CICS Transaction Server V3.1

CICS and SOA Architecture Implementation Considerations

IBM

CICS™ and SOA

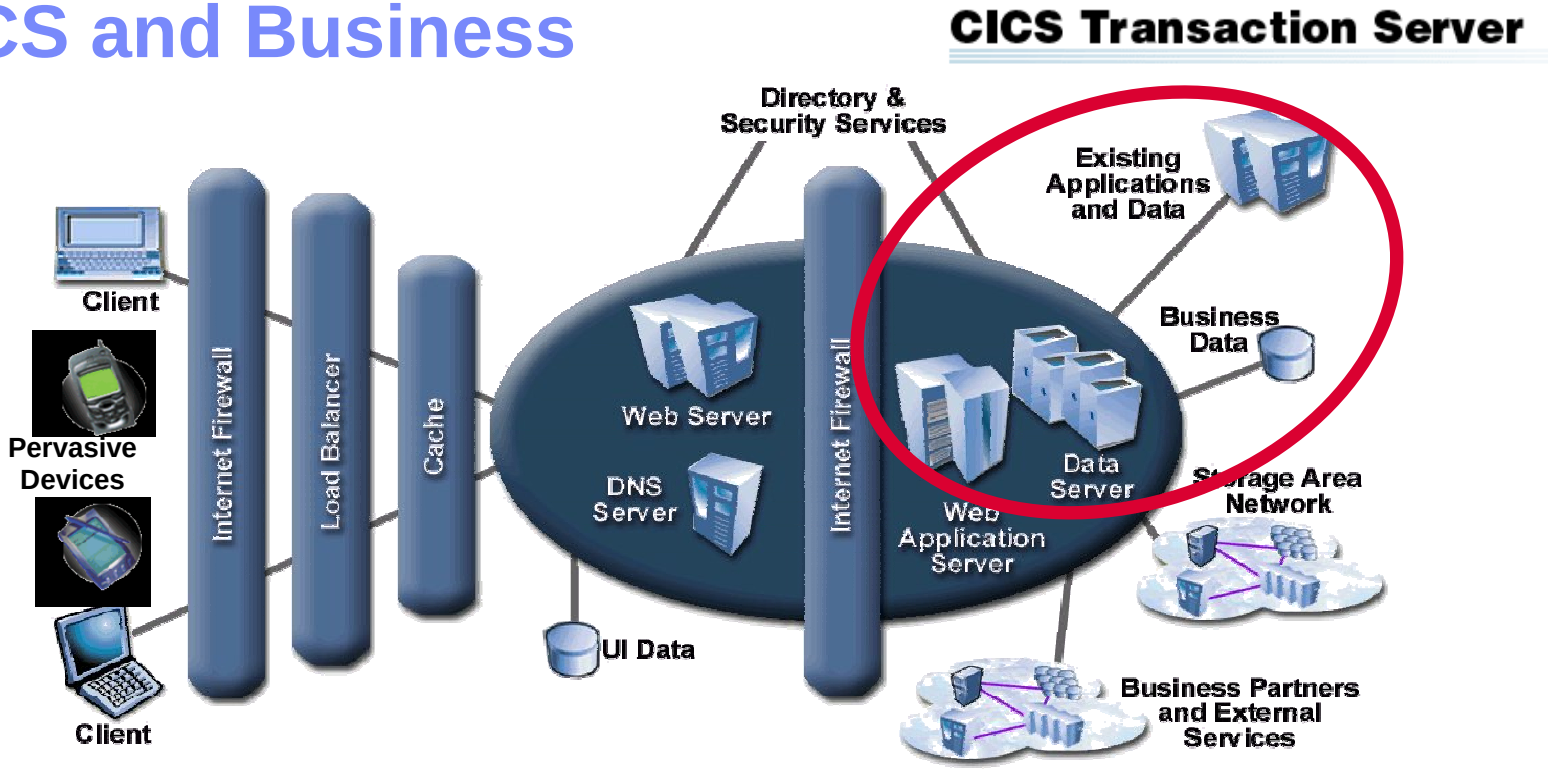
Architectural Implementation Considerations

This session addresses common architectural decisions you will have to make as you move your CICS applications to a Service Oriented Architecture. Areas discussed will include service granularity, where and when to use the CICS Transaction Gateway™, tight coupling versus loose coupling, where to expose your services, and where to process XML. Integration into the Enterprise Service Bus and integration with WebSphere Application Server™, along with availability, transactionality, and security, are of prime importance. This session will include a discussion of strategic CICS connectivity options, and when each of these options is appropriate.

Agenda

- **CICS Strategic Interoperability Options**
  - Where and when to use
  - Service Granularity
  - Tight vs. Loose Coupling
  - Where to expose your services
  - Where to process XML
  - Security and Transactionality
- **Other SOA Enablers**

CICS and Business

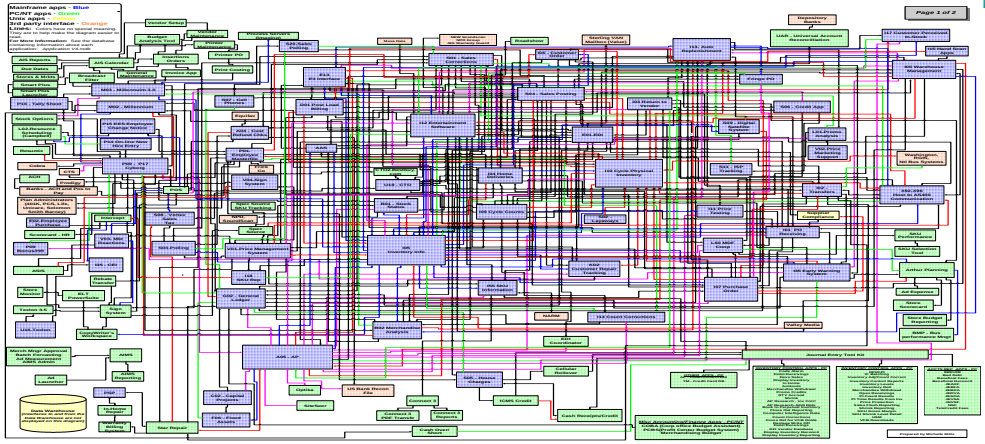


- ✓Over 35 years and $1 Trillion invested in Applications ... IDC
- ✓Over $1 trillion processed/day
- ✓Over 30 billion transactions/day
- ✓Most people use CICS

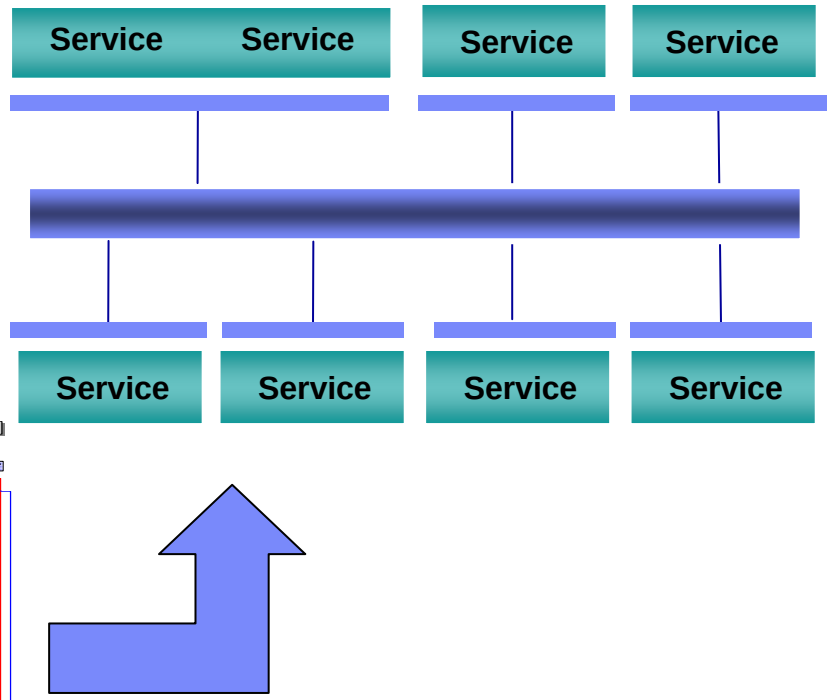
Combining the reliability and security of CICS software with the flexibility of SOA

Integration

- Complexity
- Multiple APIs
- Hidden Interfaces
- Custom coded connections



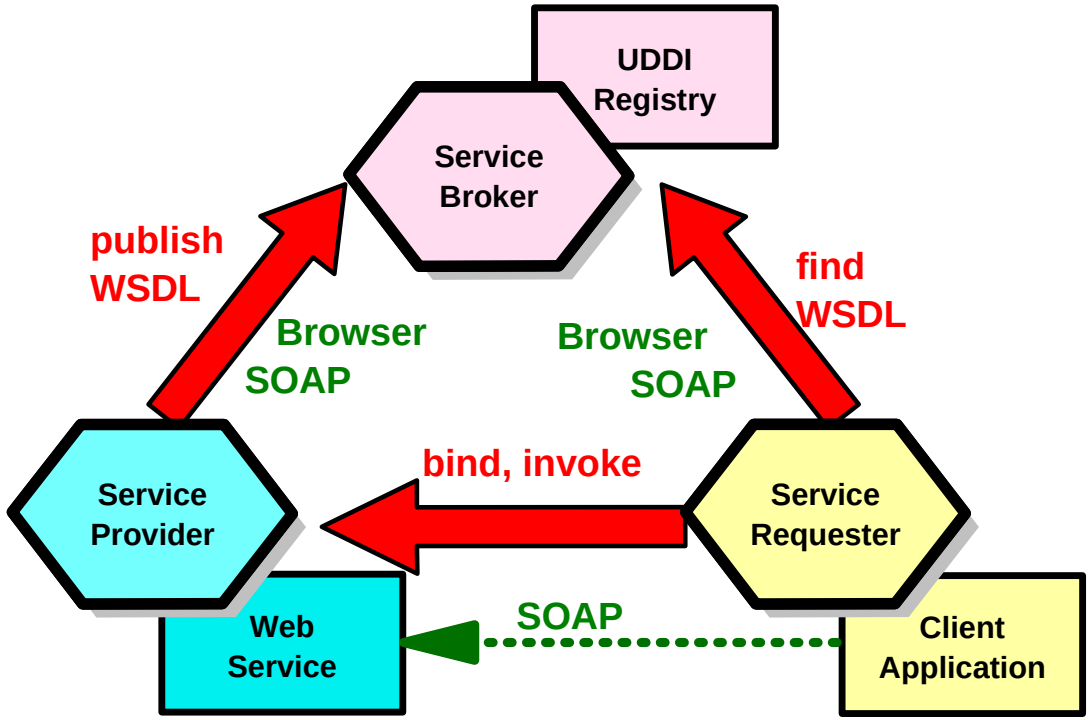
Actual application architecture for consumer electronics company



RESULT → Greater Business Responsiveness

Web Services

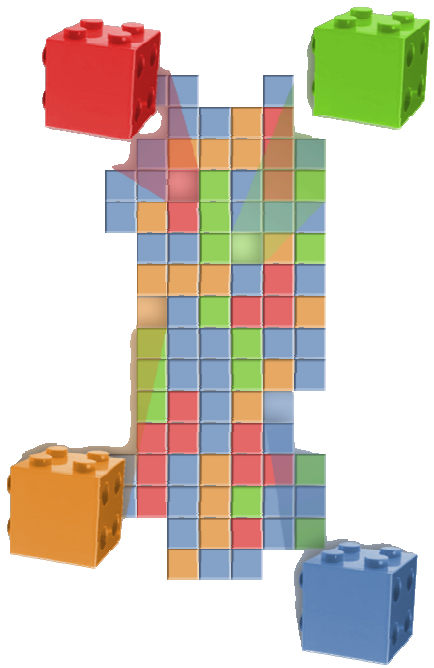
- **Architecture for**
  - Application to application
    - Communication
    - Interoperation
- **Definition:**
  - Web Services are **software components described via WSDL** that are capable of being accessed via **standard** network protocols such as SOAP over HTTP
- **WS-I.org (Web Services Interoperablity Organization)**
  - Ensure interoperability



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

SOA (Service Oriented Architecture)

- **Align with Business processes to respond faster to Business needs**
- **Composing new applications by combining Services**



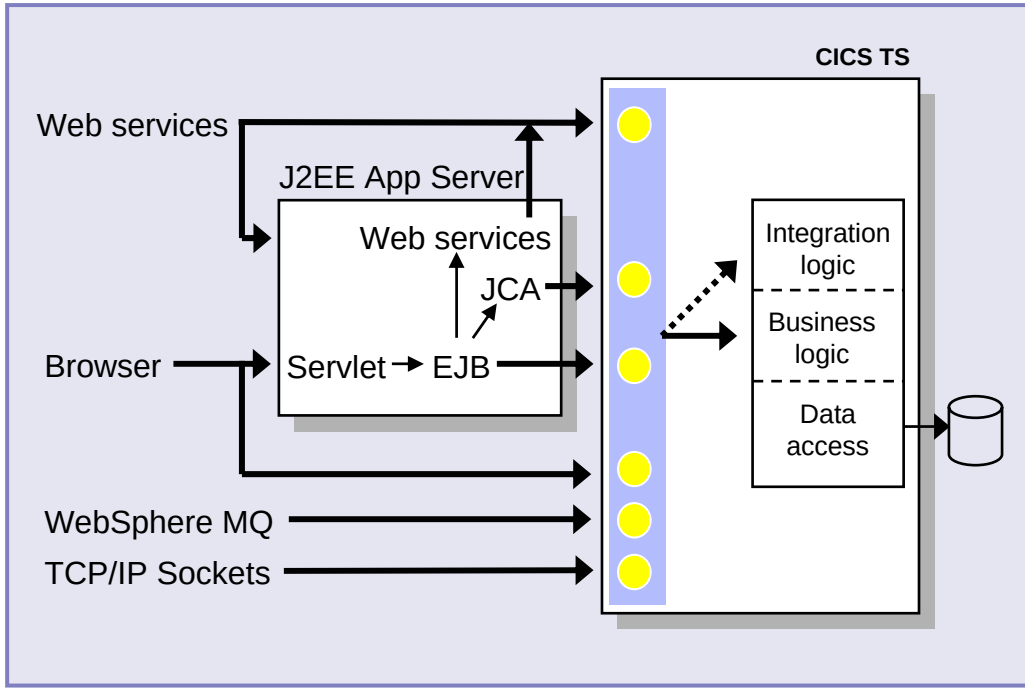
SOA Levels

- **Service Enablement - Transform existing applications to services**
- **Service Integration –Align with business, abstract integration layer, look into ESB**
- **Process Integration – Composite applications with process choreography and service aggregation**

CICS: Designed for mixed transformation and workload styles

Key Benefits

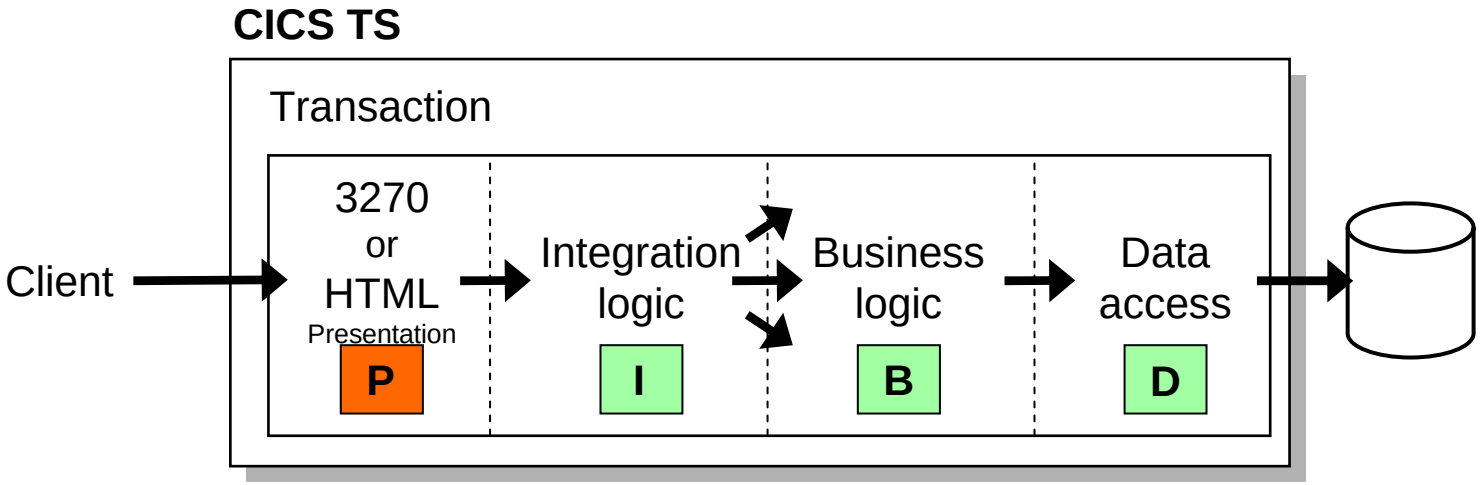
- **Standards based integration enables:**
- Programmatic integration with minimal change to existing applications
- High Qualities of Service (QoS)
- Broader range of tooling options
- System managed business workflow



Technology Characteristics

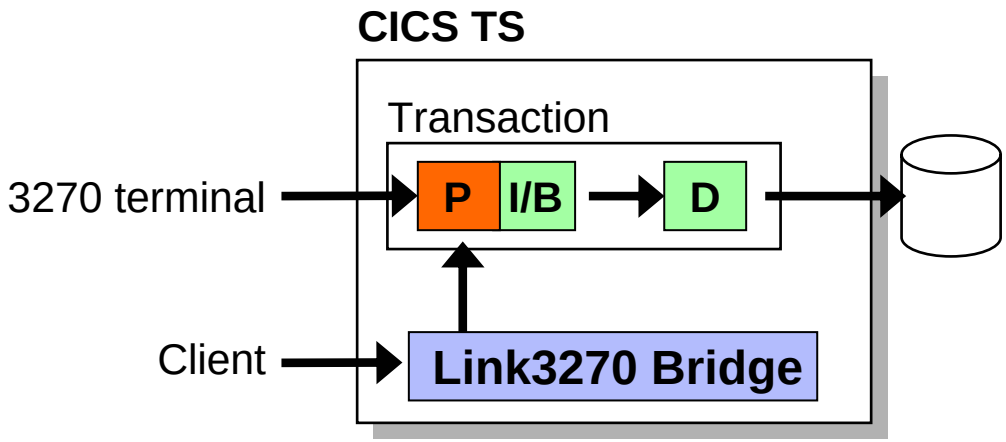
- The scope of the application solution is limited by the capabilities of the connector
- This may not satisfy the business drivers

CICS Application Architecture



- **Best practice in CICS application design is to separate key elements of the application, in particular:**
  - Presentation logic — eg. - 3270, HTML, XML
  - Integration or aggregation logic — - Menu, router, tooling
  - Business logic — - Reusable component
  - Data access logic — - VSAM, DB2, IMS, …
- **Provides a framework for reuse and facilitates separation of concerns, clear interfaces, ownership, and optimization**
- **Allows callable business logic – parameters passed via COMMAREA**

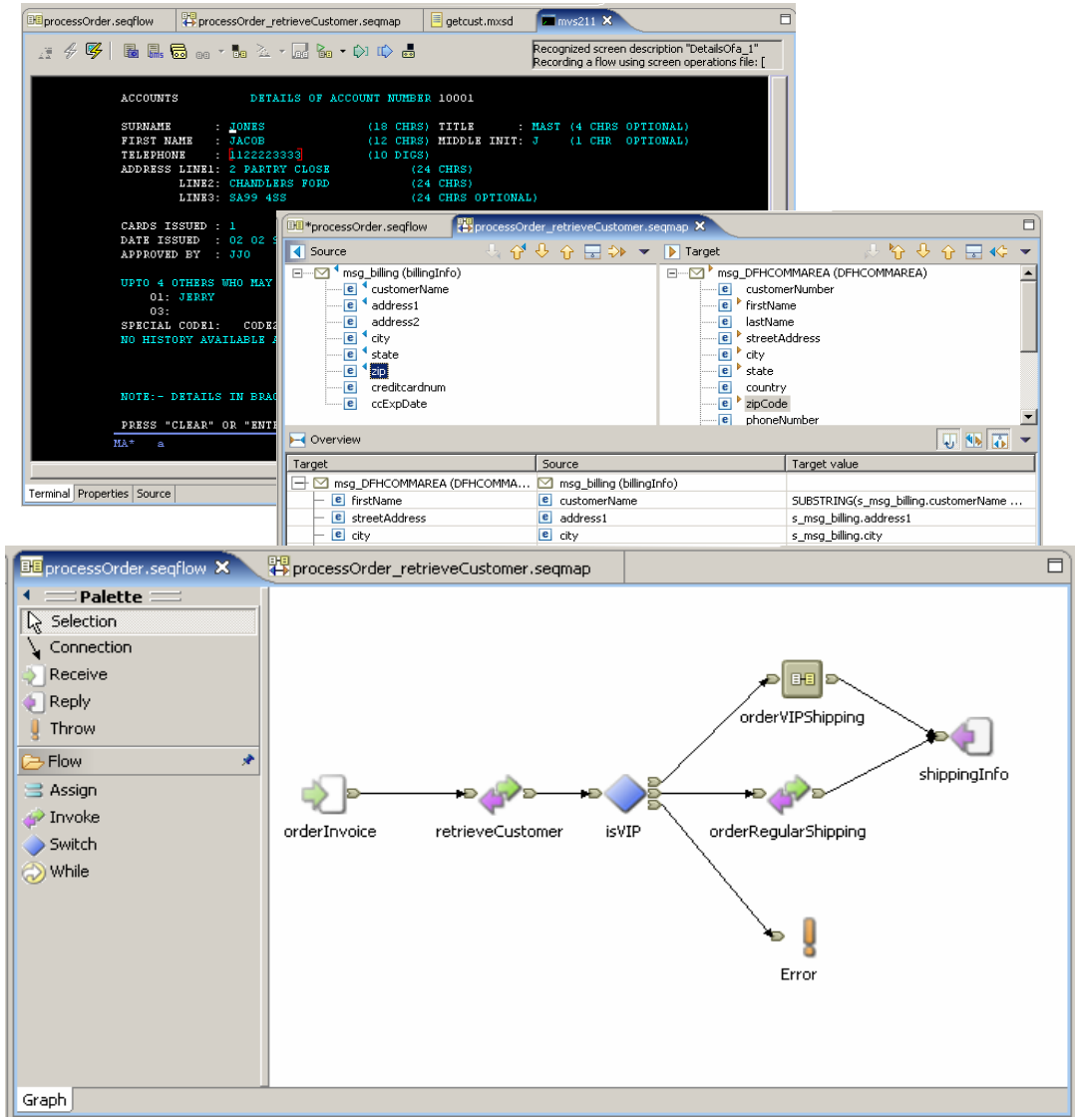
3270 Based Program Reuse



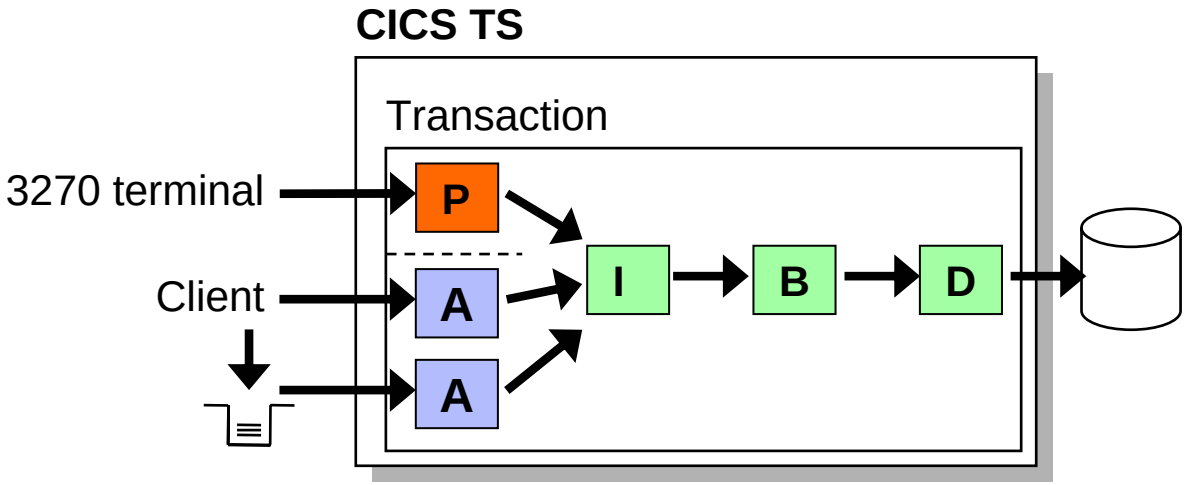
- **Some programs combine presentation, integration, and business logic**
- **Service Flow Modeler (SFM) and Link3270 Bridge provide a callable, COMMAREA interface to many BMS and terminal-oriented programs**
  - Information in the COMMAREA is passed to the BMS application
  - Does not use VTAM or screen scraping
  - No changes required to existing BMS application

Service Flow Modeler

- **Tool is part of WDz**
- **Can graphically choreograph 3270 transactions and/or COMMAREA based programs into callable COMMAREA programs**
- **Can generate artifacts necessary to expose the choreography as a Web Service in CICS**
- **Extreme productivity**



Connectivity to CICS



- **Typical clients…**
  - Web service requester
  - Java servlet or EJB running in a J2EE app server
  - C# application running in a Microsoft .NET VM
  - Web browser
  - Messaging middleware

- **Transforming technologies…**
  - External connectors

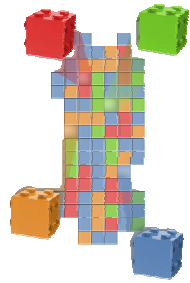
A Internal adapters (user written or generated by tools)

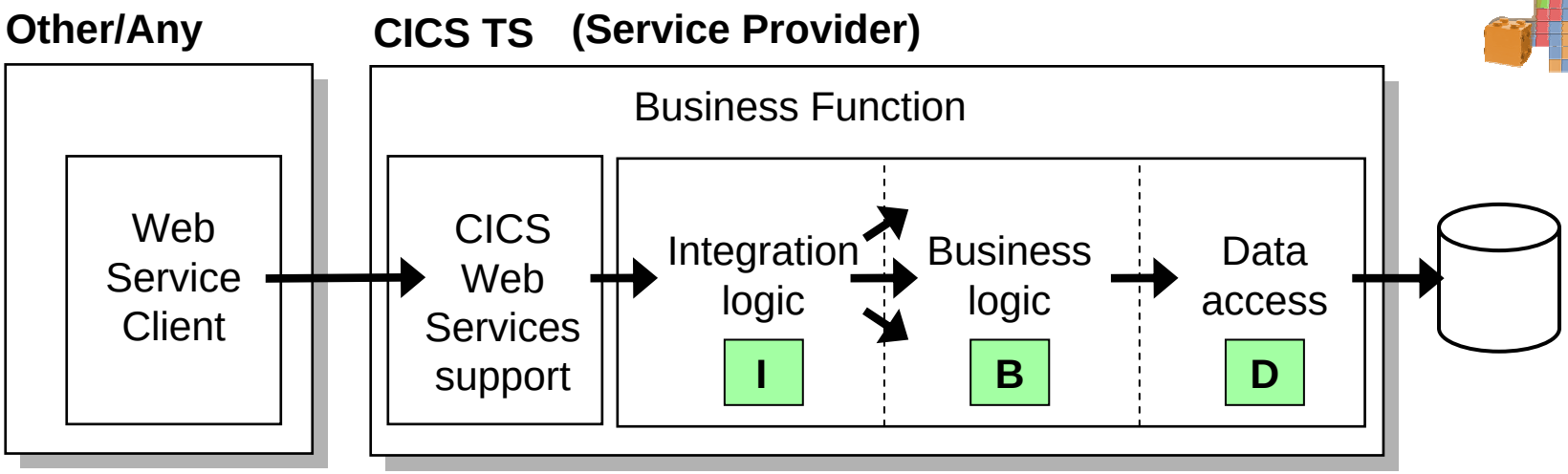
  - Standard IP-based protocol

Factors Influencing your Integration Choices

- **Business factors**
  - Agreed company standard or reference frameworks
  - Preferred application development environment and tools
  - Availability of skills
- **Technical factors**
  - Security
  - Transactional scope
  - Performance
  - Reliability, availability and scalability (RAS)
  - Application interface consumability
  - Synchronous or asynchronous invocation
  - Inbound and outbound capability
  - Client/server coupling
  - Data conversion
  - State management
- **Applications today are typically delivered across several e-business clients**

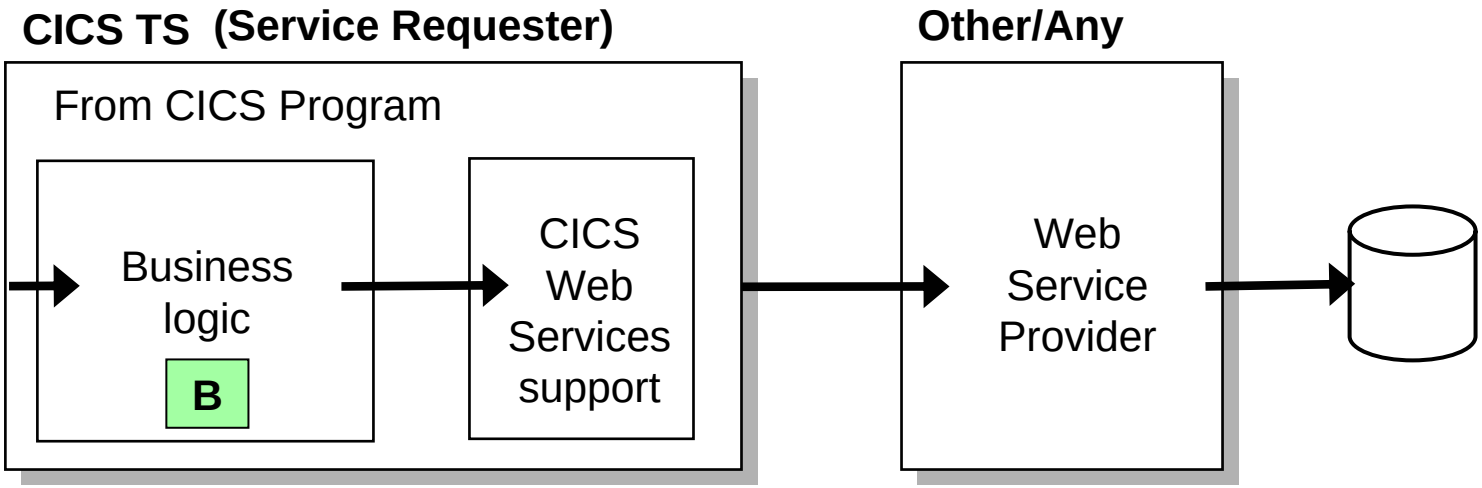
CICS Web Services Support (1 of 2)



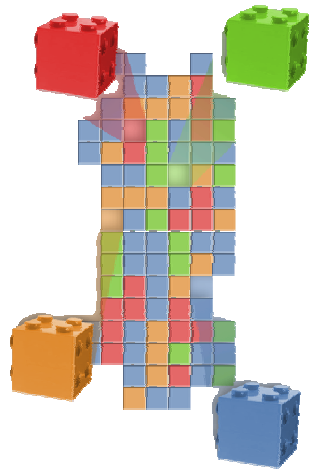


- **Web Services Clients (examples):**
  - Another program in CICS (invoke web service)
  - BPEL process (Process Choreography – WPS/WID)
  - WebSphere Web Services Gateway
  - .NET assembly
  - WebSphere MQ client
  - Anything that can invoke a Web Service

CICS Web Services Support (2 of 2)



- **Invoke Web Services from CICS programs**
  - Any Language (COBOL, Assembler, PL/I, C, C++, Java)
  - EXEC CICS INVOKE WEB SERVICE …
- **Web Service Could be (examples):**
  - A CICS based Web Service
  - BPEL process (Process Choreography – WPS/WID)
  - WebSphere Web Services Gateway
  - .NET assembly
  - Any Web Service (SOAP over HTTP or MQ)

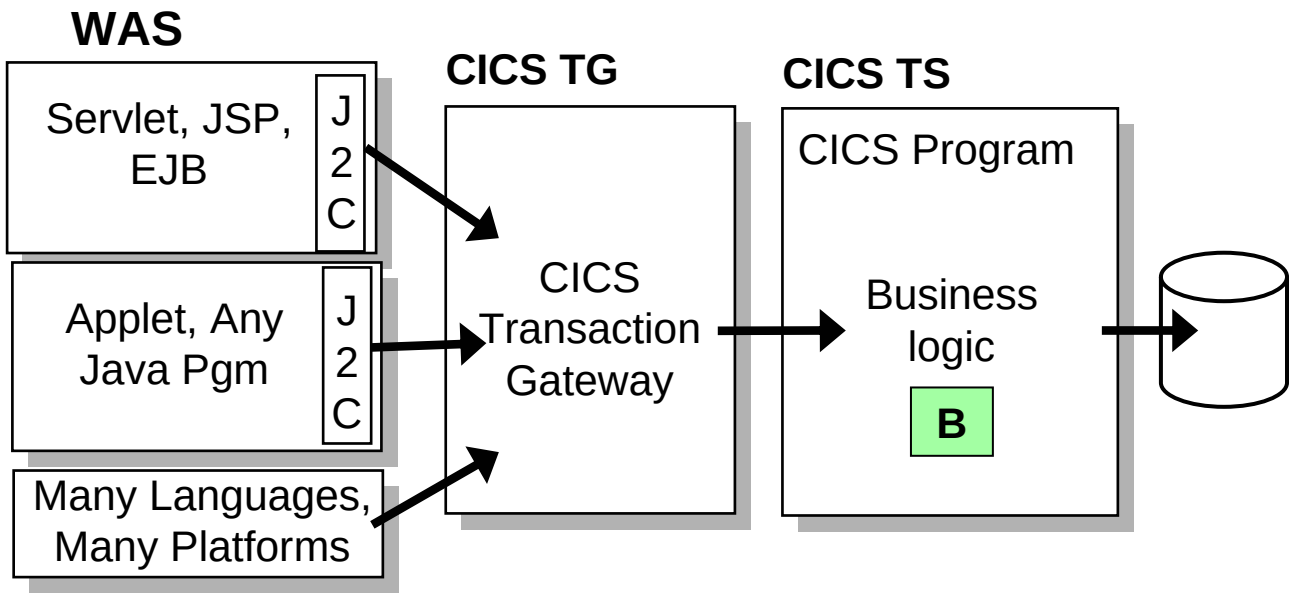


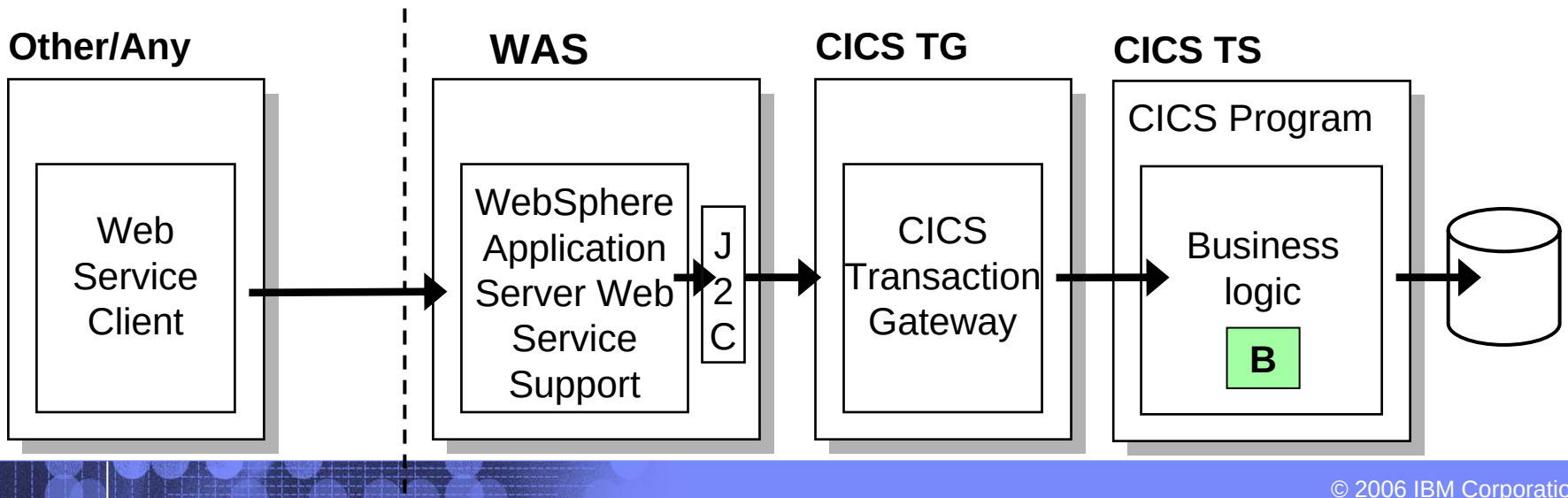
Web Service Capability

- **Capabilities**
  - Synchronous when over HTTP
  - Asynchronous when over WebSphere MQ™
  - Inbound and outbound
  - Flexible configuration using enhanced System Management
    - Multiple protocol configurations per server
    - Ease of use for header handlers
- **Security to zSeries**
  - Web services security (WS-Security)
  - Transport level security
    - Basic Authentication and SSL when over HTTP
    - User ID and password when over WebSphere MQ
- **Transactional scope**
  - Web services distributed transactions (WS-AtomicTransaction)
  - CICS local transaction
- **Interface**
  - Language structure (COBOL™, C, C++, PL/I) in a COMMAREA
  - Web services body message (XML) in a CONTAINER
  - XML transformation performed on System z
- **Coupling**
  - Loose coupling and interoperability are inherent in a service-oriented architecture (SOA) and make it a natural choice for many enterprise applications
- **Tools**
  - Tool enhancements (some CICS TS, some WebSphere Developer for zSeries™)
  - Broader language support (CICS TS)
  - Flexible COMMAREA mapping (WD/z)
  - Batch converter generation
  - CICS TS Assistant will merge with WebSphere Developer for zSeries

WAS=WebSphere Application Server

CICS Transaction Gateway

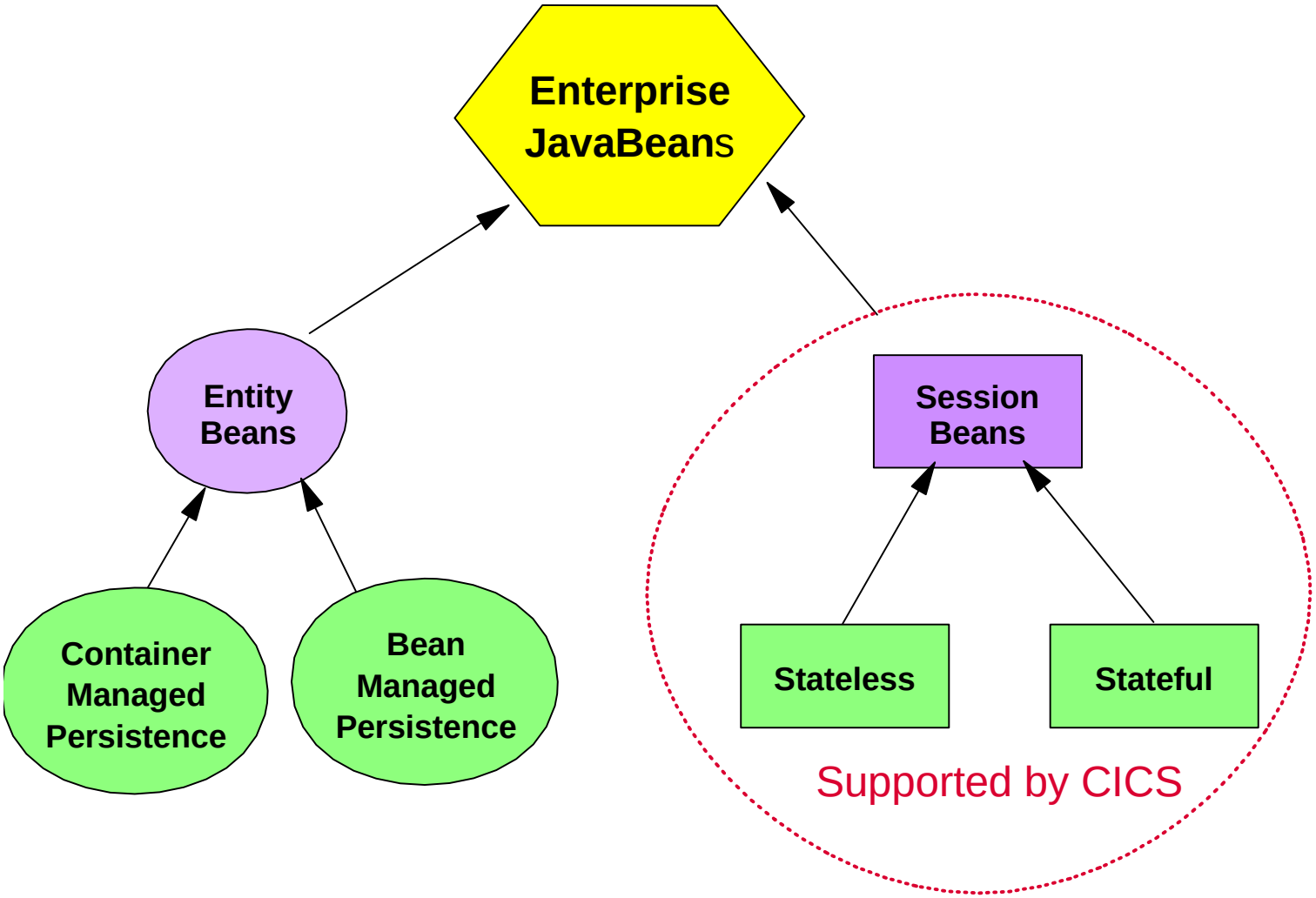




CICS Transaction Gateway

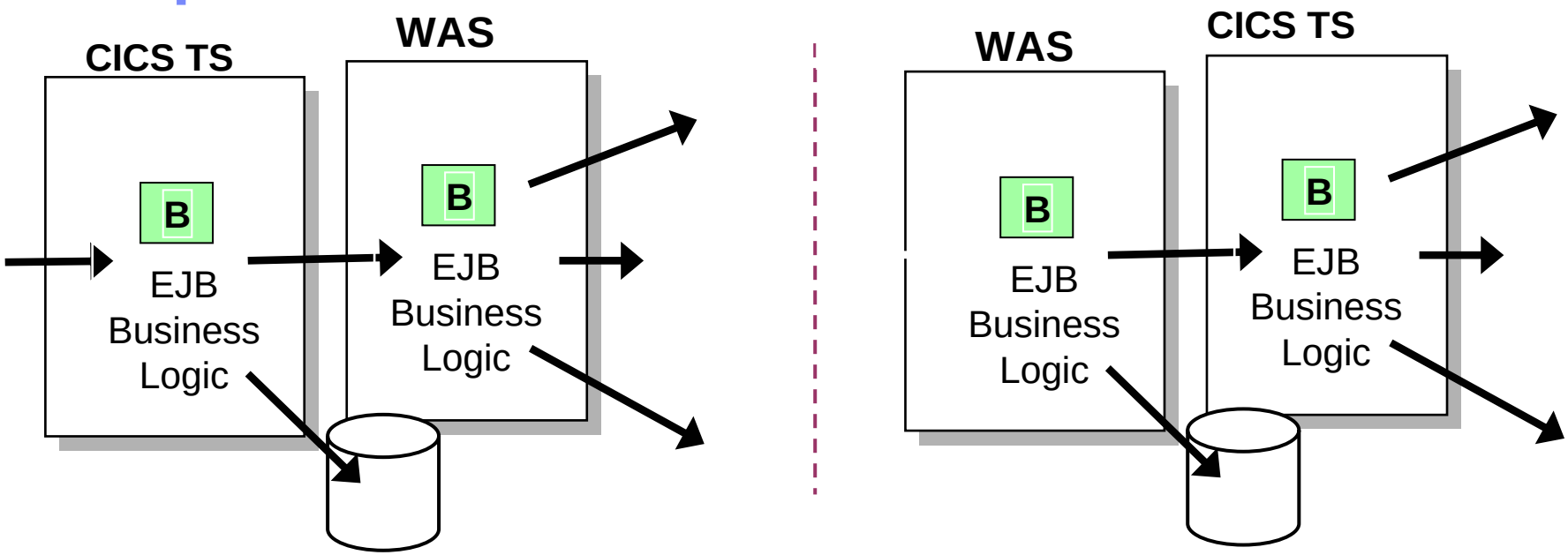
- **Capabilities**
  - Enabler for J2C (Java Connecter Architecture)
  - Synchronous
  - Inbound
  - 32K COMMAREA Limit
  - CICS TG daemon runs on many platforms
  - Proprietary flows over TCP/IP or SNA
  - Highly optimized, very fast
- **Security to zSeries**
  - SSL to CICS TG daemon
  - Userid and trusted relationship from CICS TG daemon to CICS TS
- **Transactional scope**
  - 2-PC (XA resource) when CICS TG daemon on z/OS
  - 2-PC (last participant support with WAS V6)
  - 1-PC when CICS TG daemon on a distributed platform
- **Interface**
  - J2C = Java (CICS TG daemon)
  - Other languages (direct to CICS)
  - COMMAREA-like interface
  - ASCII to EBCDIC performed on client or CICS (zSeries)
- **Coupling**
  - Medium coupling, programs on both sides need to know data layout. Layout not described via industry standard
- **Tools**
  - Can generate J2C capable object to use the CICS TG from Rational Application Developer™ (RAD), WebSphere Integration Developer™ (WID) or WebSphere Developer for zSeries (WDz)

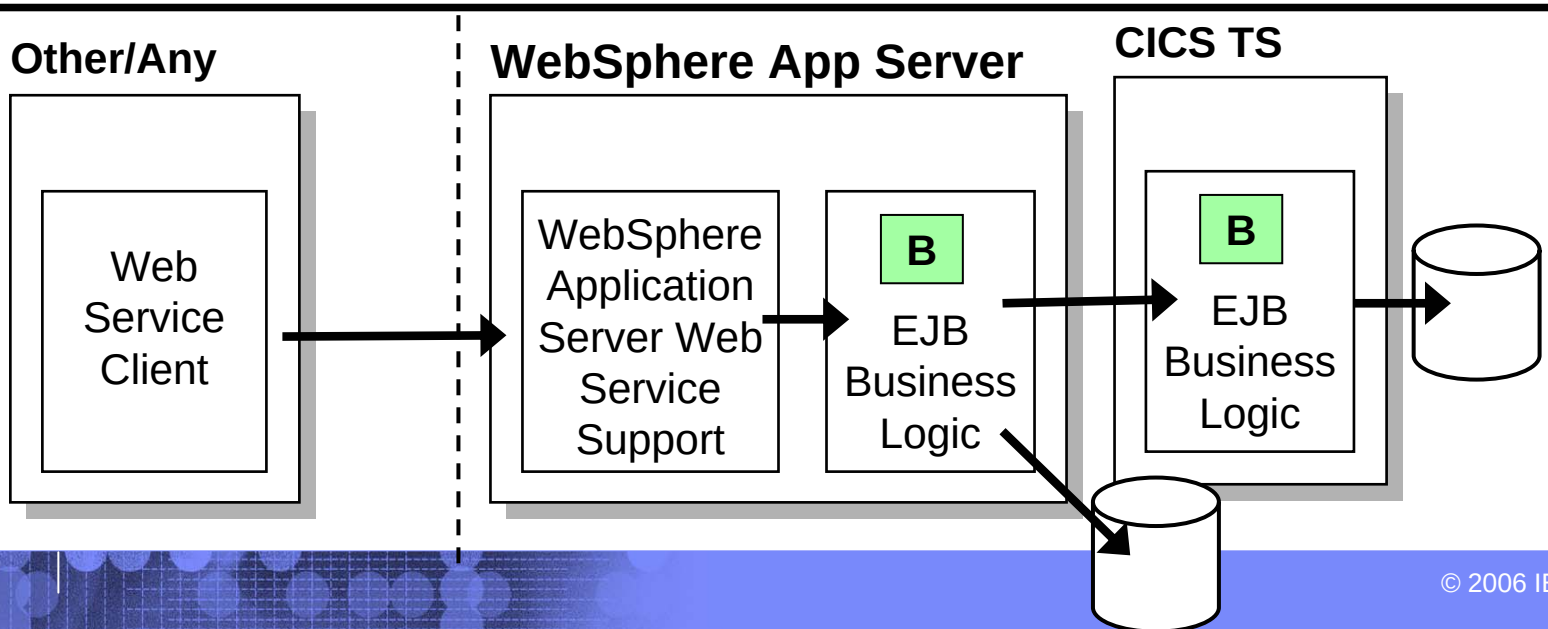
EJBs: Types in EJB 1.1



Enterprise JavaBeans

WAS=WebSphere Application Server



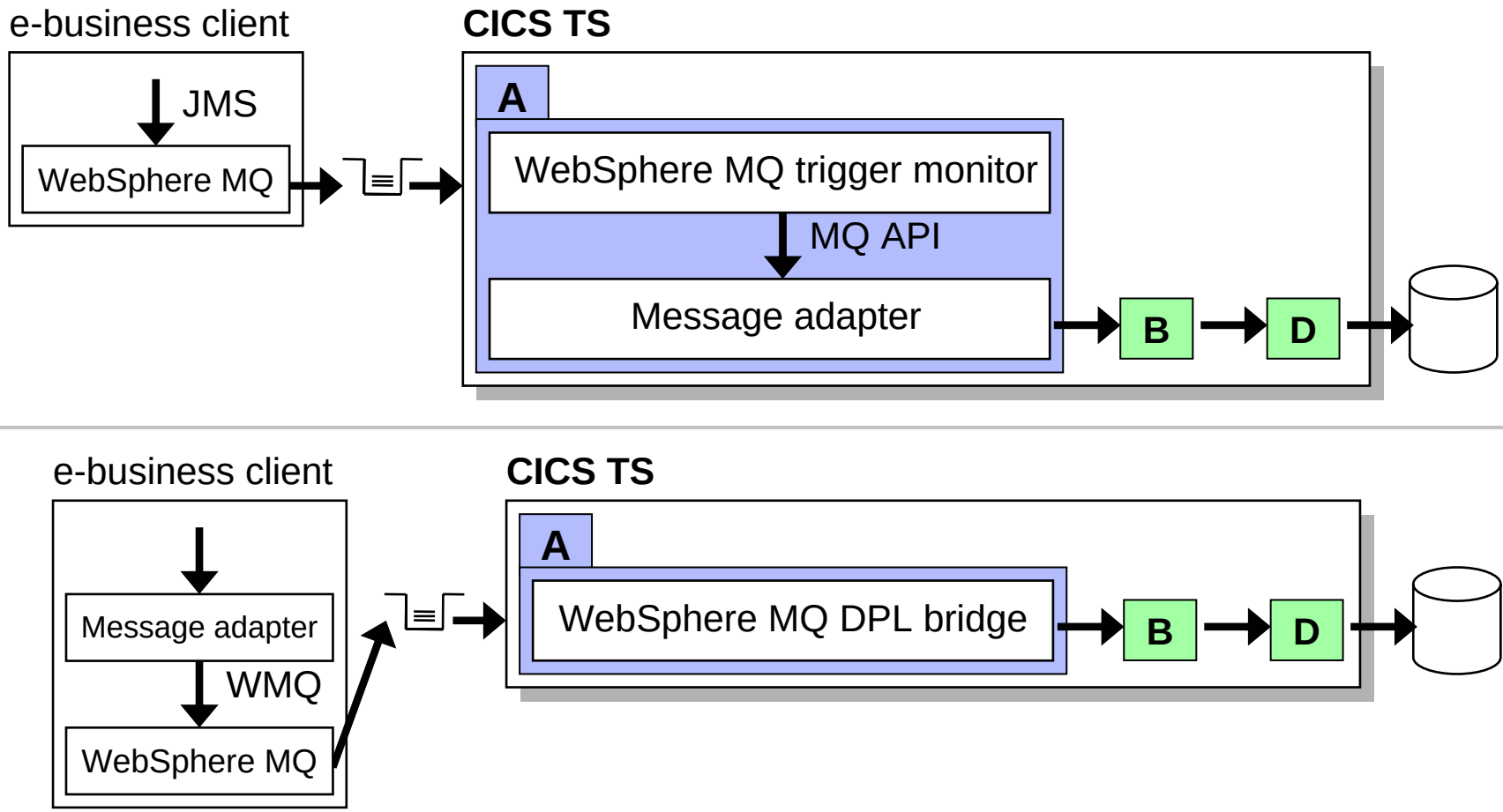


Java COMPATIBLE

Enterprise JavaBeans

- **Capabilities**
  - Synchronous (RMI over IIOP)
  - Inbound and outbound
  - Flexible configuration
  - Easy deployment
  - Can use LDAP or COS naming JNDI server
  - Can be dispatched to zAAP processor
- **Security to zSeries**
  - Method level security
  - SSL
  - Asserted Identity with WebSphere Application Server on z/OS
- **Transactional scope**
  - Many transactional options (specified in deployment descriptor)
  - Can be 2-PC with EJBs on other platforms
  - Allows CICS resources to be in same unit-of-work with WebSphere Application Server EJBs
- **Interface**
  - EJB 1.1 Specification
  - Session Beans only
  - Stateful or Stateless
  - Arguments marshalled and de-marshalled on both sides
  - Can be packaged as V2.0 session beans
- **Coupling**
  - Tight coupling, Java required on both sides
- **Tools**
  - Any tool that produces EJB 1.1 compliant Session beans, includes
    - WebSphere Studio Application Developer
    - Rational Application Developer
    - WebSphere Integration Developer
    - WebSphere Studio Enterprise Developer
    - WebSphere Developer for zSeries

WebSphere MQ

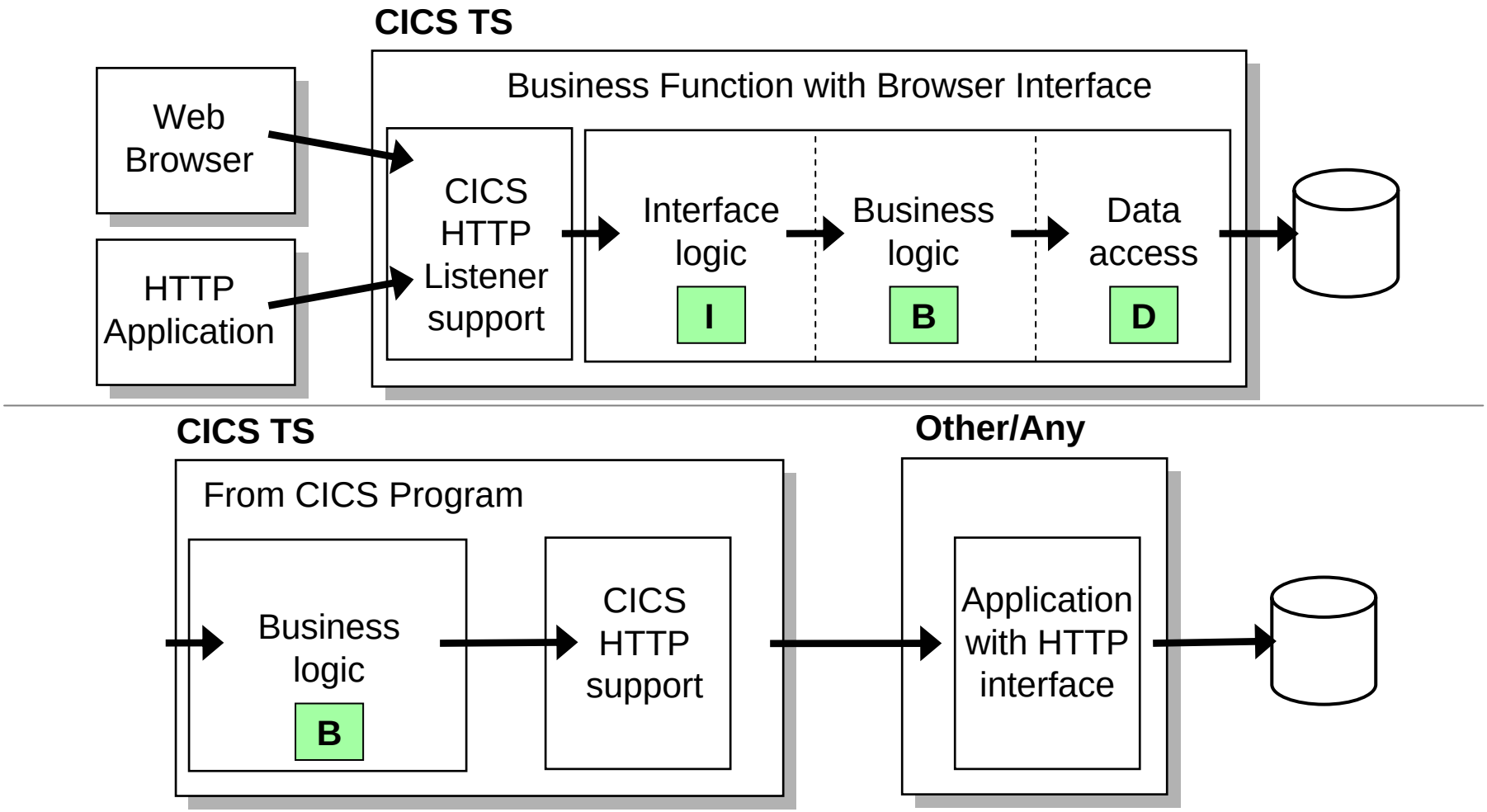


- **Assured asynchronous message delivery**
- **MQ DPL bridge links to existing CICS programs**

WebSphere MQ

- **Capabilities**
  - Asynchronous
  - Assured Delivery
  - Inbound and outbound
  - Flexible configuration
  - Large message sizes
- **Security to zSeries**
  - Userid / password as specified in the queue definition
  - SSL
- **Transactional scope**
  - Unit-of-work includes GET and PUT
- **Interface**
  - JMS
  - MQ API
  - COMMAREA when used with DPL Bridge
  - Web Service
- **Coupling**
  - Medium coupling, Both sides need to know message format
- **Tools**
  - WID can generate

CICS Web Support

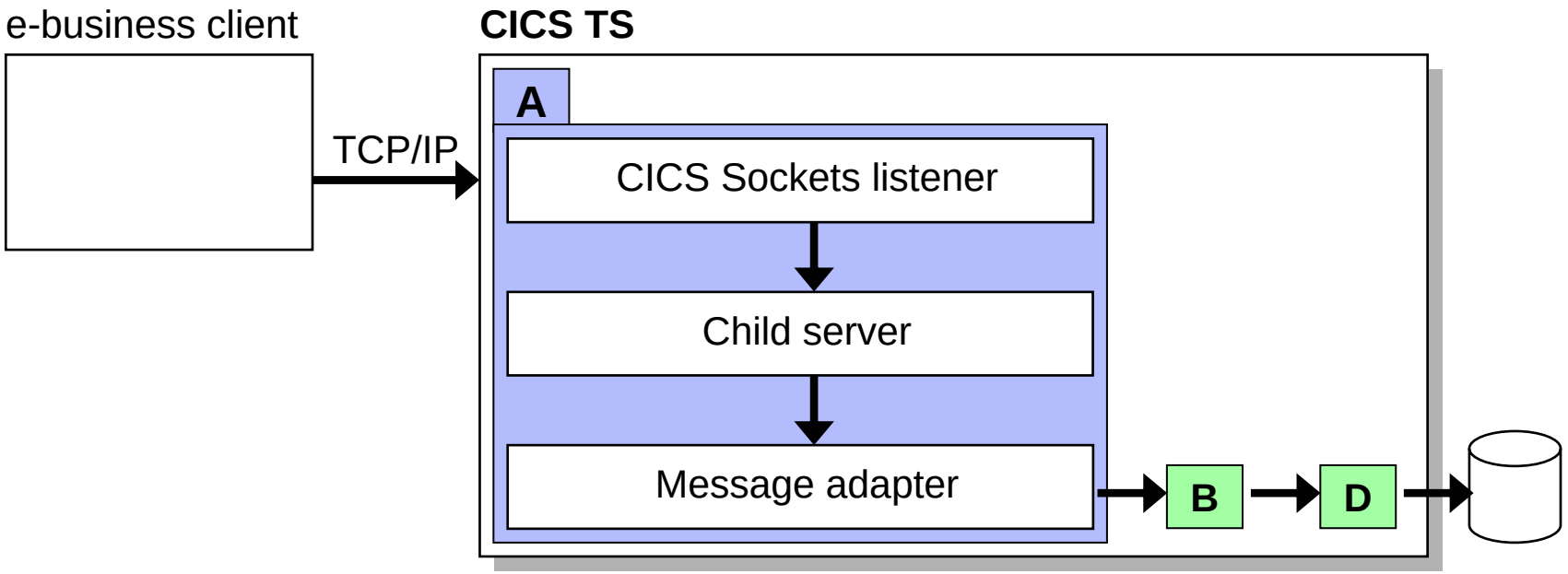


- **Interact with Web Browser or HTTP-based applications**

CICS Web Support

- **Capabilities**
  - Synchronous
  - Inbound and outbound
  - HTTP 1.1
  - Conditionally compliant with HTTP 1.2
  - Allows Interaction with Web browser
  - Uses z/OS conversion services
- **Security to zSeries**
  - Basic Authentication
  - SSL
- **Transactional scope**
  - Local CICS Transaction
- **Interface**
  - CICS WEB API
  - CICS DOCUMENT API
  - HTTP 1.1 and 1.2
  - Chunking, Pipelining
  - New URIMAP Definition
- **Coupling**
  - Medium coupling, both sides need to know message layout. Layout not documented with WSDL.
- **Tools**
  - Many tools to design Web pages, graphics, CSS, JavaScript, etc. including
    - WebSphere Studio Application Developer
    - Rational Application Developer
    - WebSphere Integration Developer
    - WebSphere Studio Enterprise Developer
    - WebSphere Developer for zSeries

TCP/IP Sockets



- ***CICS Sockets provides a completely programmable solution where other access options are not suitable***

TCP/IP Sockets

- **Capabilities**
  - Synchronous, possibly Asynchronous
  - Inbound and outbound
  - Flexible Programming model
  - Code your own listener and dispatcher
- **Security to zSeries**
  - Userid / password
  - SSL
- **Transactional scope**
  - Local CICS Transaction
- **Interface**
  - CICS Sockets API (variant of Berkeley Software Distribution 4.3 Sockets)
  - Flexible – do anything you can program
  - You handle conversion when required
- **Coupling**
  - Very tight coupling – low-level API required on both sides, although various languages can be used
- **Tools**
  - Some tools exist depending on chosen language
    - IDE coding assistance
    - No tools to generate code

CICS Interoperability Summary

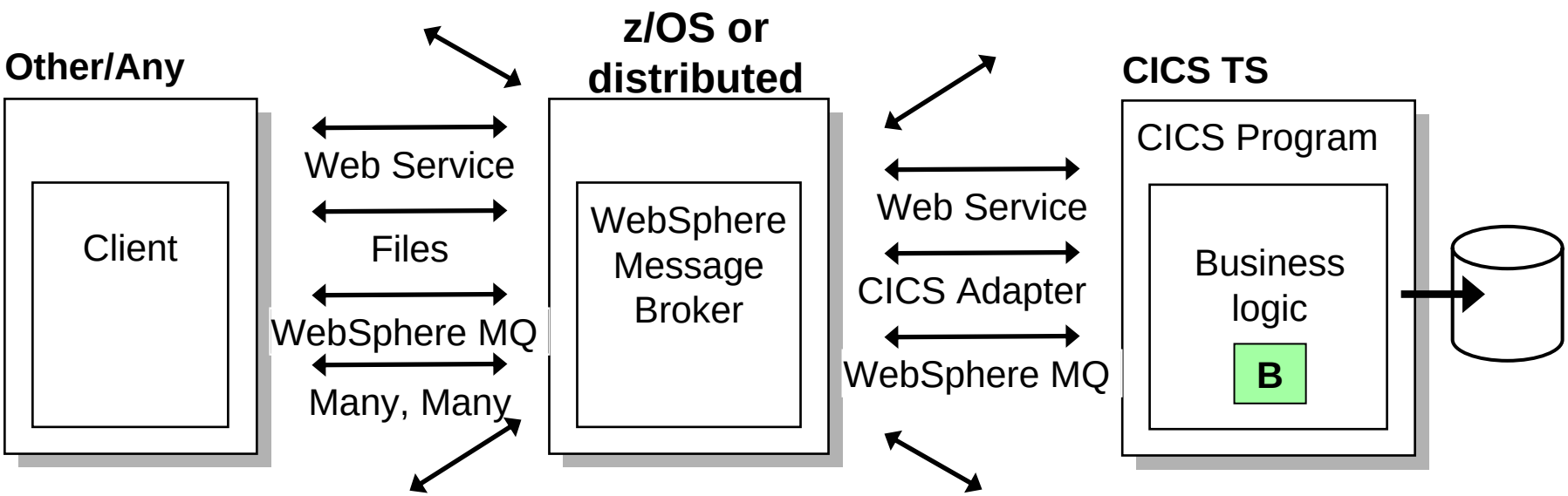
CICS Transaction Server

Standard architecture	Capabilities	Security to zSeries	Transactional scope	Interface	Coupling
1. Web Services	Synchronous (HTTP) Asynchronous (WebSphere MQ) Inbound and outbound	User ID + password SSL	Local CICS transaction Global transaction	CONTAINER COMMAREA XML	Loose
2. JCA	32KB max message size Inbound only Synchronous and Async	User ID + password Thread identity SSL	Local transaction Global transaction	COMMAREA	Medium
3. Enterprise JavaBeans	EJB state management Inbound and outbound Synchronous	EJB security roles SSL	CICS transaction Global transaction	Enterprise JavaBean session bean	TIght

Standard transport

4. WebSphere MQ	Inbound and outbound Asynchronous Assured delivery	User ID + password SSL	CICS transaction	WebSphere MQ API or COMMAREA	Medium
5. HTTP	Inbound and outbound Synchronous	User ID + password SSL	CICS transaction	CICS WEB API	Medium
6. TCP/IP sockets	Inbound and outbound Synchronous and Async	User ID + password	CICS transaction	CICS sockets API	Tight

Advanced ESB – WebSphere Message Broker™

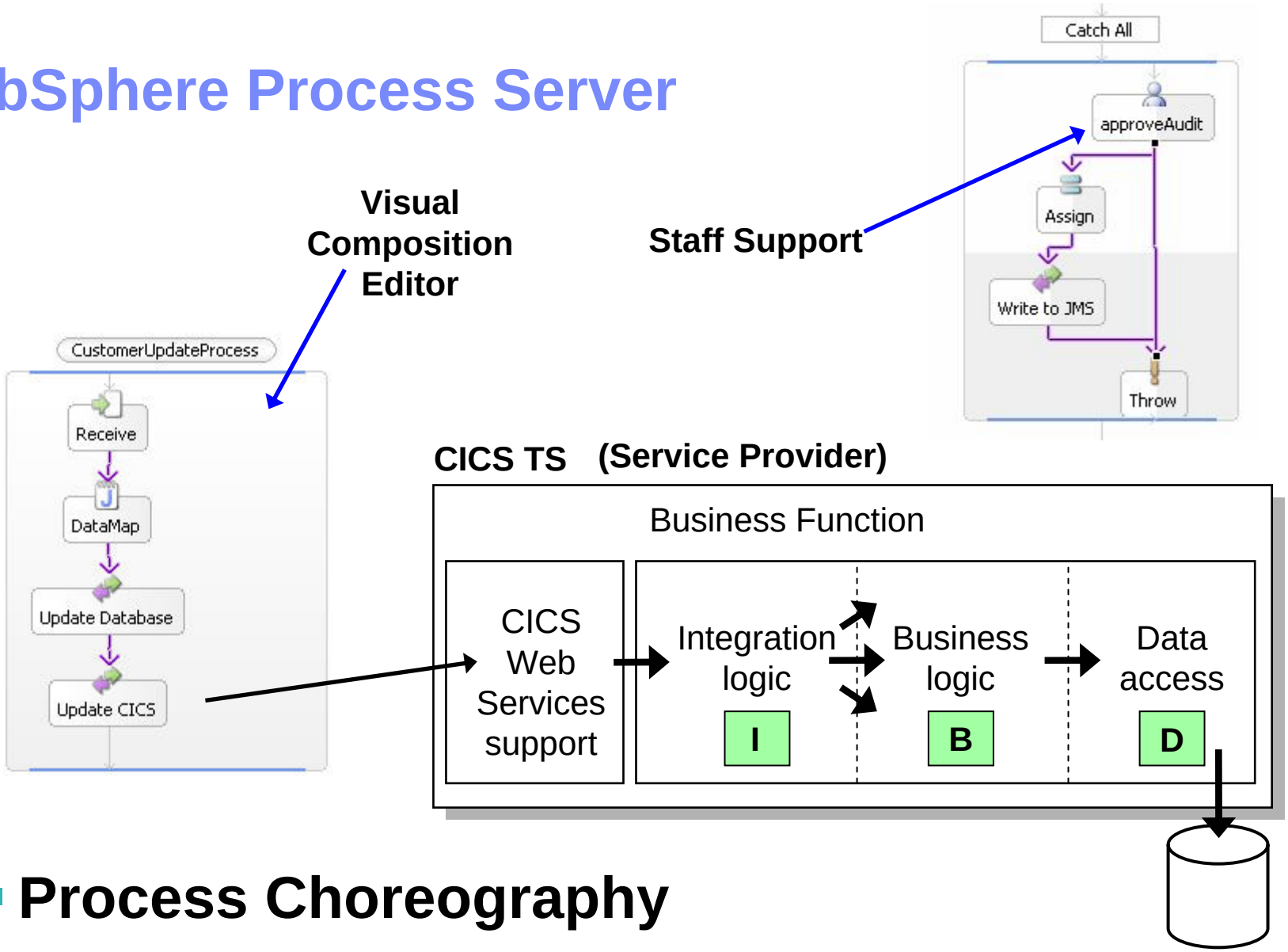


- **Accepts most any input format**
- **Delivers most any output format**
- **Provides transformation services**
- **Provides choreography services**
- **Visual development environment**

Advanced ESB – Message Broker

- **Capabilities**
  - Asynchronous (end-to-end)
  - Can support Synchronous communications
  - Messages moved internally via message queues
  - Inbound and outbound
  - Mediation / Transformation
  - Provides choreography capabilities
  - Easy deployment
- **Security to zSeries**
  - SSL
  - Depends on type of connection
- **Transactional scope**
  - Transactions include Get and Put from queues
- **Interface**
  - Many interfaces
  - Provides single place for all service requests
- **Coupling**
  - Loose to Tight depending on transport used
- **Tools**
  - Visual development environment
    - Rational Application Developer

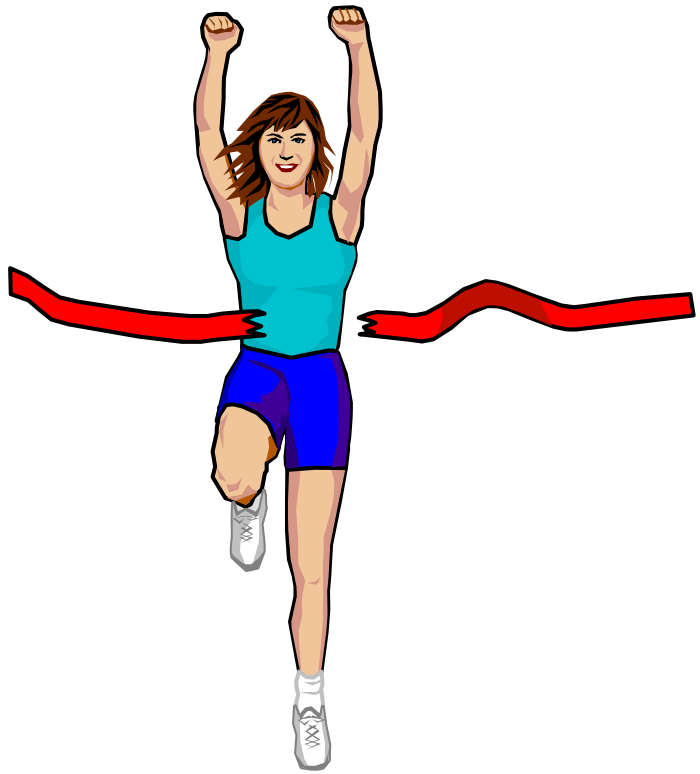
WebSphere Process Server



- **Process Choreography**

Summary

- **CICS Strategic Interoperability Options**
  - Where and when to use
  - Service Granularity
  - Tight vs. Loose Coupling
  - Where to expose your services
  - Where to process XML
  - Security and Transactionality
- **Other SOA Enablers**



Resources

- **IBM CICS Transaction Server for z/OS**
  - CICS TS V3.1 Product documentation
  - Whitepaper “e-business access to CICS: strategic options”
    - **ibm.com**/software/htp/cics/library/cicstsforzos23.html#wpapers
  - Companion whitepaper “Integrating WebSphere Application Server and CICS using the JCA”
    - **ibm.com**/software/htp/cics/library/cicstgv5.html#wpapers
  - SOAP for CICS Feature - **ibm.com**/cics/soap/
  - Redbook “Revealed! Architecting Web Access to CICS
    - redbooks.**ibm.com**/redbooks/pdfs/sg245466.pdf
- **IBM CICS Transaction Gateway –** ibm.com**/software/htp/cics/ctg/**
- **WebSphere Application Server**
  - **ibm.com**/software/webservers/appserv/was/
- **WebSphere MQ**
  - **ibm.com**/software/integration/mqfamily/index.html
- **WebSphere Studio Enterprise Edition**
  - **ibm.com**/software/awdtools/studioenterprisedev/
- **Java Applications in CICS SC34-6238-00**
- **Persistent Reusable Java Virtual Machine Users Guide SC34-6201**
- **Diagnostic Guide: http://www.ibm.com/developerworks/java/jdk/diagnosis/**
- **Performance: www.ibm.com/software/htp/cics/library/whitepapers/java_benchmark_whitepaper.pdf**

Acknowledgements