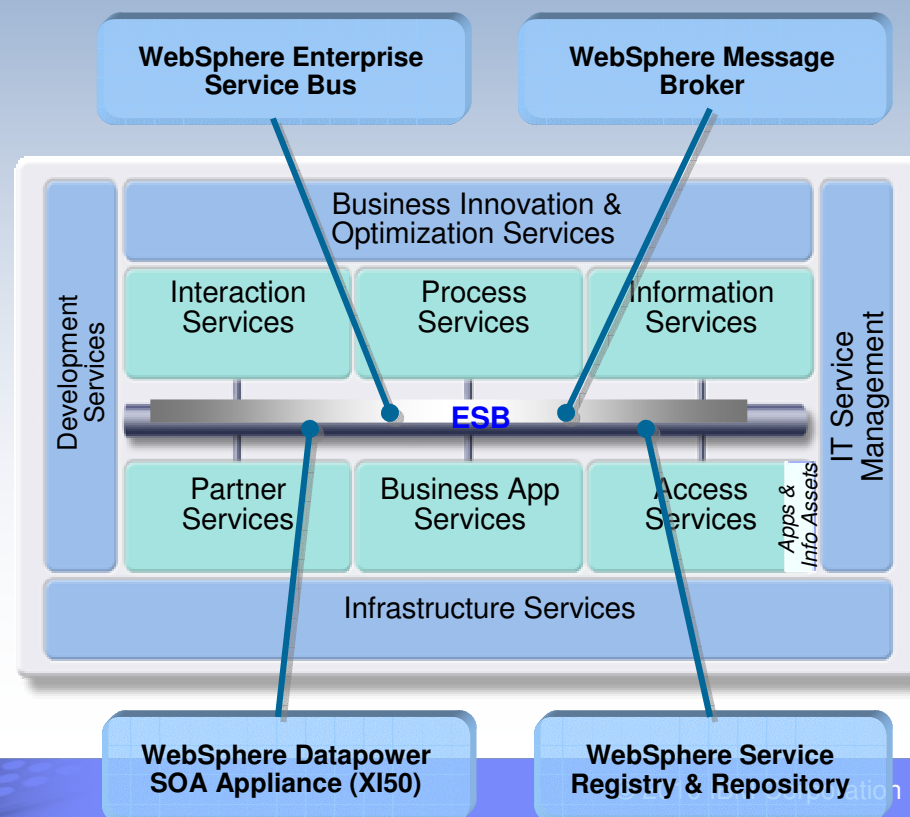
WebSphere DataPower SOA Appliance (XI50)

- **External device that parses SOAP/XML, security headers**
- **Unique routing/transformation engine**
- **Delivers wirespeed performance, high reliability, improved security, and easy-to-use administrative interfaces**
- **Includes the functions of WebSphere DataPower XML Security Gateway (XS40) and WebSphere Datapower XML Accelerator (XA35)**

See Stuart Jones' [“ESB Product Positioning Guide”](#) for guidance on which product works best and where

WebSphere Service Registry & Repository

- **Discover and reuse services that could serve as building blocks for new composite applications.**
- **Publish newly developed services.**
- **Govern deployed services to ensure changes are authorized and service integrity is maintained.**
- **Manage efficiency by providing detailed information about service interaction endpoints being monitored.**



Interaction Services

WebSphere Portal Server

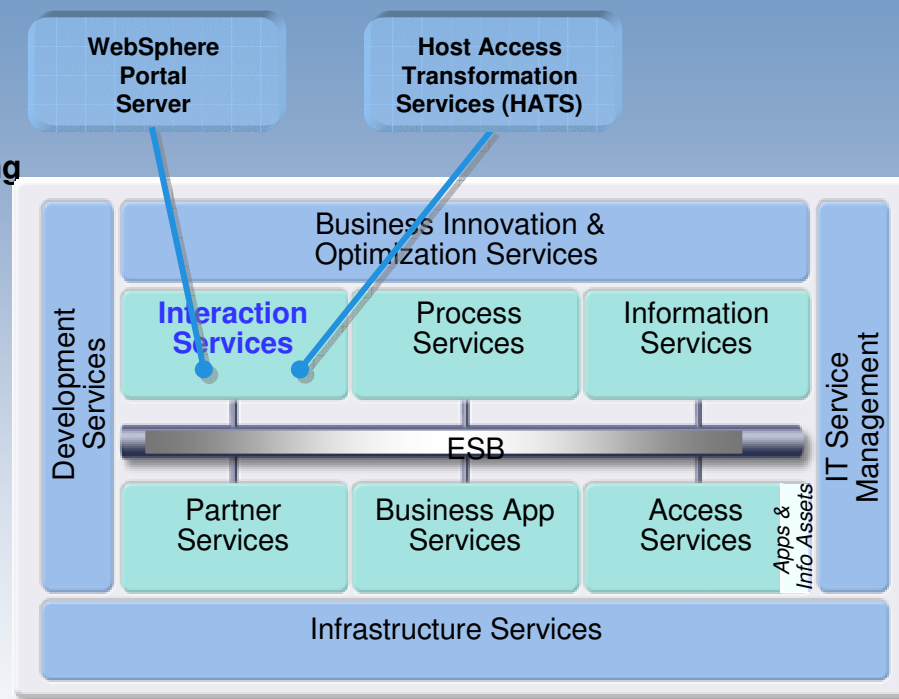
- Scalable secure portal infrastructure for B2E, B2B, and B2C portal solutions.
- Provides a single, unified point of access and interaction with people, information, applications and business processes.
- Includes the most complete set of portal services including personalization, collaboration, web content management, search, workflow, web services and many more

WebSphere Portlet Factory

- Toolbox for building composite portal applications
- Includes wizard-driven adapters for many back-end data sources, including SAP, SAP Business Warehouse, PeopleSoft, Siebel, or any JDBC-compliant database
- Architected for reuse and flexibility

Host Access Transformation Services (HATS)

- Rules based Web-to-host transformation engine
- Provides customized access to multiple host applications
- Dynamically generates Web HTML user interfaces
- **Dynamically creates a reusable web service**
- Significantly improves usability



Information Services

DB2 V9

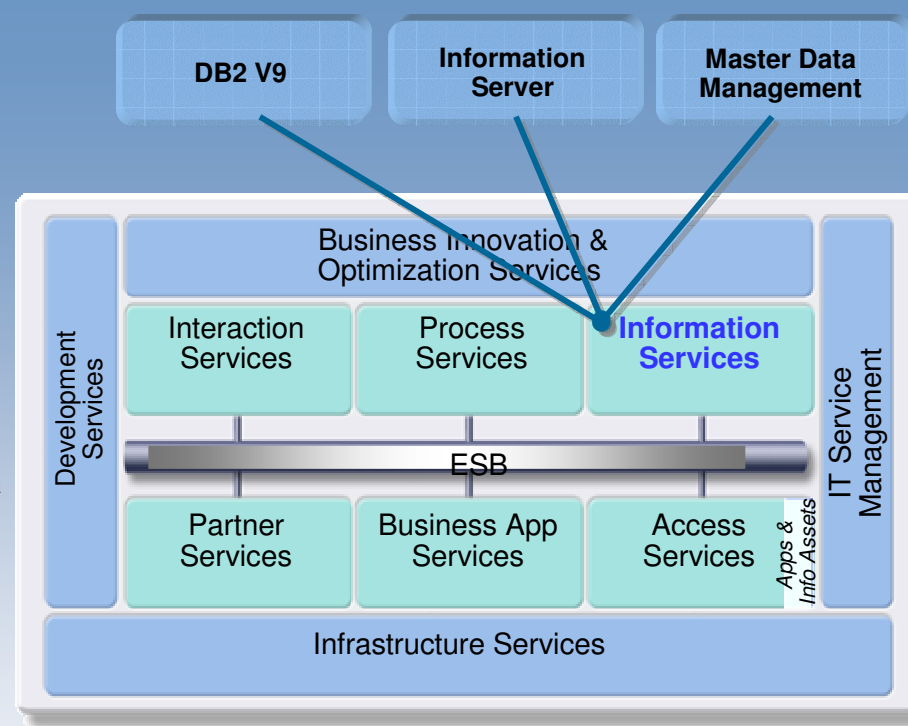
- First hybrid data server includes native XML and relational data services for improved data governance
- Fast development of high performance SOA solutions on Linux, Unix, Windows & z/OS

WebSphere Information Server

- Deploy information services enabling use of data in line and in context within business processes
- Trusted, consistent view of information spanning heterogeneous sources
- WebSphere Metadata Server links business & technical metadata to service metadata

Enhanced Master Data Management Offerings

- Integration of WebSphere Customer Center with the IBM Industry Models for Banking providing a business-oriented “SOA blueprint” for customer data
- Updated IBM WebSphere Product Center to easily collect, author, rationalize and publish product information in a SOA



Access Services

InfoSphere Classic Federation Server

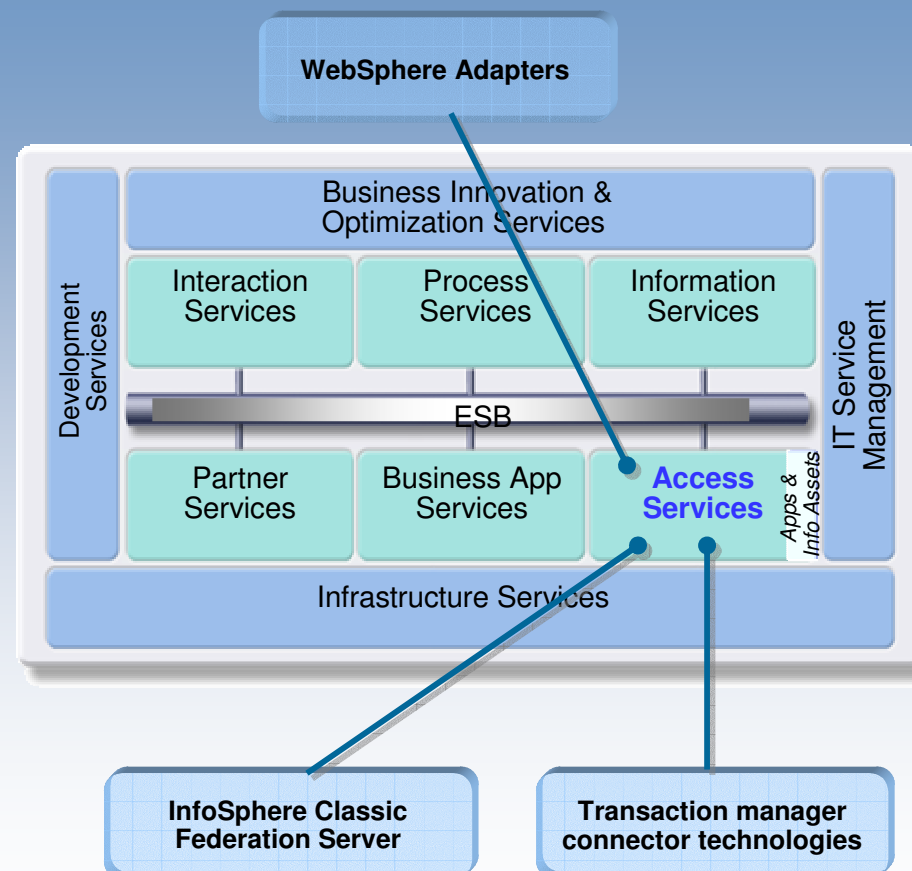
- **Read from and write to federated data sources using SQL**
- **Standards-based access via JDBC, ODBC, or Call Level Interface**
 - Including VSAM & other non-relational
- **Multithreaded with native drivers for scalable performance**

Transaction manager connector technologies

- **Provides connectivity to CICS via J2EE Connector Architecture (J2C), WebSphere MQ Bridges, or native Web Services**
- **Allows reuse of existing transactions without rewrite or modification**
- **“Plugs in” to commonly-used APIs and existing programming languages**

WebSphere Adapters

- **J2C-based adapters to accelerate connectivity to back-end systems and data sources including:**
 - JD Edwards, PeopleSoft, Siebel, SAP
 - Flat files, e-mail, JDBC, FTP ...



Development Services

WebSphere Integration Developer

- Import and work with business process models directly from the business analyst
- Delivered graphical user interface where everything can be done through the GUI
- Provided a single way to define all types of processes for business rules
- Enabled ability to leverage existing services and develop for future reuse

Rational Software Architect

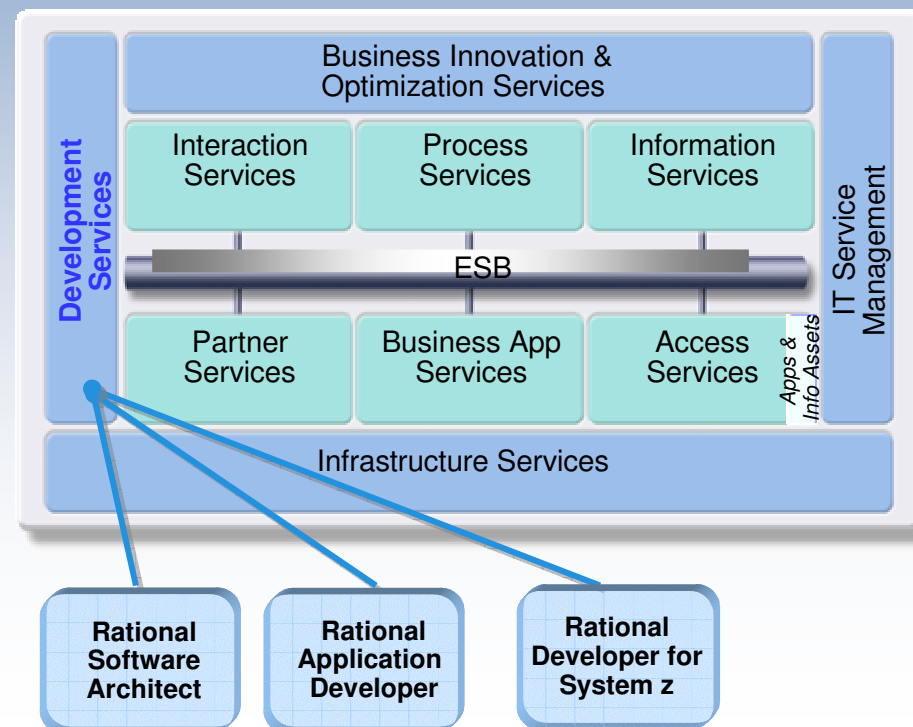
- Model in UML and transform to Web service
- Delivered patterns to help automate development of applications and promote reuse
- Enable reuse through best practices

Rational Application Developer

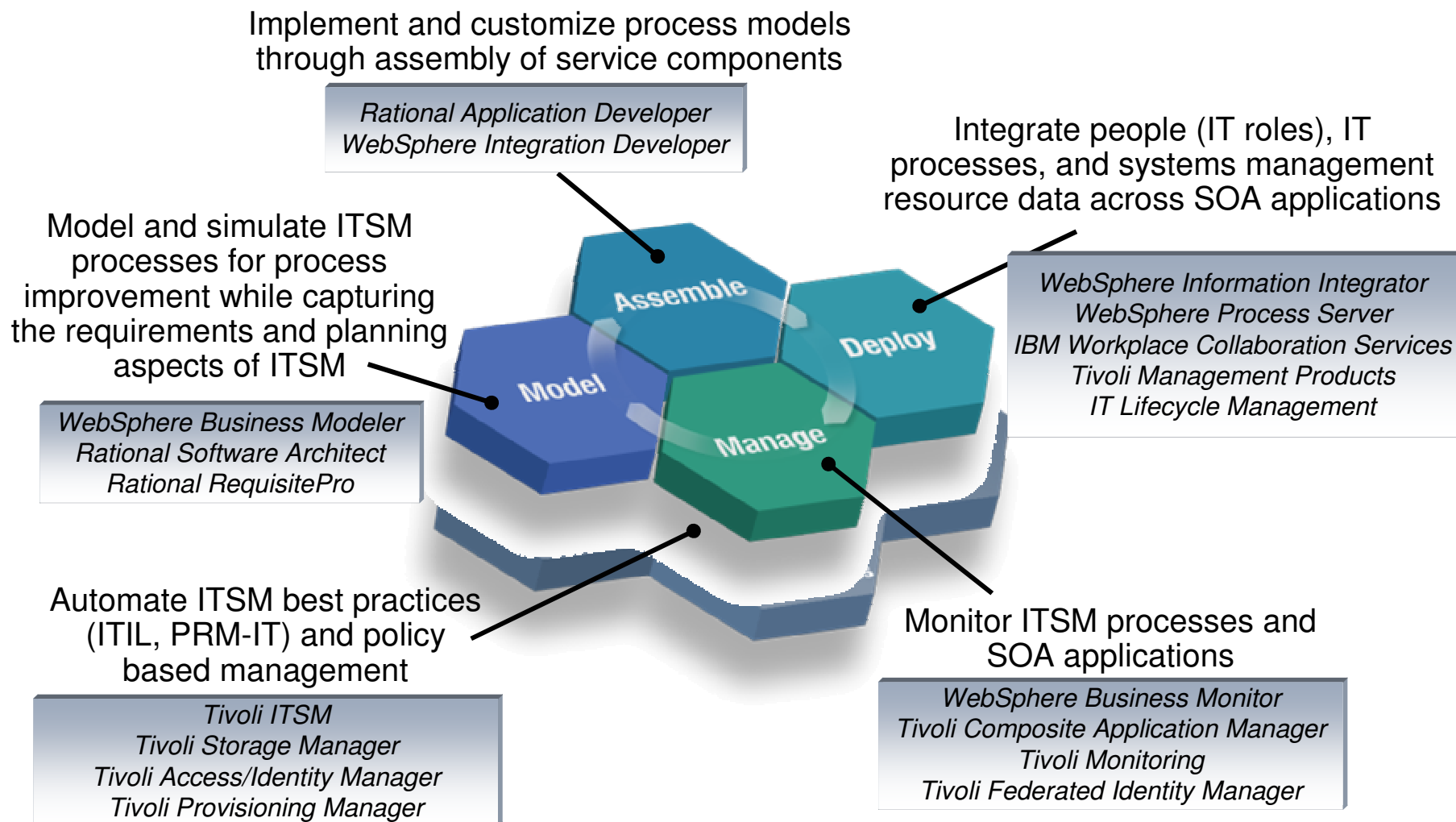
- Automated Services testing
- Built new Web services from scratch
- Discover and consume existing Web services
- Allowed to generate and use services with no Java required using EGL
- Extended the power of Eclipse

WebSphere Developer for System z

- Develop applications in traditional languages – COBOL, PL/I, Assembler, in addition to Java
- Wizards to integrate existing host applications or expose as services
- Tools to support CICS Service Flow Feature, BMS or MFS application integration
- Integrated host/workstation debugging



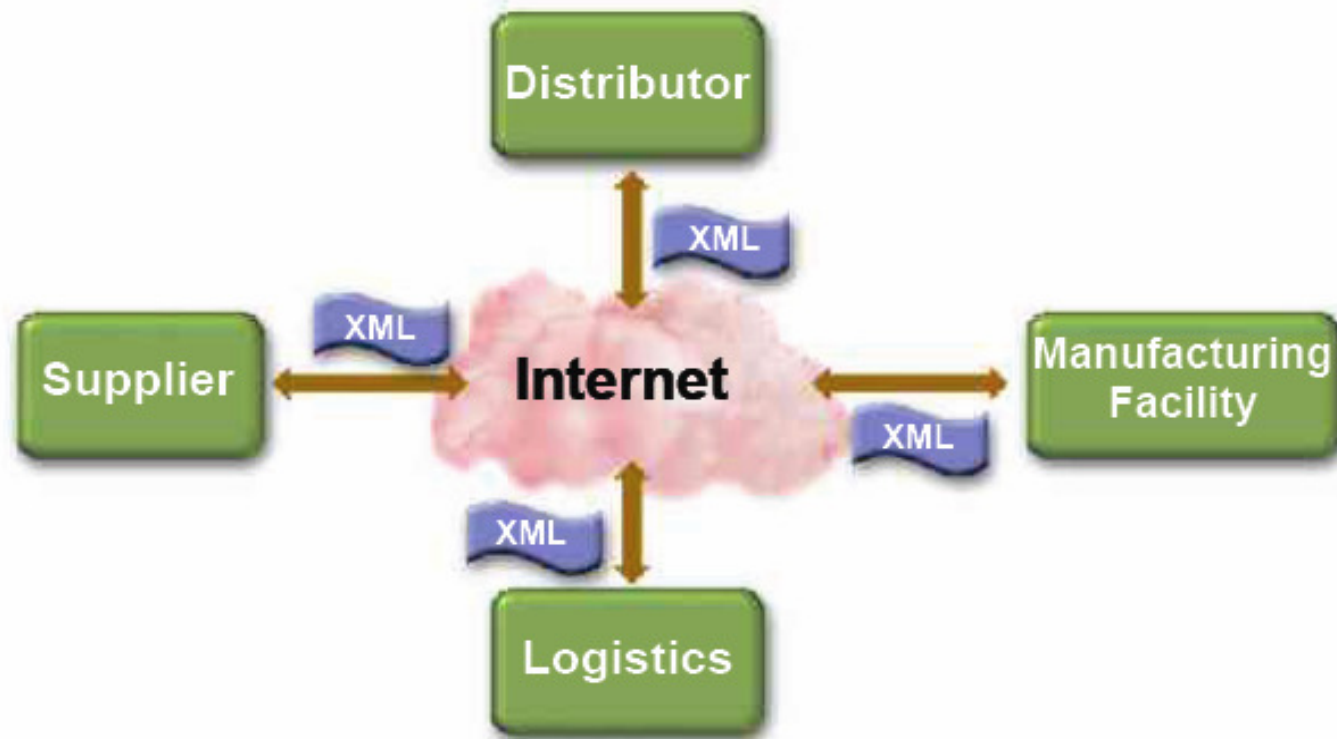
IT Service Management: Built on the SOA Foundation



Why would a VSE customer do SOA ?

- SOA is modern (hype) and strategic
 - The management says: We also have to do SOA
- Easy integration of existing VSE programs with other platforms
 - Reducing the interface complexity
 - Reuse of existing applications as services
 - Use of standard protocols (XML, SOAP, HTTP)
- Encapsulation of VSE programs
 - Disconnecting business and display logic
- Integration of VSE into a Microsoft .Net environment
 - You do not want to use Java
 - You already have a Microsoft environment

Web Services Usage



Growing need for a standard infrastructure for application communication independent of platform or programming language.

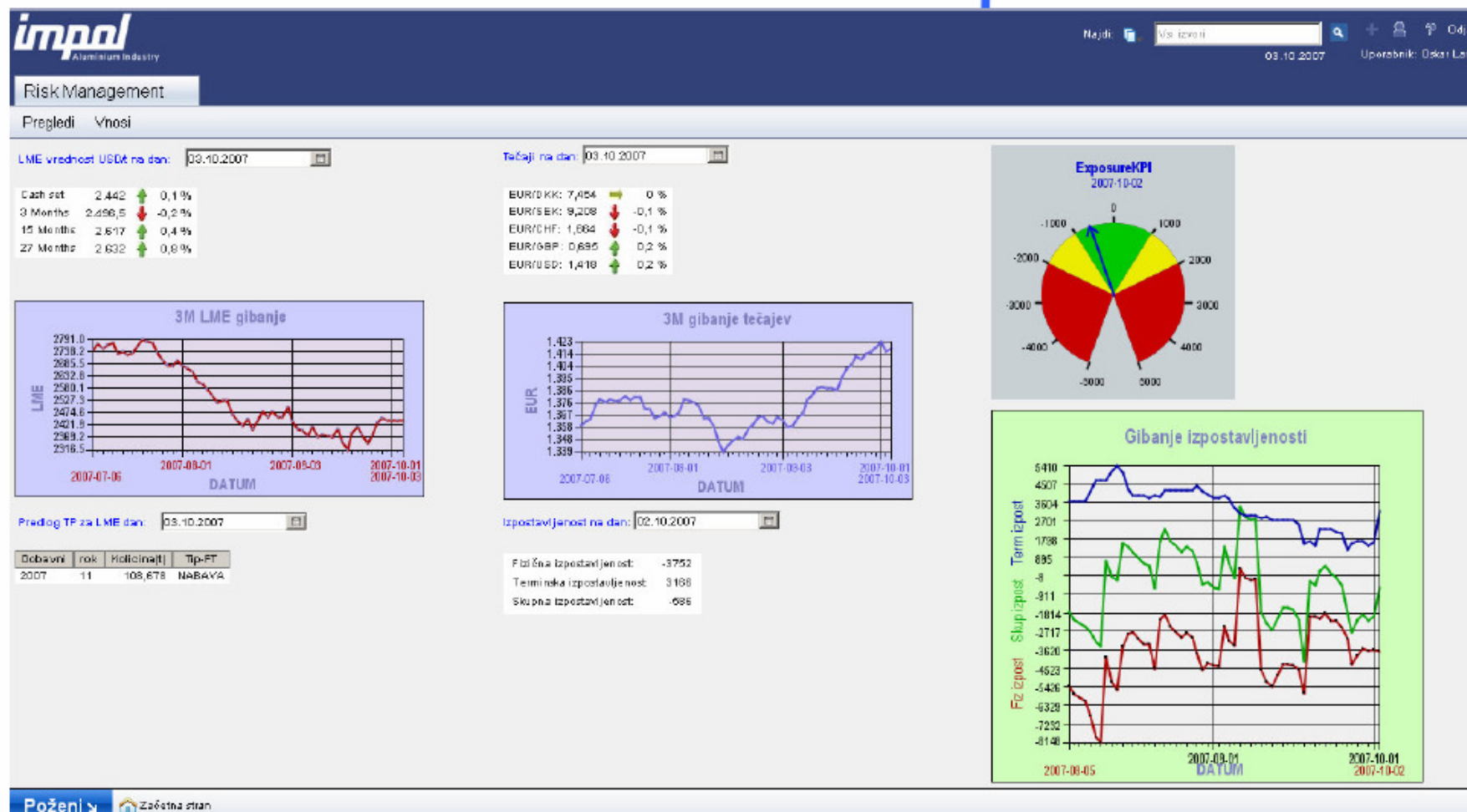
SOA Sample: Impol produces and sells Aluminum in Slovenia

Project “Protection of Sales Margin”

- Business Need: Buy and sale based on Stck Rates
 - Business success is depending upon favorable LME price movements and we have to create a business environment which can be controlled and commanded by us. The main risk is the constantly increasing Aluminum price volatility on the London Metal Exchange.

- SOA Solution:
 - Each deal is calculated based on actual LME Value – realized with a SOA
 - Alcad and IBM described the process with **WebSphere Business Modeler**. In the course of optimizing and simulating this process we saw the high flexibility of IBM WebSphere and we knew why IBM is market leader.
 - A key tool was also the **WebSphere Integration developer** which enables us to transform many of the existing IT assets into highly efficient, flexible and reuseable services.
 - Having also **WebSphere Process Server** in place, it is very easy to quickly deploy those services enterprise wide!
 - And for the first time we now are able to monitor the process flows in real-time with IBM **WebSphere Business Monitor**, immediately react on process gaps and use this information as input for better strategic decisions and planning.

Portal RM Impol



SOA with z/VSE, z/VM and Linux
on System z

SOA Sample 3: Ball State University Indiana, USA „Student Address-System Integration“

- **Business Need: Reduce Administrative effort**
 - “Coordinate 40 name and address systems to streamline administrative processes and ensure information integrity for users”

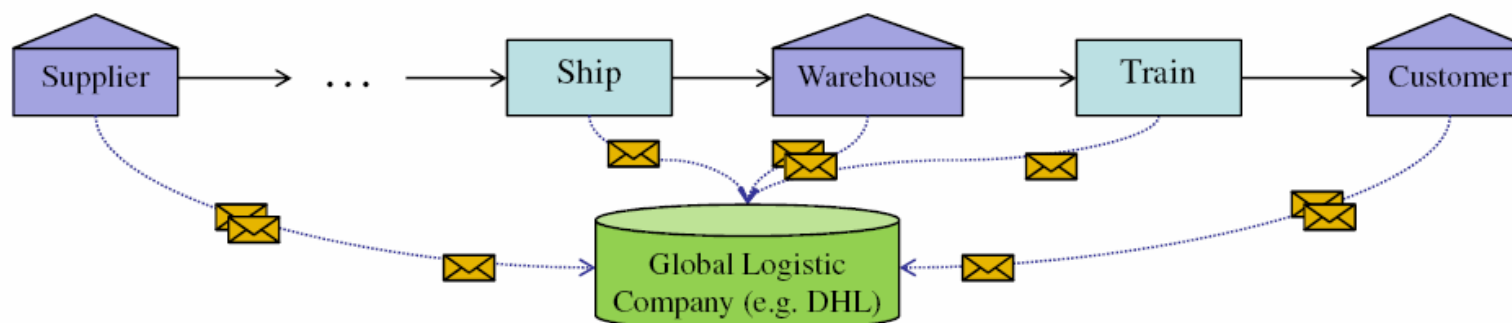
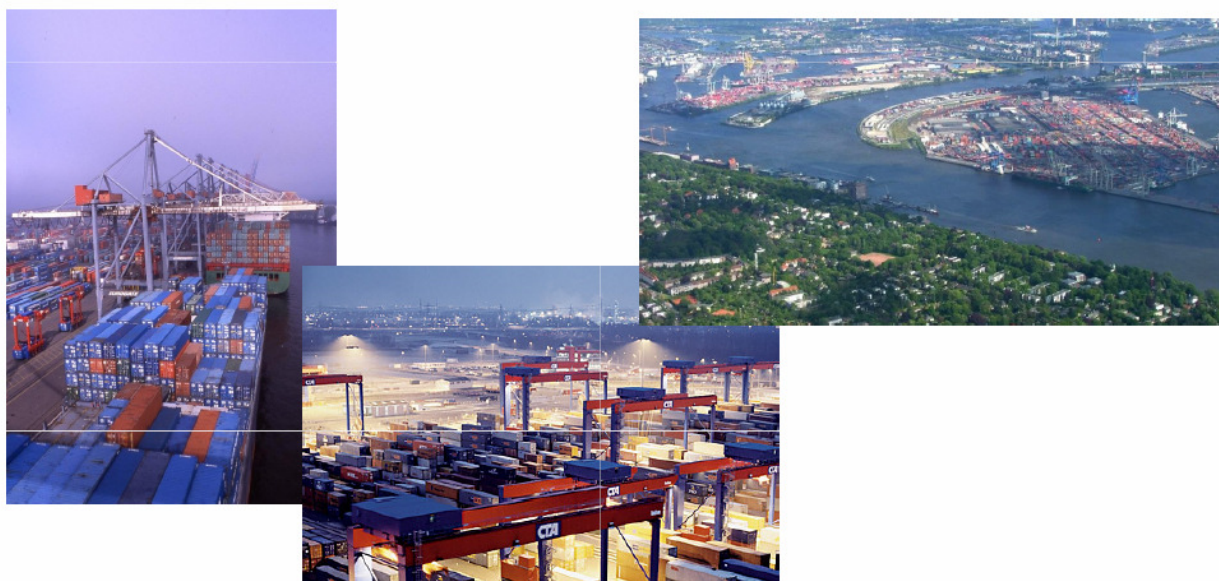
- **SOA Solution: Building reusable components with SOA**
 - The IBM SOA solution runs in the mainframe environment utilizing **IBM CICS Transaction Server**. This version strengthens the application development capabilities of CICS and extends CICS applications to an SOA. In a scenario in which information is provided as a service and applications are decoupled rather than hard-coded together, CICS makes it possible for transactions to readily connect with multiple interfaces and repositories to provide authoritative data.
 - IBM WebSphere Enterprise Service Bus**, works in conjunction with IBM WebSphere Application Server, to communicate between decoupled back and front ends, choosing the destination for a message and transforming it into the correct format.
 - IBM WebSphere Host Access Transformation Server (HATS)**, Version 7, generates Web Services Definition Language (WSDL) that is callable by BSU's .NET front end to publish information to the user interface.
 - IBM WebSphere Integration Developer** enables developers to assemble complex applications across the ESB by connecting reusable components.
 - IBM DB2**, is the database for the SOA.

So, the advent of SOA and Web services is not just an evolutionary step, to me it's a revolutionary step.”

–Fred Nay, IT Director, Ball State University

SOA Sample 4: Logistic at Hamburg Haven

- Daily:
 - >20 Container Ships
 - >25.000 Containers
 - Container stored:
 - >100.000
 - Stay time: 4 days



Increase your business – Get greater Value through SOA

Create Flexible, Service-based Business Applications

Value

- Flexibility and elimination of duplication for reduced cycle times
- Expanded access to core applications
- Consultant studies have found it 5X less expensive to re-use existing applications than to write new applications*

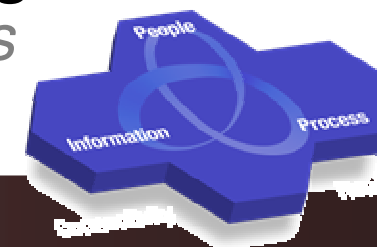
Start with

- What services are needed to run your business?
- Identify high-value existing IT assets and service-enable them for reuse
- Fill in gaps by creating new services for today's business needs and future reuse
- Registry/repository to facilitate centralized access and control of reusable services

“With reuse, solving the next business problem can be done more quickly and efficiently.”

- Amy Wohl

** Software Productivity Research (SPR)*



Documentation

- **What is SOA?**
 - <http://www-01.ibm.com/software/solutions/soa/>
 - <http://www.ibm.com/developerworks/webservices/newto/>

- **Web Services**
 - <http://www.ibm.com/servers/eserver/zseries/zvse/documentation/ebusiness.html#soap>

- ***z/VSE e-business Connectors, User's Guide (SC33-8231)***
 - <http://www-03.ibm.com/servers/eserver/zseries/zvse/>

- **Web Services in VSE (zJournal.com)**
 - <http://www.zjournal.com/index.cfm?section=article&aid=281>
 - <http://www.zjournal.com/index.cfm?section=article&aid=320>
 - Includes COBOL sample code