2009 System z Expo

October 5 – 9, 2009 – Orlando, FL

Session Title: SOA Roadmap and Web Services with z/VSE

Session ID: ZEO02

Speaker Name: Wilhelm Mild

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

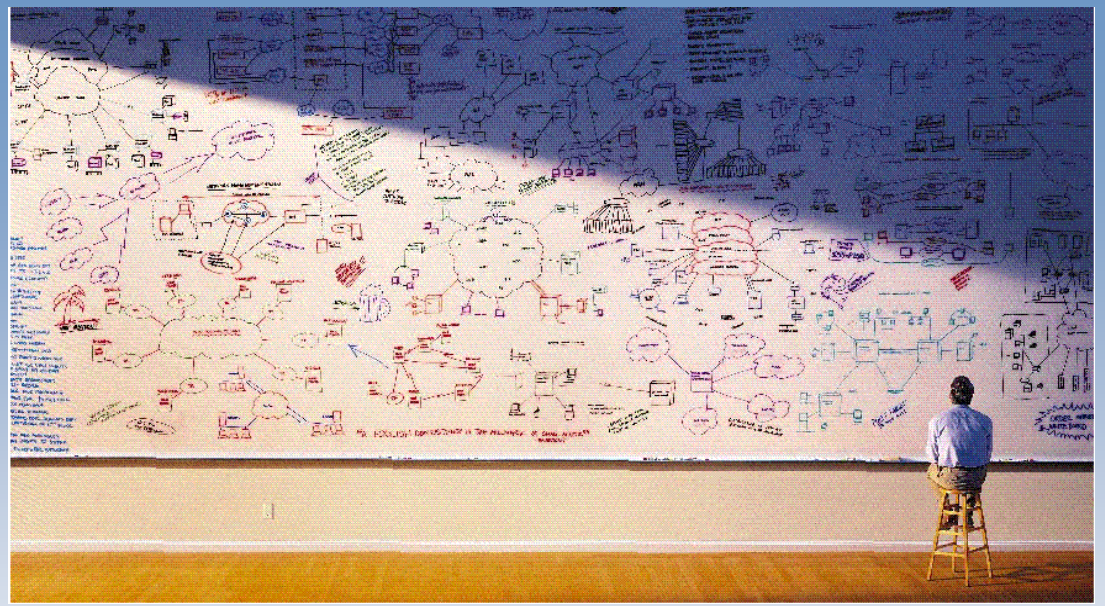
Top Innovation Priorities:

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

*87% of CEOs believe fundamental **change** is required in next two-years to drive innovation*

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 with SOA:

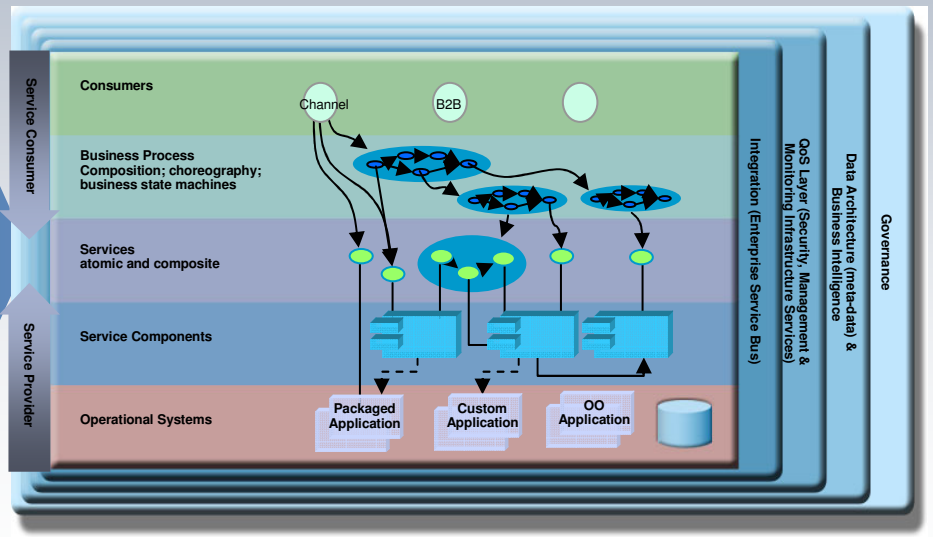
With SOA to a Business Process driven IT

Traditional Thinking

IT manages IT **assets** that support the business

Business Thinking

IT manages **Services** and **Components** which **reflect** the Business processes



Silos, static

Flexibility, dynamic, virtualised

Customers & Analysts Agree:

SOA Enables Rapid & Incremental Change Leading to Innovation

Innovation That Matters *

*"The IBM and GenXus SOA-based solution has made our product **more innovative, expanded our market** and made us **more competitive** It will let us grow our business significantly in the years to come."*

MADDENCO INC.

*"SOA is the **heart of the next wave of innovation**. The leaders that do this well are able to **rapidly change** …"*

AMR

*"SOA is critical for … executing the on-demand vision and in preparing … for the **incremental changes** … over time. Companies … make better decisions."*

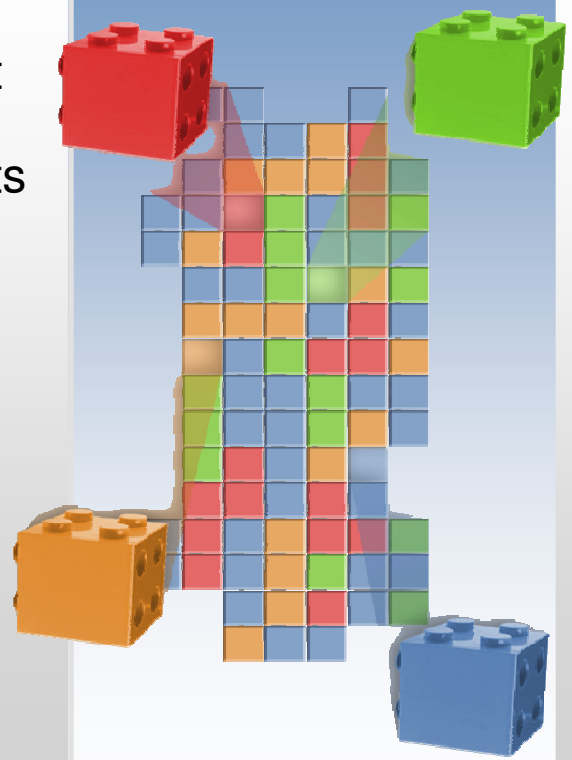
IDC

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 application functions 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.

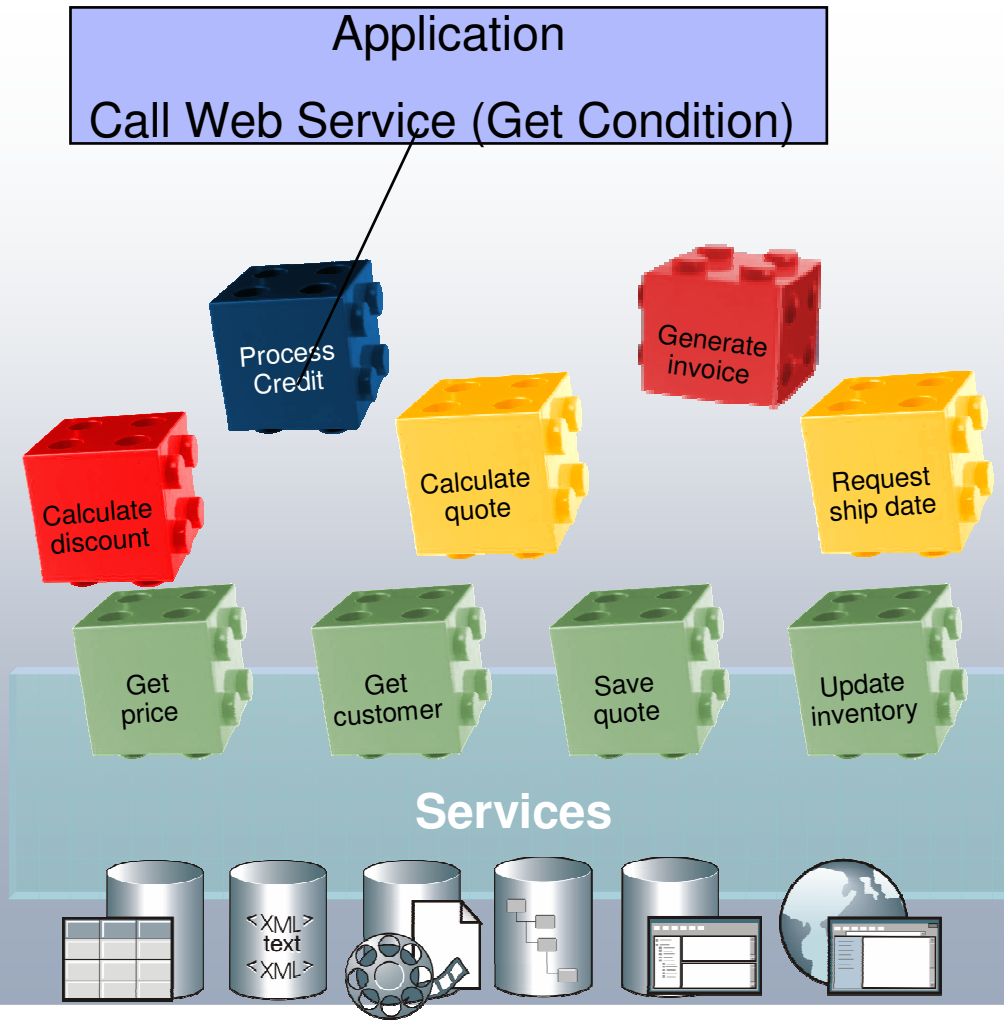
4-3-3

4-4-2

3-5-2

3-6-1

Integrating Logic in an SOA

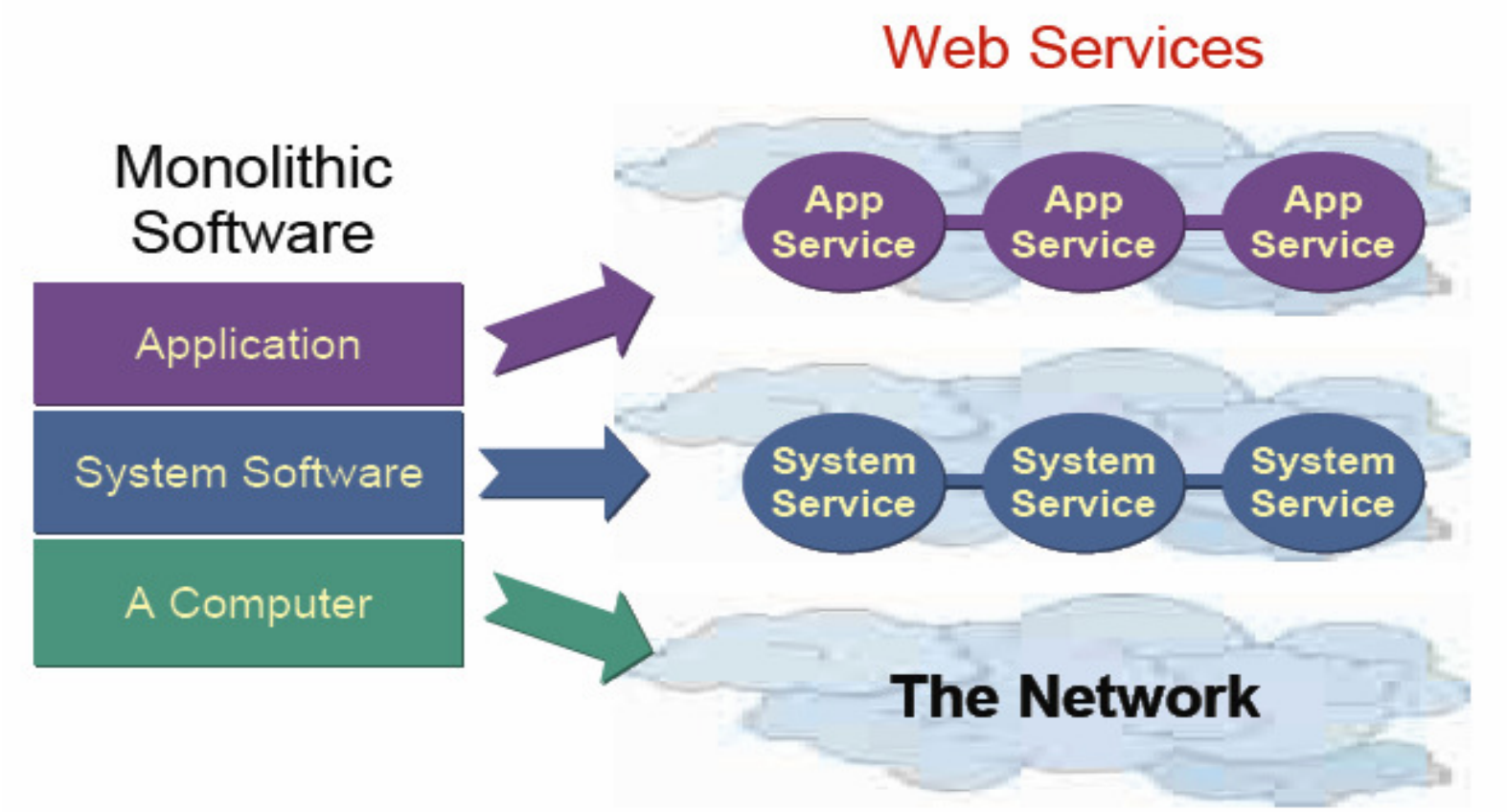


Information as a service makes information more accessible, consistent, and flexible

Publishing consistent, reusable services for information that make it easier for processes to get the information they need from across a heterogeneous landscape of application and data.

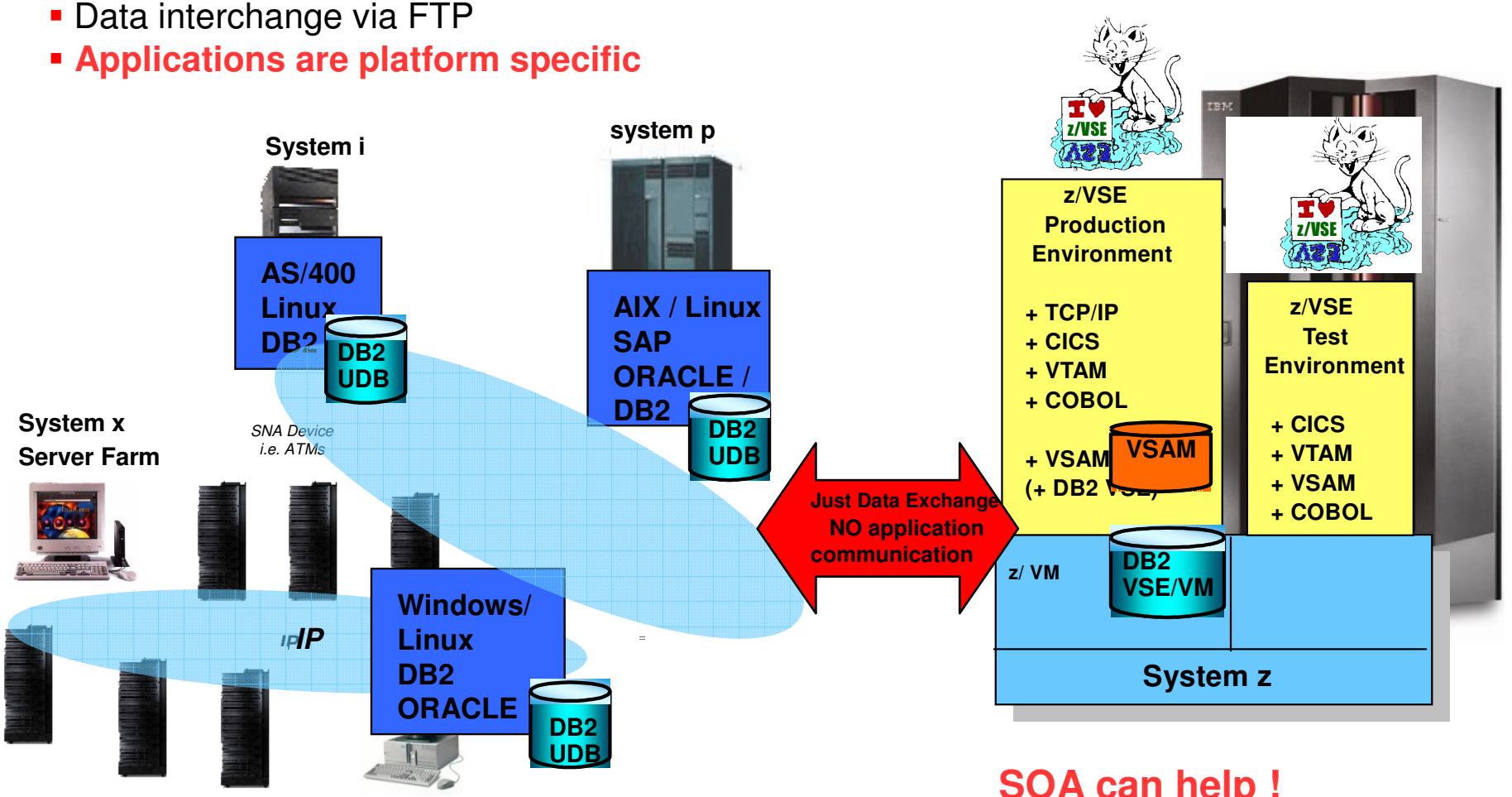
- Select data from sources
- Run Business logic
- Transform data to target

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 !

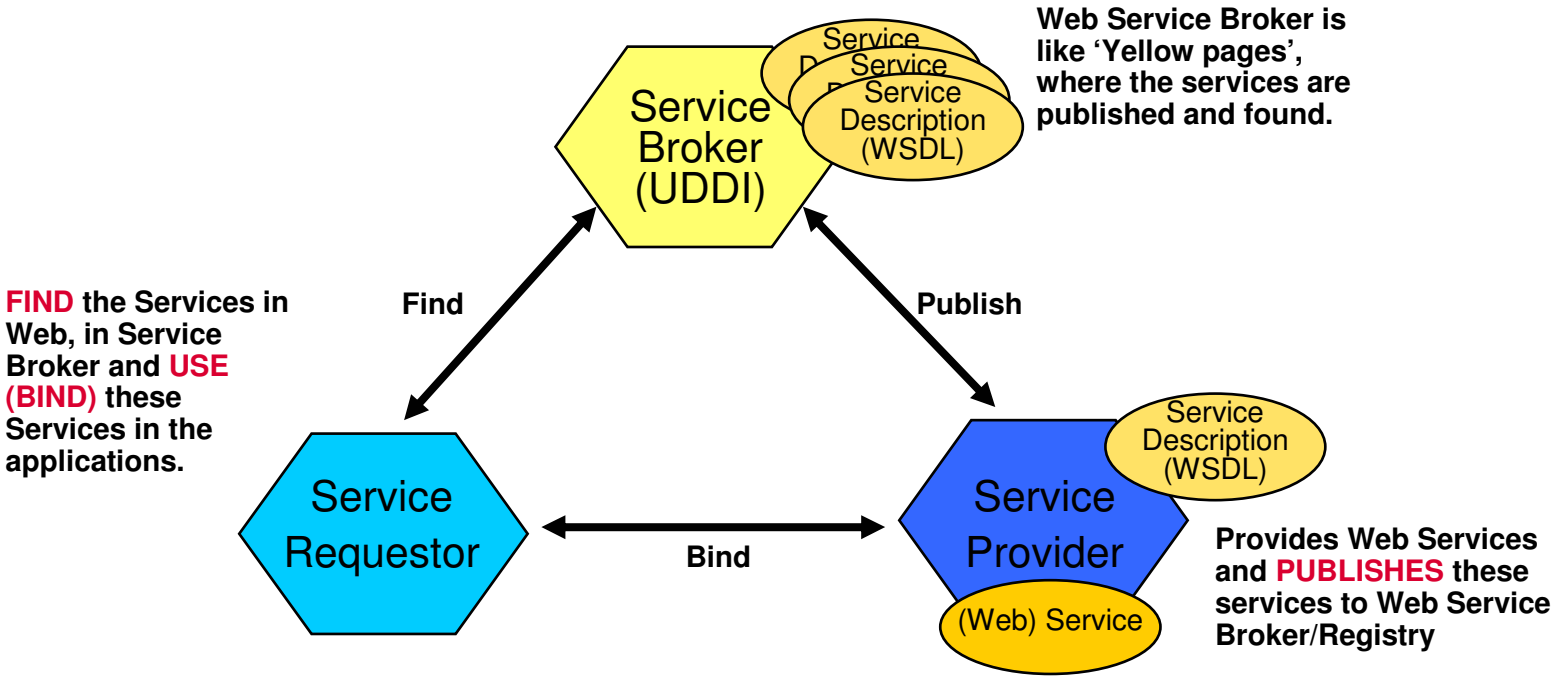
The SOA Architecture - Standards

- **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 ?

„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**

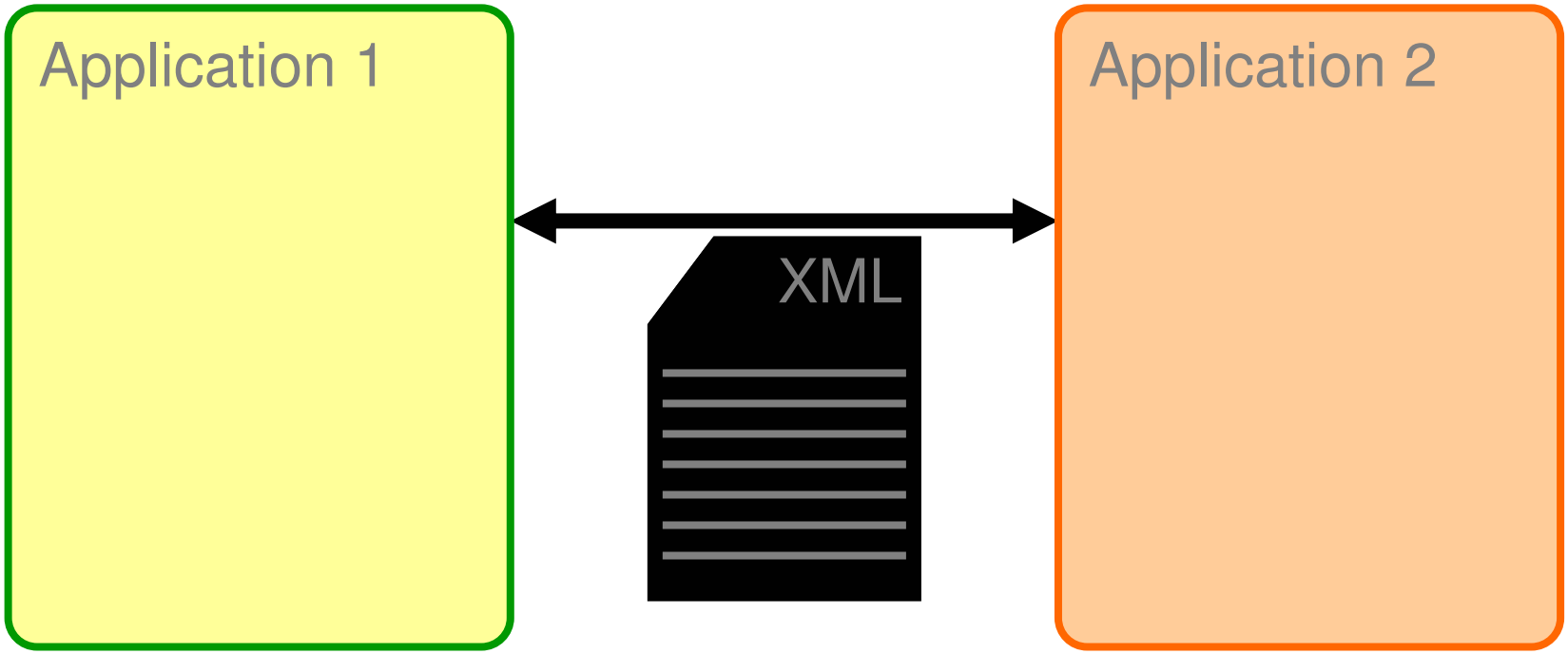


What are Web Services?

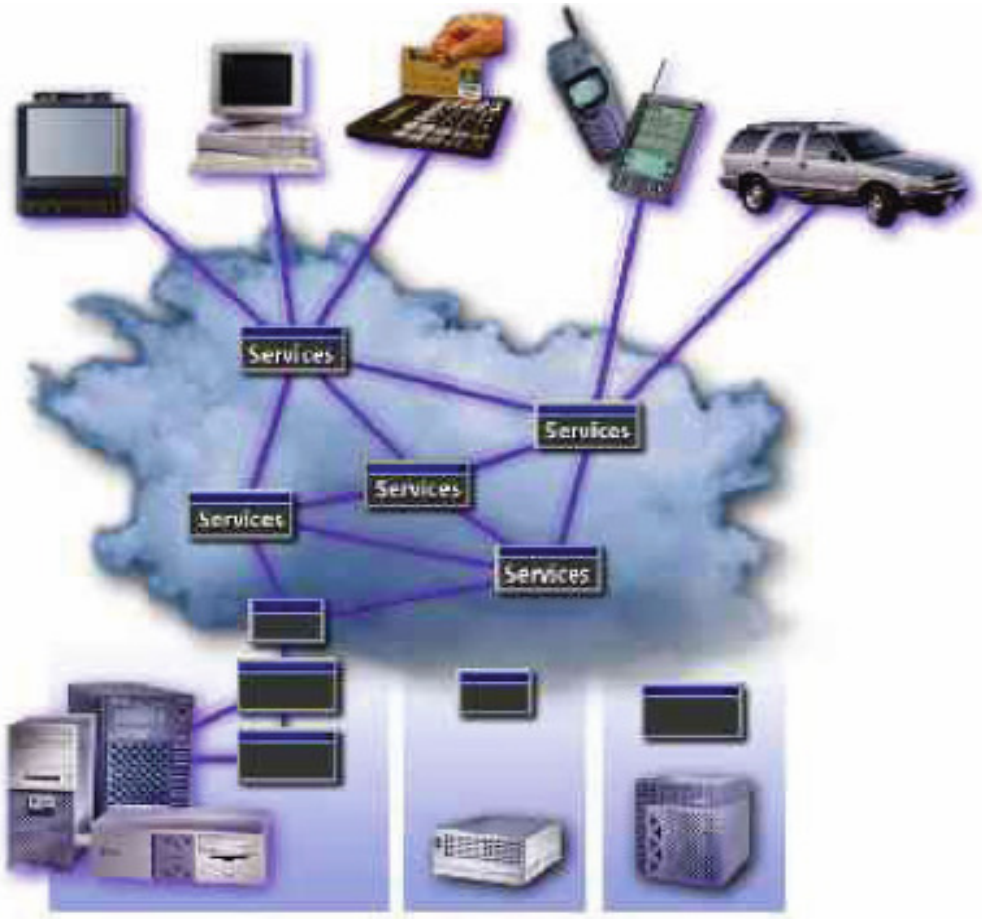
- Identified by a URI
- Interfaces defined in a WSDL using XML
- Can be discovered by other systems
- Interact using XML based messages conveyed by Internet protocols

Source: W3C Web Services Glossary

What are Web Services? Applications !



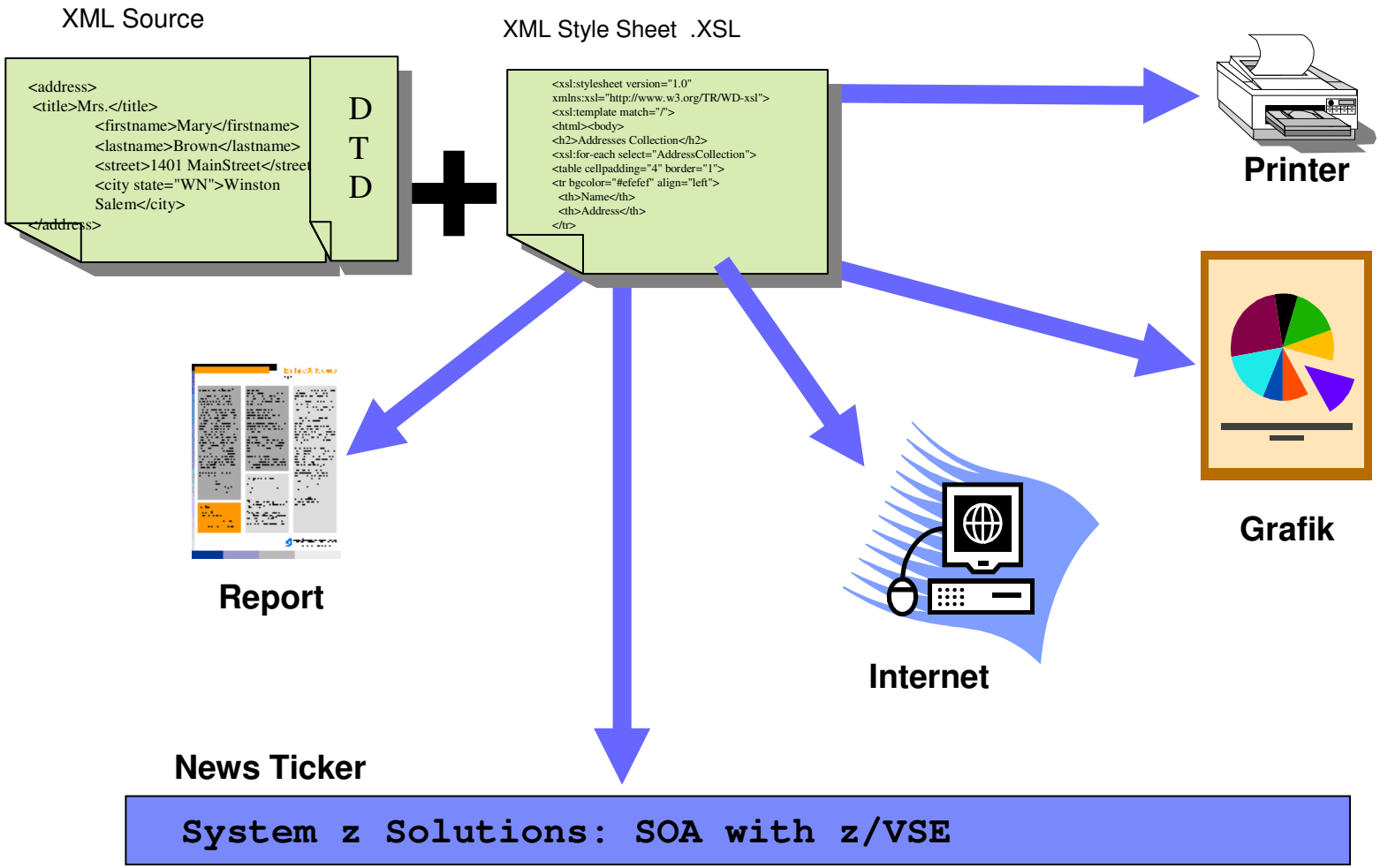
Why should we choose Web Services ?



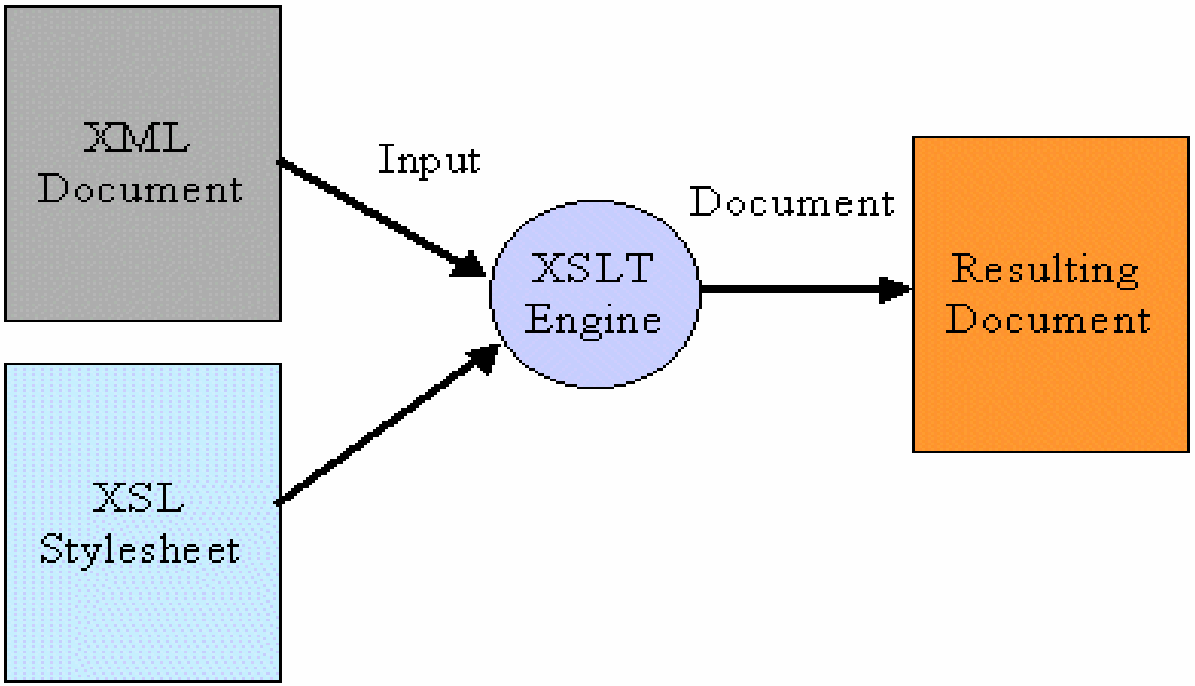
Web Services:

- 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

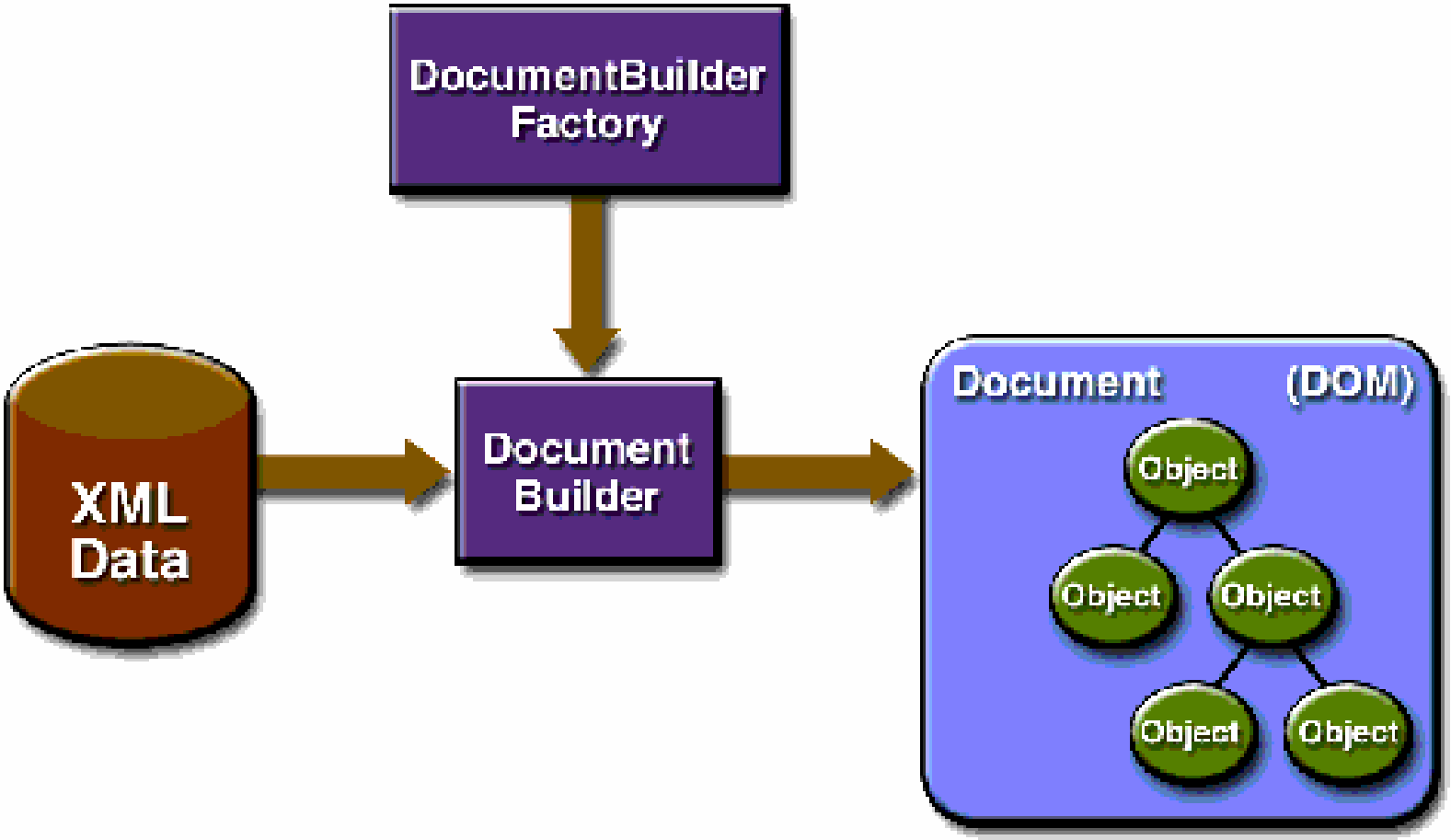
What is XML ?



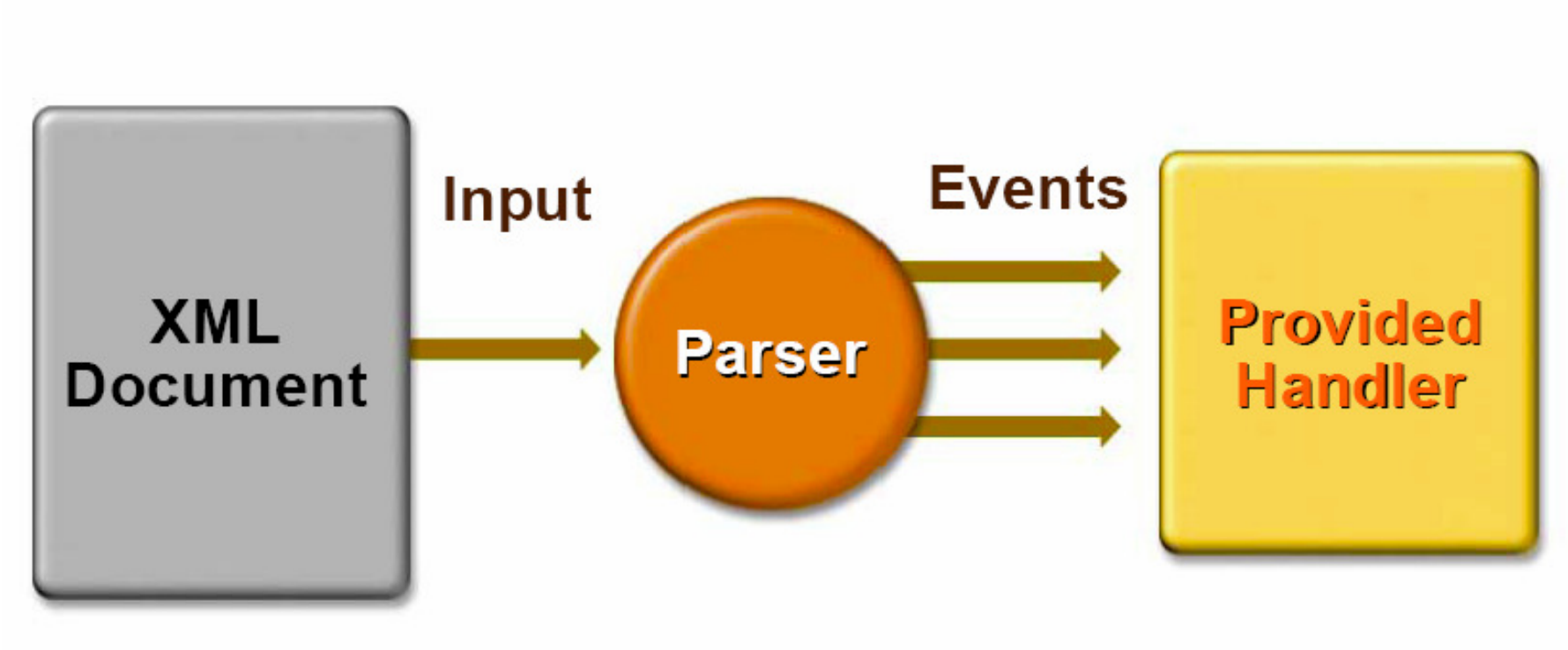
XLST Process



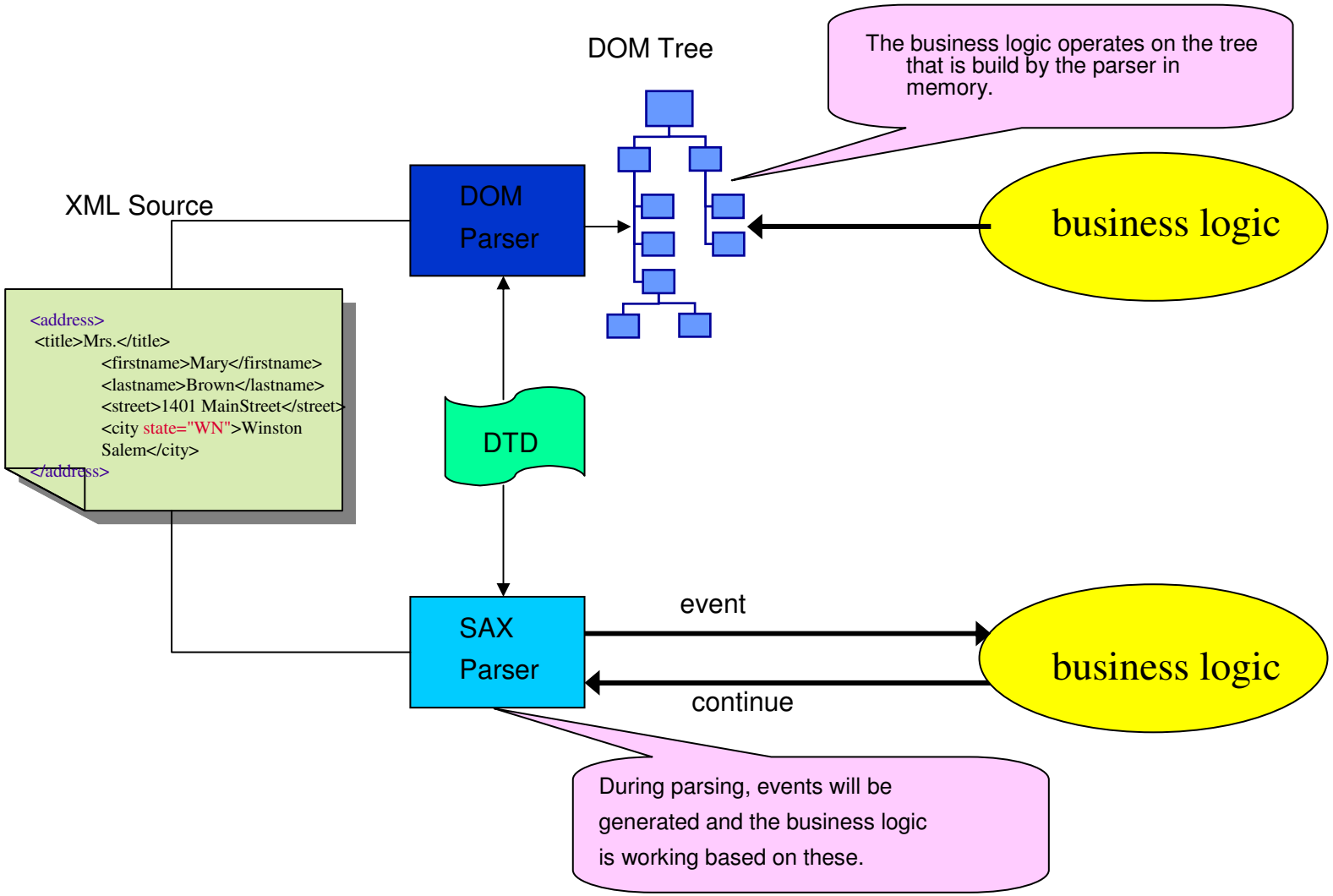
DOM Parser Process



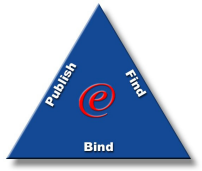
SAX Parser Even Process

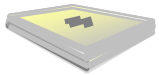


XML parser Technology (DOM,SAX)

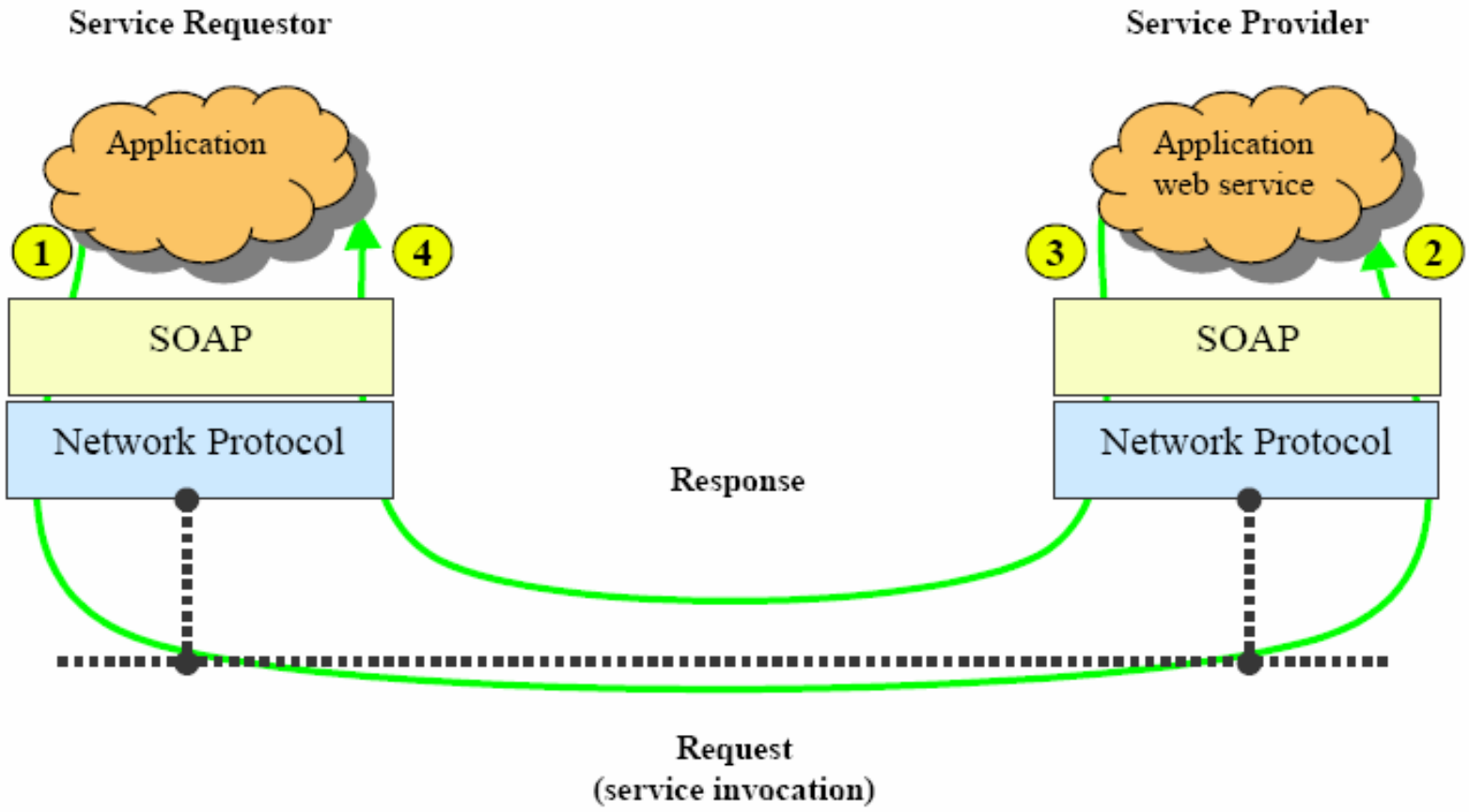


What is SOAP?

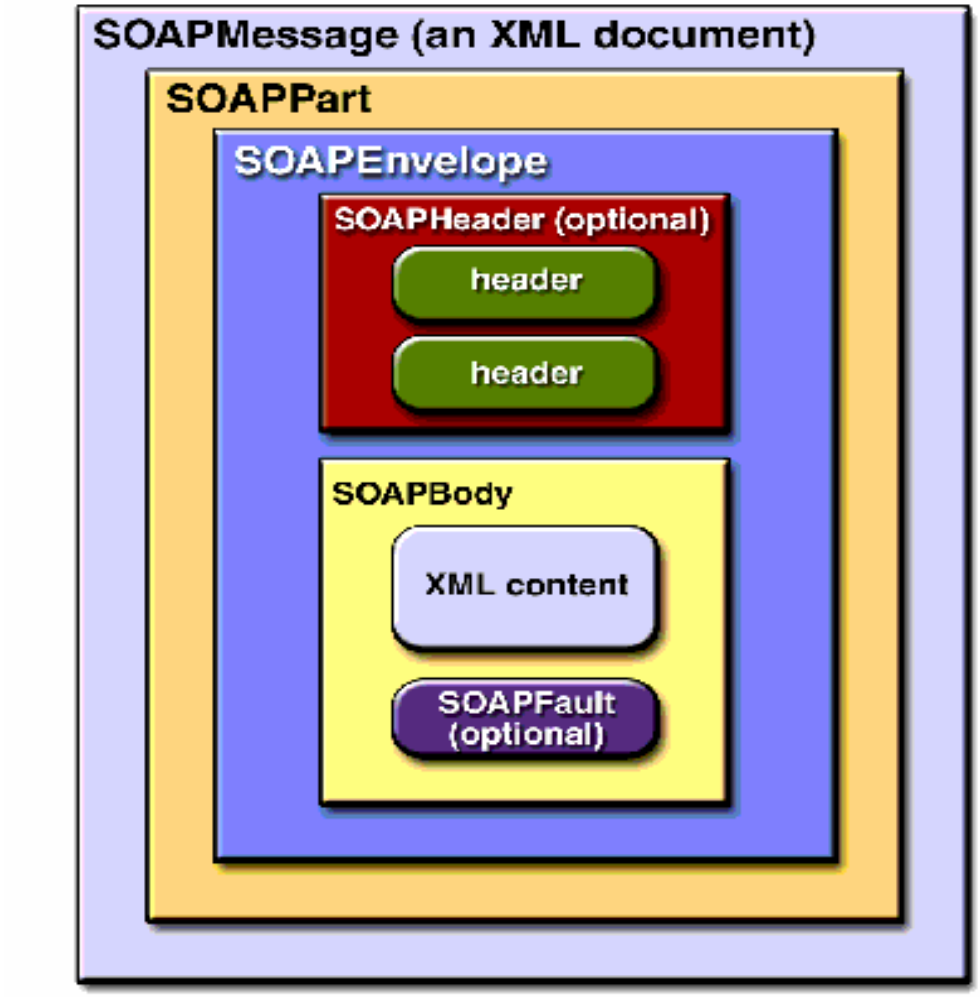


- Simple Object Access Protocol
- SOAP is an XML based protocol for communication between two remote applications:
  - is based on RPC messaging
  - is language independent (de-couples interface from implementation)
  - represents remote procedure calls and responses
- A SOAP message consists of:
  -  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

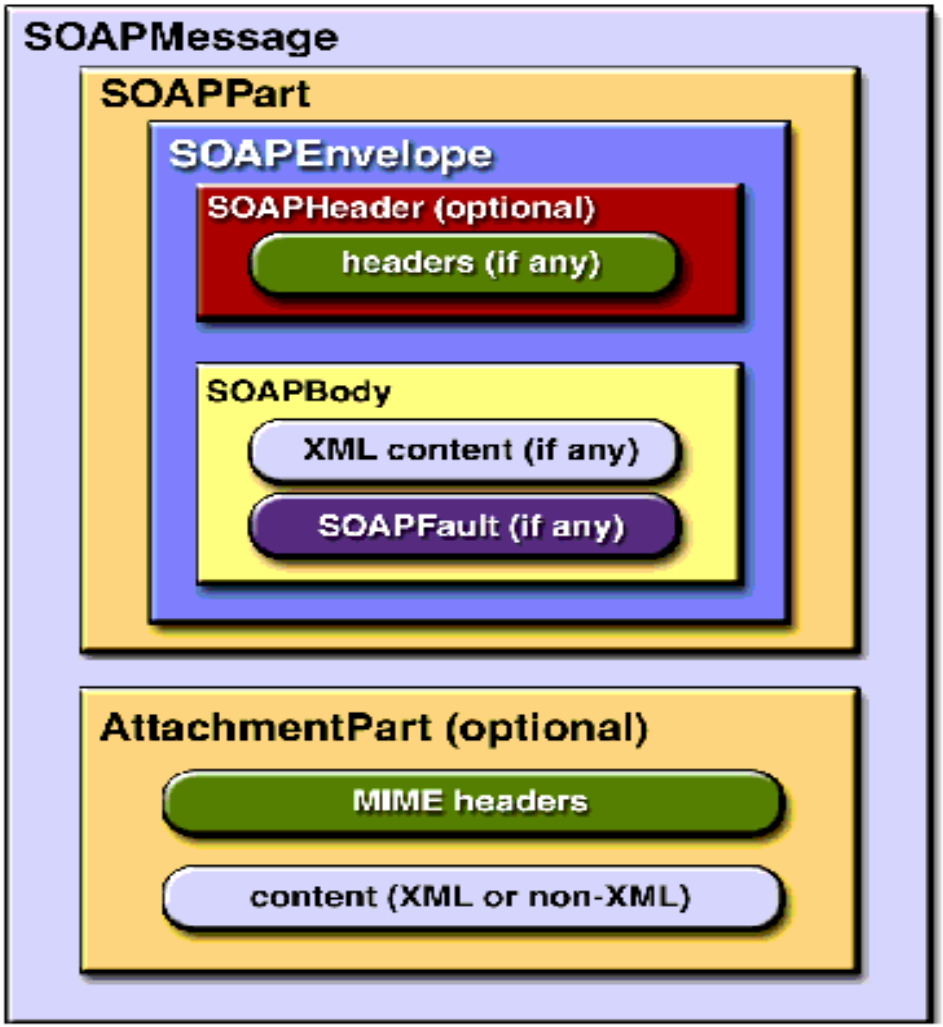
XML Messaging using SOAP



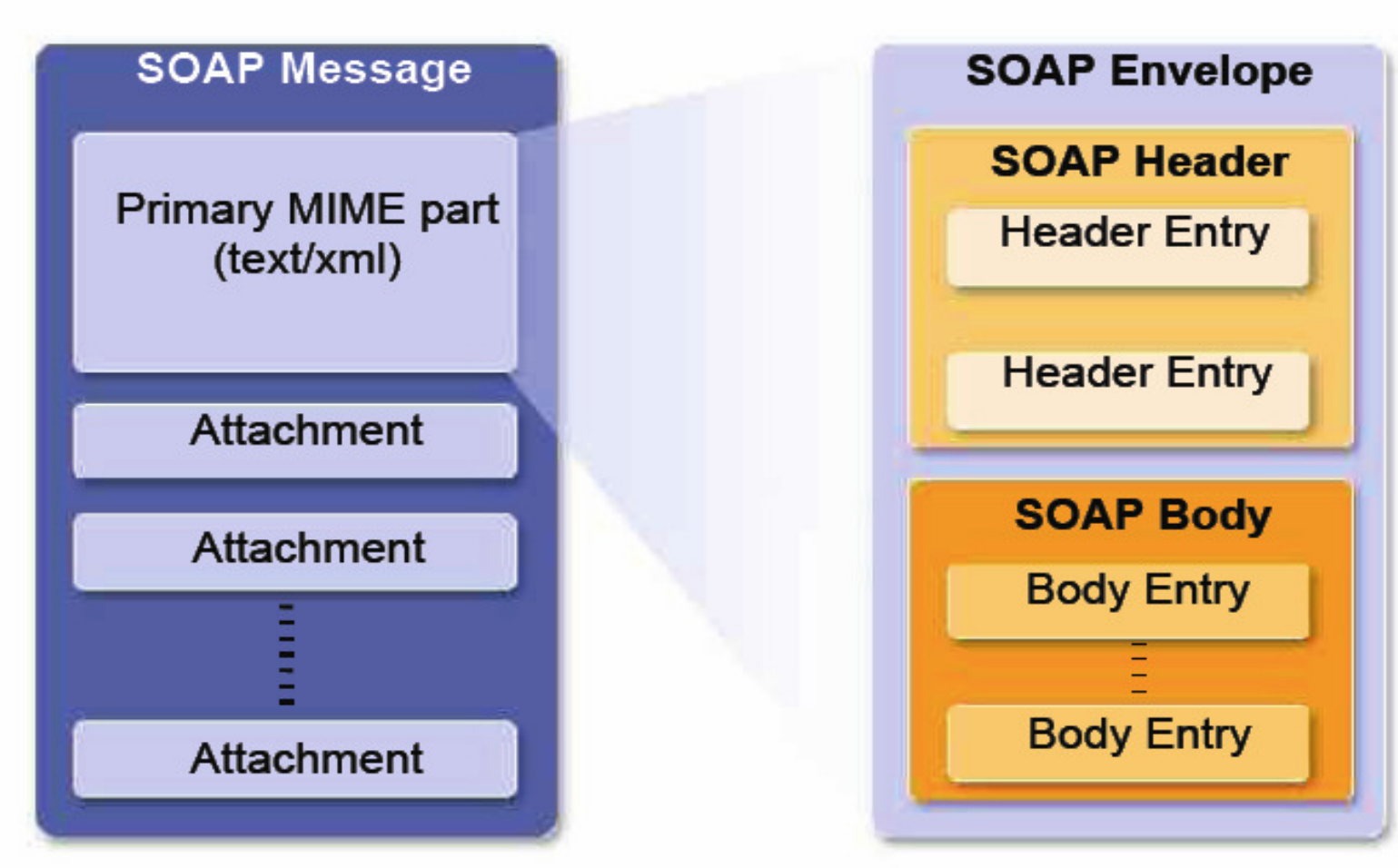
SOAP without Attachments



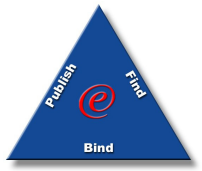
SOAP with Attachment



SOAP Message



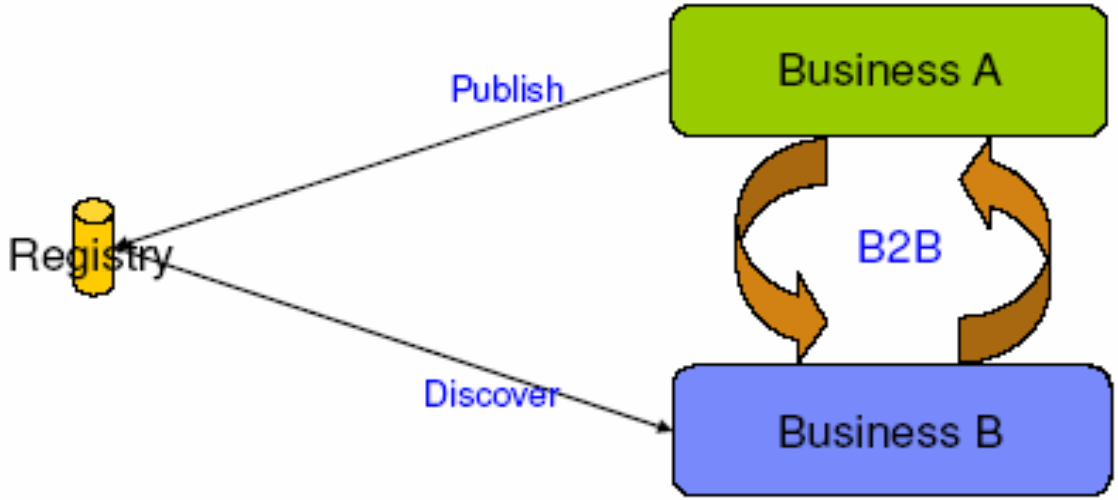
What is UDDI or Registry ?



- Universal Description, Discovery and Integration
- UDDI is a specification for publishing and discovery of businesses and the services they provide
- UDDI specifications define how to construct UDDI Business Registries
- UDDI specifications are based on XML and SOAP:
  - API to communicate with a UDDI Registry are SOAP based
    - UDDI4J (UDDI for Java) - Open Source implementation in Java
    - JAXR (Java API for XML Registries) - Sun
  - data structures that define Web Service in UDDI Registry are XML based

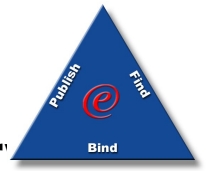
What is a Registry?

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

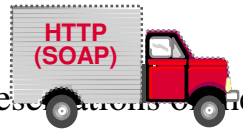


What is WSDL?

- **Web Services Description Language**

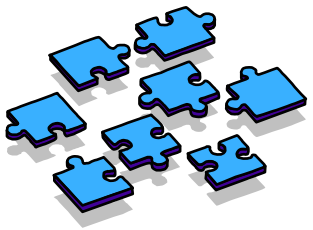


- **WSDL is an XML based vocabulary for defining a Web Ser**
  - **interfaces**
    - **operation types (i.e. one-way, request-response, notification)**
    - **messages defining a Web Service interface**
    - **definition of data types (XML Schema)**
  - **access protocol (i.e. SOAP over HTTP)**
  - **contact endpoints (i.e. Web Service URL and URNs[1])**



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

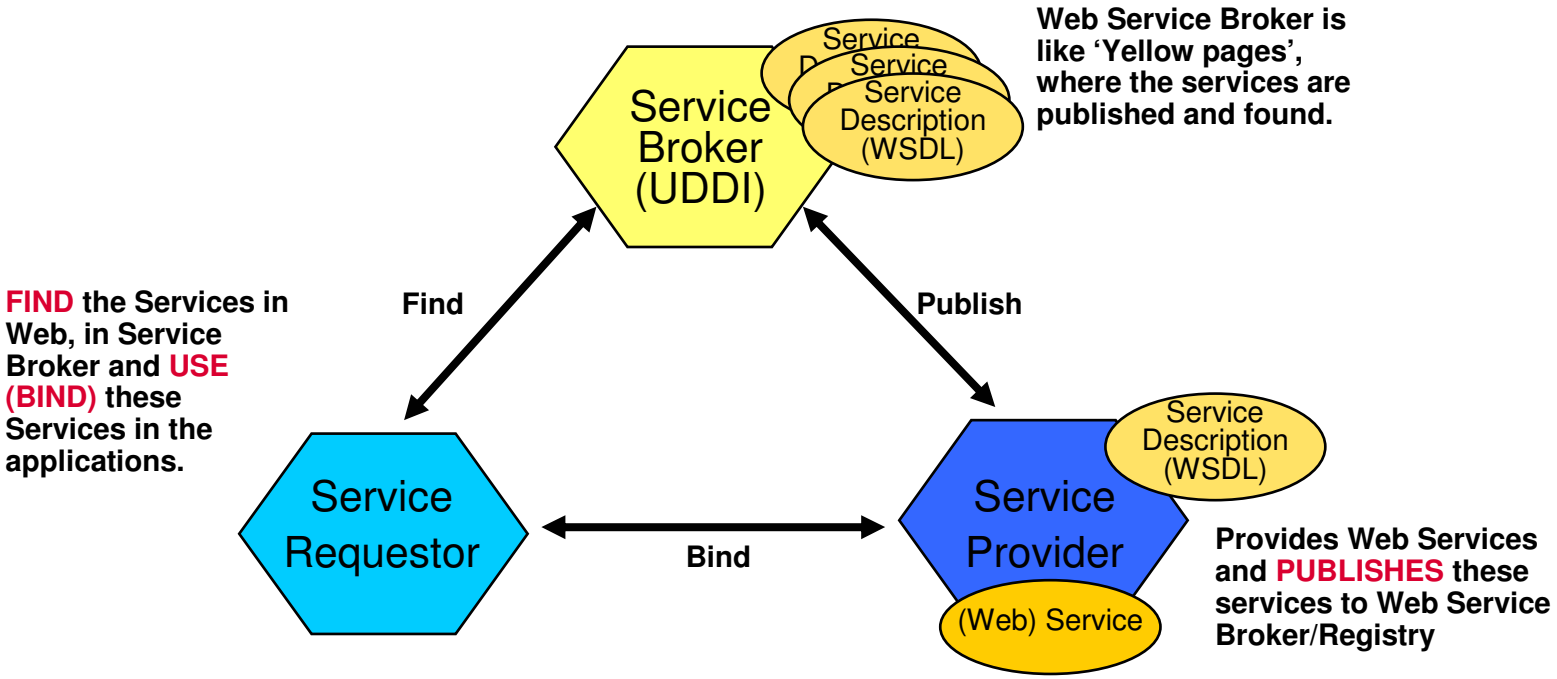
- **A Web Service URL returning WSDL makes Web Services self-describing**
- **Similar in purpose to IDL (Interface Definition Language)**
  - **From a WSDL file, wizards can generate:**
  - **proxy classes for calling Web Service**
    - **skeleton classes to implement a Web Service**



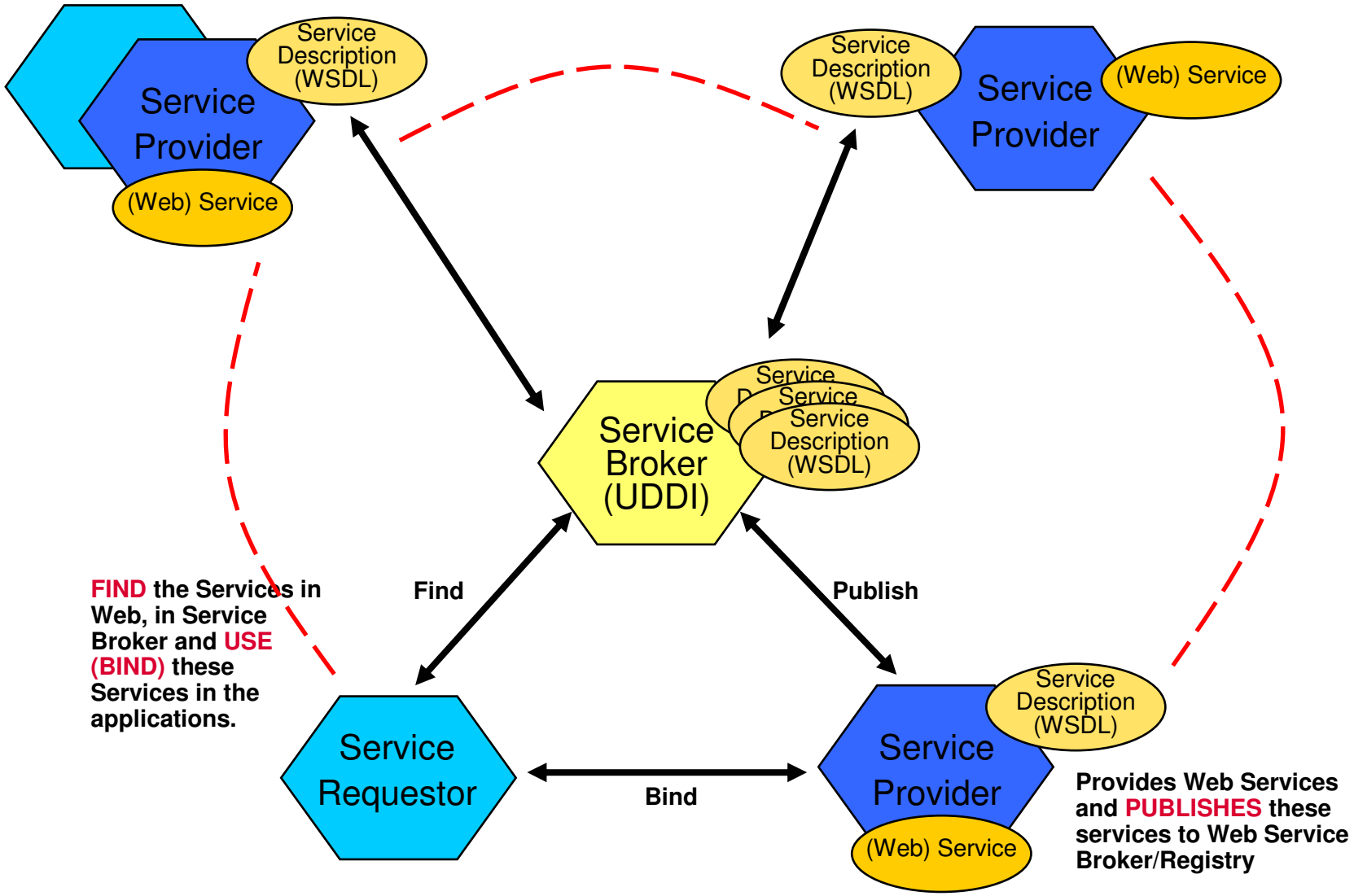
What are Web Services ?

„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**



What are Web Services ?



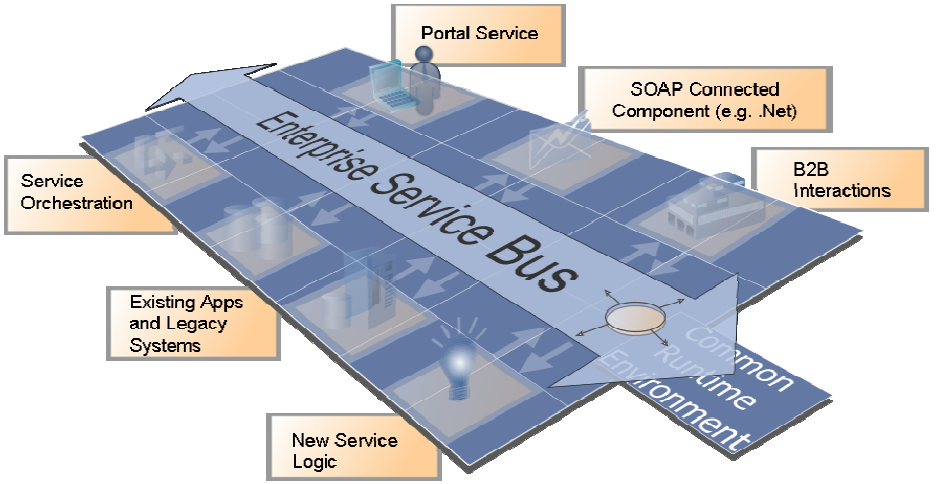
What is an Enterprise Service Bus?

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 und service

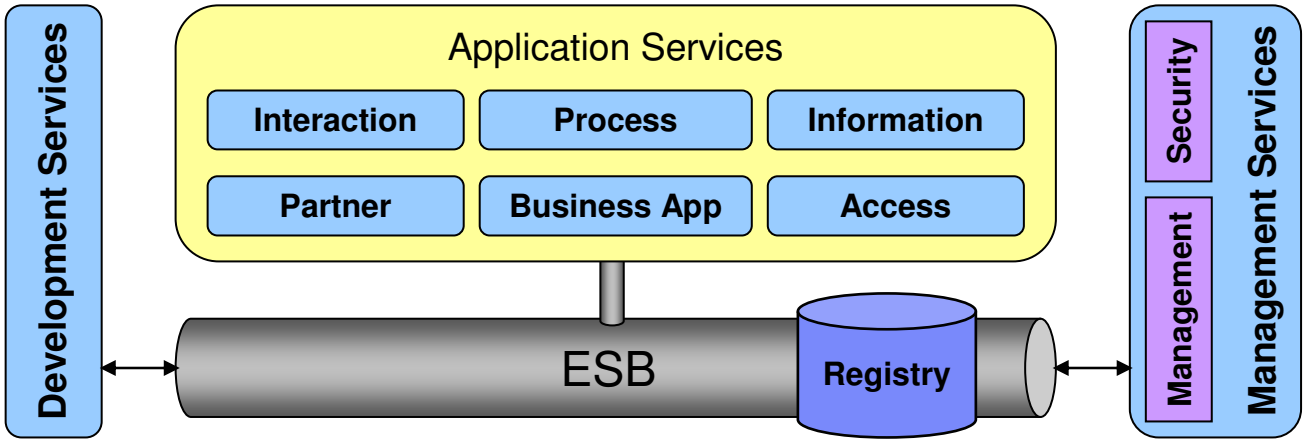
- **ROUTING of messages between Services**
- **CONVERTING the transport protocol between requestor and service**
- **TRANSFORMING message formats between requestor and service**
- **HANDLING of business events between different types of services**



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

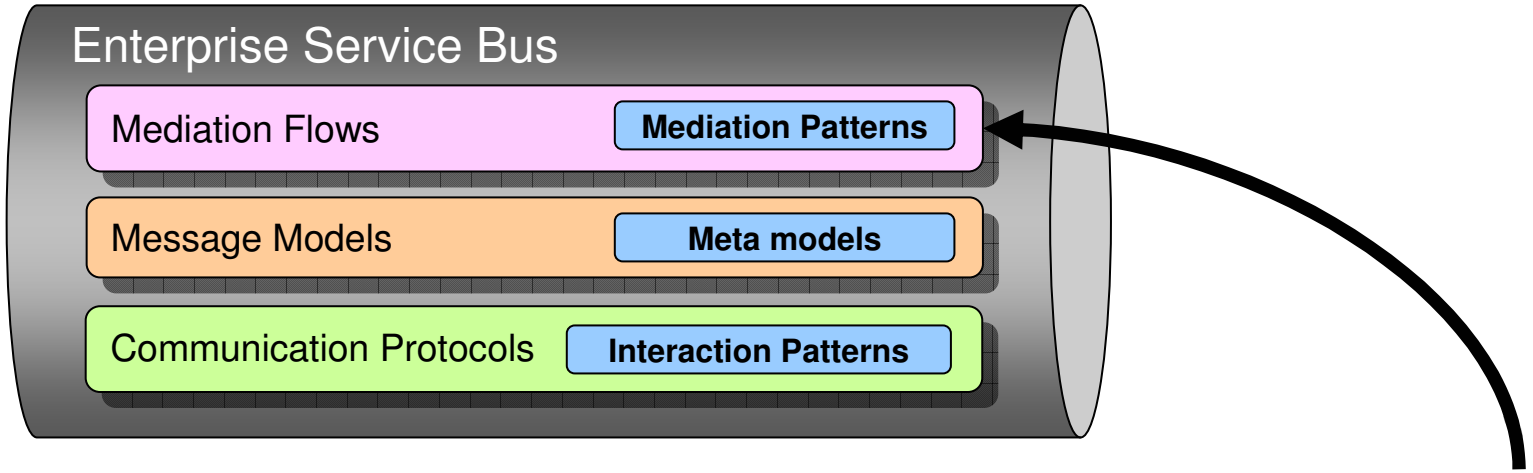
An ESB-centric view of the Logical Model



- 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/

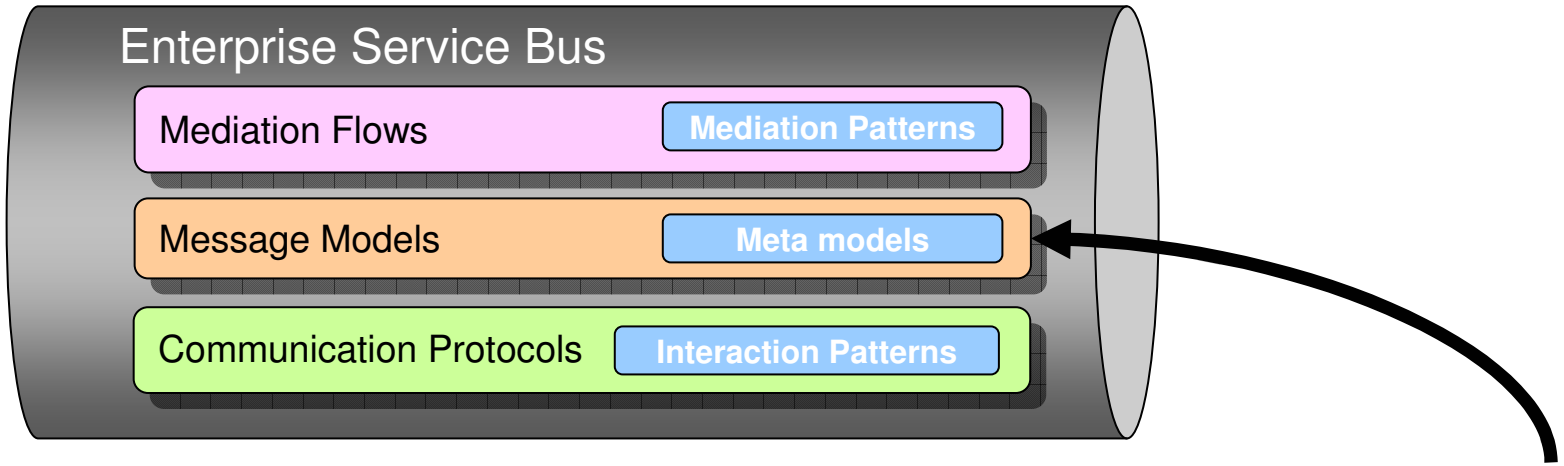
Expanded view of the ESB



- Mediation Flows
  - Process messages exchanged between requester and provider via ESB
    - Large grained
    - Moderately reusable
    - Constructed from *Mediation Patterns*
  - Mediation Patterns define processing “steps” of a mediation flow
    - Small to middle grained
    - Highly reusable
    - ESB products include pre-built “mediation primitives”

- Typical requirements
  - Dynamic routing
  - Logging

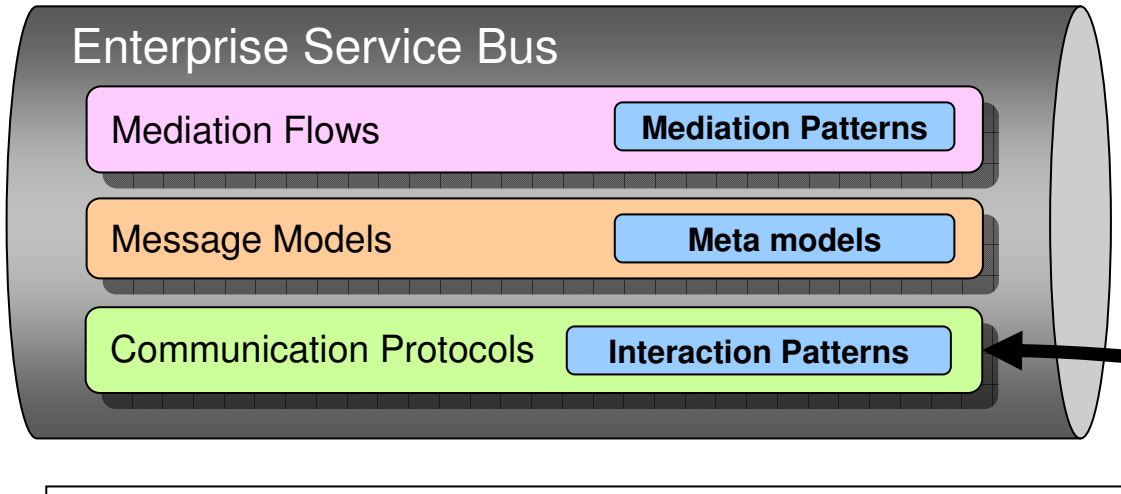
Expanded view of the ESB



- Message Models
  - Describe message content exchanged with requesters and providers
    - For example, XML schema
  - Based on Meta-models
    - Fundamental means of describing messages
    - For example, XML Schema language
- An ESB supports one or more message meta-models
- An ESB supports multiple message content models
  - Can include industry standard models as well as enterprise specific models
  - Can include weakly-typed models

- Typical requirements
  - XML schema definition
  - Industry specific content models

Expanded view of the ESB

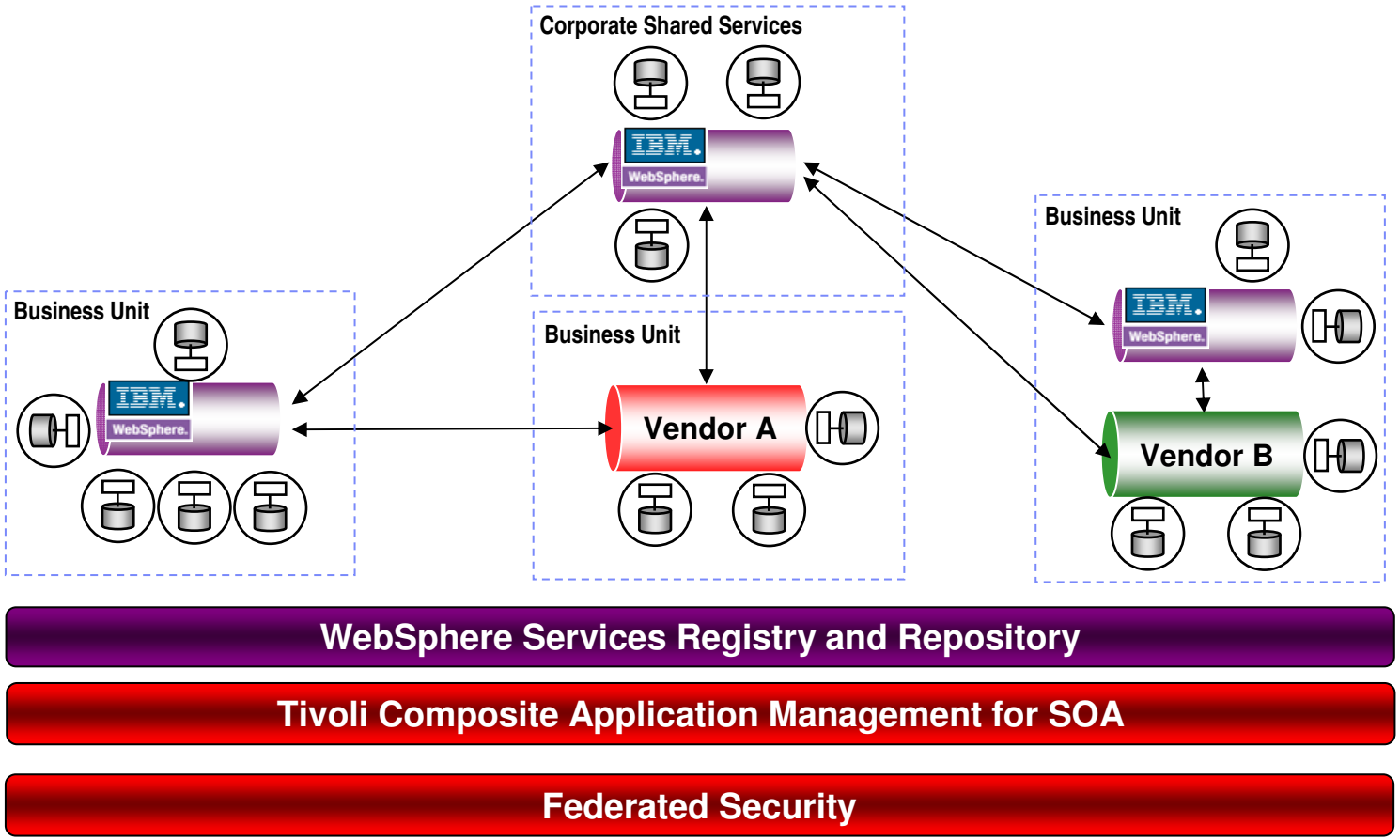


- Communication Protocols
  - Supply basic connectivity to requesters and providers
    - Impact QoS (e.g., reliable delivery, transactions)
  - Supply inherent *Interaction Patterns* (e.g., request/reply, one-way, pub/sub)
- An ESB leverages underlying communication fabrics of SOA infrastructure
  - ESB provides *on-ramps* and *off-ramps*
- Standards are important

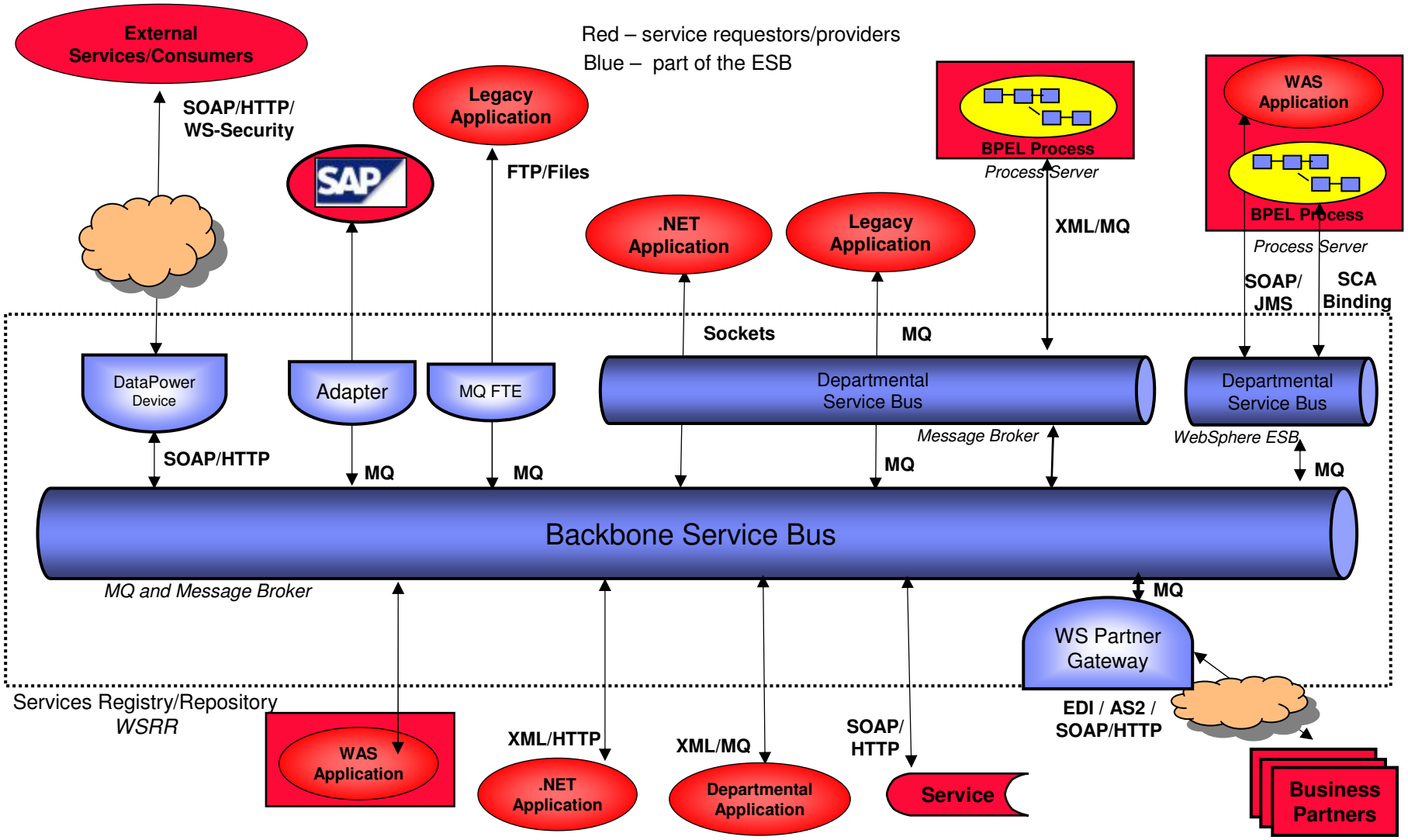
- Typical requirements
  - HTTP (SOAP/HTTP, XML/HTTP)
  - MQ (SOAP/JMS/MQ, XML/MQ, text/MQ, …)
  - Adapters (legacy, EIS)
  - WS-I, WS-Security
  - RAMP

Federated ESB Topology Patterns

A single enterprise-wide ESB is rarely attainable –
most businesses will have multiple ESBs across business units

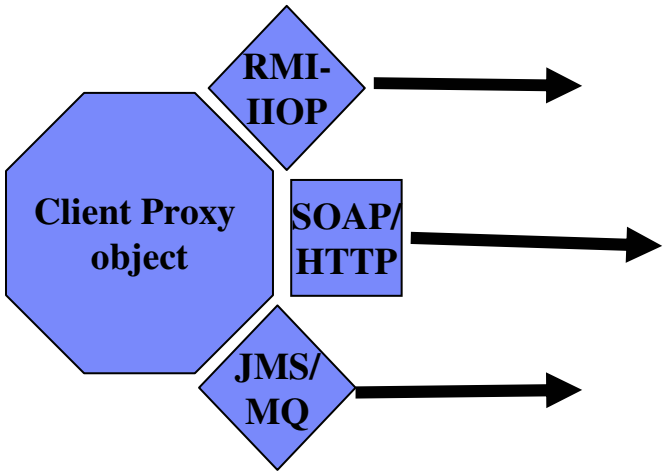


Example of Federated ESB



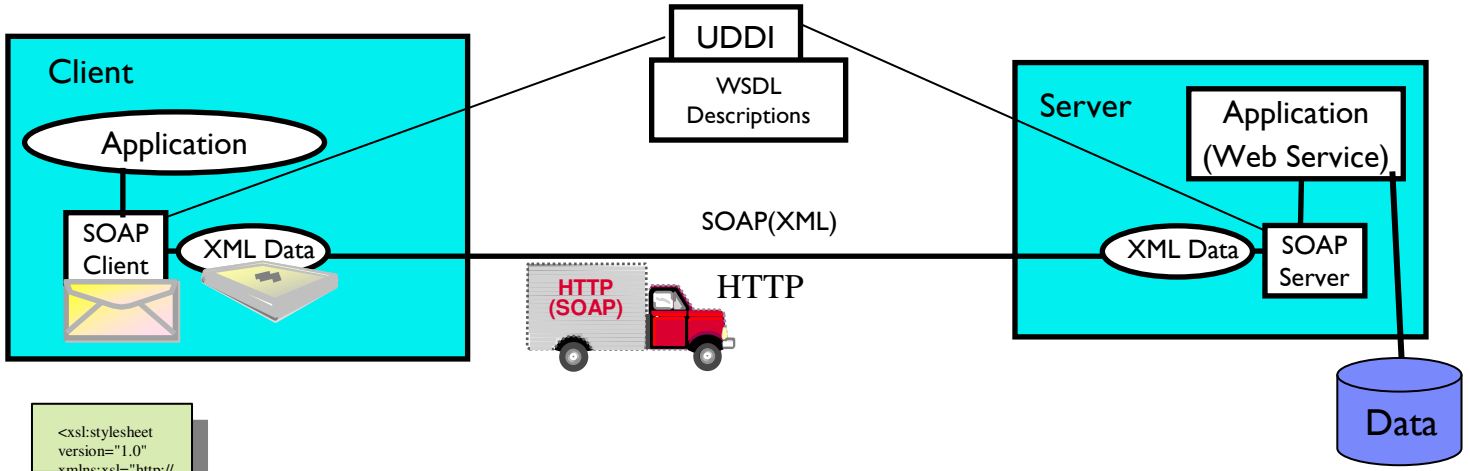
Client invocation

- Single stub can invoke services over different bindings
  - Depends only on abstract interface.
- Are independent of binding (but pluggable).
  - Add new bindings without recompiling/redeploying stub
- Allows optimisations based on the bindings of service.
- Will support extended services models if described In WSDL



Web Services in action

XML Document + SOAP Protocol = Web Services



XML -

SOAP -

HTTP - **Carrier**

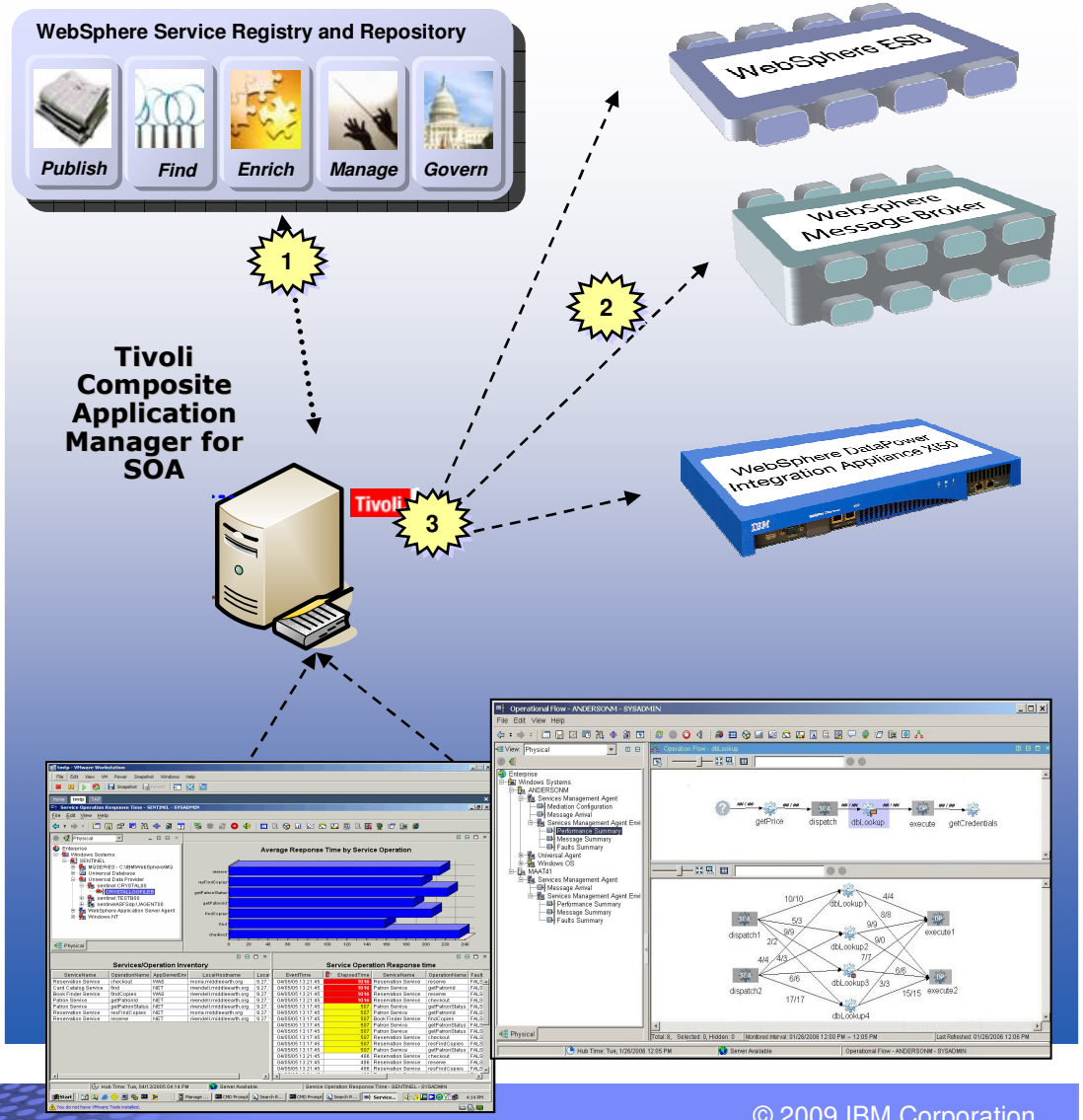
TCP/IP - **Street**

A ***web service***

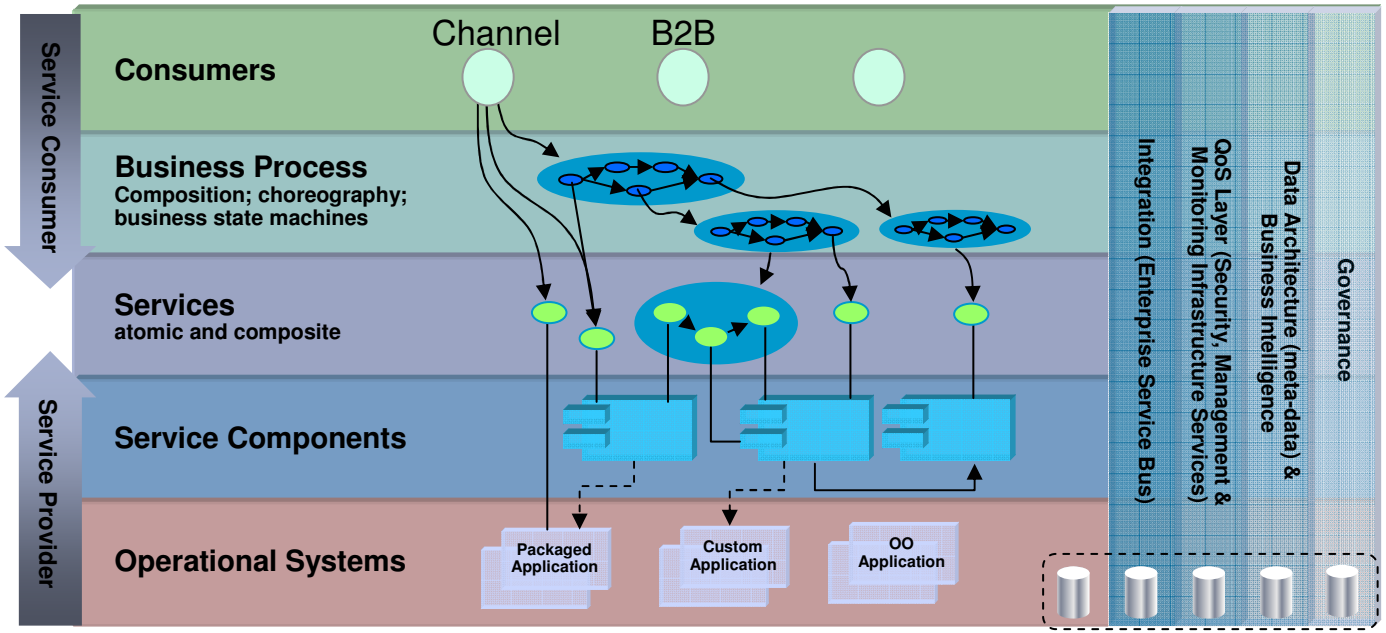
- implements a bussiness, application or system functionality
- is intended for application communication
- is useable in internet, intranet, extranet
- is useable for browser-based solutions up to the B2B integration between companies
- uses only standard internet technologies

Manage SOA:
IBM Tivoli Composite Application Manager for SOA

1. Service Discovery & Reconciliation
   - ITCAM for SOA discovers *"rogue services"* running in production by comparing with registered services in WSRR
2. Service Monitoring & Logging
   - ITCAM for SOA maintains logs of service calls in data warehouse (for historical reporting)
   - Monitors for service degradation and thresholds defined by service level agreements
   - Automated 'take action' commands to re-route services to meet SLAs
3. Views Optimized for Operations and Web Services
   - Tivoli Usage & Accounting manager creates reports for chargeback
   - Tivoli Service Level Advisor creates SLA reports for compliance



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

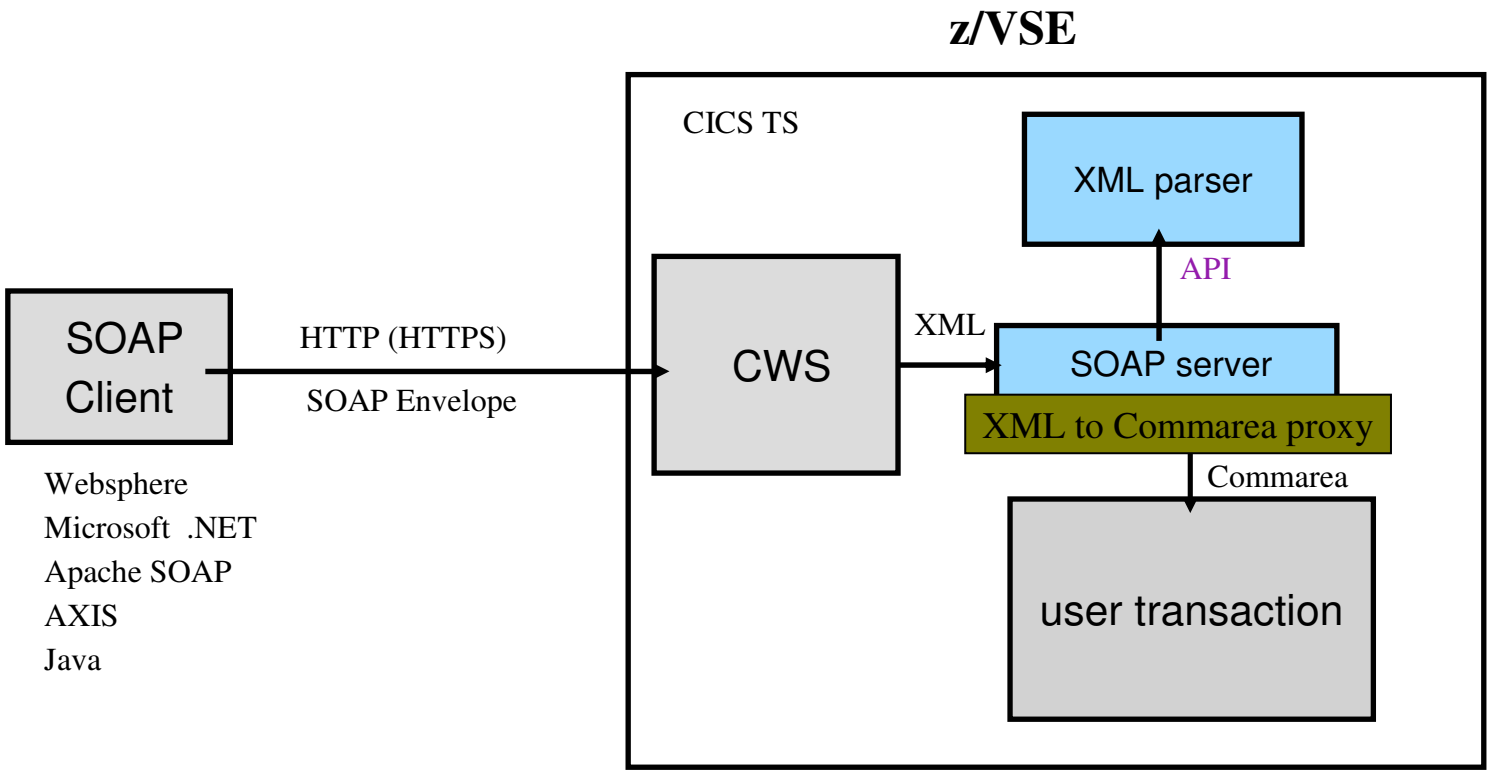
Why should VSE customers consider SOA?

- **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)

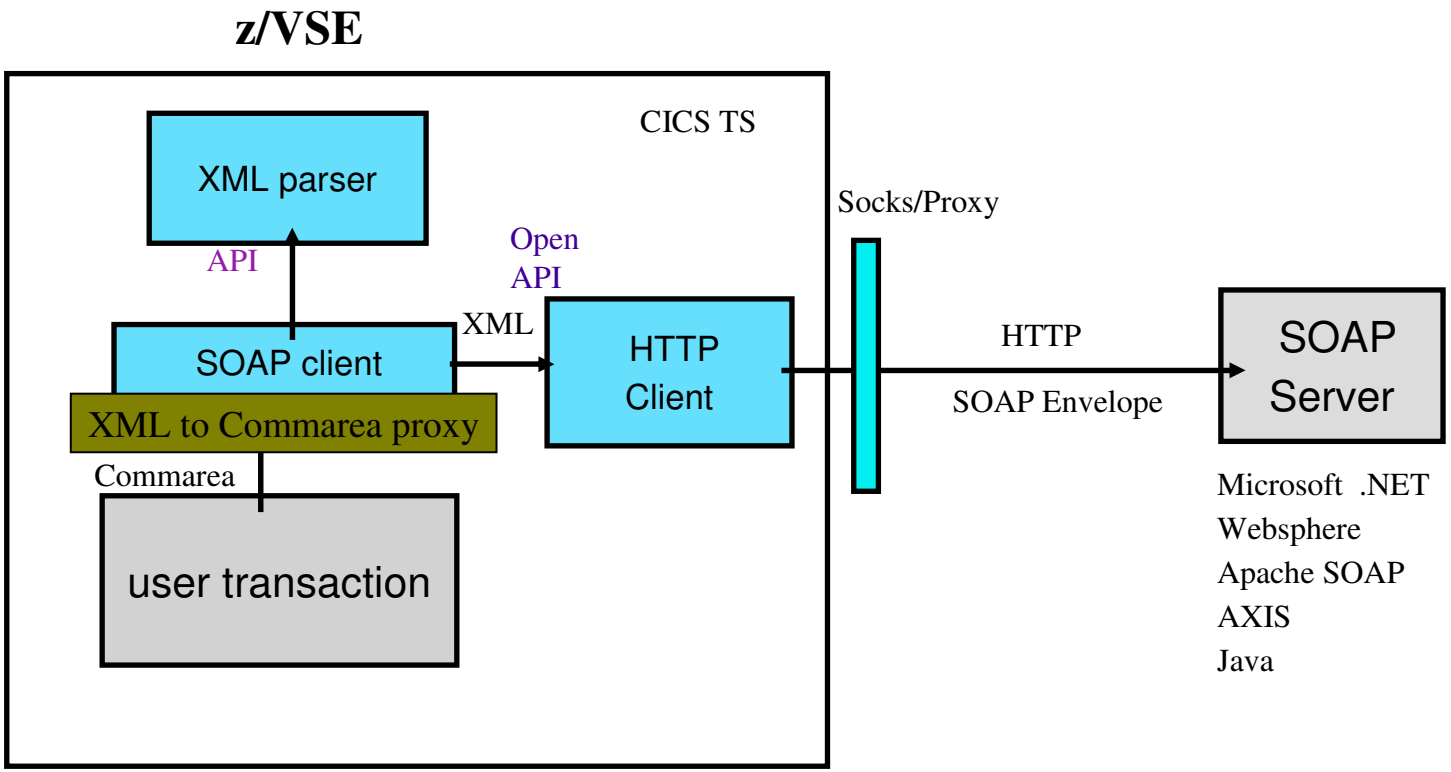
SOAP - Simple Object Access Protocol
(platform independent remote procedure call)



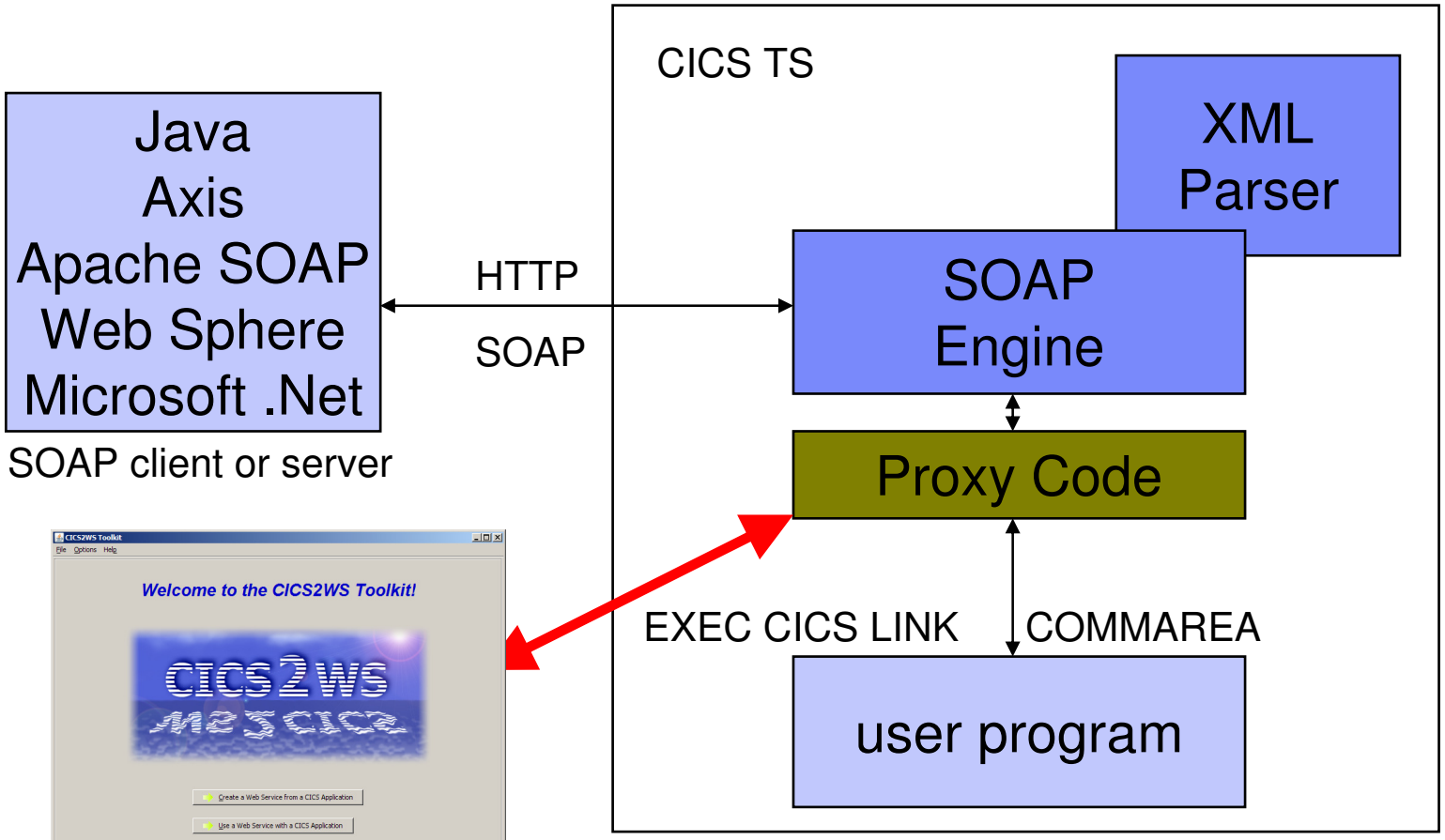
VSE as SOAP client

Web Services (SOAP)

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

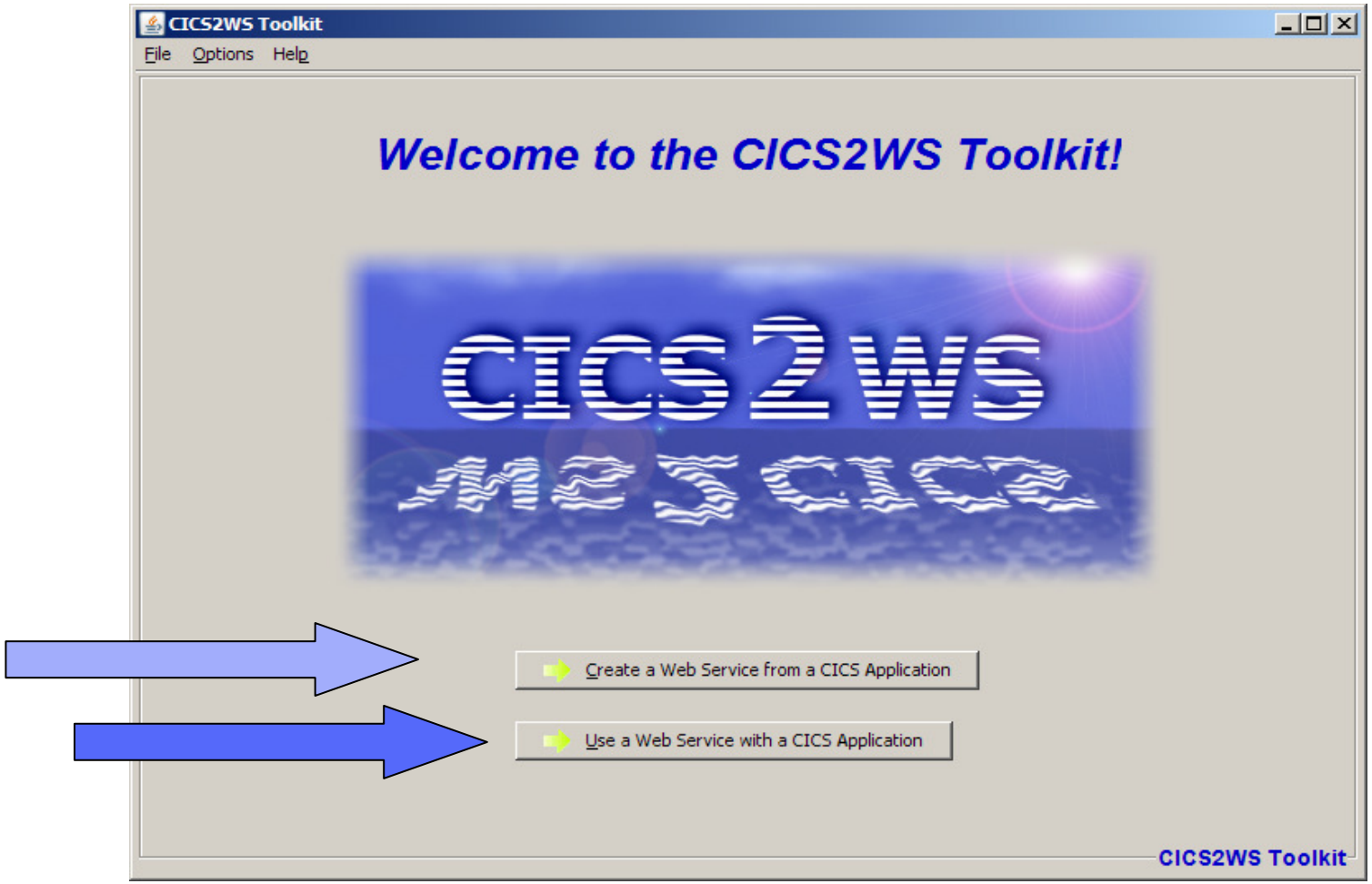


Web Services in and with VSE



Tool to generate the Proxy code

CICS to Web Services Tool

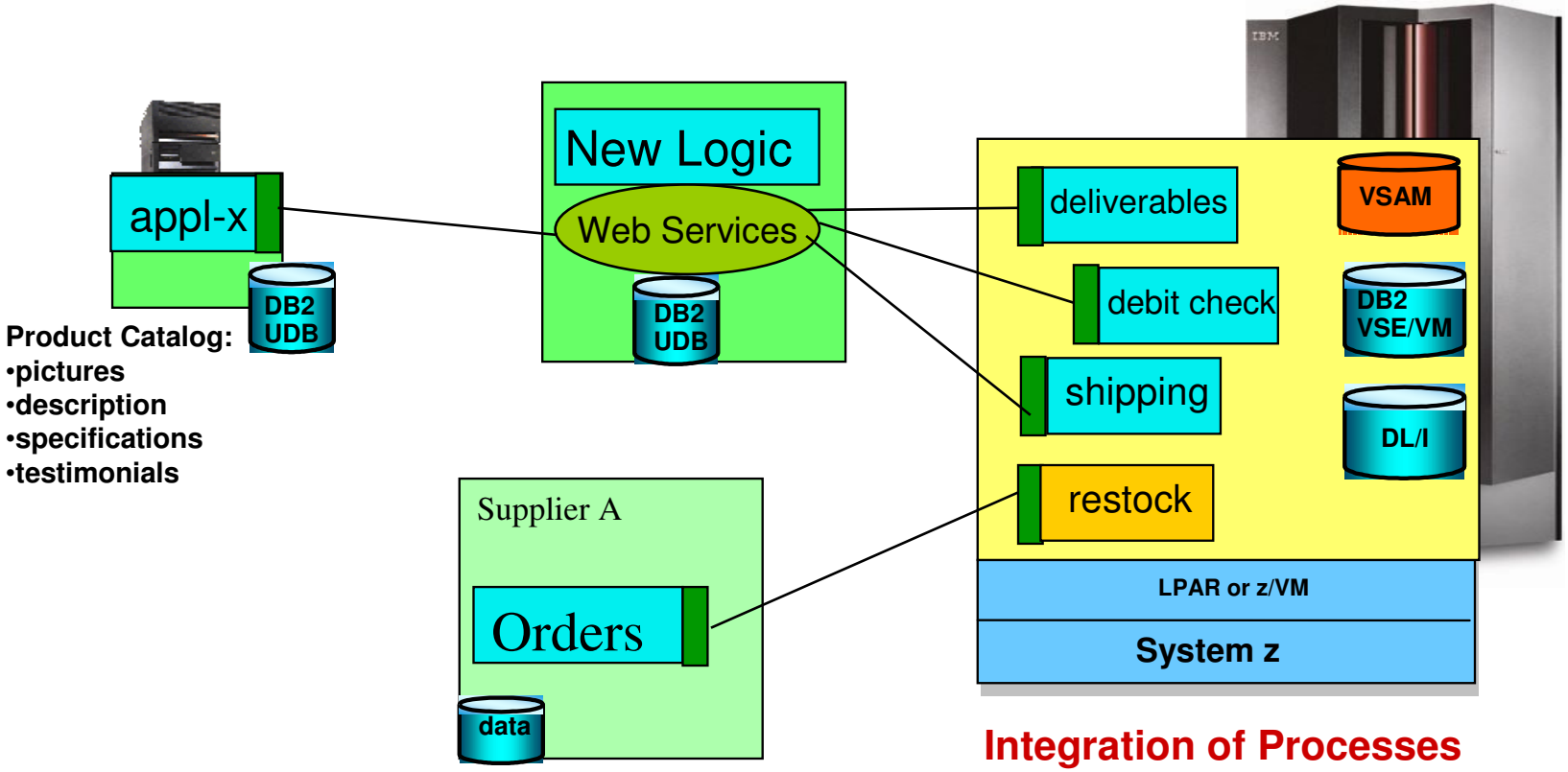


http://www.ibm.com/servers/eserver/zseries/zvse/downloads/

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

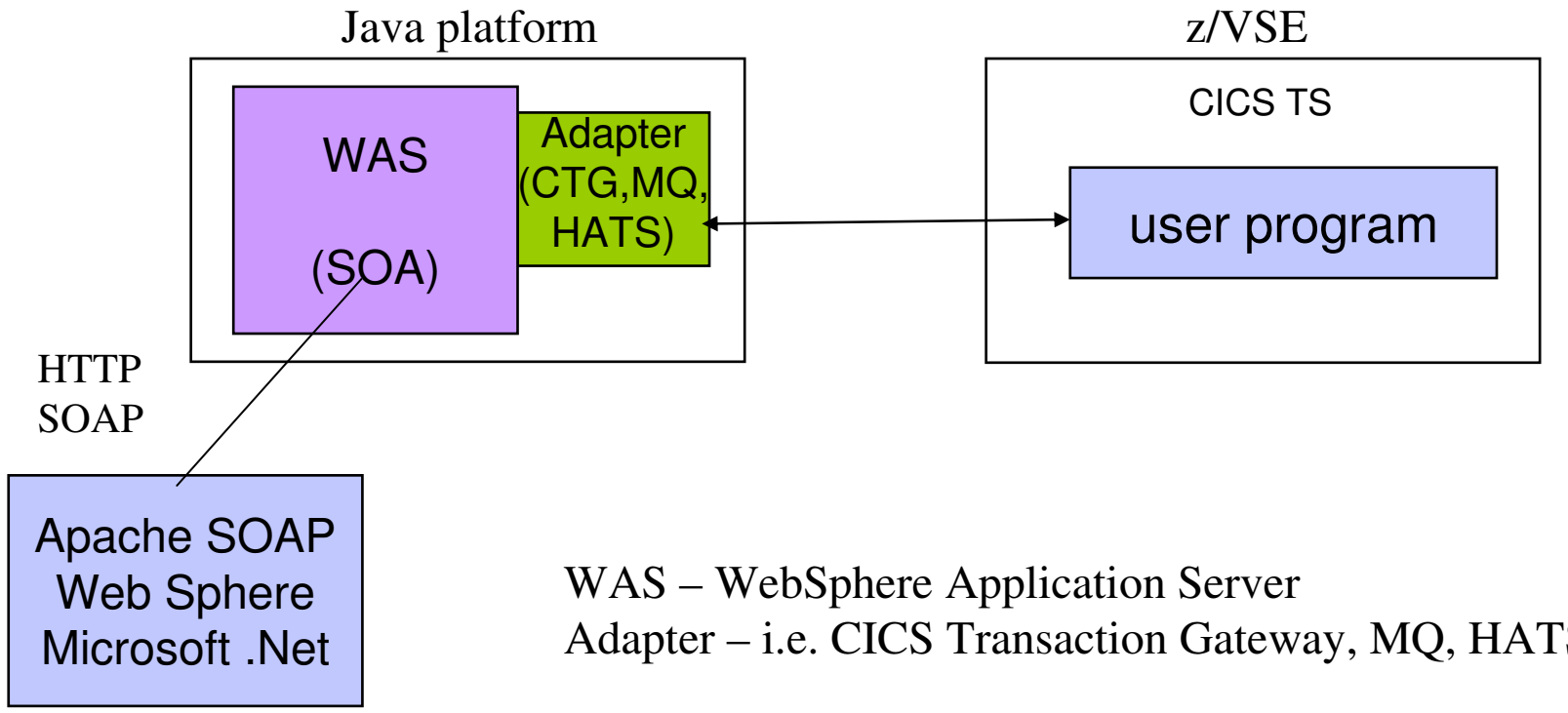
Increases business for the Company



Integration of Processes

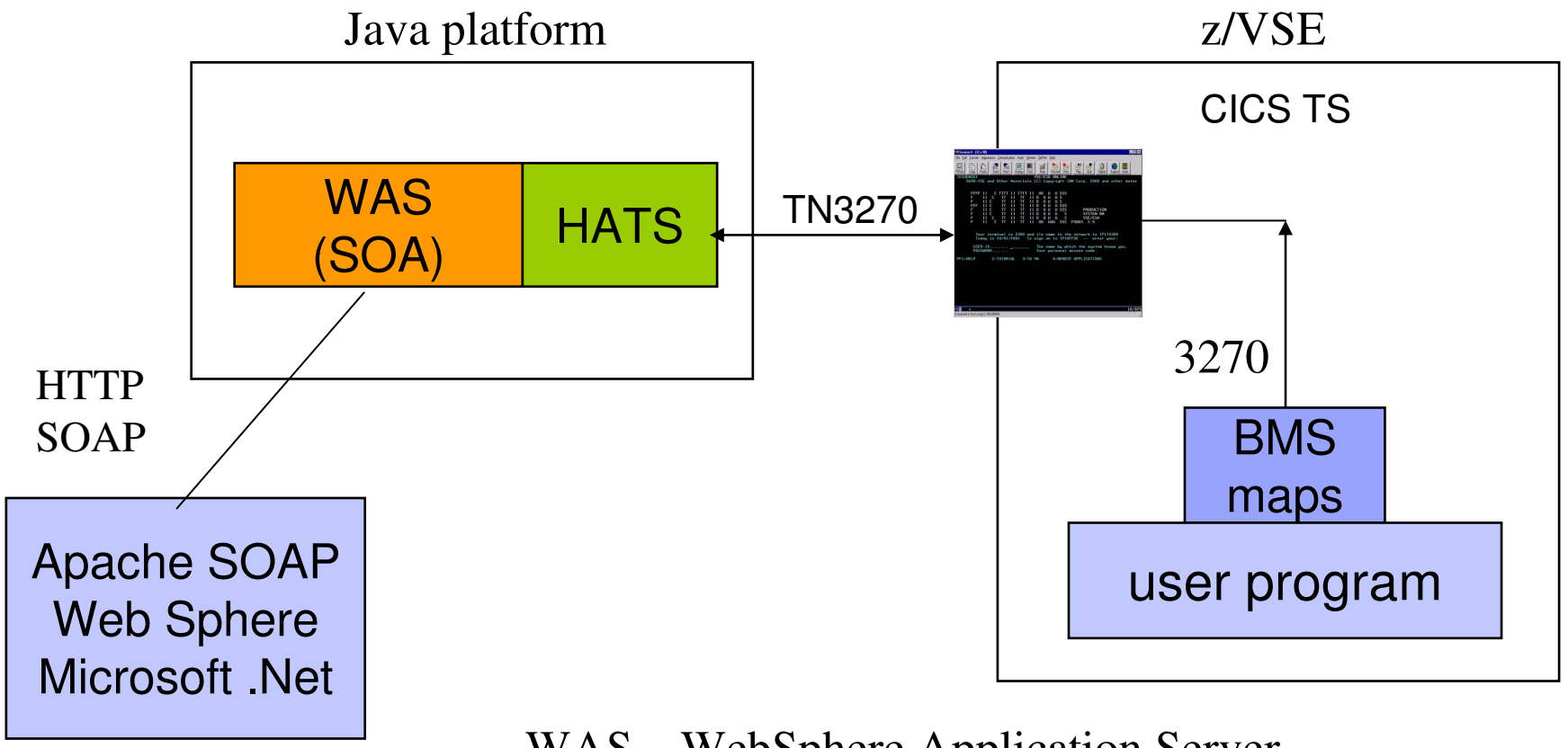
Web Services with Middle tier and z/VSE

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



WAS – WebSphere Application Server
Adapter – i.e. CICS Transaction Gateway, MQ, HATS

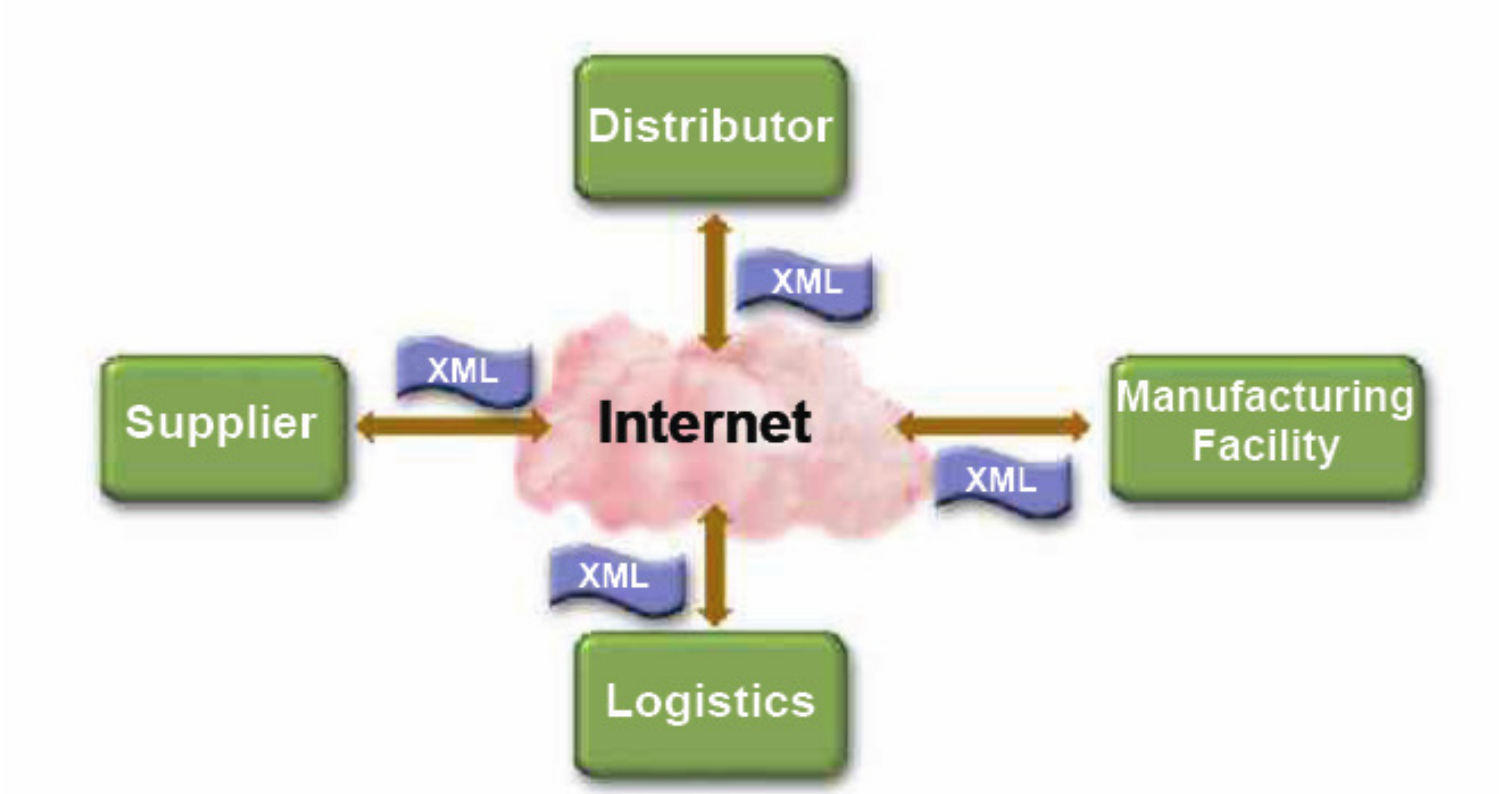
Web Services with 3270 applications



Solutions where SOA is not the best architecture

- For high performance requirements
  - Communications using SOAP/XML are time consuming
- For the transfer of large amounts of data
  - XML data can get huge
- If transaction security is required
  - Two phase commit has to be implemented in the logic
- For real time direct access to data
  - SOAP is program to program communication

Web Services Usage



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

Reusing Services

Business Challenge: Leverage existing assets to improve business agility

Sysdat & Gautzsch

Reused existing RPG service based function to integrate with Amazon.de sales portal

Online in 3 weeks. ***ROI in < 3 months.***

WebSphere Business Integration Express, Partner Gateway

Acesita

Integrates mySAP and existing backend applications with **reusable service data**

Real-time views of critical cost and profit information for **better decisions**

IBM BCS, WebSphere MQ and Message Broker

Mainsoft and Comtec

Reuse .NET applications in open standards-based SOA with IBM Business Partner Mainsoft Solutions

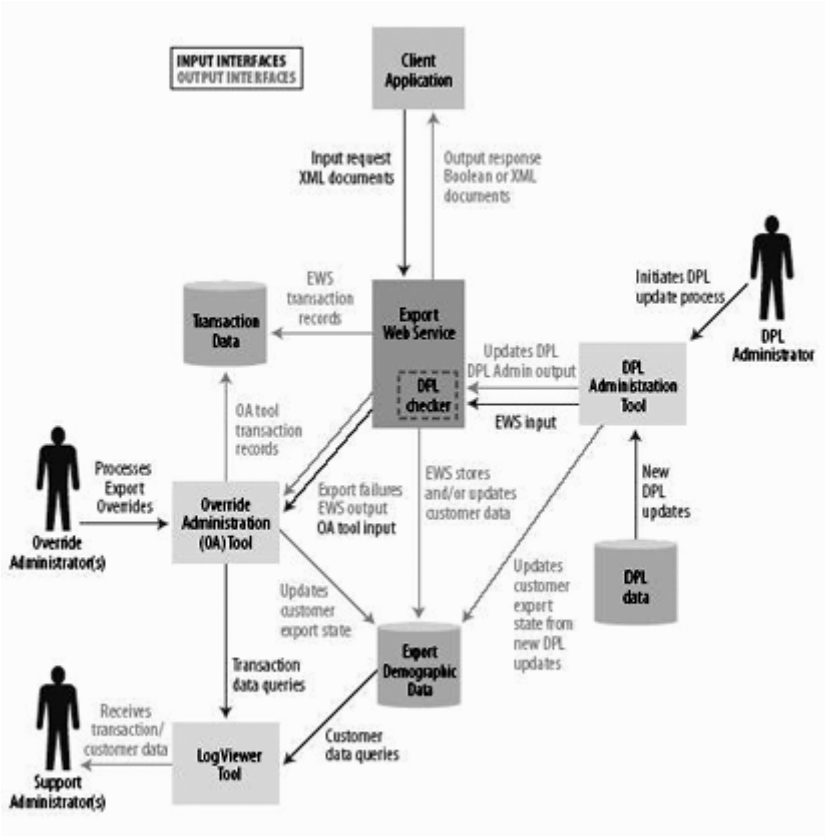
5x faster than rewriting the code from scratch

Mainsoft Visual MainWin, WebSphere Application Server

SOA Sample: US Export Validation—Regulatory Compliance

Business need:

- Worldwide access to actual Regular of the US Export Regulations
  - "IBM must comply with U.S. export regulations for product deliveries within the U.S. and abroad. This requirement is met by multiple applications performing export checks on customer demographic data and product purchases and delivery.
  - Each month the U.S. Export Regulations Office publishes a new version of its Denied Parties List."
- SOA Solution:
  - A unified standard interface was developed for all Systems using a SOA implementation



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

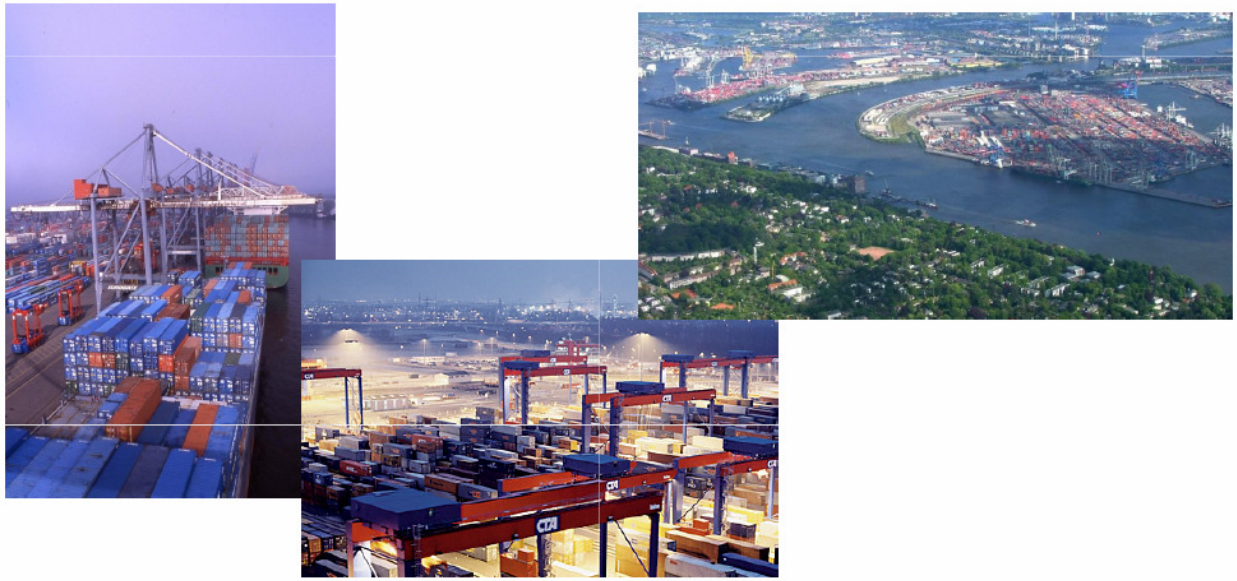
- Business Need: Reduce Administrative effort
  - “Coordinate 40 name and address systems to streamline administrative processes and ensure information integrity for users”

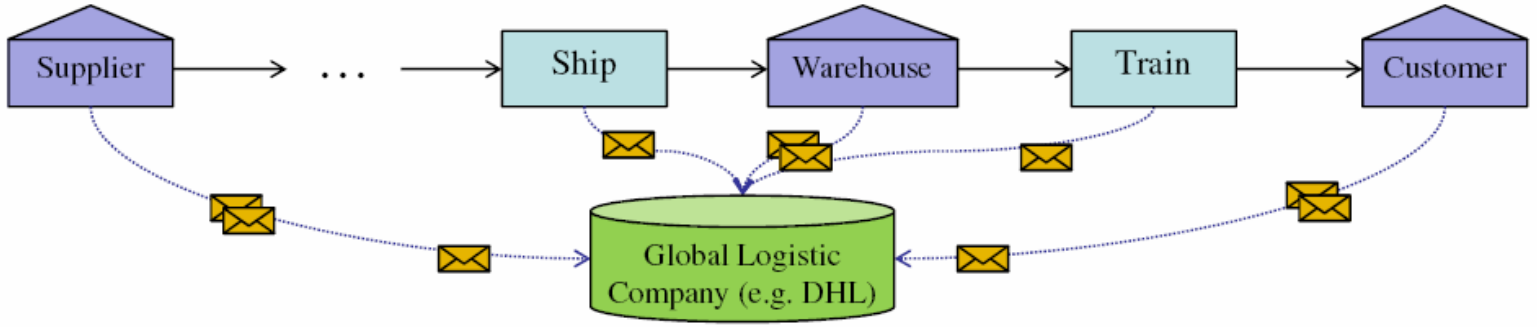
- **SOA Solution: Building reusable components with SOA**
  - The IBM SOA solution runs in the mainframe environment utilizing **IBM CICS Transaction Server**. This version strengthens the application development capabilities of CICS and extends CICS applications to an SOA. In a scenario in which information 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, Version 8**, 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

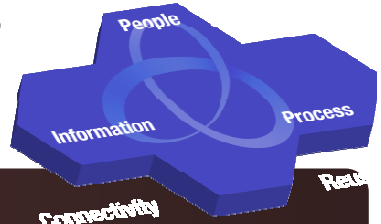
- Daily:
  - >20 Container Ships
  - >25.000 Containers
  - Container stored:
    - >100.000
  - Stay time: 4 days





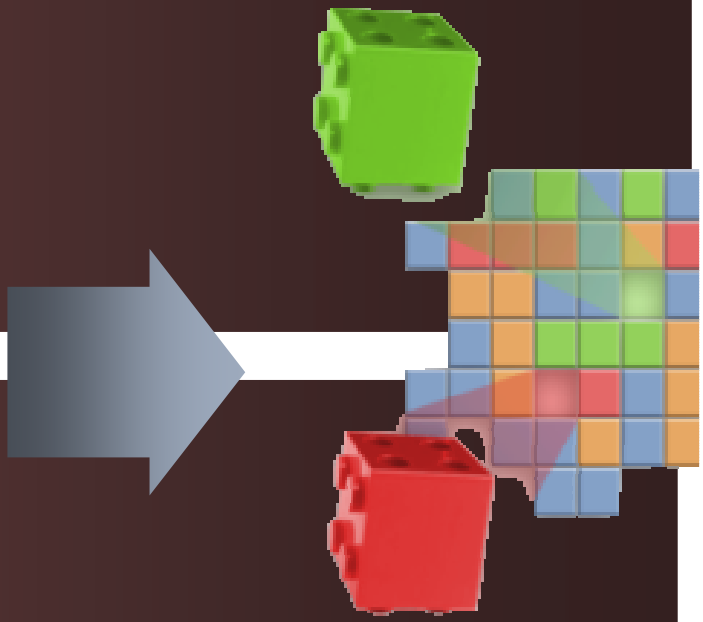
Creating & Reusing Services - Greater Value through SOA

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