IBM Software Group

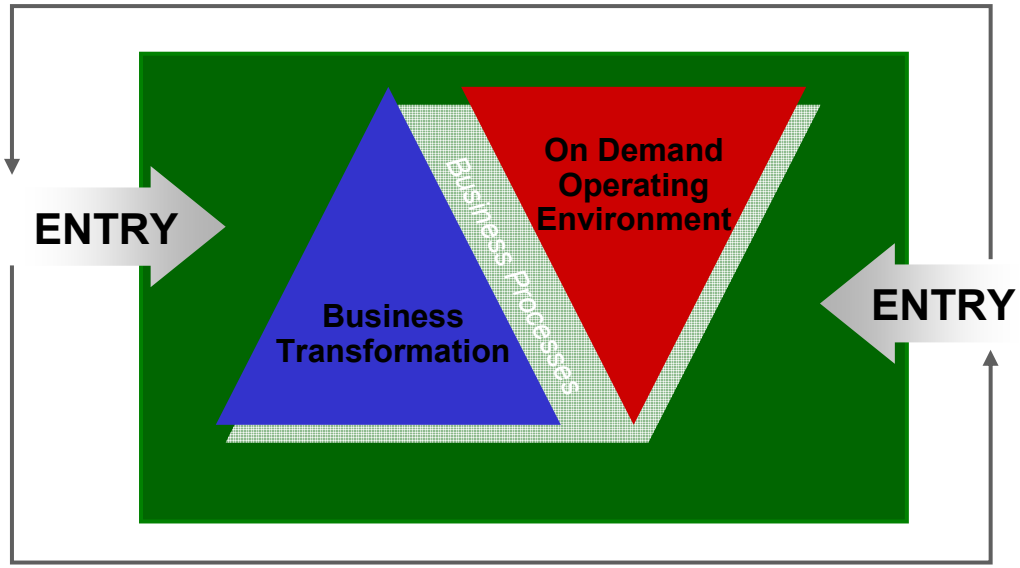
On Demand access to CICS and IMS Assets
or How to integrate existing 'legacy' applications into an On Demand infrastructure

e business on demand software

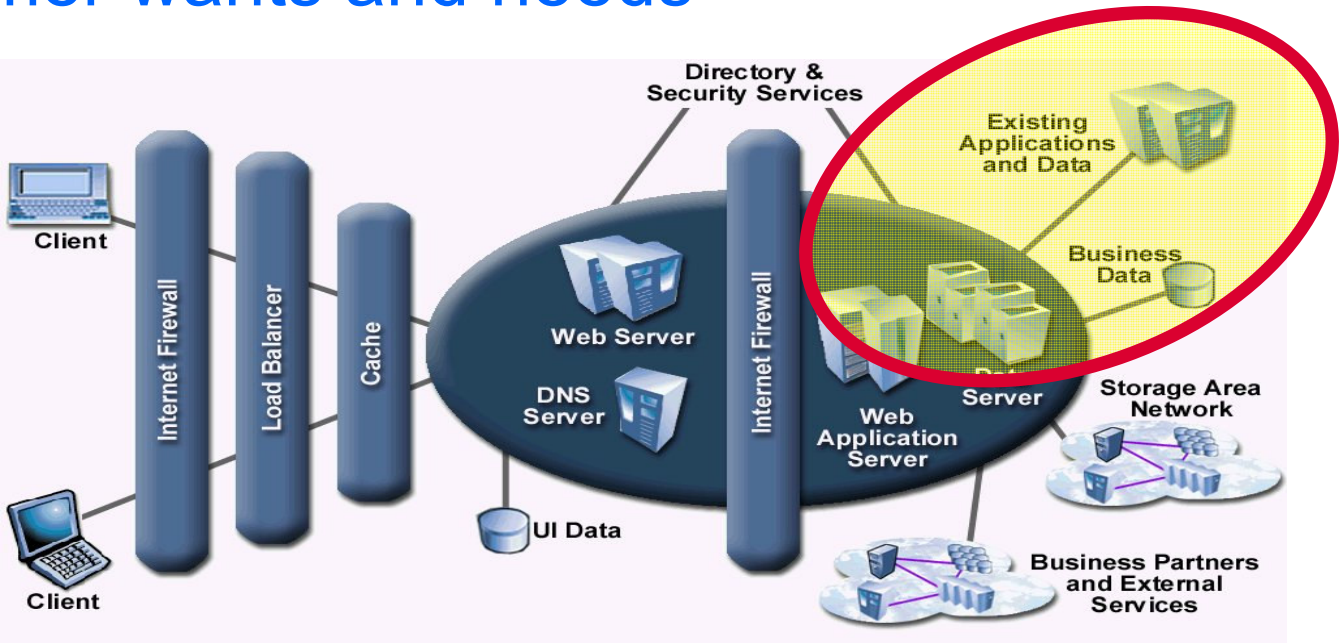
Hélène Lyon - IMS Consultant, Technical Sales SWG EMEA
helene.lyon@fr.ibm.com
Bertrand Tyl – CICS Consultant, Technical Sales SWG West Region
Bertrand.tyl@fr.ibm.com

Agenda

- **On Demand Operating Environment Introduction**
- **CICS Assets and Implementations**
- **IMS Assets and Implementations**
- **Summary**

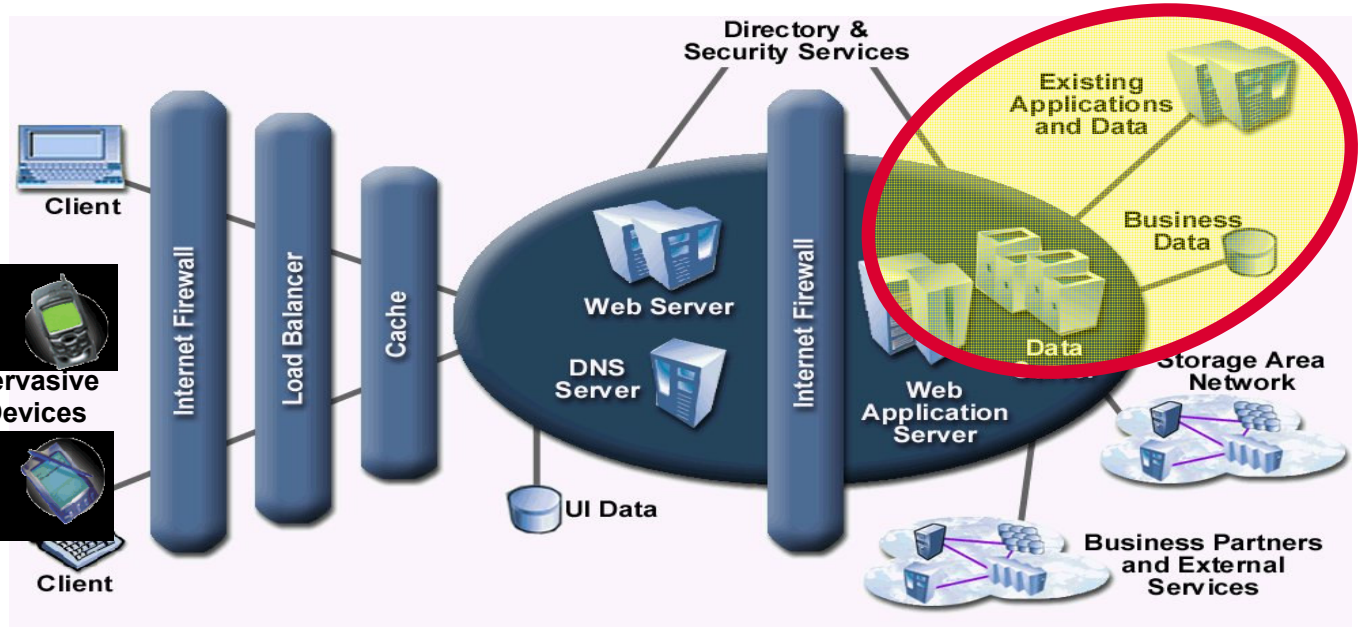


Customer wants and needs



- **Increased return on existing investments**
  - improved reuse of legacy assets
  - simplified access & better integration with Websphere Application Server
- **Lower total cost of ownership**
  - improved AD productivity
  - easier systems management
- **Solutions which are scalable, available, reliable and secure**

CICS – Fulfilling these Needs



Over 30 years and $1 Trillion invested in Applications ... IDC

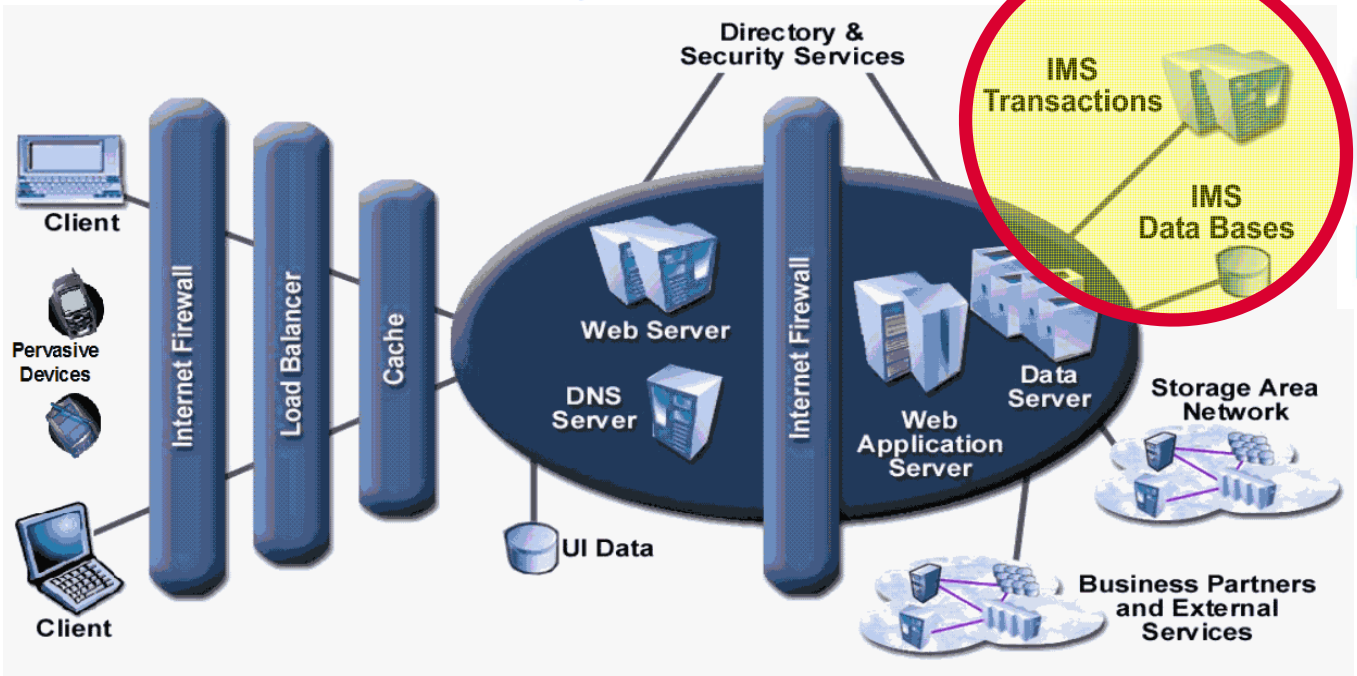
Over $1 trillion processed/day

Over 30 billion transactions/day

Most people use CICS

Combining the reliability and security of CICS software with the flexibility of e-business technology

IMS TM/DB – Fulfilling these Needs



IMS Today

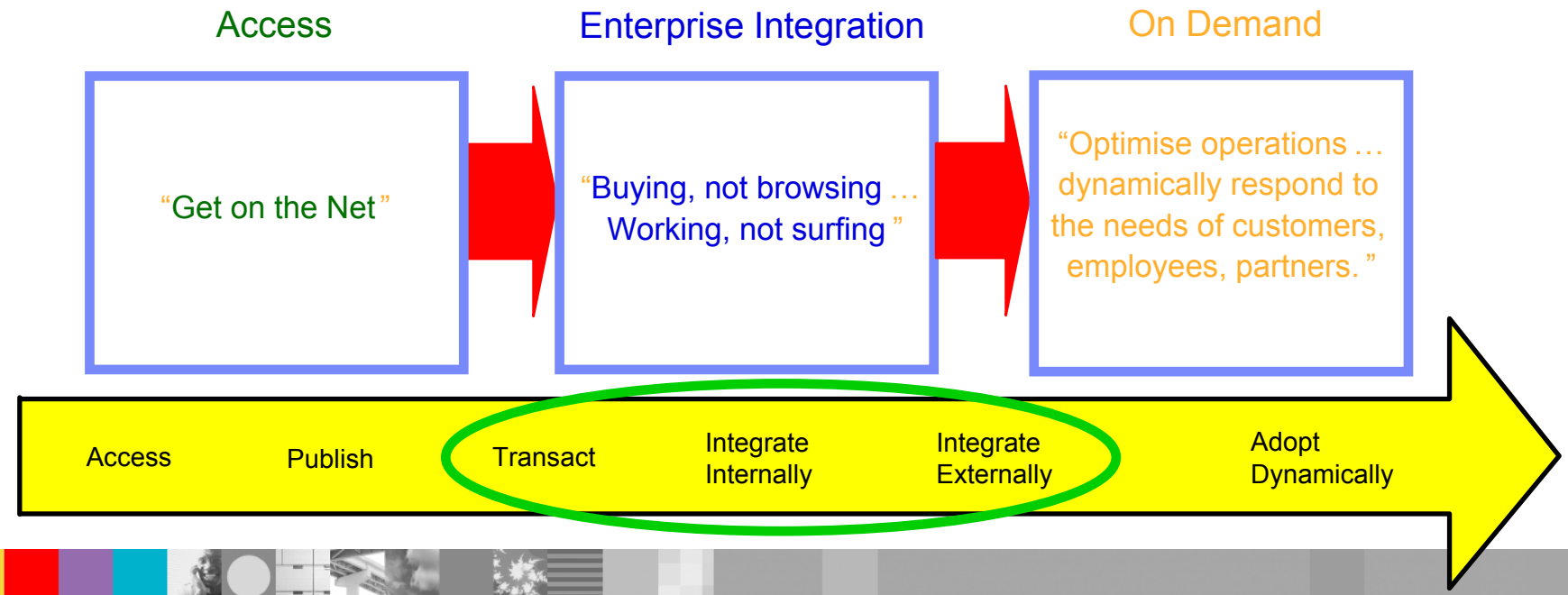
- Over 95% of Fortune 1000 Companies use IMS
- IMS Manages over 15 Million GBs of Production Data
- Over 50 Billion Transactions a Day run through IMS
- IMS Serves Close to 200 Million Users a Day
- Over 100 Million IMS Trans/Day Handled by One Customer
- 6000 Trans/sec across TCP/IP to single IMS

IMS Strategy

- Easier Systems Management (Common Service Layer, Dynamic Resource Definition, etc.)
- Availability and capacity (Shared Queues, IMSPlex, etc)
- Exploit new programmer skills (IMS Java)
- Enhanced connectivity (IMS Connect and Extensions)
- Improved development productivity provided by WebSphere Studio (Application Developer Integration Edition and Enterprise Developer)

Phases of On Demand Adoption

- **Provide static information to a browser**
- **Provide dynamic information to a browser**
  - with calls to backend systems (eg. CICS or IMS TM)
- **New Internet Applications**
- **Business to business "transactional" processing**
- **Publishing of Web Services**

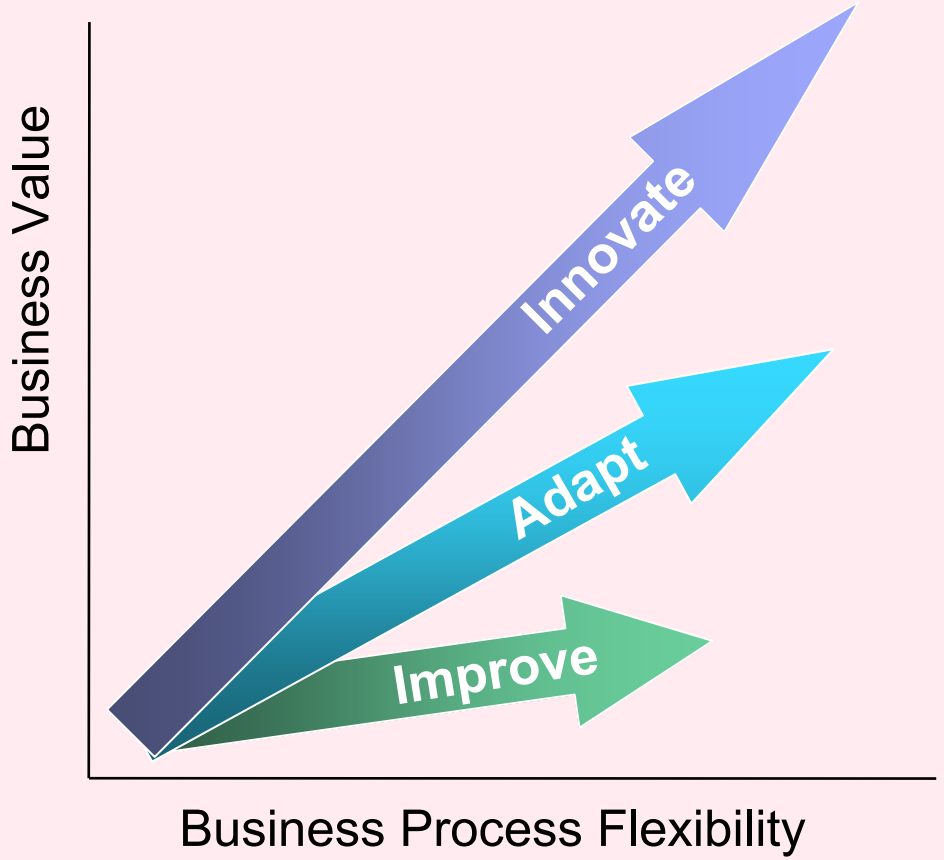


Three Styles of Transformation

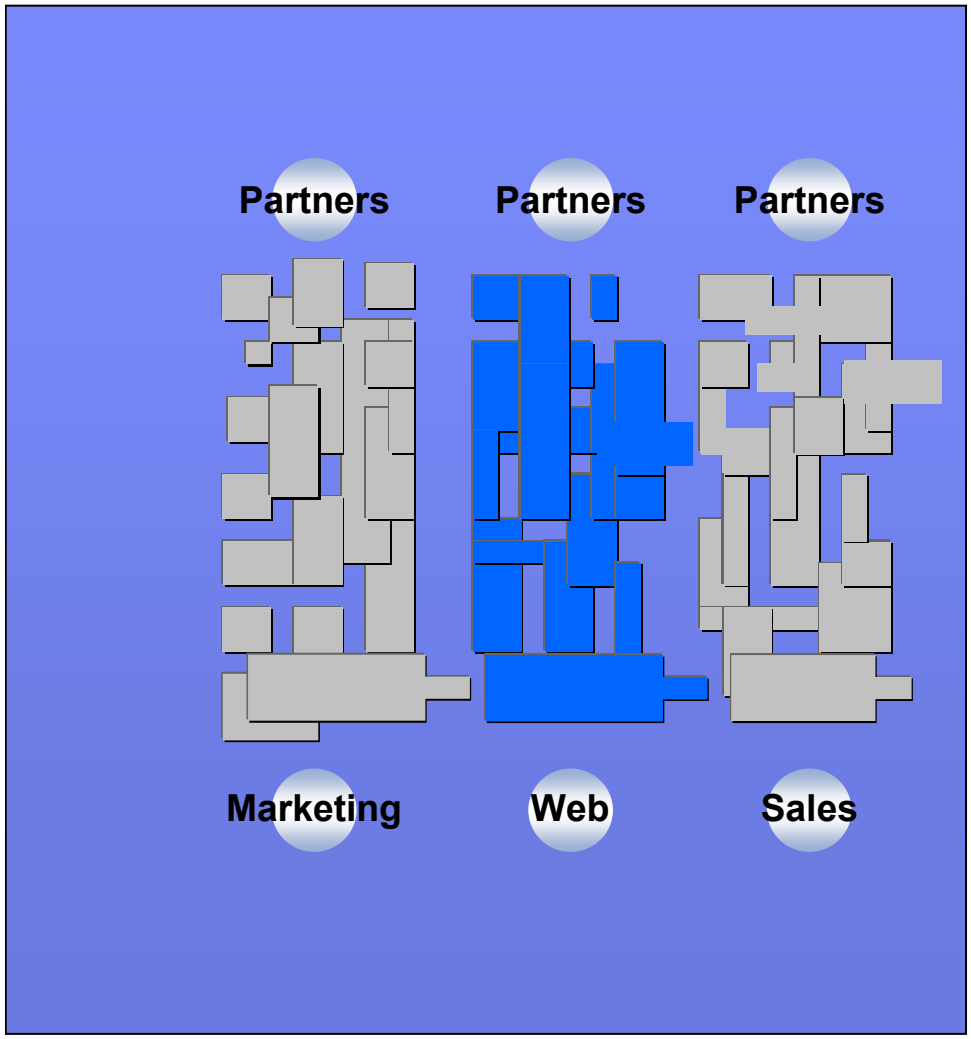
- **Improve**
  - re-face applications to enhance the user experience
- **Adapt**
  - re-use applications as part of a larger solution
- **Innovate**
  - re-engineer applications to reflect business processes

The basic business processes have not changed. So all three styles of transformation should exploit existing applications.

"To move towards an on-demand business, customers need to transform their technical infrastructure from unique, single purpose applications to shared resources"



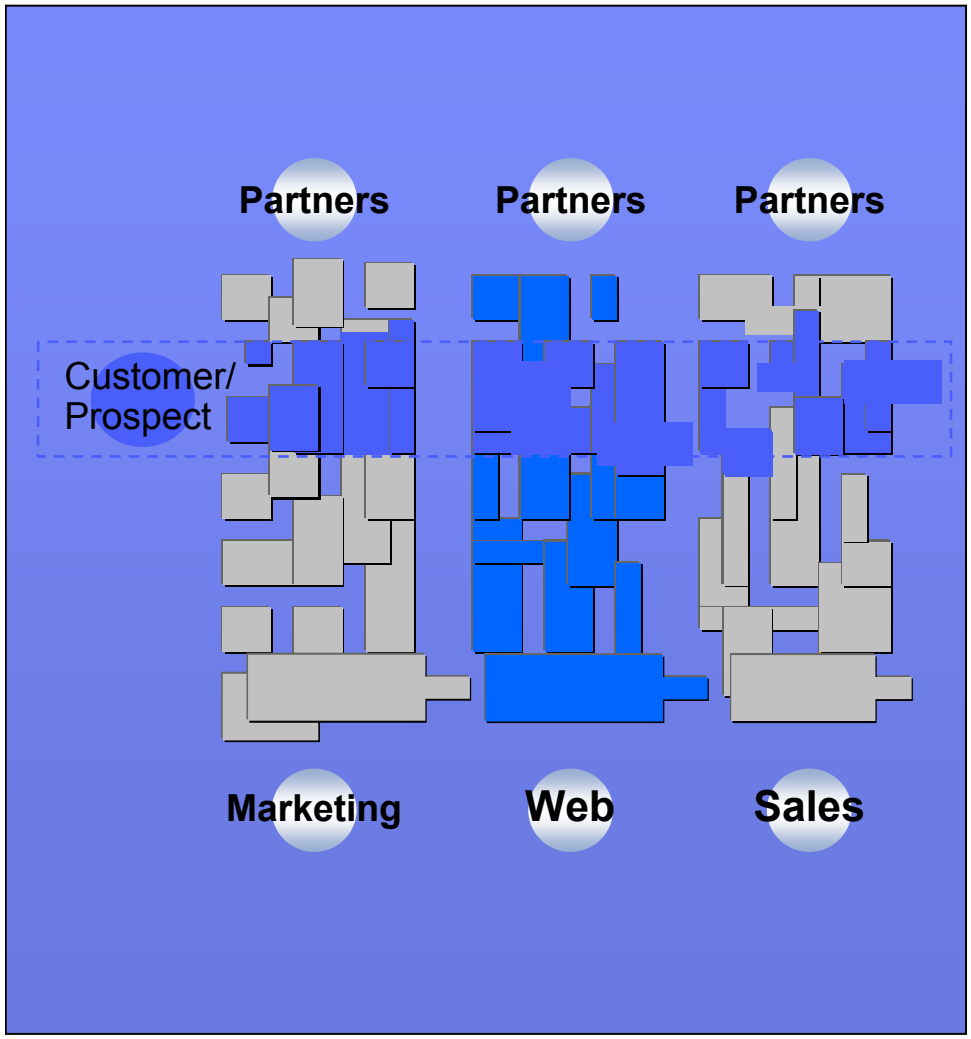
Your Reality: Functional Automation



Historical limitations:

- Monolithic applications can't be reused
- Ad hoc integration creates connections that are difficult to change/maintain
- Lack of standards limits ability to deliver meaningful interoperability

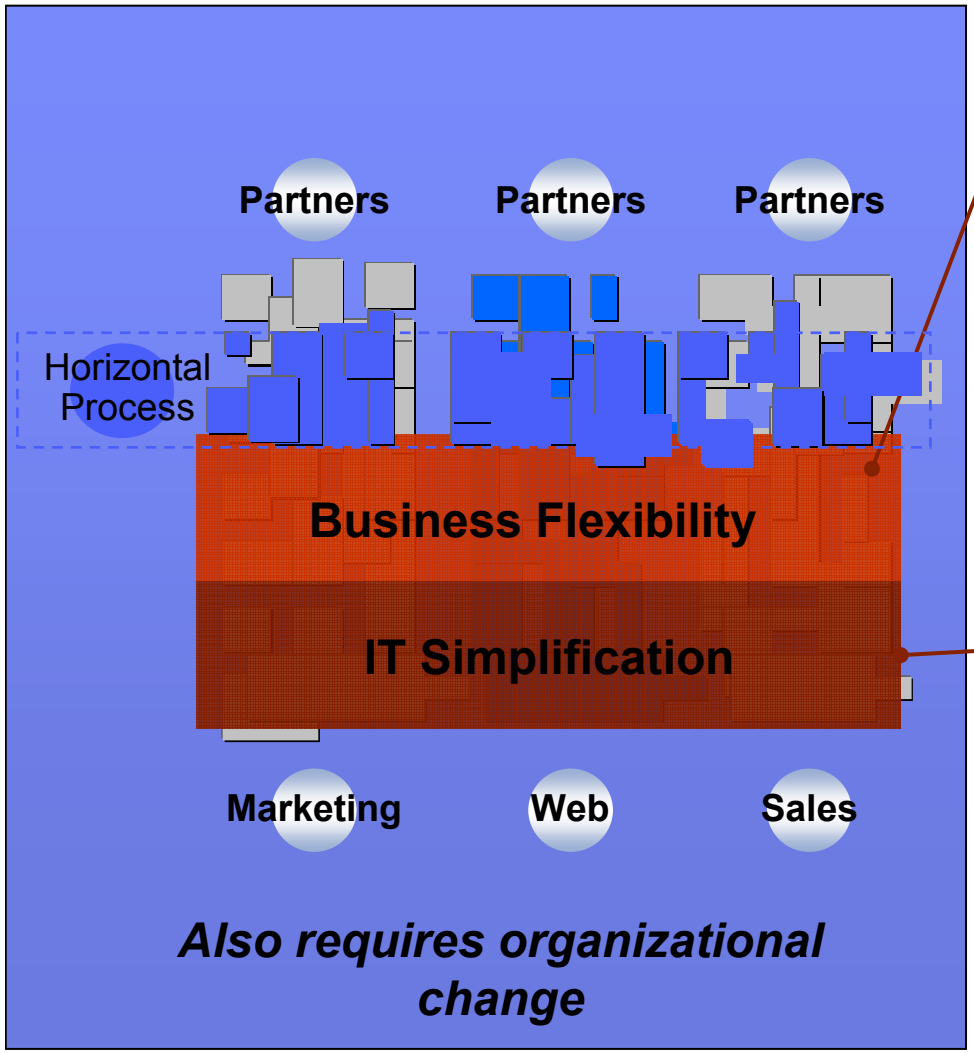
The Growing Reality: Horizontal Process Integration



Advances that make it possible:

- Standards for creating services and enabling them to communicate are agreed upon by major vendors
- Infrastructure that supports self-defined, loosely coupled services has emerged
- Tools to incorporate existing assets are available
- Automation and virtualization of systems resources readily available

On Demand Operating Environment and Legacy



On Demand Operating Environment

Integration

Business flexibility through integration of people, processes and information within and beyond the enterprise

Adapters/connectors for legacy

QoS considerations (performance, transactions, security …)

Services Oriented Architecture

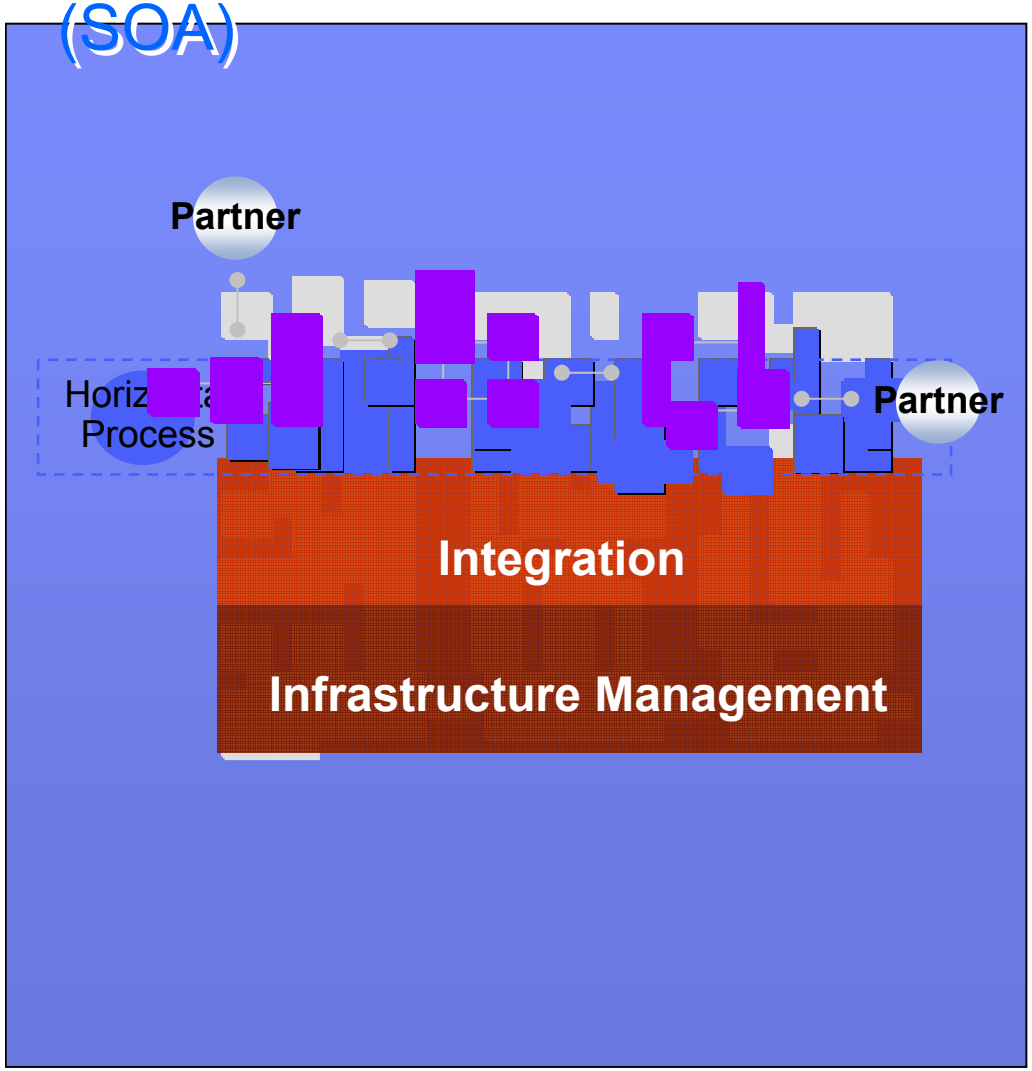
Infrastructure Management

IT simplification through automation and virtualization of resources …

Automation (availability, security, optimization, provisioning …)

Consolidated view of legacy and new

What makes this possible? A service-oriented architecture (SOA)



What is SOA?

SOA enables flexible connectivity of applications or resources by:

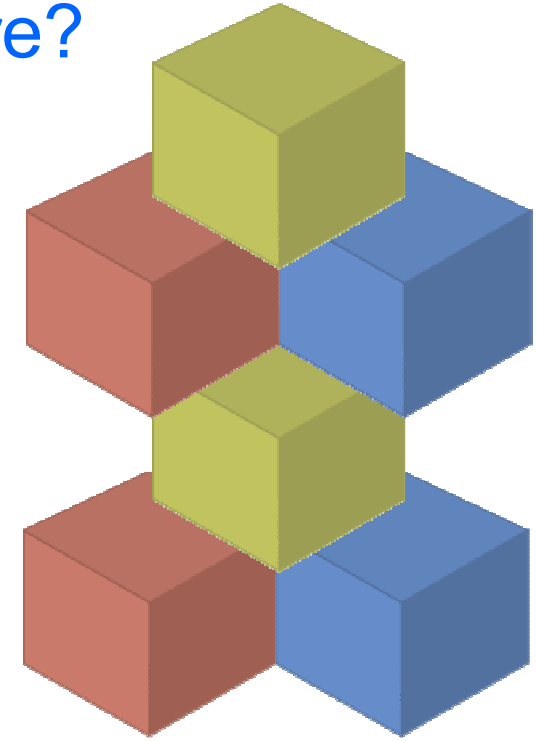
- Representing every application or resource as a service with a standard interface
- Enabling them to exchange structured information

Why do you care?

Provides the flexibility to treat business processes and the underlying infrastructure as defined components that can be mixed and matched at will

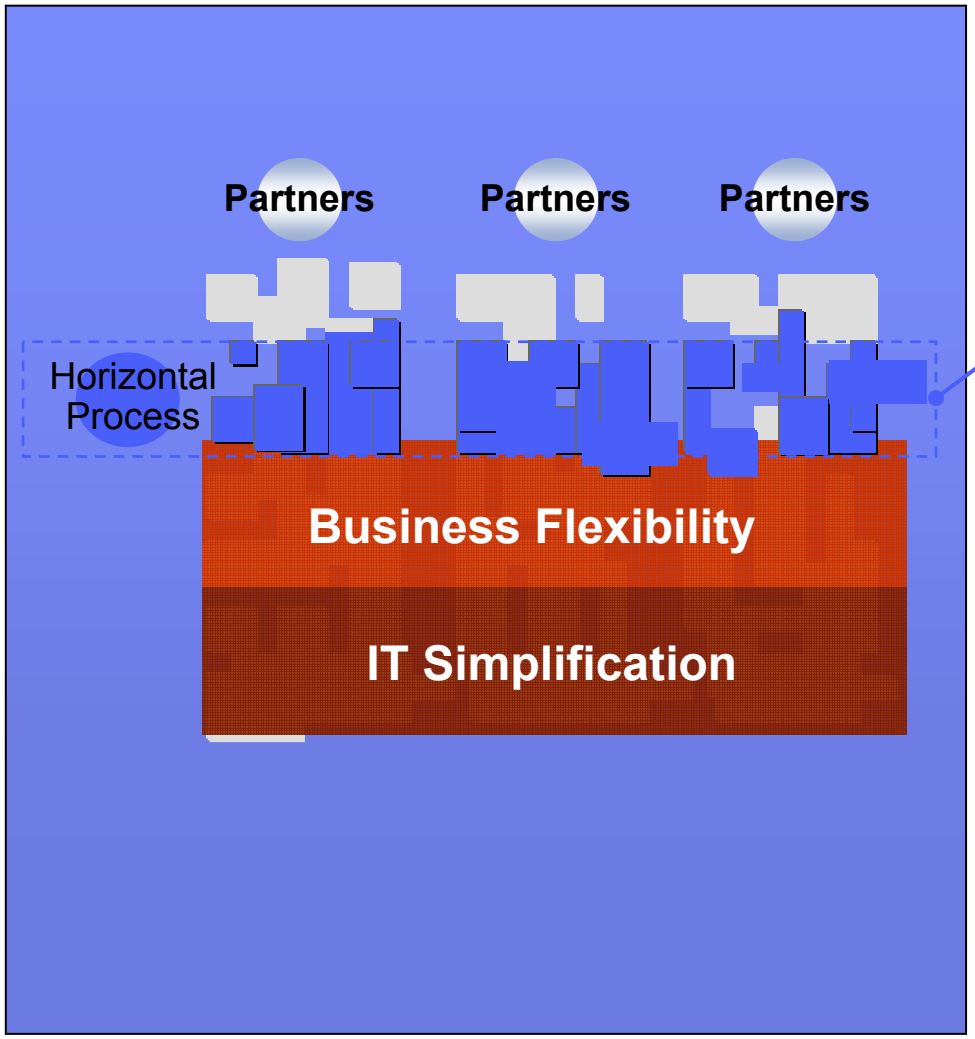
What is a Service Oriented Architecture?

- **An approach for building distributed systems that deliver application functionality as services to either end-user applications or other services**
  - An architecture that leverages open standards to represent software assets as services.
  - Provides a standard way of representing and interacting with software assets
  - Individual software assets become building blocks that can be reused in developing other applications
  - Shifts focus to application assembly rather than implementation details
  - Used externally to integrate with applications outside of the Enterprise



"A system architecture in which application functions are built as components (or services) that are loosely-coupled and well-defined to support interoperability and to improve flexibility and re-use".

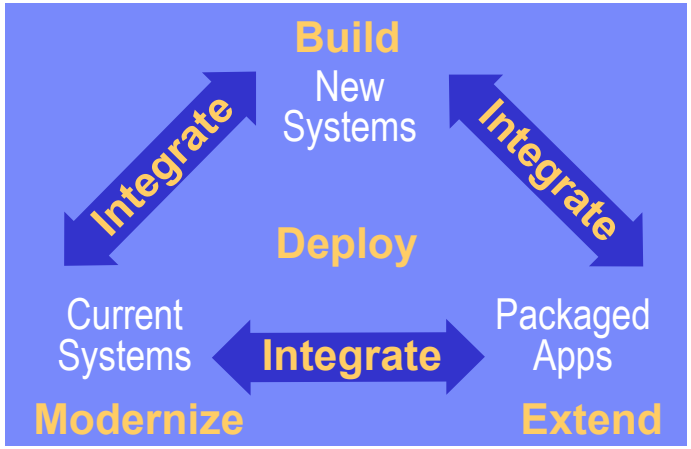
Software Development is Key



Software Development

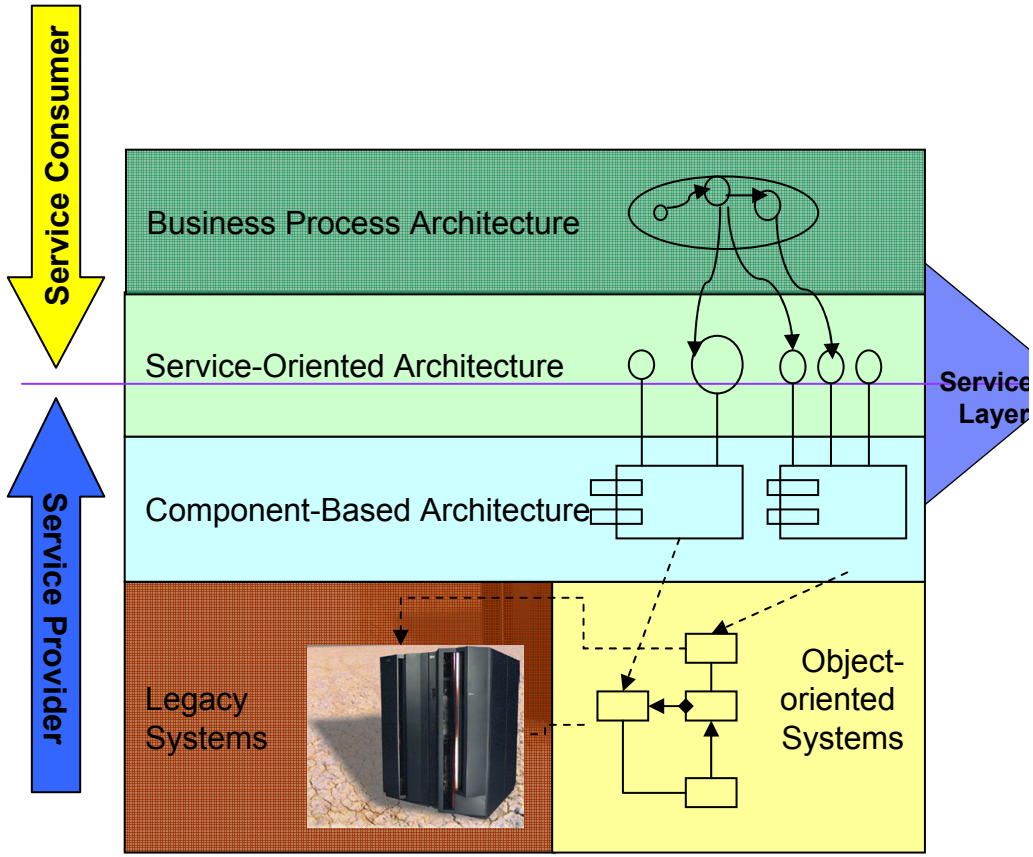
Integrating & automating horizontal business processes:

- Unique to each business
- Competitive advantage



Why legacy services ?

- **Leverage existing assets**
  - layer of abstraction that wraps existing assets as services that provide business functions
- **Easier to integrate and manage complexity**
  - integration point is the service specification and not the implementation
- **More responsive and faster time-to-market**
  - ability to compose new services out of existing ones
- **Reduce cost and increase reuse**
  - loosely coupled core business services can be more easily used and combined based on business needs
- **Be ready for what lies ahead**
  - better flexibility and responsiveness

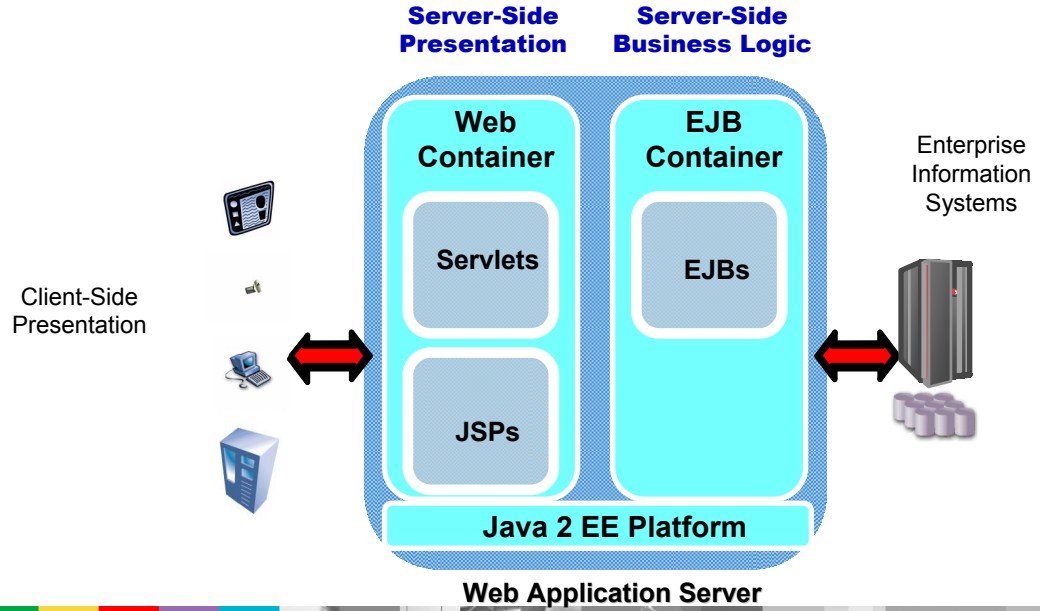


The WebSphere Software Platform

- **Web infrastructure software that helps companies at each stage of On Demand development**
  - from startup,
  - to integrating and exploiting business processes,
  - to handling high volume Web transactions.

WebSphere® Application Server

A next-generation application server that simplifies build-to-integrate tasks, accelerates application development, and enables dynamic application flexibility.

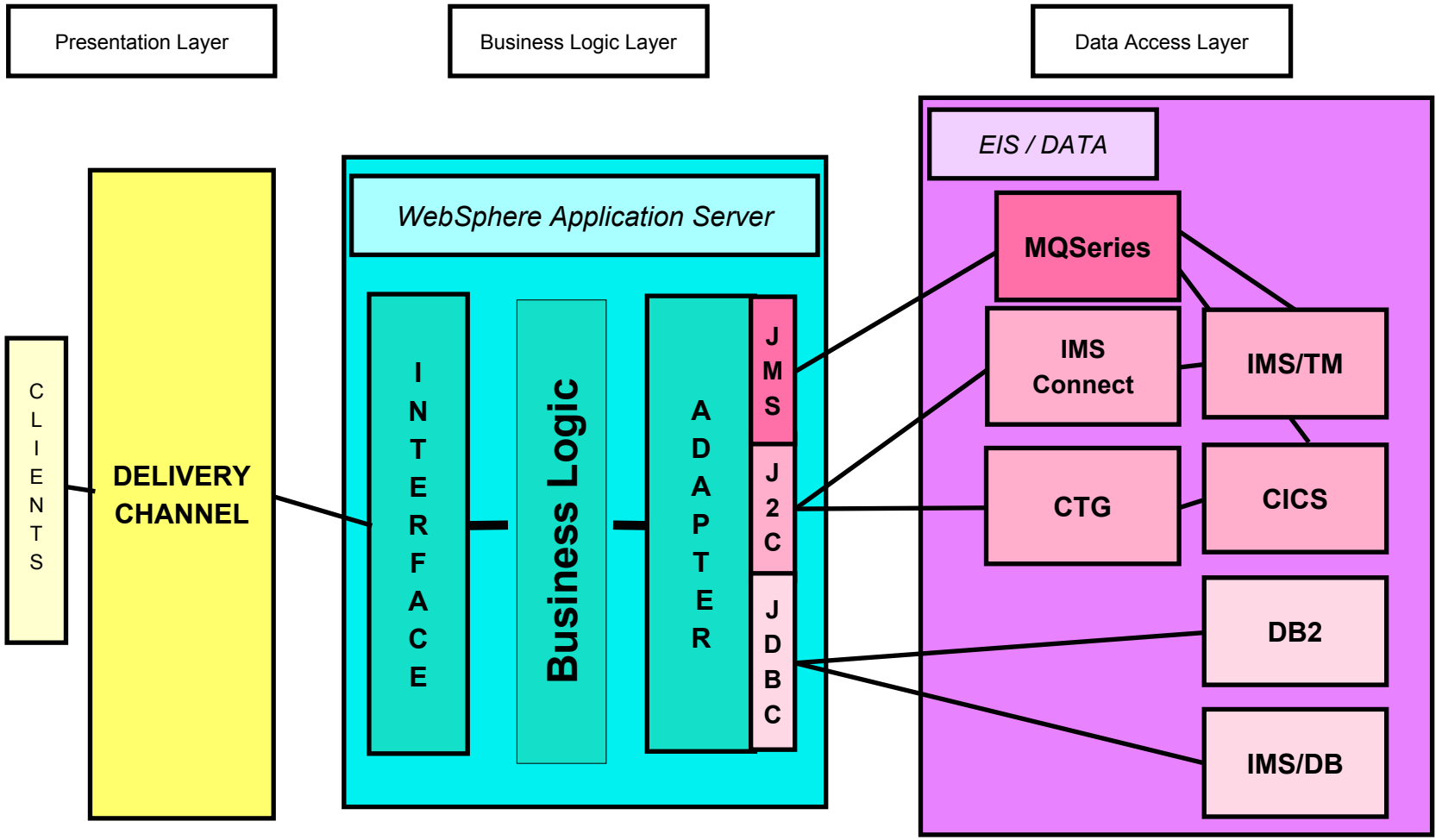


Application developers focus on business logic (components)

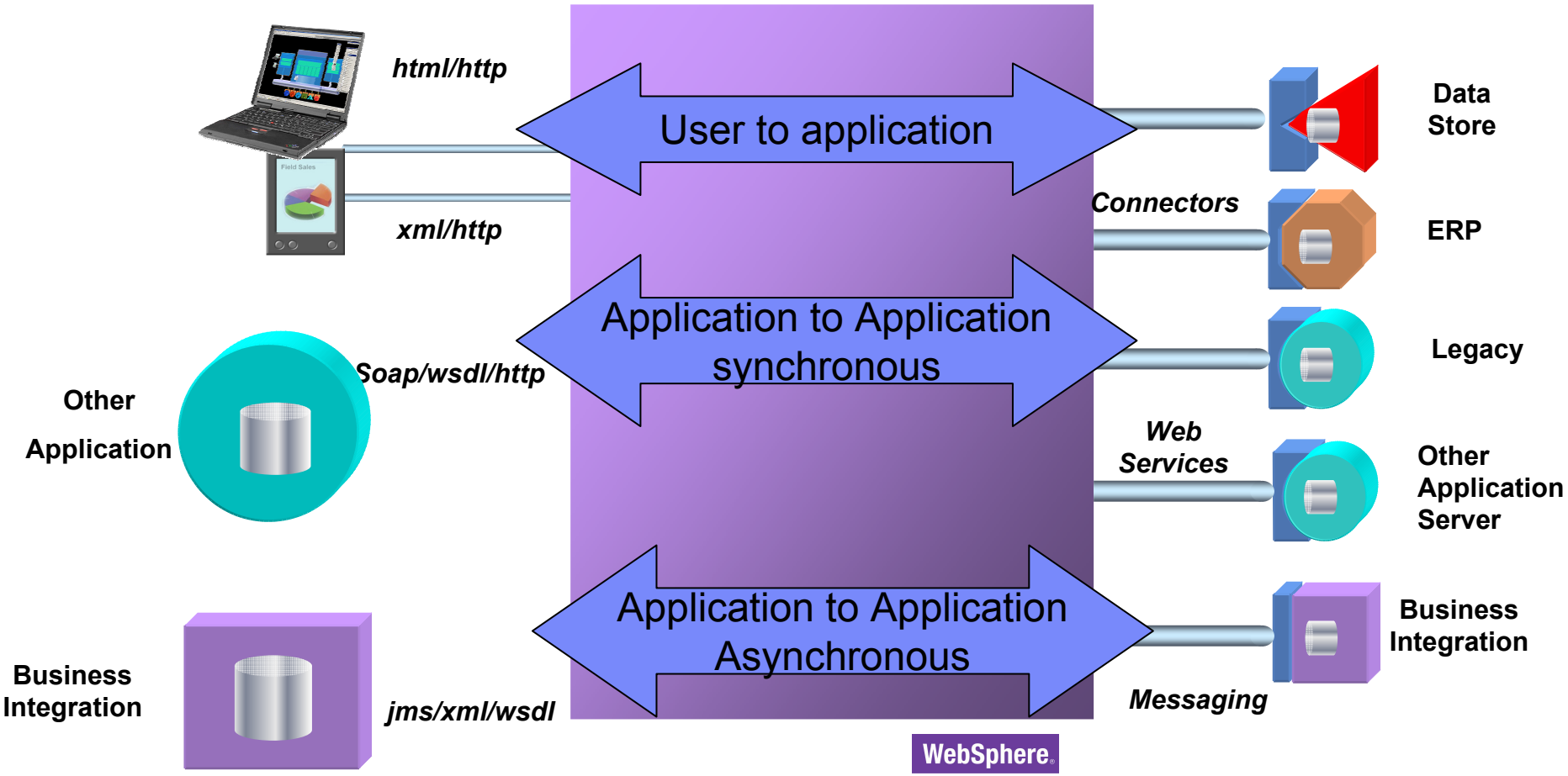
Containers and connectors conceal complexity and promote portability

Components inherit qualities of service of the underlying platform

J2EE - 3-tiered Distributed Computing Architecture



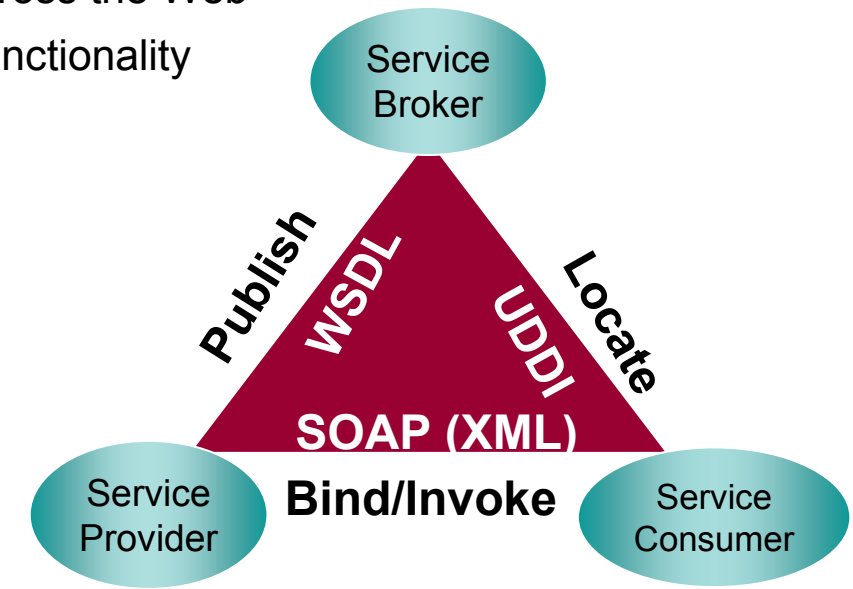
WebSphere - Foundation for Integration



Enterprise Web Services

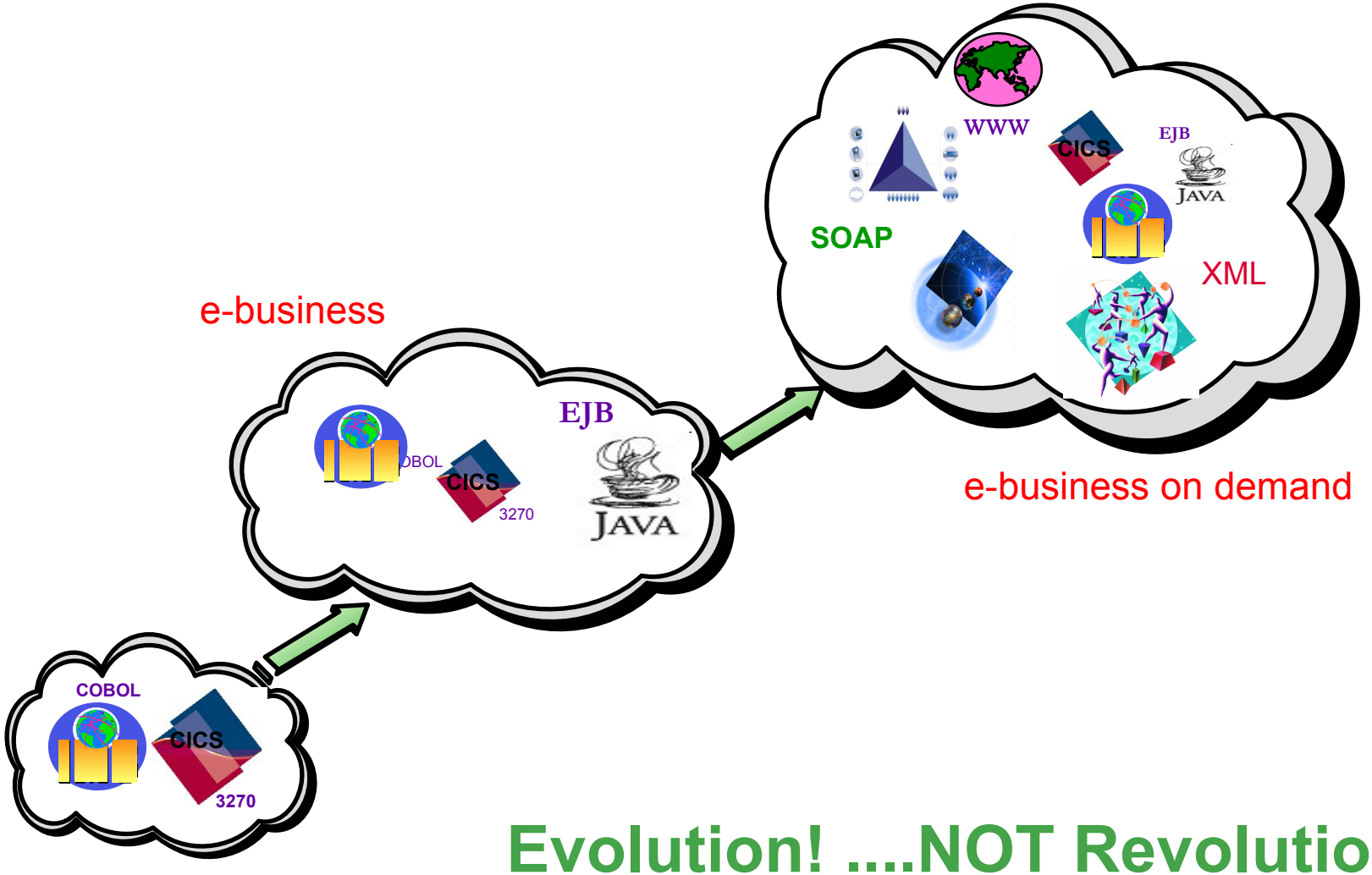
The building blocks of the future

- **Service model used by WSADIE and WSED**
- **Self-contained, self-describing modular applications**
  - Can be published, located, and invoked across the Web
  - Are reusable building blocks of business functionality
- **Use simple, standard interfaces for**
  - Publishing and discovering services
  - Communicating between service provider and service consumer in a platform-independent way (XML and SOAP)



- **Web Service is an extension of the EJB enablement**

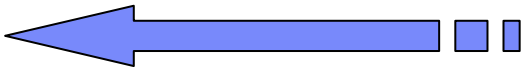
CICS and IMS On Demand Strategy



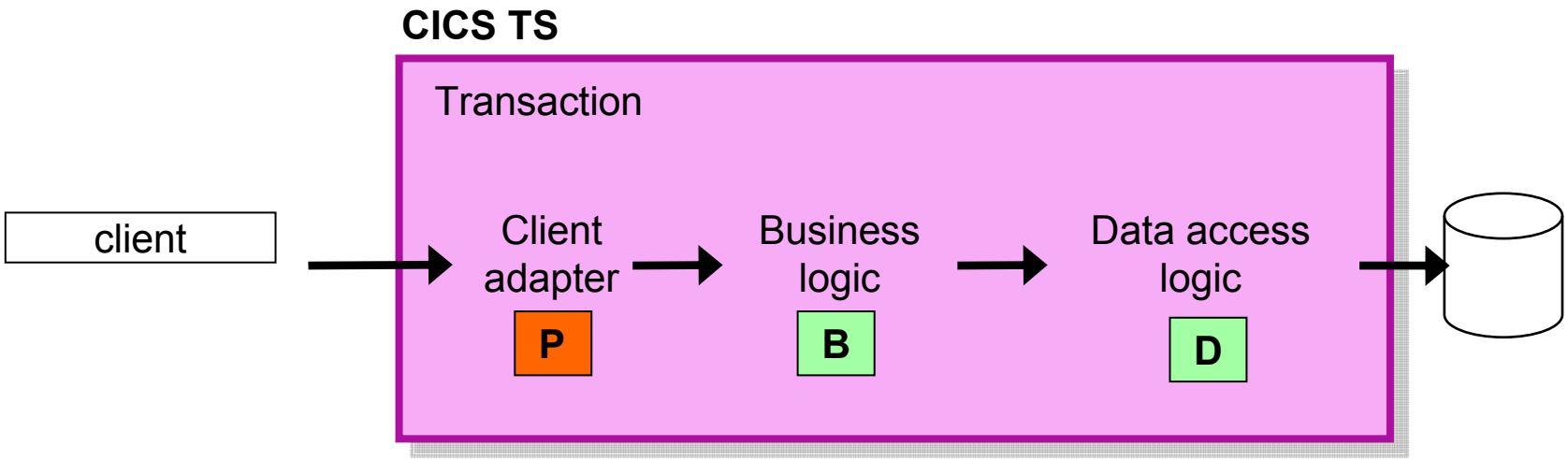
Evolution! ....NOT Revolution

Agenda

- **e-business Introduction**
- **CICS Assets and Implementations**
- **IMS Assets and Implementations**
- **Summary**



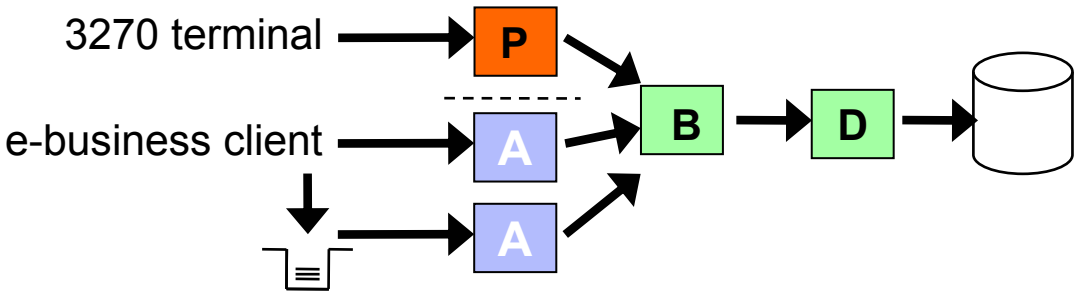
What assets in CICS can be transformed?

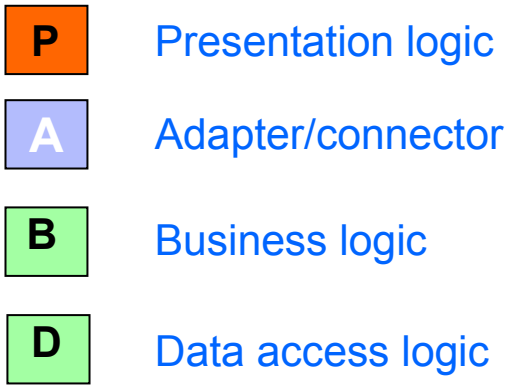


- **Best practice in CICS application design is to separate key elements of the application, in particular:**
  - Client adapter or presentation logic
  - Business logic
  - Data access logic

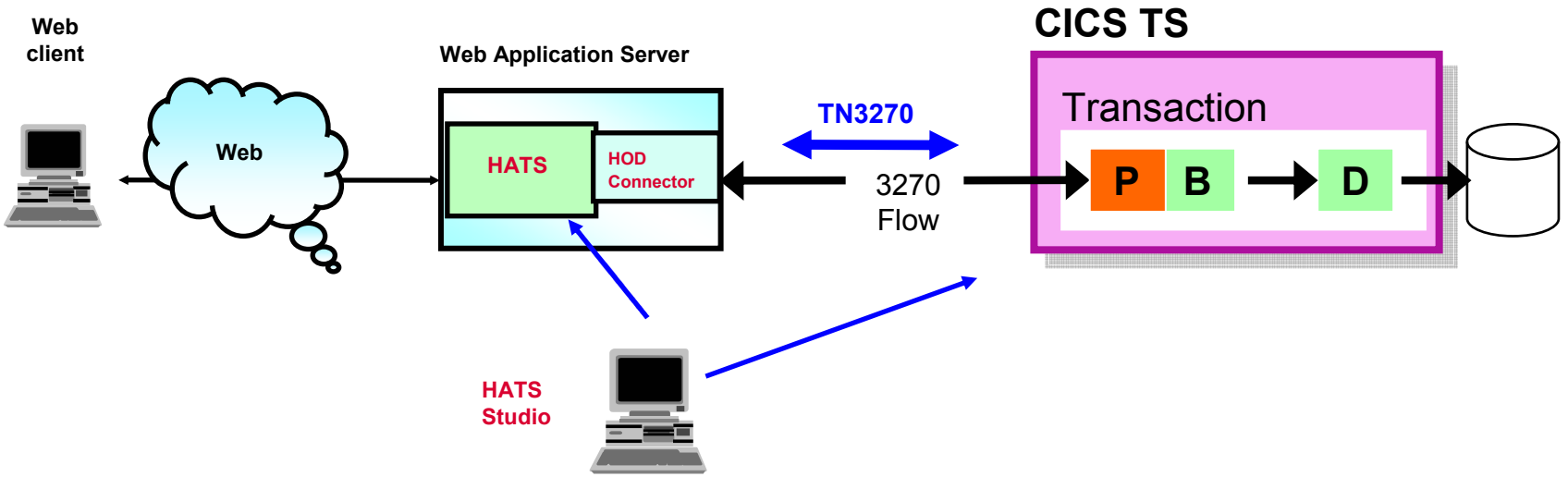
Access to CICS programs – Adapter Style

- **Typical e-business clients**
  - Web browser
  - Java servlet or EJB
  - Web Services SOAP client
  - C# client in Microsoft .NET
  - WebSphere MQ client
- **Adapter/connector can be**
  - External (e.g JavaBean using CICS Transaction Gateway classes)
  - Internal (e.g CICS XML-aware program)
  - Written or generated by tools





Web-to-Host using HATS



- **Host Access Transformation Server (HATS)**
  - Rules-based Web-to-Host transformation engine which dynamically converts 3270 screens in HTML pages
  - Can generate Web services or other Java objects from host transactions
  - Server based on WebSphere and 3270 HOD Connector
  - HAST Studio based on WebSphere Studio (WSAD)

*http://www.**ibm.com/software/webservers/hats***

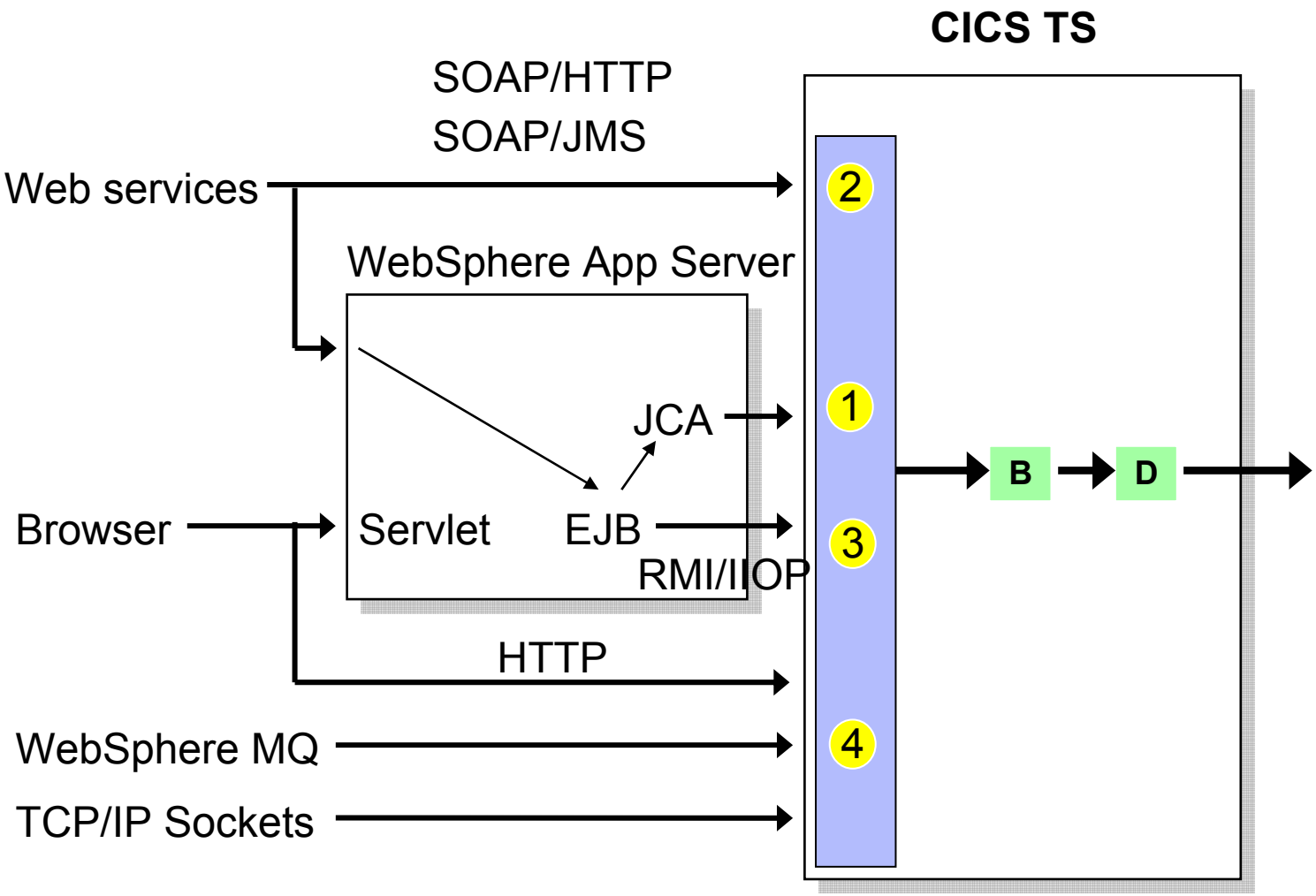
Which architecture should I use to connect to CICS?

- **Standard architectures provide a choice of options and support in CICS and tools**
  1. JCA (J2EE Connector Architecture)
  2. SOAP (Simple Object Access Protocol)
  3. Java RMI (Remote Method Invocation)
  4. JMS (Java Messaging Service)

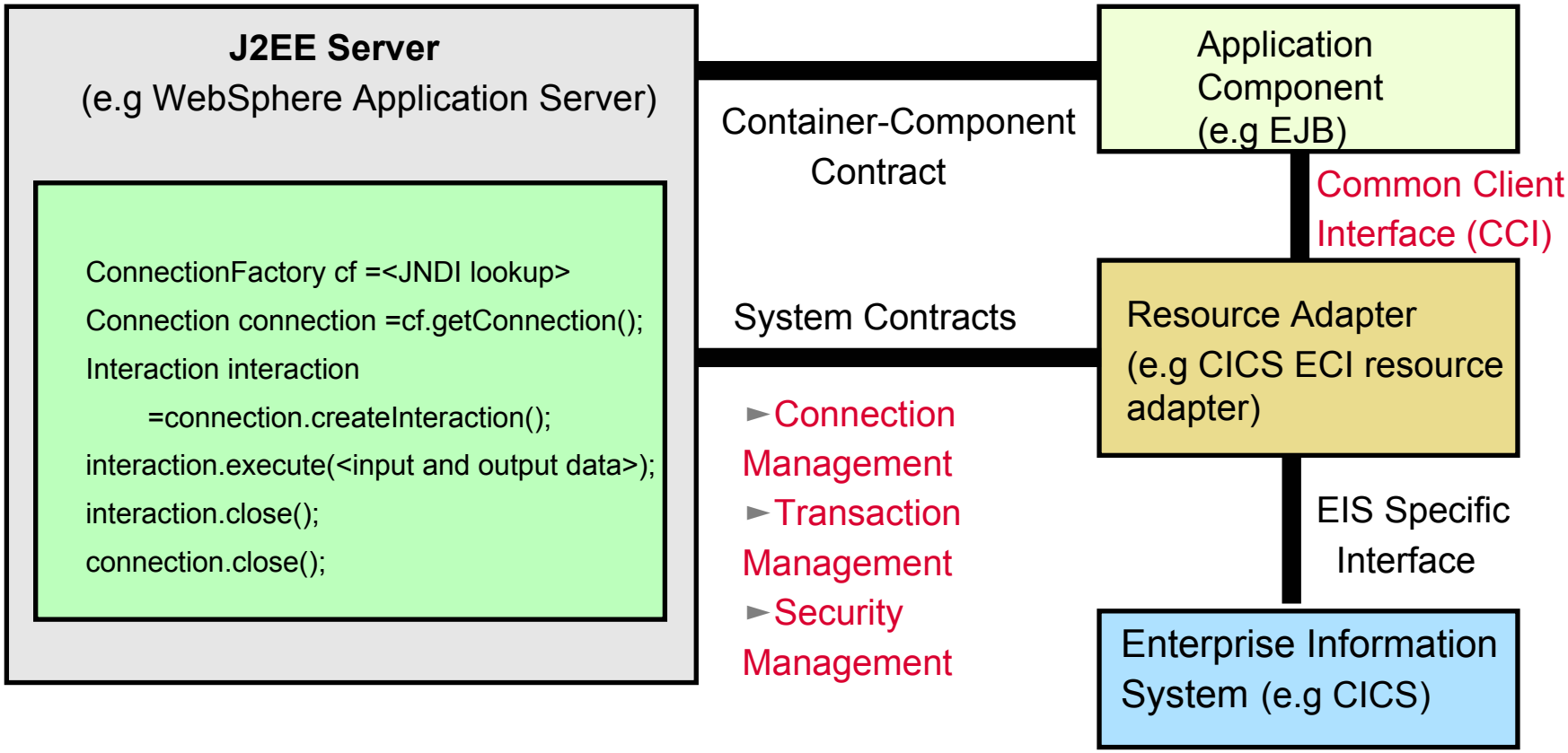
- **Lots of factors including …**
  - Security
  - Transactionality
  - Performance
  - Architectural limits
  - Synchronous or asynchronous invocation

- **See redbook ‘Architecting e-business Access to CICS’**

Which architecture should I use to connect to CICS? …

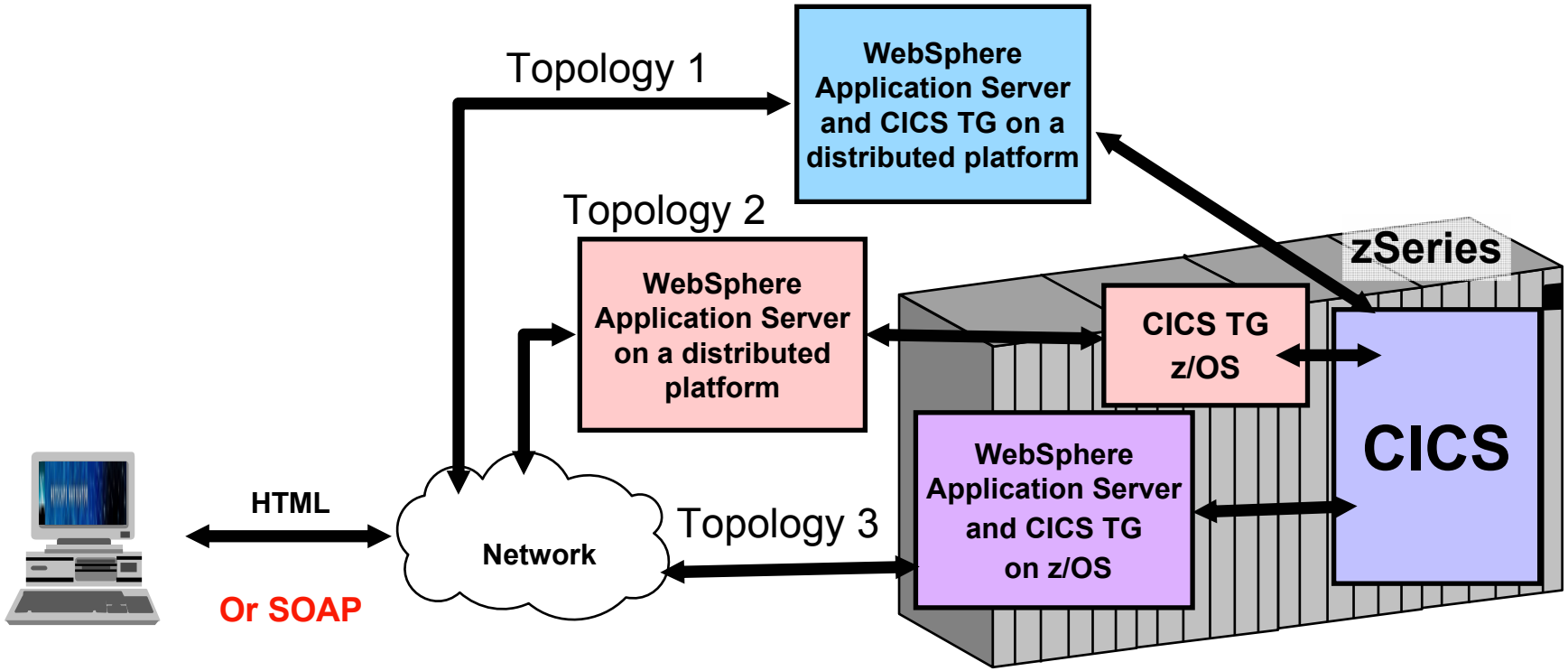


J2EE Connector Architecture (JCA) for CICS



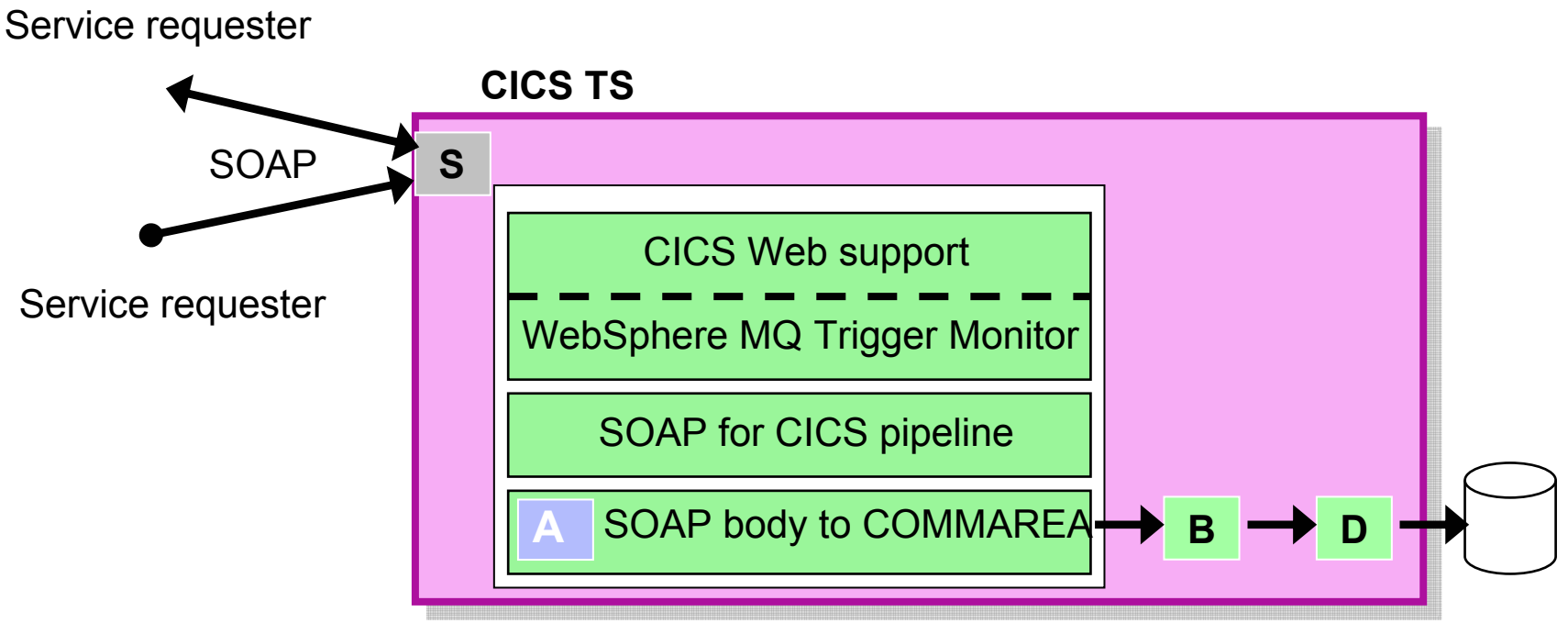
- **Standard for connecting from J2EE to EIS**
- **Resource adaptor is provided by EIS**
- **CICS resource adapters are provided by CICS Transaction Gateway**

JCA platform and topology options



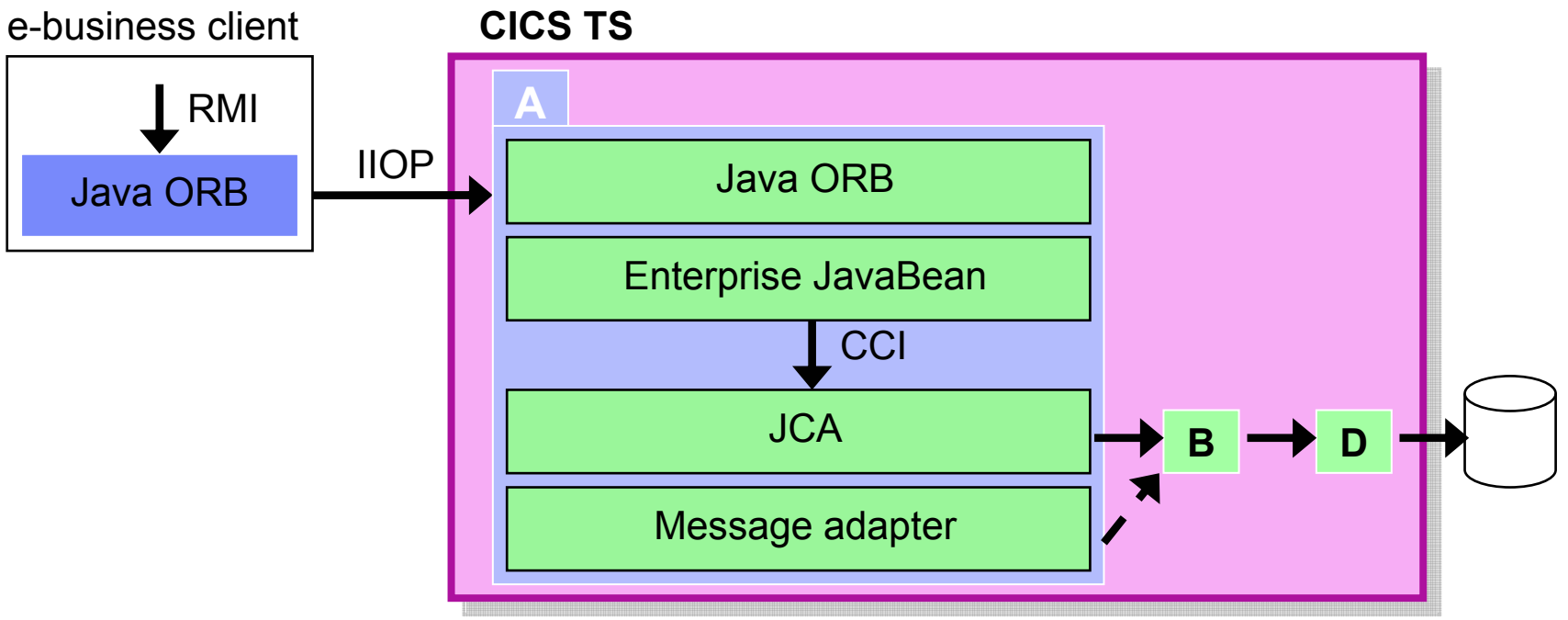
- **Qualities of service vary according to topology**
- **See white paper** '*Integrating WebSphere Application Server and CICS using the JCA*'

SOAP – Web Service deployed in CICS

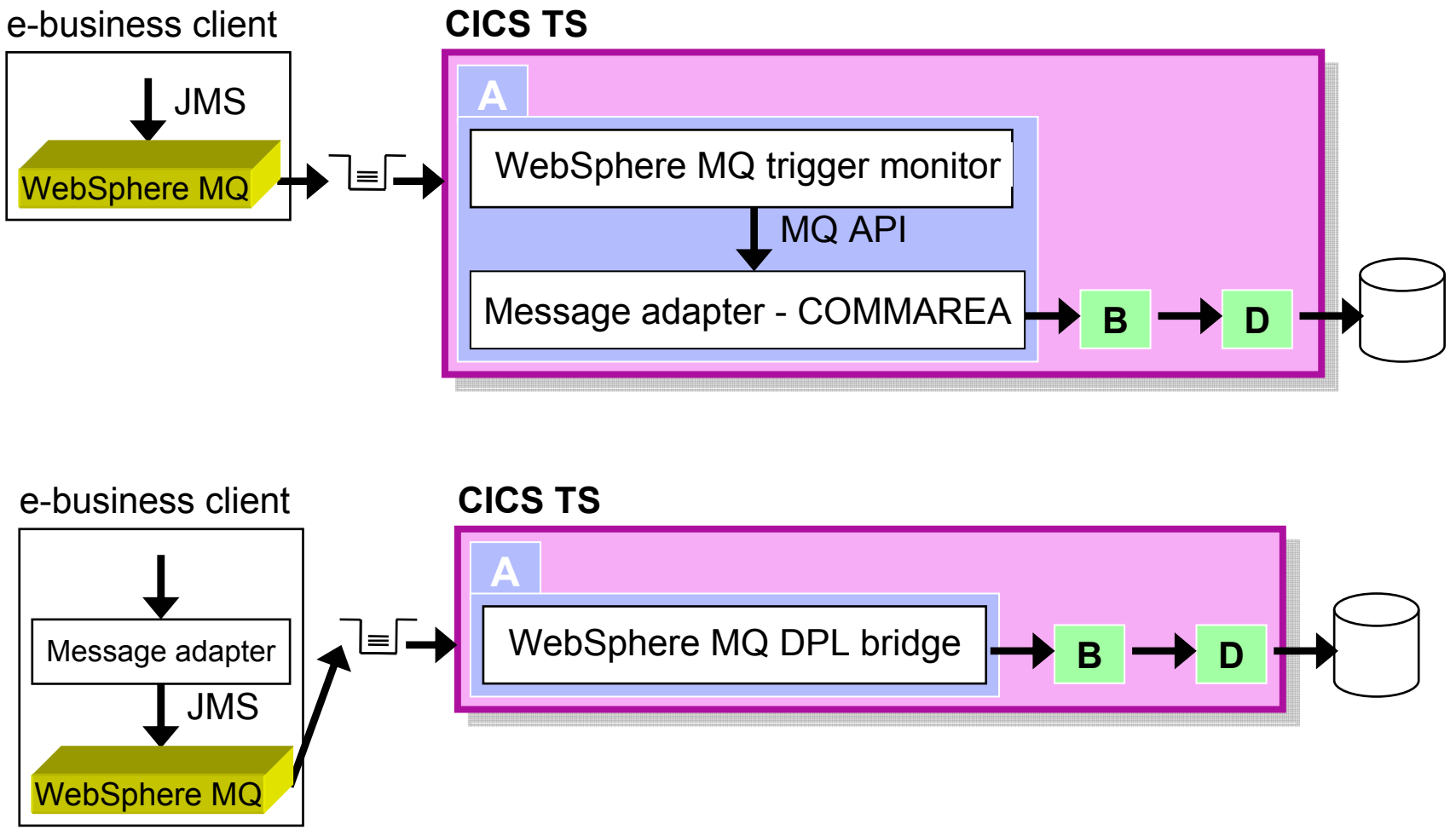


- **Service can be accessed using SOAP/HTTP or SOAP/JMS**
- **SOAP for CICS pipeline is a Business Transaction Services (BTS) process**
- **Adapter is COBOL program**
  - converts SOAP body to COMMAREA and vice-versa

CICS Java RMI Support



JMS access to CICS Transactions

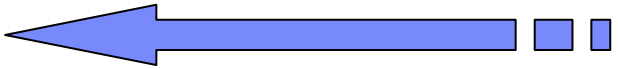


CICS Strategic Options Table

Standard architecture	Middleware	Interface
0. 3270 Revamping	HATS or Link3270bridge	3270 flow
1. SOAP	a - CICS SOAP support b - WAS CTG	XML in a CONTAINER COMMAREA COMMAREA
2. JCA	WAS CTG	COMMAREA
3. Java RMI	CICS TS V2	Enterprise JavaBean (session bean)
4. JMS	WMQ	WMQ API or COMMAREA

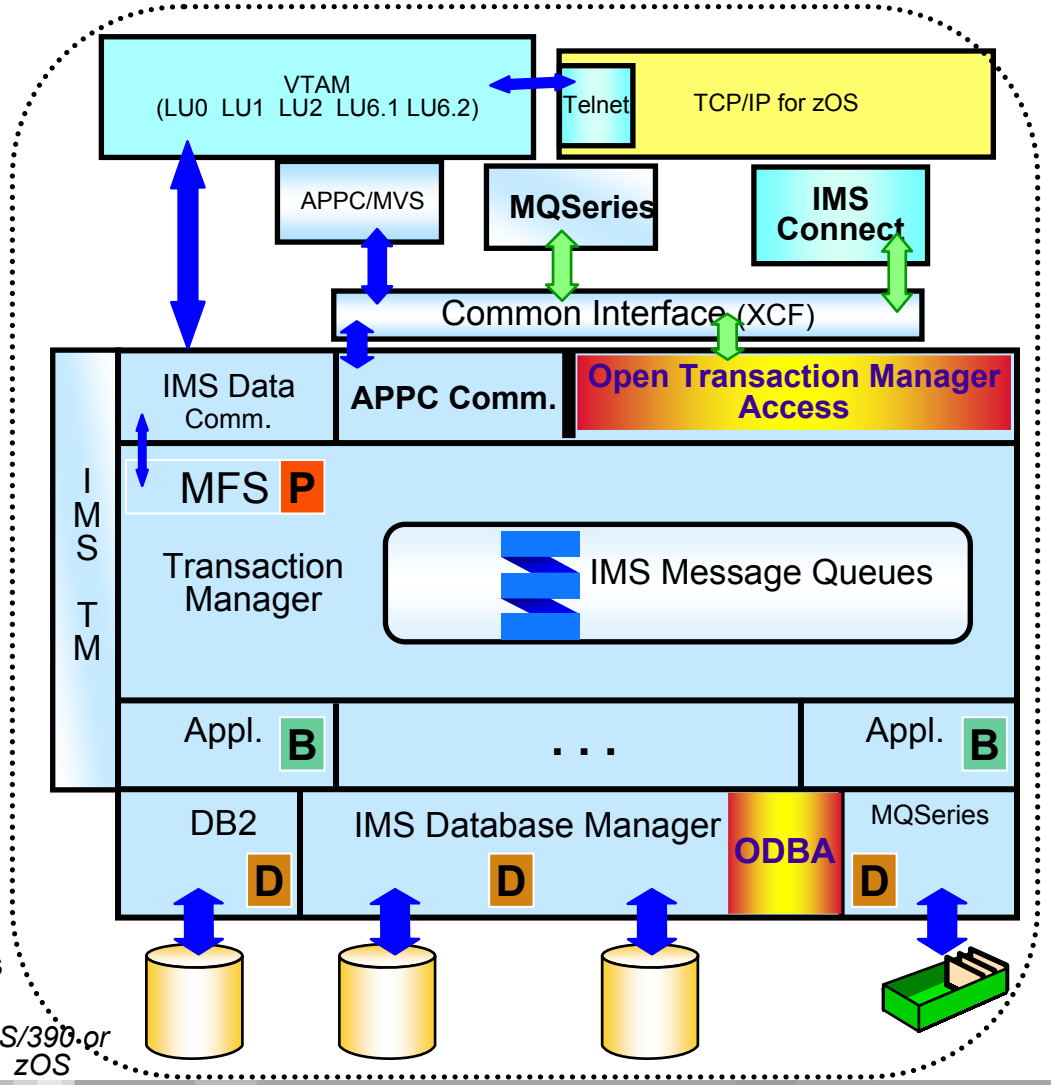
Agenda

- **e-business Introduction**
- **CICS Assets and Implementations**
- **IMS Assets and Implementations**
- **Summary**



What assets in IMS can be accessed?

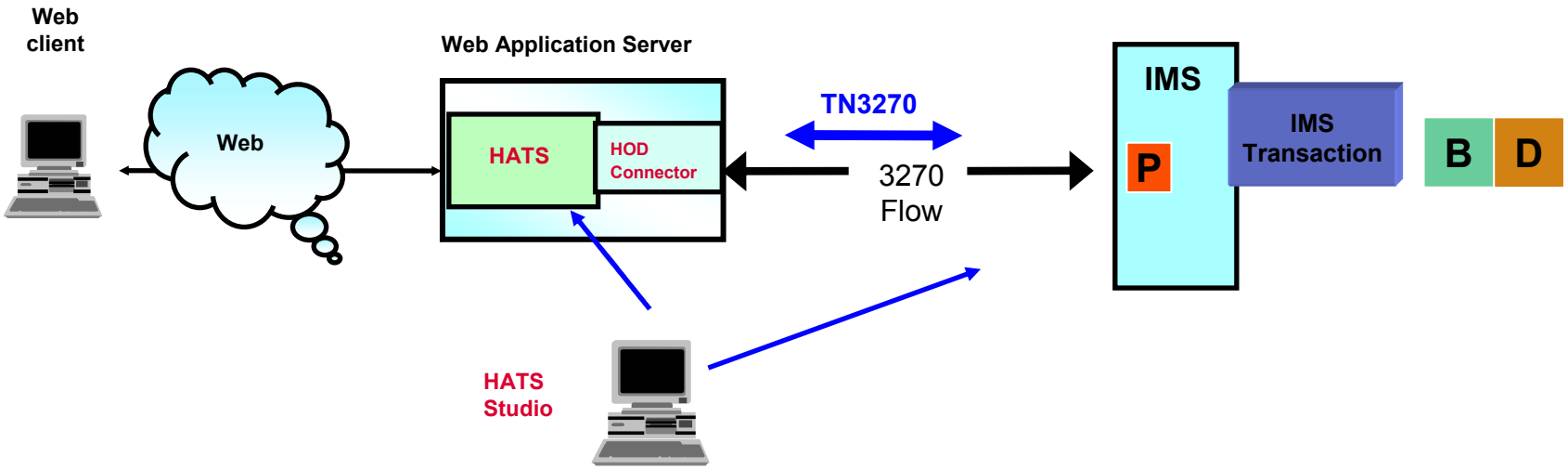
- **IMS Transaction** — **B**
  - No presentation layer
  - Access to Resource Managers (RM)
  - IMS DB, DB2, MQ
  - Very simple design
    - GU IOPCB
    - RM calls
    - ISRT IOPCB
- **IMS Database** — **D**
  - Hierarchical design
  - JDBC access
  - XML datastore (IMS V9)
- **IMS MFS** — **P**
  - Description of input and output messages and device map
  - Can be used by WSADIE to *assist* the Web Services generation
  - Not used in client/server implementations



What assets in IMS can be accessed? …

- **First IMS Asset – IMS Transactions**
  - Integrating IMS Transactions into an On Demand Application
    - Direct connection model
      - 3270 Emulation
      - JCA Connector
    - Messaging and Queuing model
    - Publishing as a Web Service
- **Second IMS Asset – IMS Databases**
  - Integrating IMS Databases into an On Demand Application
    - Direct Connection model
      - ODBA
      - JCA Connector
  - Integrating IMS Databases in a Data Integration Platform

Web-to-Host using HATS



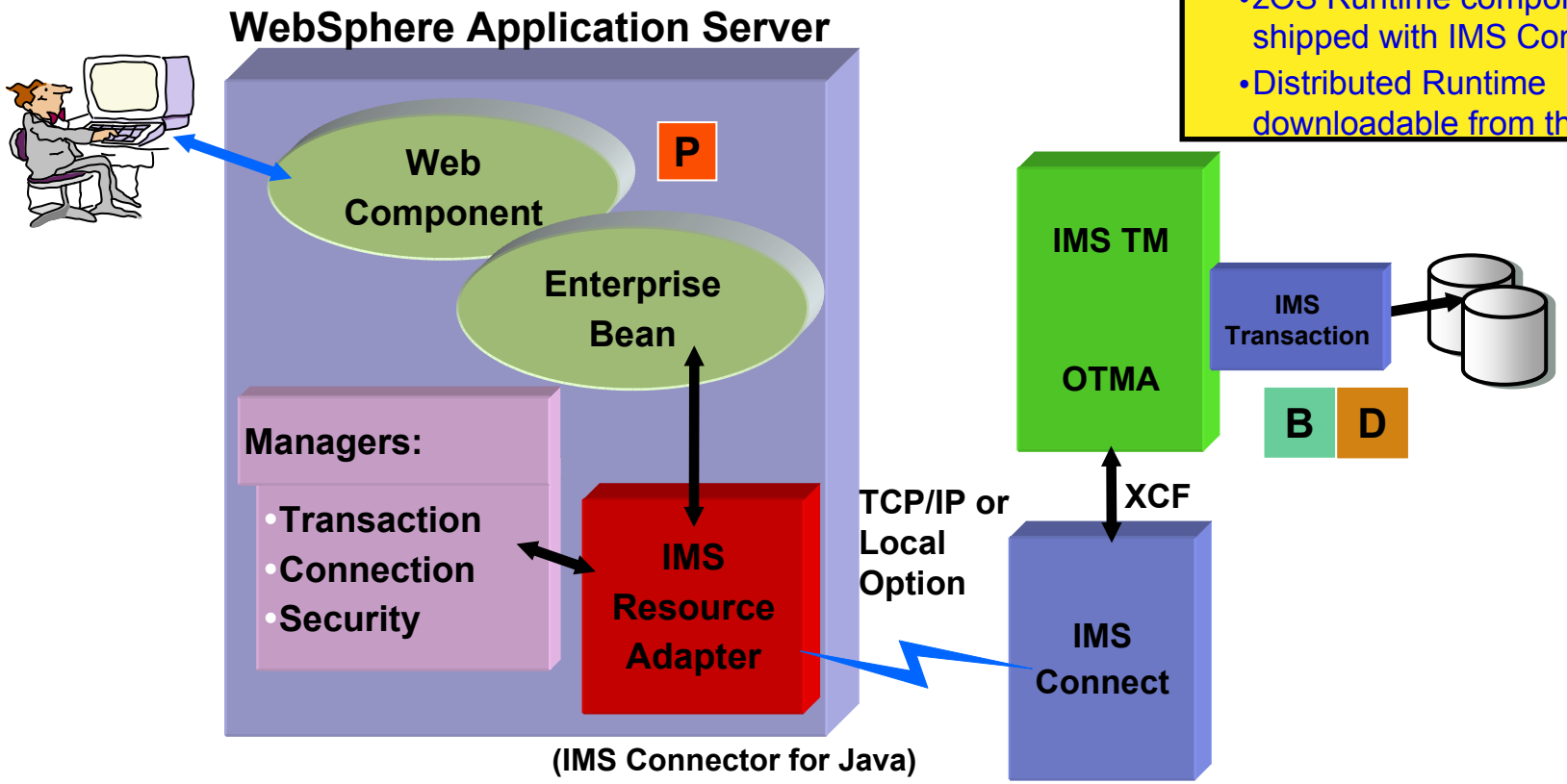
- **Host Access Transformation Server (HATS)**
  - Rules-based Web-to-Host transformation engine which dynamically converts 3270 screens in HTML pages
  - Can generate Web services or other Java objects from host transactions
  - Server based on WebSphere and 3270 HOD Connector
  - HAST Studio based on WebSphere Studio (WSAD)

*http://www.**ibm.com/software/webservers/hats***

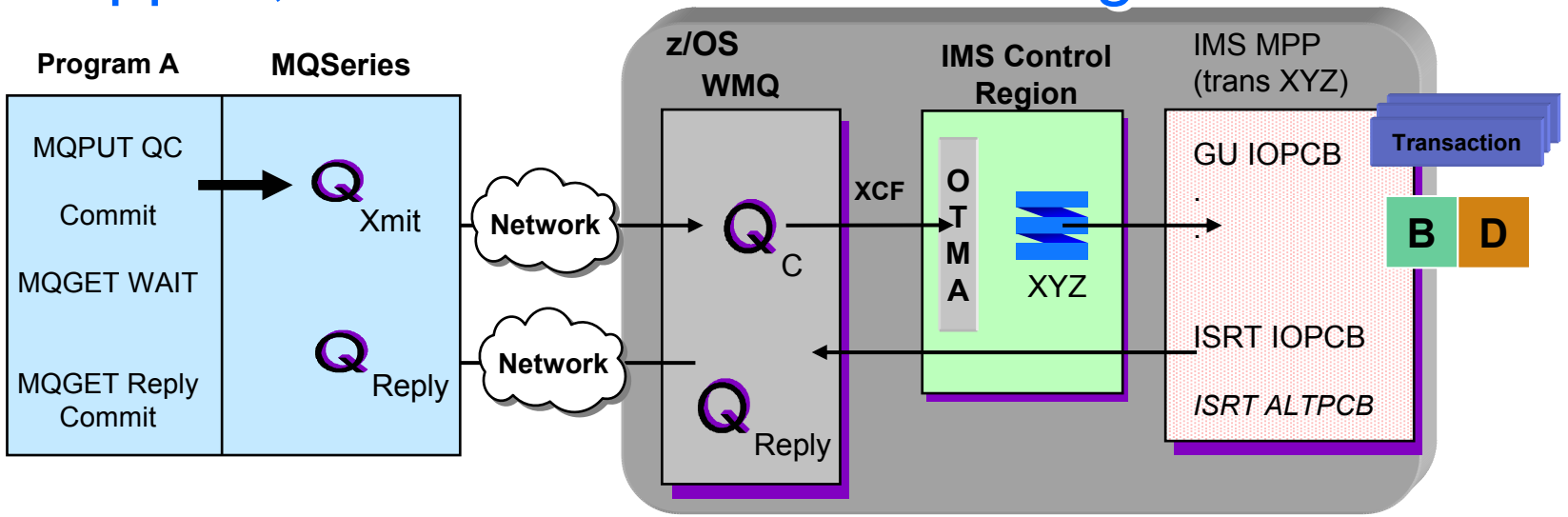
J2EE Connector Architecture (JCA) for IMS

- **IMS Connector for Java**
  - Provides connectivity to IMS Transactions

IMS Connector for Java
- Development component shipped with WSAD-IE
- zOS Runtime component shipped with IMS Connect
- Distributed Runtime downloadable from the web



JMS Support, based on WMQ IMS Bridge

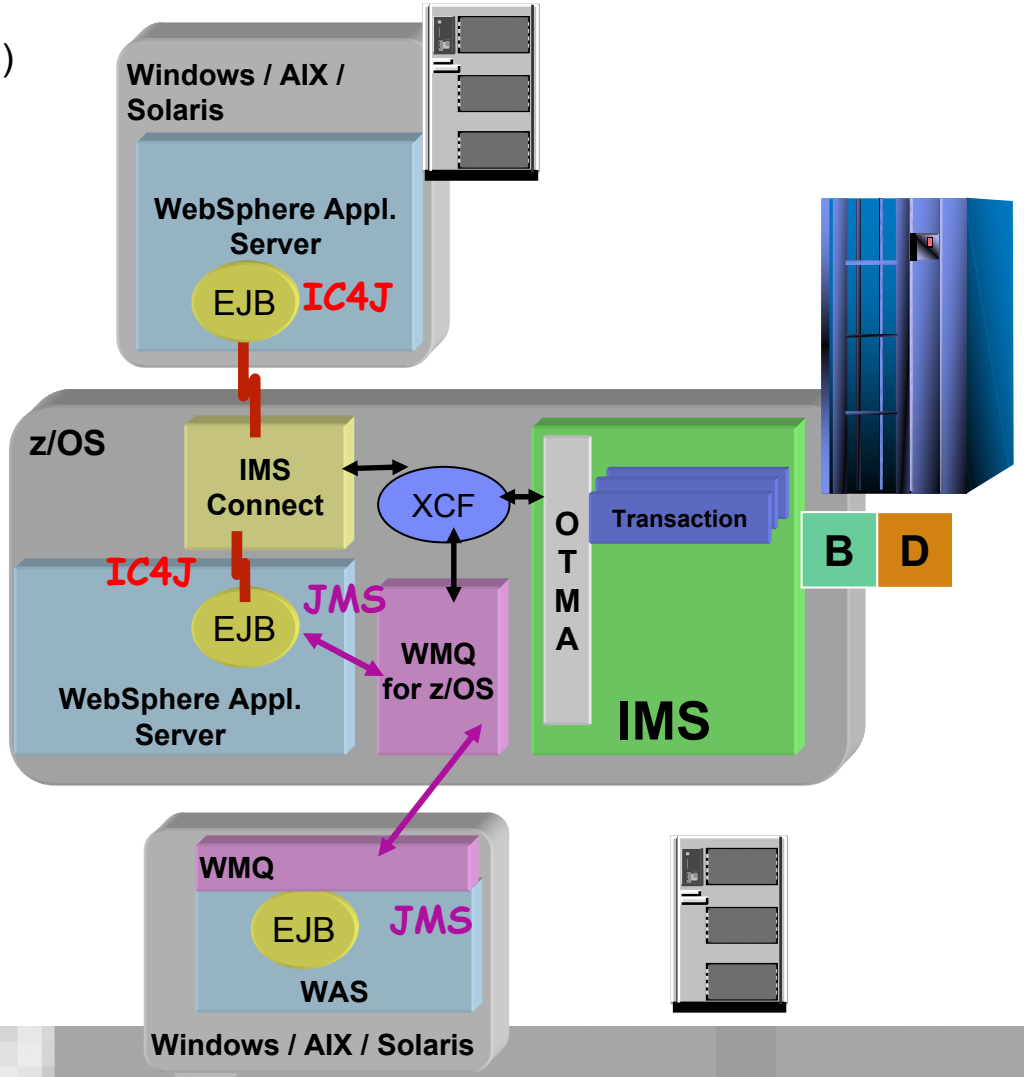


- **JMS access to IMS transaction based on the WMQ IMS Bridge**
  - Client program (or EJB) does an MQPUT to the application queue.
  - WMQ routes the message to the appropriate z/OS Queue Manager.
  - The MQ-IMS bridge transfers the message to the IMS input queue using OTMA.
  - IMS transaction (XYZ) gets scheduled, issues a GU to retrieve the message, processes the message and issues an ISRT call to send the output message

    Unchanged IMS application – No MQ API calls
  - The MQ-IMS bridge transfers the output message to the Reply_To_Queue.

EJB Access to IMS Transaction

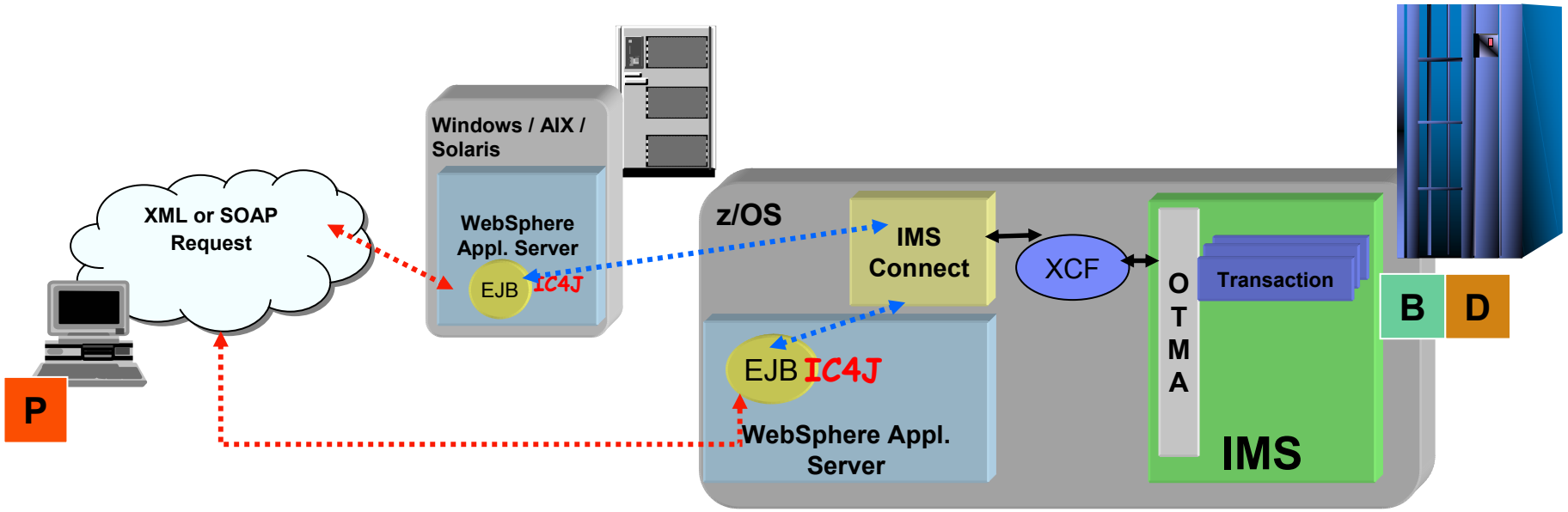
- **Thru IMS Connect**
  - Using IMS Connector for Java (IC4J)
  - From any WebSphere platform
    zOS or distributed
  - With J2EE Quality of Services
- **Thru WebSphere MQ**
  - Using MQSeries IMS Bridge



Bridging SOAP and Existing IMS applications

- **Development using WSADIE**
  - Import of the input/output message description from IMS COBOL and C applications or from IMS MFS MID/MOD formats
  - Automatic generation of WSDLs
  - Based on IC4J for IMS transaction access

- **The IMS transaction, accessed via SOAP, can be "published" as a "Web Service".**



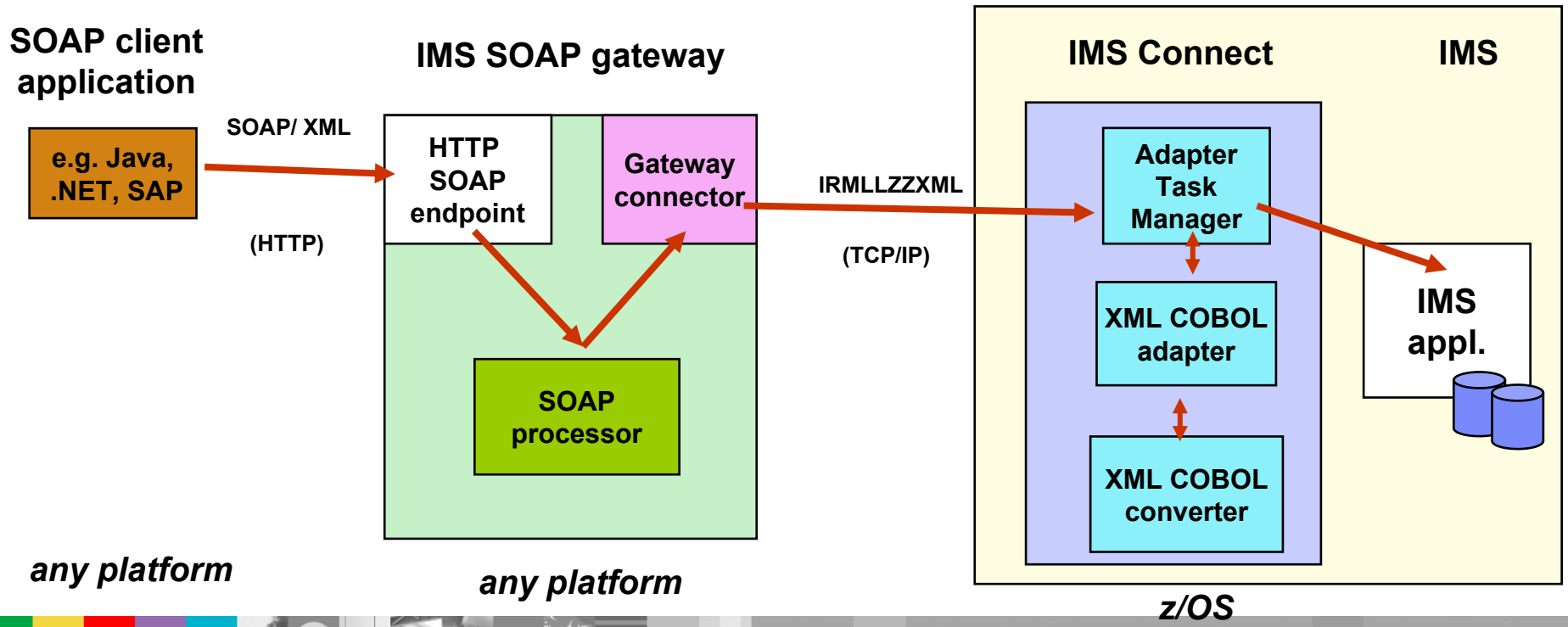
IMS SOAP Gateway – Coming Soon

- **So far ...**
  - … IMS provides services for IMS applications using (a) the WebSphere Application Server, (b) IMS Connector for Java, and (c) IMS Connect
  - transform existing IMS transactions into services by using e.g. WSAD-IE to create service definitions for IMS transactions
  - deploy these service definitions to WAS to make the IMS services available as EJB services or SOAP web services

- **The IMS SOAP Gateway is an XML based connectivity solution**
  - enable existing/new IMS applications to communicate outside of the IMS environment using SOAP
  - provide the ability for non-WebSphere customers to access IMS-based business logic
  - retrieve IMS WSDL files out of the UDDI directory and fit them into a tool (such as Microsoft .Net tools, or Apache Axis server tools) to generate SOAP messages to be sent to the host to run IMS applications
  - also use the standard APIs, such as Java API for XML-based RPC (JAX-RPC), to create both client and server code out of the generated WSDL files (RYO)

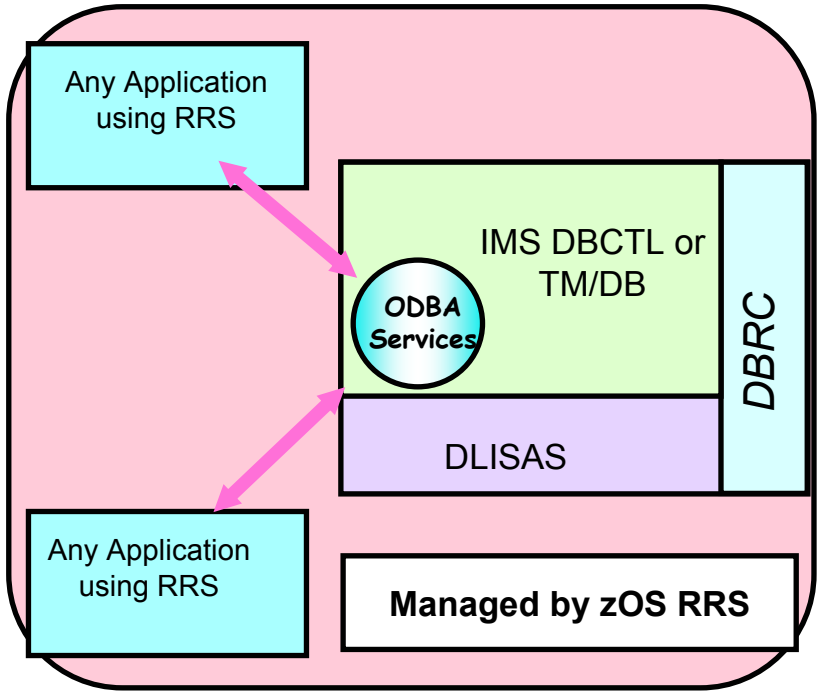
IMS SOAP Gateway – Coming Soon ...

- **Technology preview (demo) available at**
  http://www.ibm.com/ims (then look for SOAP for IMS)
- **This sample demonstrates how the input SOAP message can be sent to the IMS SOAP gateway to drive a back-end IMS transaction and to send output SOAP messages back to the client**



Access to IMS Data - Open Database Access

- **Provides a callable interface to IMS databases from any zOS programs that are not managed by IMS**
  - DB2 Stored Procedures
  - or Any applications that use zOS Resource Recovery Services (RRS) to manage their syncpoint processing
- **Connection to IMS TM or DBCTL**
  - Uses the Database Resource Adapter (DRA)
  - DL/I calls are issued using the Application Interface Block (AIB) interface



Access to IMS Data - IMS Java

- **A new feature in IMS since IMS V7**
- **A set of classes that...**
  - Offers Java support to access IMS Databases from various environments (IMS, CICS, DB2, WebSphere)
  - Enables SQL access through the JDBC interface

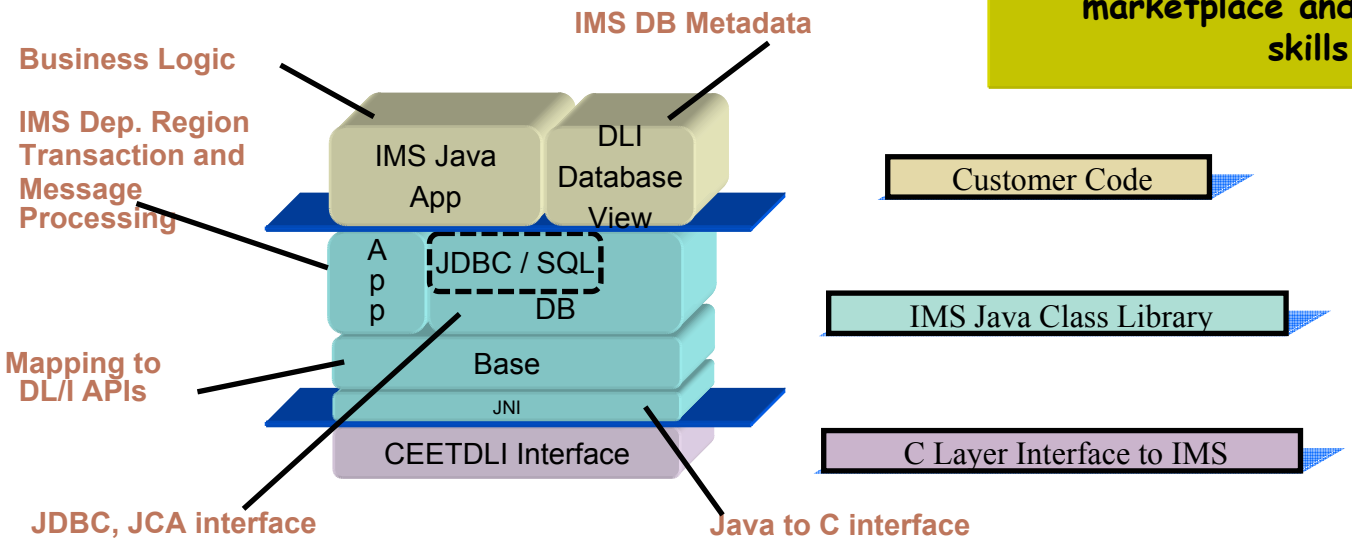
Rapid Application Development

Extend the life and scope of IMS applications and databases

Leverage existing marketplace, industry-sanctioned standards - they are the slowest changing and most persistent

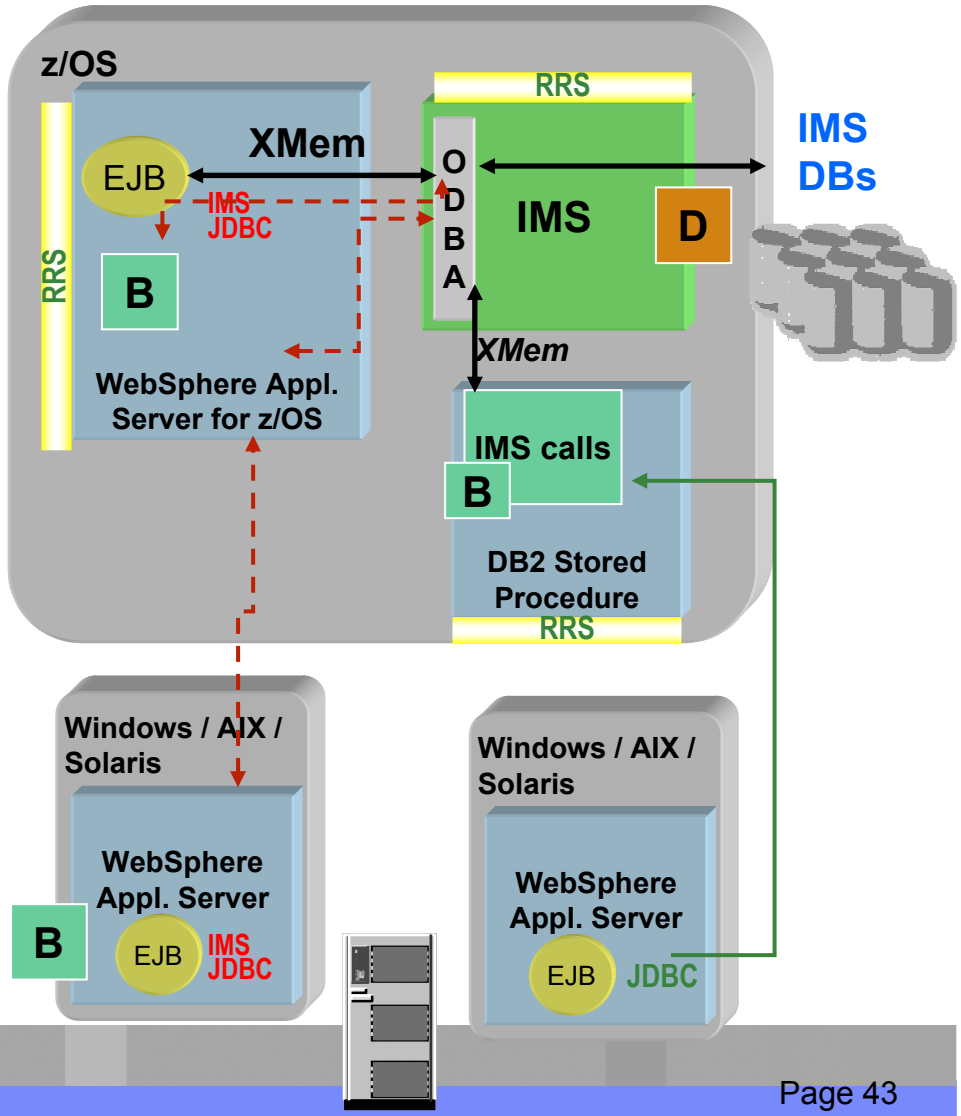
JDBC and J2EE are standards

Leverage new and abundant skills in the marketplace and mitigate the loss of 390 skills for customers



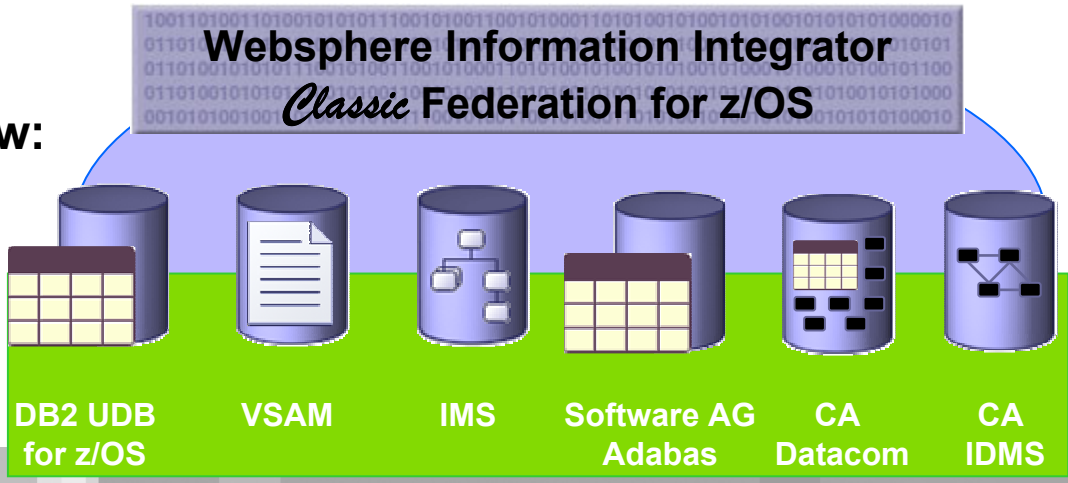
EJB Access to IMS Databases

- **Using IMS JDBC in WAS z/OS**
  - Since IMS V7
- **Using a gateway containing the IMS logic**
  - A CICS application
  - A DB2 Stored Procedure under RRS control
  - Same z/OS than the IMS Database Manager
- **Using IMS JDBC in a distributed WAS**
  - With IMS V9
  - Using a WAS for zOS Gateway
  - IMS logic in distributed EJB



Websphere Information Integrator Classic Federation for z/OS

- **Read from and write to mainframe data sources using SQL**
- **Standards-based access via ODBC, JDBC, or Call Level Interface**
- **Multi-threaded with native drivers for scalable performance**
- **Metadata-driven means:**
  - No mainframe programming required
  - Fast installation & configuration
  - Ease of maintenance
- **Works with existing and new:**
  - Mainframe infrastructure
  - Application infrastructure
  - Toolsets



IMS Strategic Options Table

Standard architecture	Middleware	Interface to IMS transaction
Client/Server on TCPIP (for any TCPIP client - .NET or other-)	none	IMS Connect
Revamping 3270	WAS + HATS + TCPIP for zOS + VTAM	3270 flow
JCA	WAS + IMS Resource Adapter	IMS Connect
JMS	WMQ	MQSeries IMS Bridge
SOAP	WAS + IMS Resource Adapter Future: IMS SOAP Gateway	IMS Connect
Data Integration	Websphere II Classic Federation + IMS Transaction Services	APPC/IMS

Standard architecture	Middleware	Interface to IMS database
On demand CICS solution	CICS Transaction	Database Resource Adapter
WAS for zOS EJB + IMS JDBC	WAS for zOS + RRS	ODBA
WAS Distributed EJB + IMS JDBC (before IMS V9)	DB2 Stored Procedure	ODBA
WAS Distributed EJB + IMS JDBC (after IMS V9)	WAS for zOS	ODBA
Data Integration	Websphere II Classic Federation	ODBA

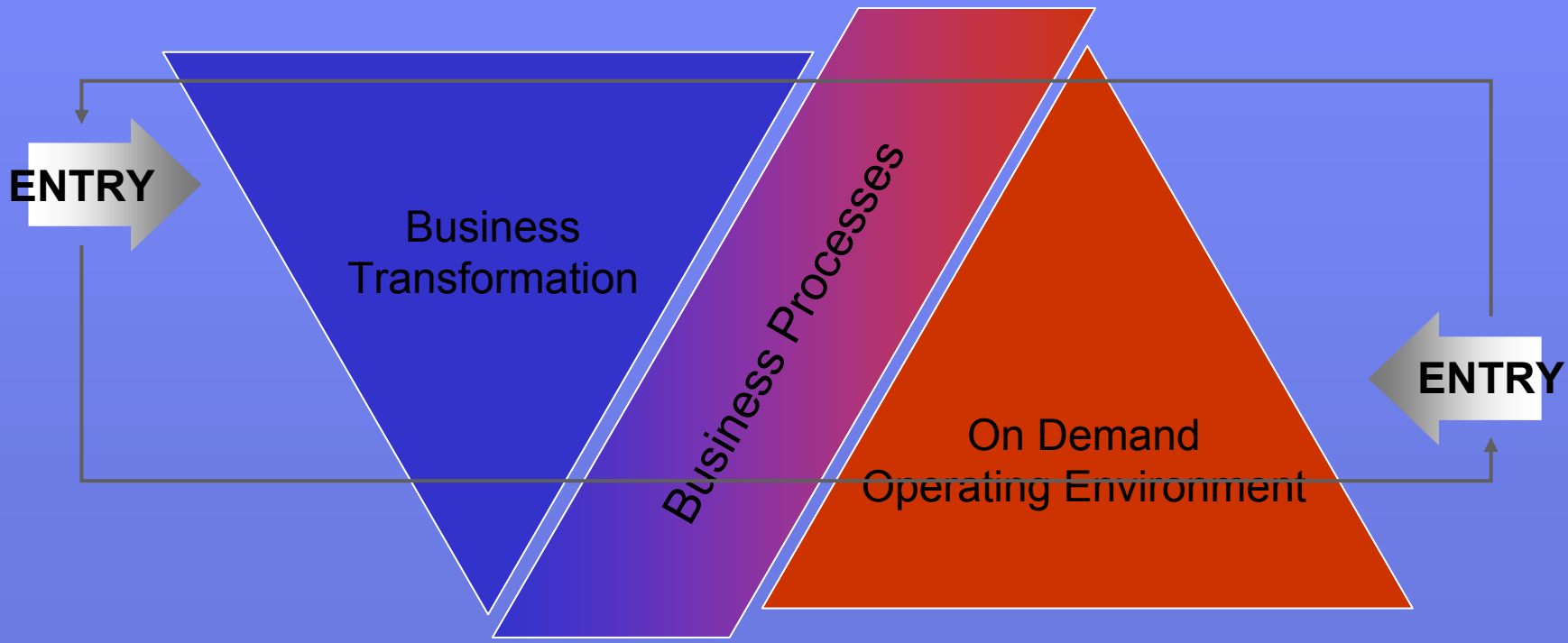
Summary

- **Legacy applications are critical to enterprise customers**
  - On-demand not interesting unless it includes legacy
  - Maintain forward progress (preserve customer investments)
  - Ratio of legacy to J2EE developers typically 5x to 10x
  - Applications must live for a long time!

An on demand business is an enterprise whose **business processes — integrated end-to-end** across the company and with key partners, suppliers and customers — can **respond with speed** to any customer demand, market opportunity or external threat.

The Essentials of an On Demand Breakthrough

Where you start depends on YOUR organization's priorities.



- Increasing flexibility is the key – business models, processes and infrastructure
- Technology acts as an enabler instead of a barrier