



Strategic options for extending IBM CICS[®] Transaction Server for z/OS[™] to a Service Oriented Architecture (SOA)

Mark Cocker

CICS Technical Strategy and Planning

Mark_Cocker@uk.ibm.com

Phil Wakelin

CICS Transaction Gateway Technical Planner

Phil_Wakelin@uk.ibm.com

Dermot Flaherty

Lead Architect, Messaging Products and Technologies

Dermot@uk.ibm.com

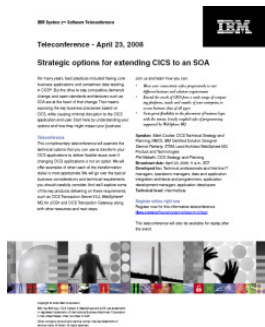
Hursley Lab, IBM United Kingdom Limited

© 2008 IBM Corporation

Acknowledgements

- The following are trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM, CICS, CICS TS, CICS Transaction Server, CICS Transaction Gateway, WebSphere MQ, DB2, WebSphere, z/OS, zSeries, Parallel Sysplex.
- Java, JavaBeans, and all Java-based trademarks and logos 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.
- Other company, product, and service names and logos may be trademarks or service marks of others.

Abstract



- For many years, best practices included having core business applications and sometimes data residing in CICS. But the drive to stay competitive demands change, and open-standards architectures such as SOA are at the heart of that change. That means exposing the key business processes based on CICS, while causing minimal disruption to the CICS application end-user.
- In particular we examine the technical options to transform your CICS applications to deliver flexible reuse, even if changing CICS applications is not an option. We will offer examples of when each of the transformation styles is most appropriate. We will go over the typical business considerations and technical requirements you should carefully consider.
- We will explore some of the key products delivering on these requirements, such as CICS Transaction Server, WebSphere MQ for z/OS, and CICS Transaction Gateway, along with other resources and next steps.

Agenda

- Analyse your integration requirements
- Strategic integration technologies
 - Web services
 - WebSphere MQ
 - CICS Transaction Gateway
- Typical customer deployments
- Conclusion and comparison tables

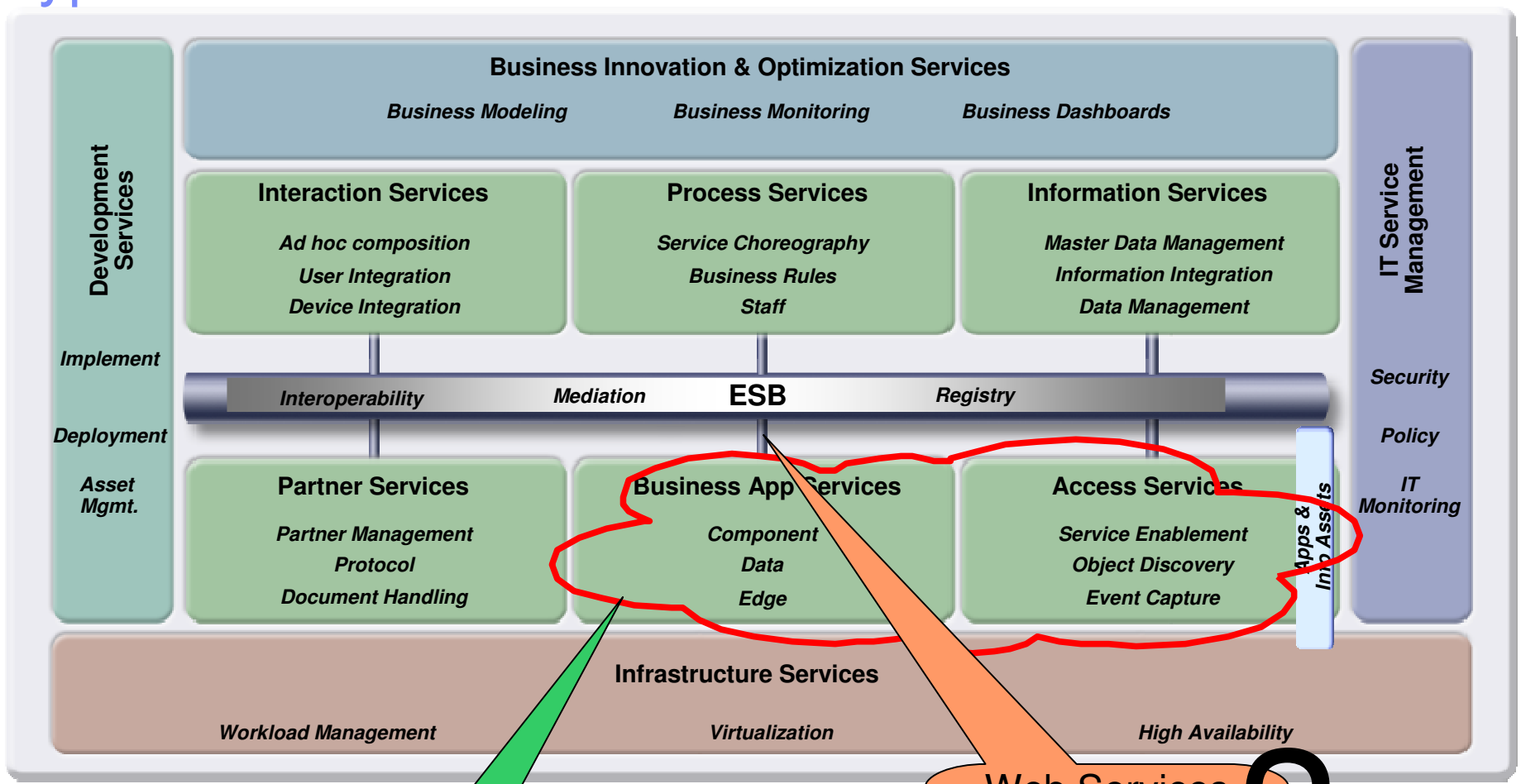
Analyse your integration requirements

- Business considerations
 - Your organization's standards or reference frameworks
 - Your organization's preferred application development environment and tools
 - The availability of skills
 - Time constraints and required development effort for a solution
- Solution technical requirements
 - Security
 - Transactional scope
 - Performance
 - Reliability, availability and scalability (RAS)
 - Granularity, Client-server coupling
 - Synchronous or asynchronous invocation
 - Inbound and outbound capability
 - Data conversion
 - State management
- Application reuse - today are typically delivered across multiple channels

Key technologies driving integration requirements between services in an SOA and CICS

- **Service Oriented Architecture (SOA)**
 - An approach to building multi-tiered systems that delivers application functionality as services to end-user applications or to other services
- **Interaction services**
 - Modern user application, typically simple Web, portals, Wikis, REST, mashups, devices, ...
- **Process services**
 - Modelling, running, monitoring, analyzing and adapting a business process on a continuous base. A business process is preferably executed using workflow concepts and technology.
- **Enterprise Service Bus (ESB)**
 - Intermediaries between service providers and requester providing eg. network location independence, mediation between service interfaces + qualities of service, logging ...
- **Infrastructure services**
 - Industry standardising on TCP/IP as the backbone for physical connectivity and network layering
 - Qualities of service, such as workload management, security, and high availability provided by combination of devices (eg. Routers) + specialist devices (eg. Sysplex Distributor) and higher level protocols built on standards (eg. SOAP)

Typical services in an SOA environment and CICS

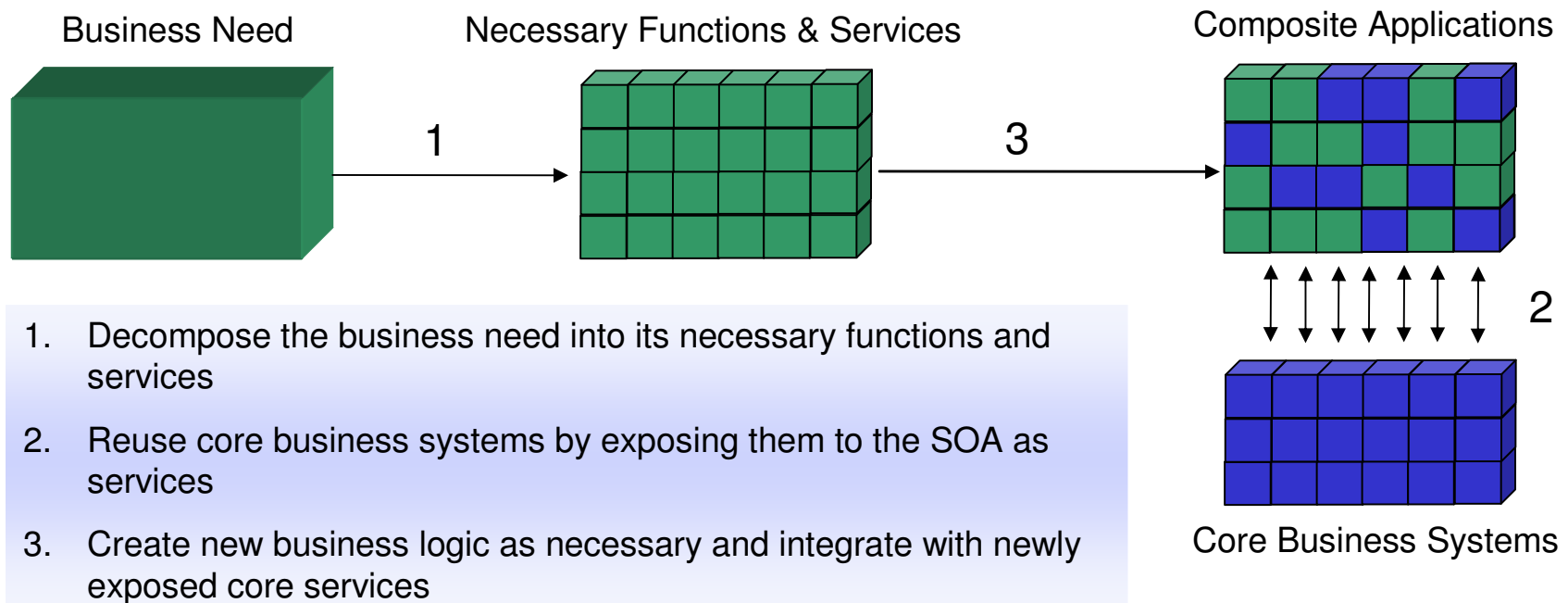


CICS TS

Web Services
CICS TG
WebSphere MQ ?

Why expose CICS applications as services in an SOA?

- Existing CICS applications and the data they manage embody the business applications enterprises have relied upon for years
- Already provides qualities of service and service level agreements required for the business
- Integrating CICS application is low risk and fast

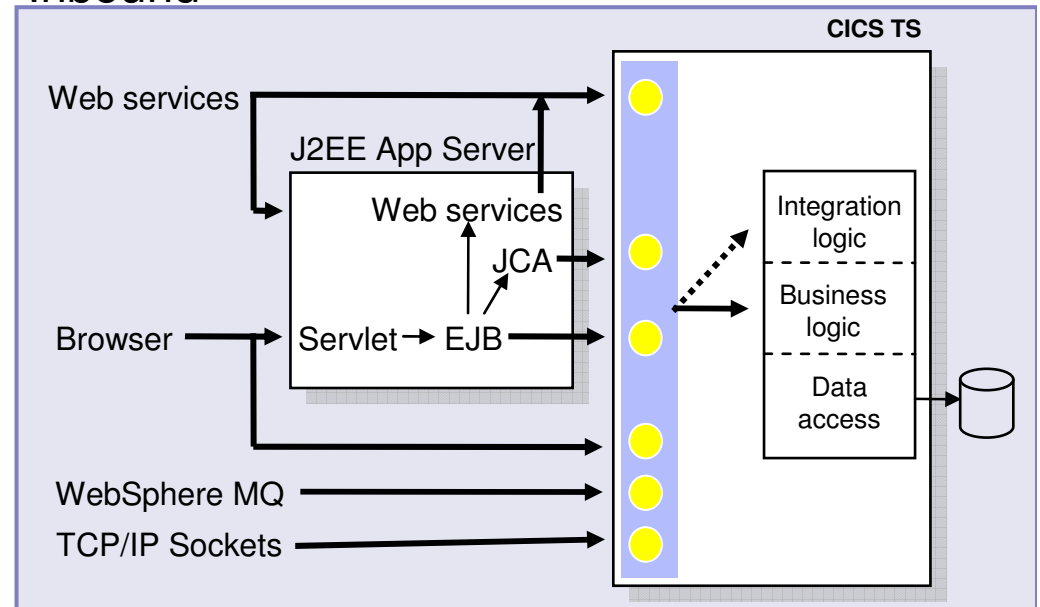


- Decompose the business need into its necessary functions and services
- Reuse core business systems by exposing them to the SOA as services
- Create new business logic as necessary and integrate with newly exposed core services

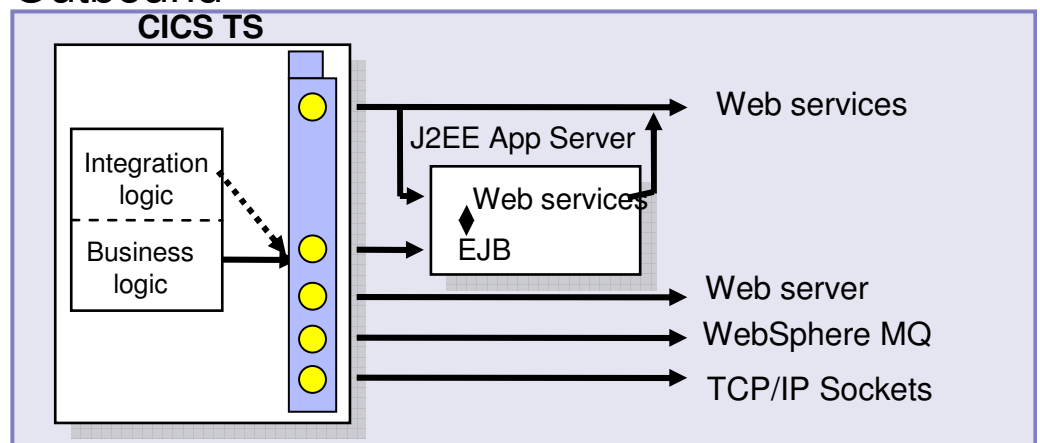
CICS is designed for mixed integration styles and interaction patterns

- Over many years CICS has made available a large choice of protocols, interfaces and APIs to connect to and from CICS applications
- Web services
 - Often used when portability of service or standard interfaces or simplicity are a prime concern
- CICS Transaction Gateway (JCA)
 - Often used when high Quality of Service and synchronous invocation are needed
 - Provides adapter based non-invasive solution
- WebSphere MQ
 - Often used when assured delivery, loose coupling, and/or asynchronous invocation is needed
 - Efficiently handles large data sizes

Inbound



Outbound



What CICS applications can be a service in an SOA

- CICS programs with COMMAREA or CHANNEL interfaces
 - Request and response, similar to remote procedure call
 - Typically stateless
 - Typically CICS manages transactional scope + security
 - COMMAREA typically optimised binary as limited to 32KB size
 - CHANNEL more flexible - eg. mixture of binary, text, XML - as not limited in size

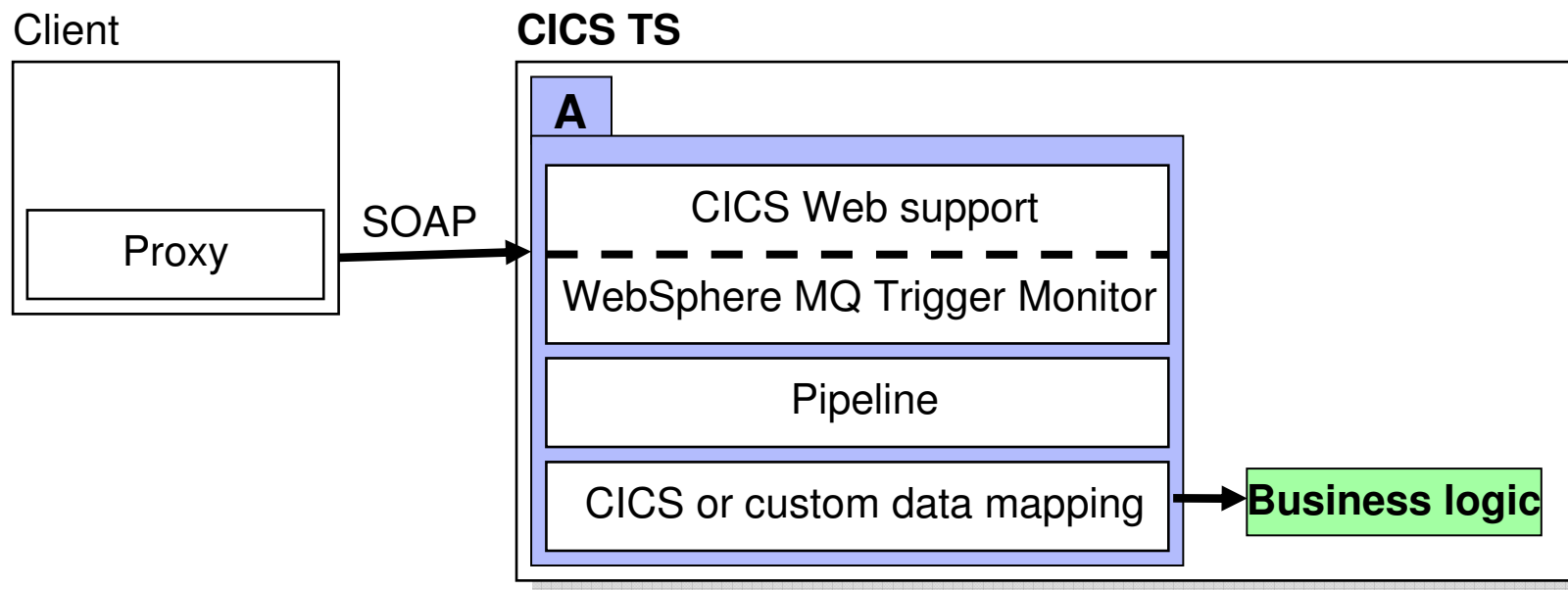
- CICS programs with a terminal-oriented 3270 interface
 - Single interaction or pseudo-conversational
 - Basic Mapping Support (BMS) used to abstract static + dynamic data
 - Transform terminal-oriented application into CHANNEL or COMMAREA using the CICS Service Flow Feature + Rational Developer for System z

Strategic options for extending CICS to an SOA

- Standard architectures provide comprehensive development tools and runtime support in CICS
 - ① – Web services over SOAP (Simple Object Access Protocol)
 - ③ – JCA (J2EE Connector Architecture)
 - Enterprise JavaBeans (J2EE EJB)

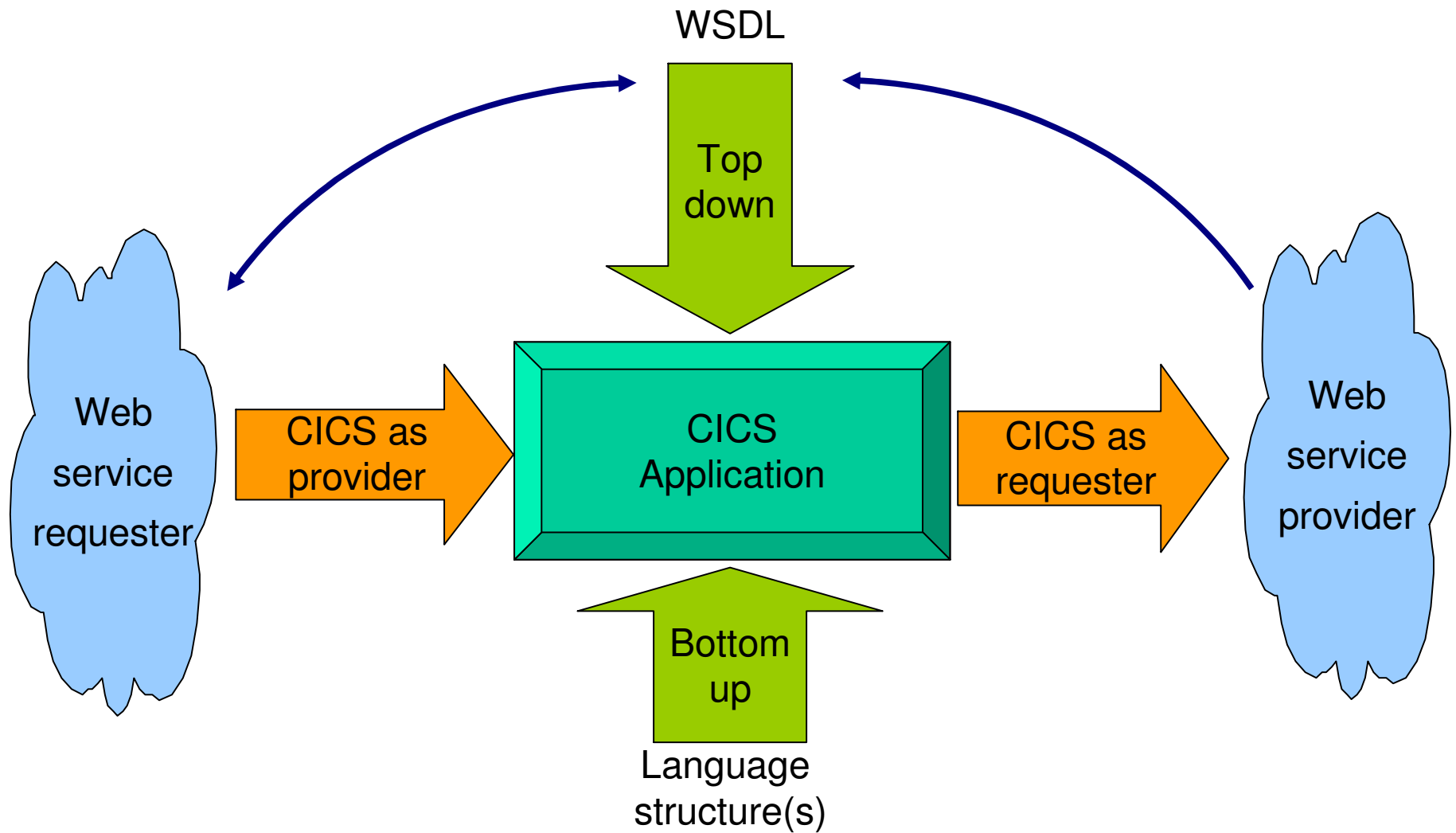
- Standard transports are suitable for use by applications that require greater control of the protocol and do not need the development tools or qualities of service provided by the standard architectures. These applications will assume more responsibility for systems management, security, and recovery
 - HTTP (HyperText Transfer Protocol)
 - ② – WebSphere MQ (MQ APIs or JMS - Java Messaging Service)
 - TCP/IP sockets

1 Web services support in CICS TS V3

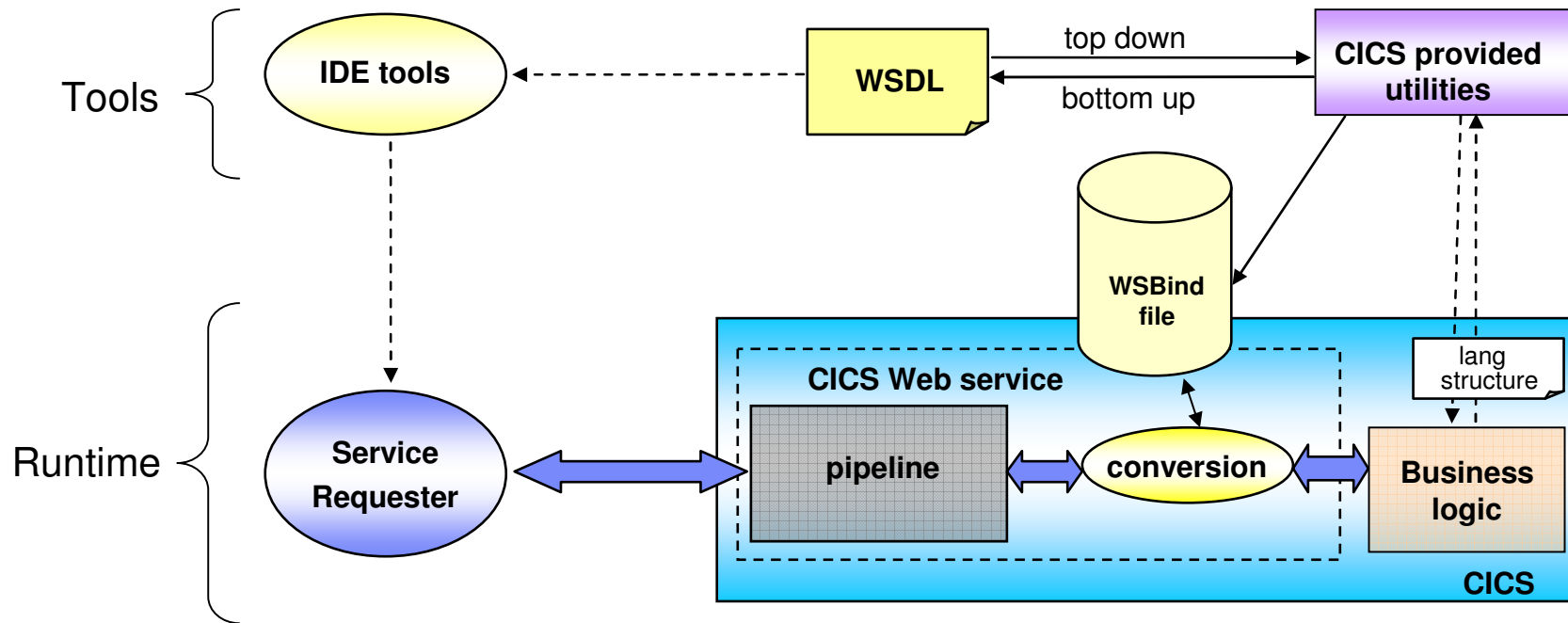


- *CICS provides direct access from Web service requesters and to Web service providers*
- WSDL (Web Services Definition Language) is used to describe the service
- Tools are typically used to generate the client proxy and message adapter

1 Web services terminology



1 Web services support tooling and runtime



1 Web services support in CICS TS V3 capabilities

- 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 System z
 - 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
 - or XML body provided in CONTAINER
- 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 (CICS TS and Rational Developer for System z)
 - Broader language support (CICS TS)
 - Flexible COMMAREA mapping (RDz)
 - Batch converter generation

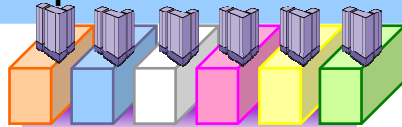
2 Universal Messaging Backbone with WebSphere MQ

- Extends the reach of your Enterprise to deliver Business data with a range of QoS for processing by CICS

Support for virtually every commercial IT System

80+ platform configurations

- AIX
- Linux x86, x86-64
- Linux POWER
- Linux for System z
- Windows x86, x64
- Solaris x86 / SPARC
- iSeries i5/OS
- HP-UX Itanium/PA-RISC
- HP NonStopServer
- HP OpenVMS



80+ platforms

Exploit System z for messaging powerhouse

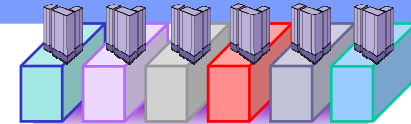
- Unique code-base designed natively for tight z/OS integration
- Runs as formal MVS sub-system
- Leverage Parallel Sysplex to provide Shared Queues for continuous availability, scalability and capacity
- Support for event-driven publish-subscribe delivery



z/OS

Enable popular applications and environments

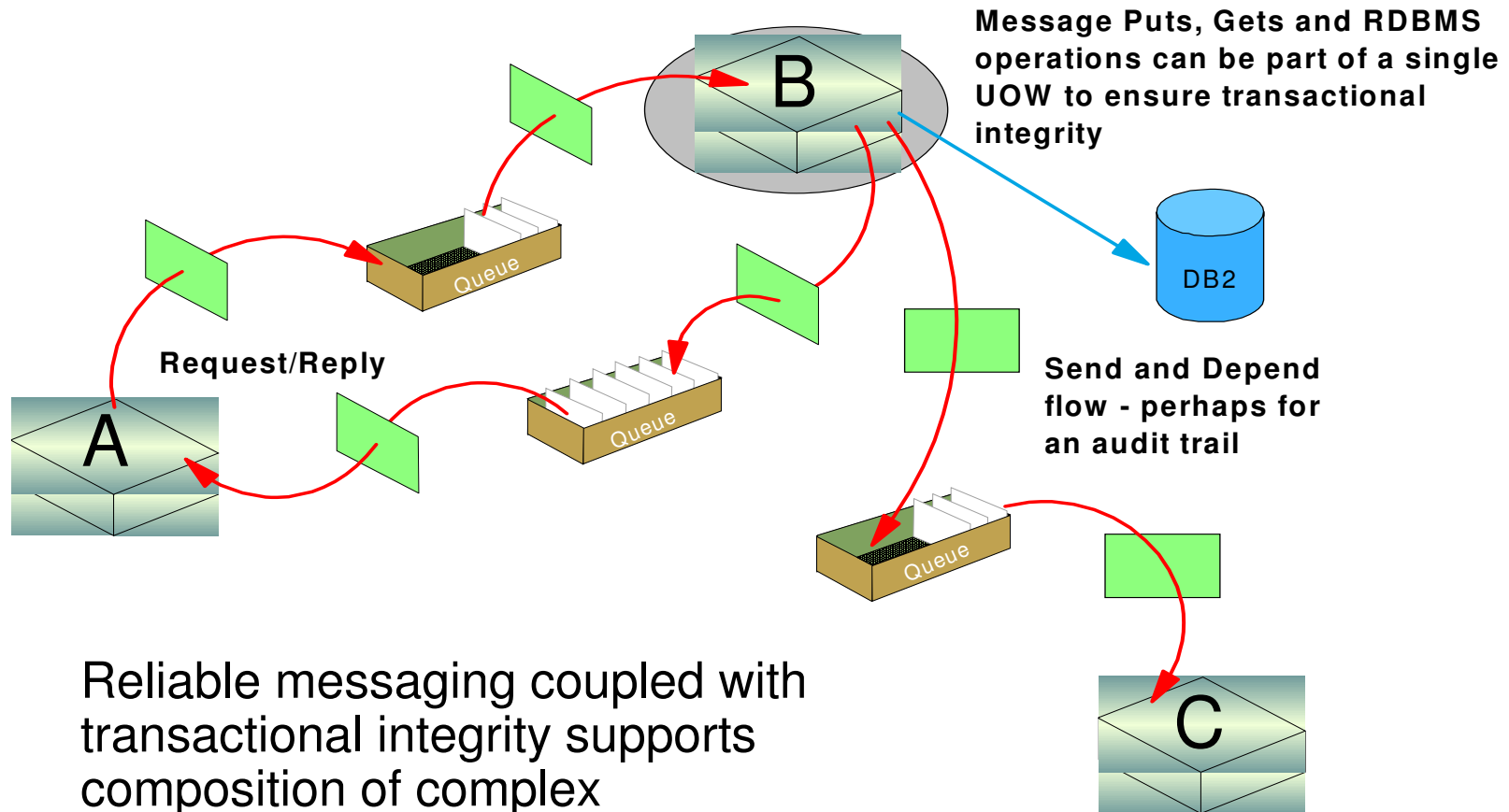
- Built-in bridge for CICS, IMS, Batch/TSO
- JCA support to provide JMS services for JEE App Servers
- Support for Databases DB2, Oracle etc
- Support for External Transaction Coordinators
- Over 950 ISVs with enabled Packaged Applications



Applications

← UNIVERSAL MESSAGING BACKBONE →

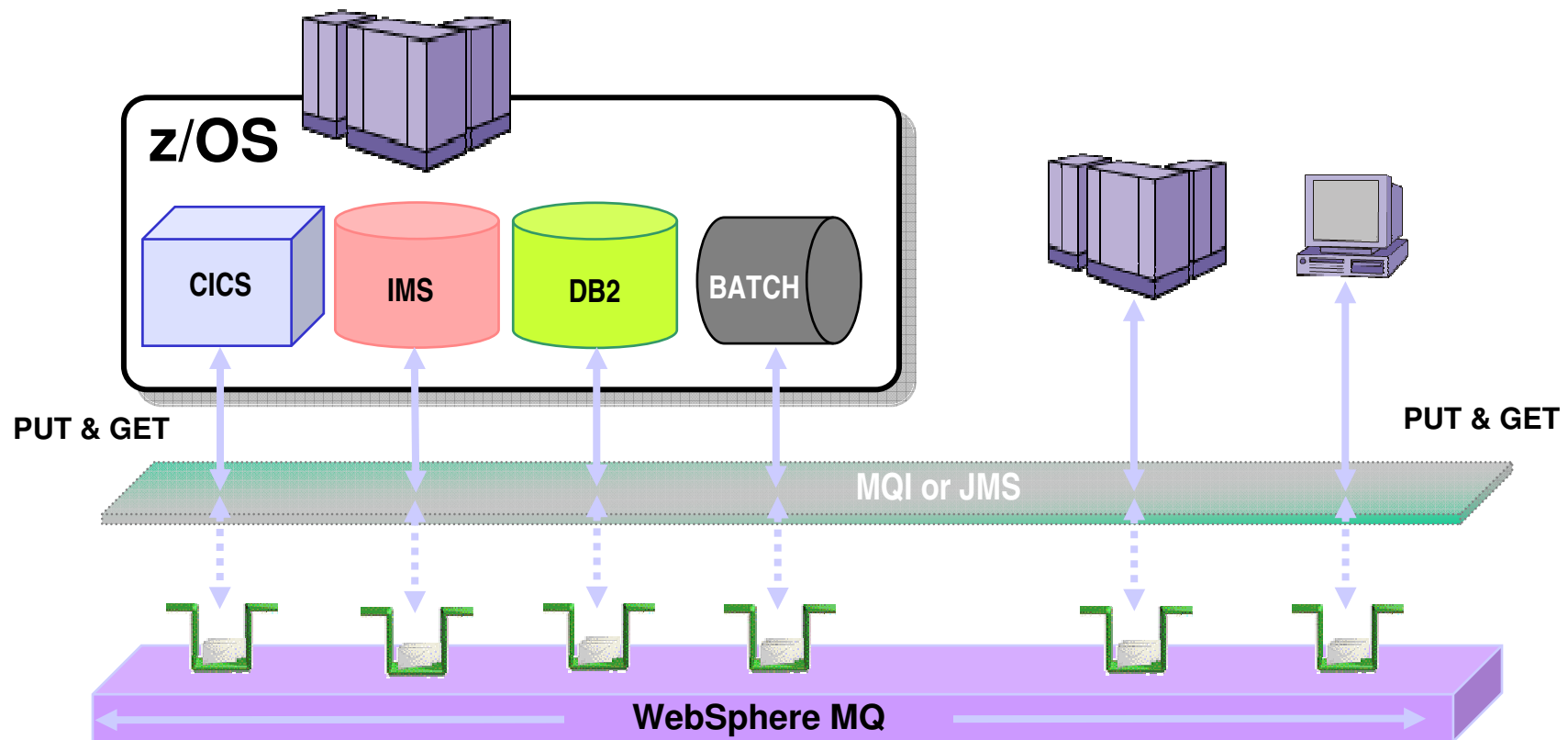
2 WebSphere MQ in a nutshell



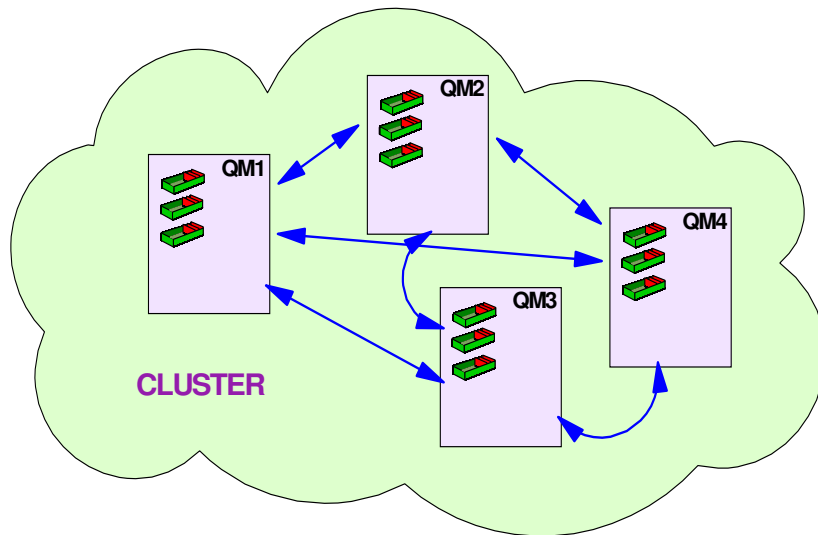
Reliable messaging coupled with transactional integrity supports composition of complex applications to fully support parallel processing.

2 WebSphere MQ access to CICS

- Can access business data from CICS from ANYWHERE – either on z/OS or elsewhere in the Enterprise either explicitly via the MQI from CICS or implicitly via the MQ-CICS Bridge



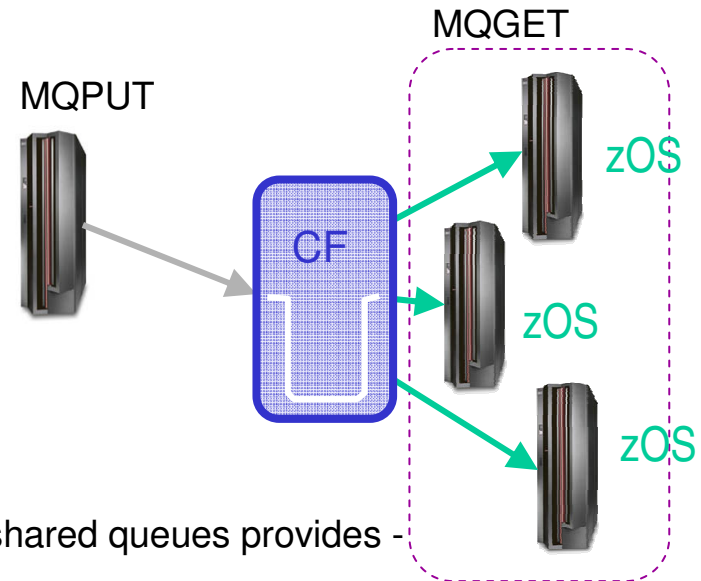
2 WebSphere MQ – Clustering and Shared Queues



MQ clustering provides -

Automatic advertising of queues within a cluster to reduce MQ administration

Workload balancing by spreading messages destined for a single logical queue over multiple queues of the same name on different Queue Managers



MQ shared queues provides -

Queues not owned by Qmgrs
Automatic load-balancing
Scalable throughput
Sysplex-wide access to shared message data
VIPA support

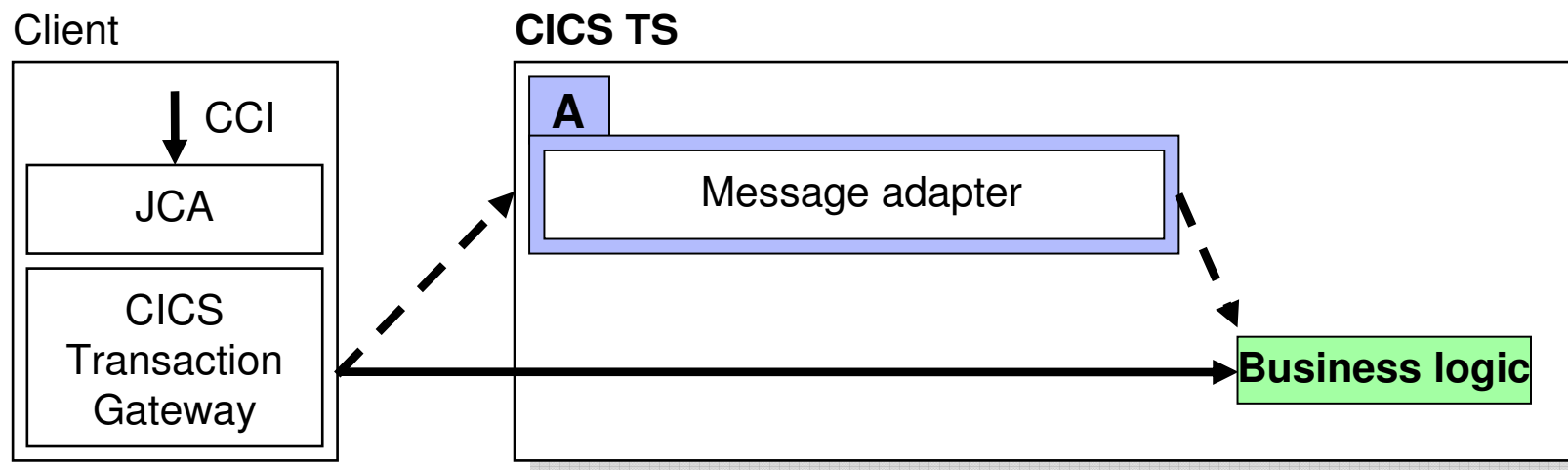
Queue Sharing Group

Failure isolation
Peer Recovery for failing Queue Managers
24x7 Availability

2 WebSphere MQ and the wider Web

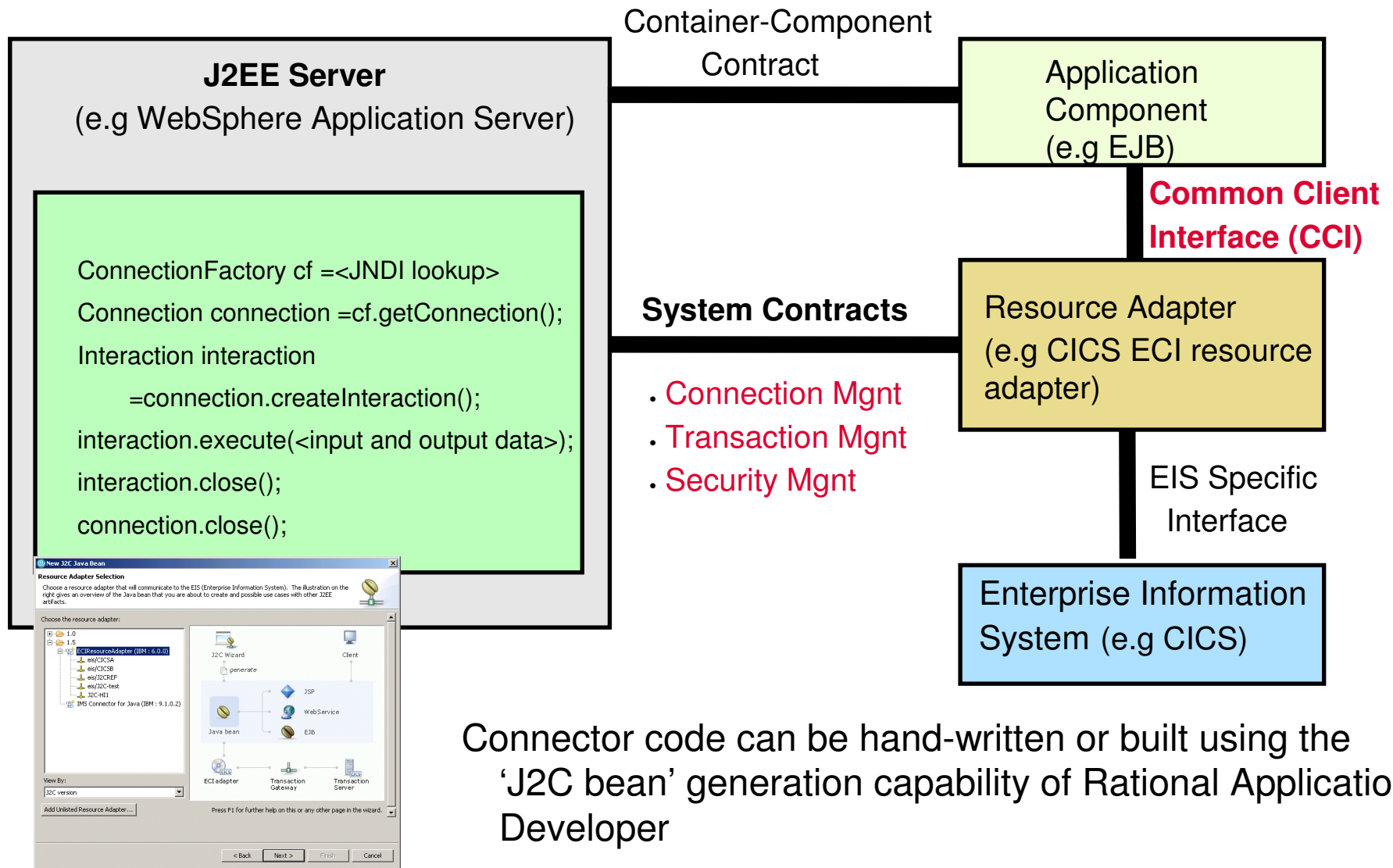
- SOAP Transport capability available since WMQ V6
 - Uses JMS message formats (RFH2 etc.)
 - Options to control other WMQ details such as queue manager, persistence etc
 - Interoperable with WAS and CICS can use WMQ as a SOAP/JMS transport
- A Standardisation effort is underway to make possible JMS provider-neutral deployment of SOAP/JMS services
 - Initial participants: IBM, BEA, Sonic, and TIBCO
 - In Oct '06 the specification reached v1.0 RC1
 - Five more vendors were invited to provide feedback (WebMethods and WSO2 subsequently joined the core team)
 - Specifications have been submitted to W3C and IETF
- SOAP/JMS was one of the most requested features by participants at W3C Enterprise Web Services Workshop in Feb. 2007
- RESTful HTTP-MQ Bridge support
 - Supports zero-footprint client requirements
 - Provides simple mapping of MQ to HTTP verbs
 - Supportpac MA0Y (delivered with WMQ V7) provides servlet for J2EE container
 - Supportpac MA94 provides native MQ listener on Distributed platforms
- MQ Service Definition allows standalone MQ Applications to be described as Services
 - To be inventoried and catalogued in Service Registry
 - To be re-used as Services in composite SOA applications
 - To be managed and traced with SOA tools
 - Service definition and SOAP binding available in supportpac MA93

3 CICS Transaction Gateway (JCA connector)



- The J2EE Connector Architecture (JCA) defines the common client interface (CCI) for the client to use to drive interactions with enterprise information systems such as CICS
- The CICS Transaction Gateway provides the JCA access to CICS


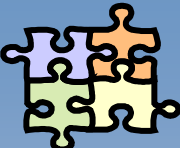

3 J2EE Connector Architecture (JCA)



Connector code can be hand-written or built using the 'J2C bean' generation capability of Rational Application Developer

3 CICS Transaction Gateway

High-performing, security-rich and scalable J2EE standards-based access to CICS

 <p>Primary connector into CICS</p> <p>Interconnectivity</p> <ul style="list-style-type: none"> High performing and scalable inbound connector to CICS applications Provides connectors to COMMAREA, container and 3270-based CICS applications 	 <p>Java and non-Java API's</p> <p>Interfaces</p> <ul style="list-style-type: none"> Standard JCA interface is strategic and provides best Qualities of Service Base Java, C, C++, COBOL and COM interfaces are supported but stabilized 	 <p>WebSphere, CICS and others</p> <p>Integration</p> <ul style="list-style-type: none"> Every in support CICS server on every platform to WebSphere SOA foundation servers 5 SNA servers (AIX, Windows, Linux)
--	---	--

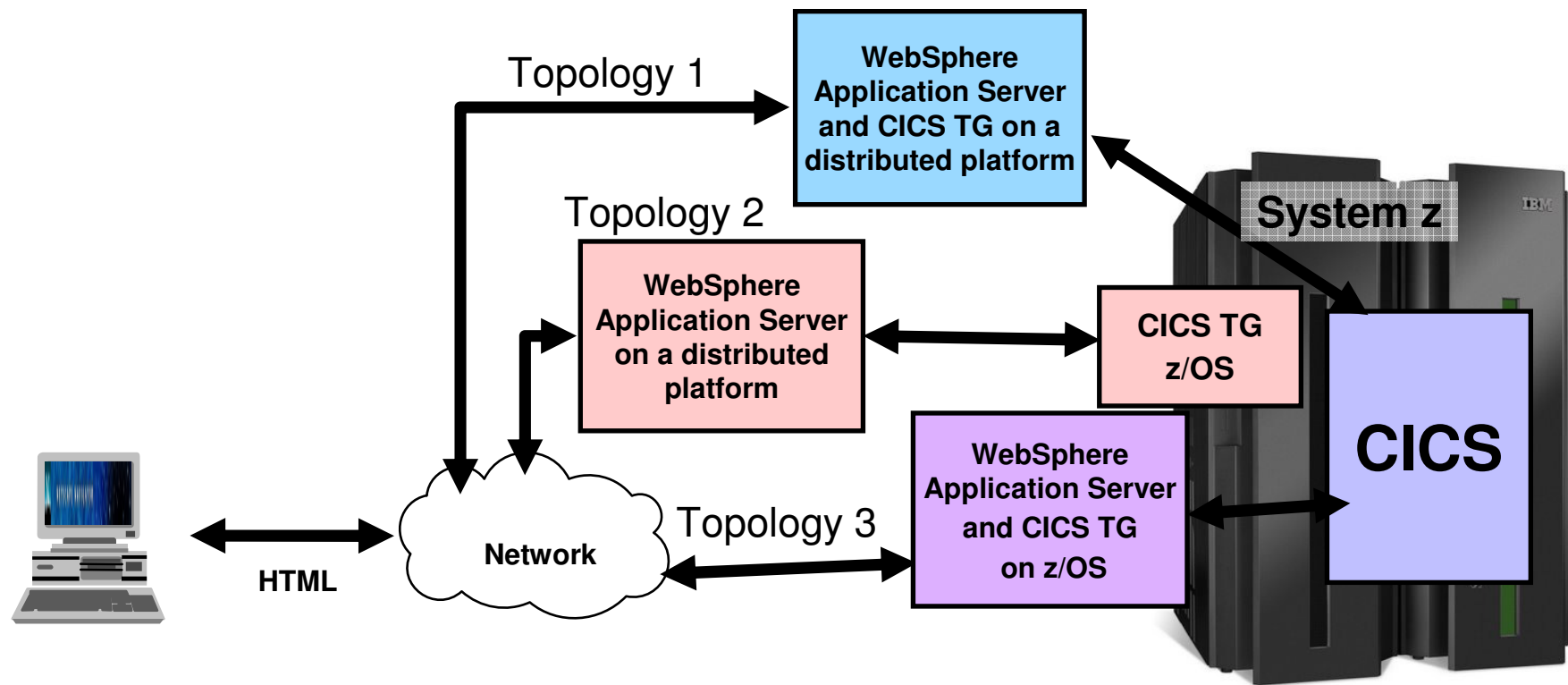
<table border="1"> <tr><td>Dist</td><td>Dist</td></tr> <tr><td>WAS</td><td>CICS</td></tr> <tr><td colspan="2">CICS TG</td></tr> </table>	Dist	Dist	WAS	CICS	CICS TG		<table border="1"> <tr><td>Dist</td><td>z/OS</td></tr> <tr><td>WAS</td><td>CICS</td></tr> <tr><td colspan="2">CICS TG</td></tr> </table>	Dist	z/OS	WAS	CICS	CICS TG		<table border="1"> <tr><td>Dist</td><td>z/OS</td></tr> <tr><td>WAS</td><td>CICS</td></tr> <tr><td colspan="2">CICS TG</td></tr> </table>	Dist	z/OS	WAS	CICS	CICS TG		<table border="1"> <tr><td>z/OS</td><td>z/OS</td></tr> <tr><td>WAS</td><td>CICS</td></tr> <tr><td colspan="2">CICS TG</td></tr> </table>	z/OS	z/OS	WAS	CICS	CICS TG	
Dist	Dist																										
WAS	CICS																										
CICS TG																											
Dist	z/OS																										
WAS	CICS																										
CICS TG																											
Dist	z/OS																										
WAS	CICS																										
CICS TG																											
z/OS	z/OS																										
WAS	CICS																										
CICS TG																											
CICS TG for Multiple Platforms		CICS TG for z/OS																									

Good ← Qualities of Service → Best

Supported Platforms

- IBM's **flagship z/OS**
- Linux on Intel, POWER, & System z
- AIX, HP-UX and Solaris UNIX support
- Windows: 2003, XP, Vista

3 JCA platform and topology options

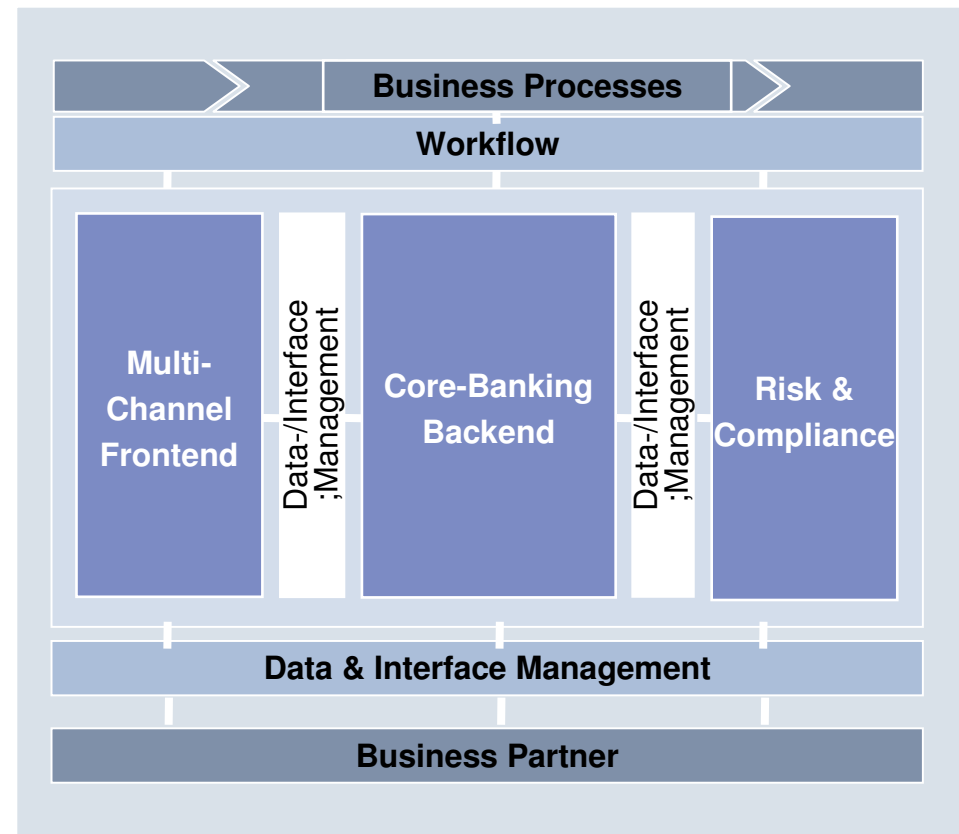


3 CICS Transaction Gateway capabilities

- Capabilities
 - Inbound only
 - Synchronous and asynchronous (within a client request!)
- Security to z/OS
 - SSL to the Gateway daemon or CICS TS.
 - User ID and password
 - Asserted identity options for z/OS
- Transactional scope – see next chart
 - Local transactions
 - (XA) Global transactions
- Interface
 - Language structure (COBOL, C, C++, PL/I) in a COMMAREA or CHANNEL
- Coupling
 - Medium – typically the COBOL formatted records are exposed to the client
 - RAD J2C tools can abstract the COMMAREA interface by generating easier to use proxies

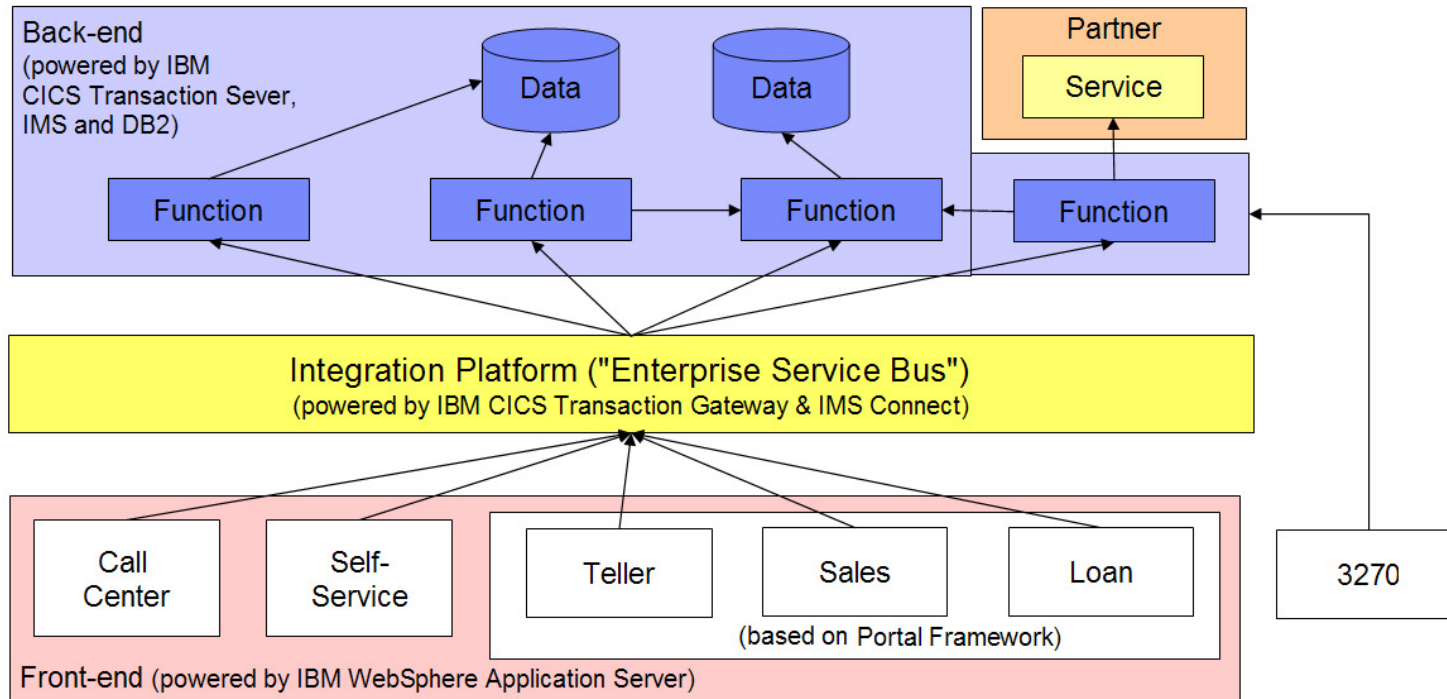
Typical customer deployment: A large European Retail Banking Solution

- Complete solution portfolio to support all kinds of business processes
- Scalability: 229 banks
- Banks from < 300 M € up to > 30 Bn € Total Balance Sheet
- High-end processing of large transaction volumes
- 30 Bn technical transactions in 2004
- 7 x 24 Operating & Support
- Real Time Core-Banking Solution
- High-Availability
- Average availability of centralized applications during last 3 years was 99.94%



A large European Retail Banking Solution

Solution Approach - A Service Oriented Architecture



IBM CICS Transaction Server

- Transaction per year = 11Bn.
- Peak day Transactions = 74M

IBM CICS Transaction Gateway

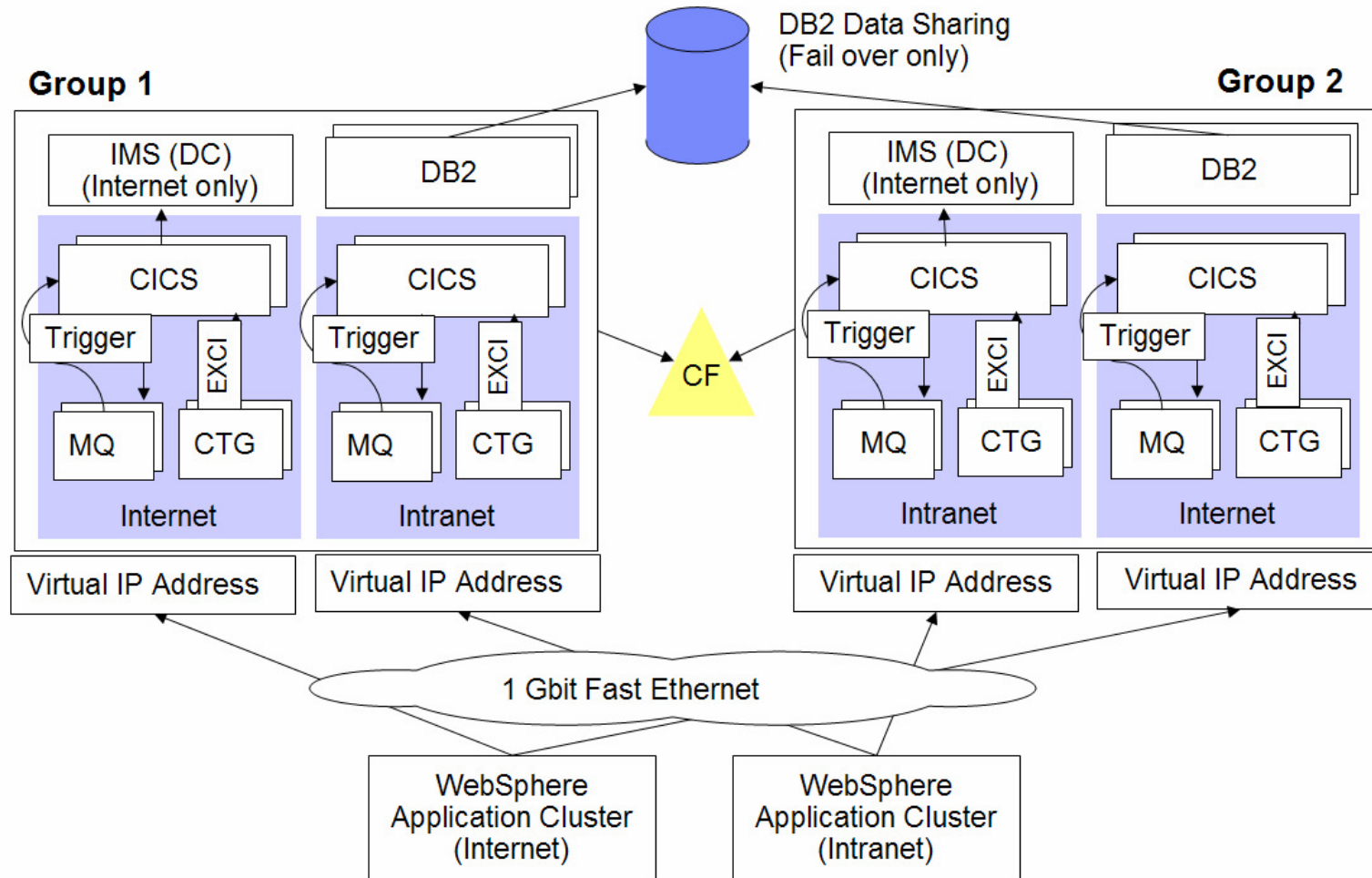
- Transaction per year = 1.7Bn.
- Transaction on peak day = 12.5M
- Over 500 Transaction per second

IBM WebSphere Application Server

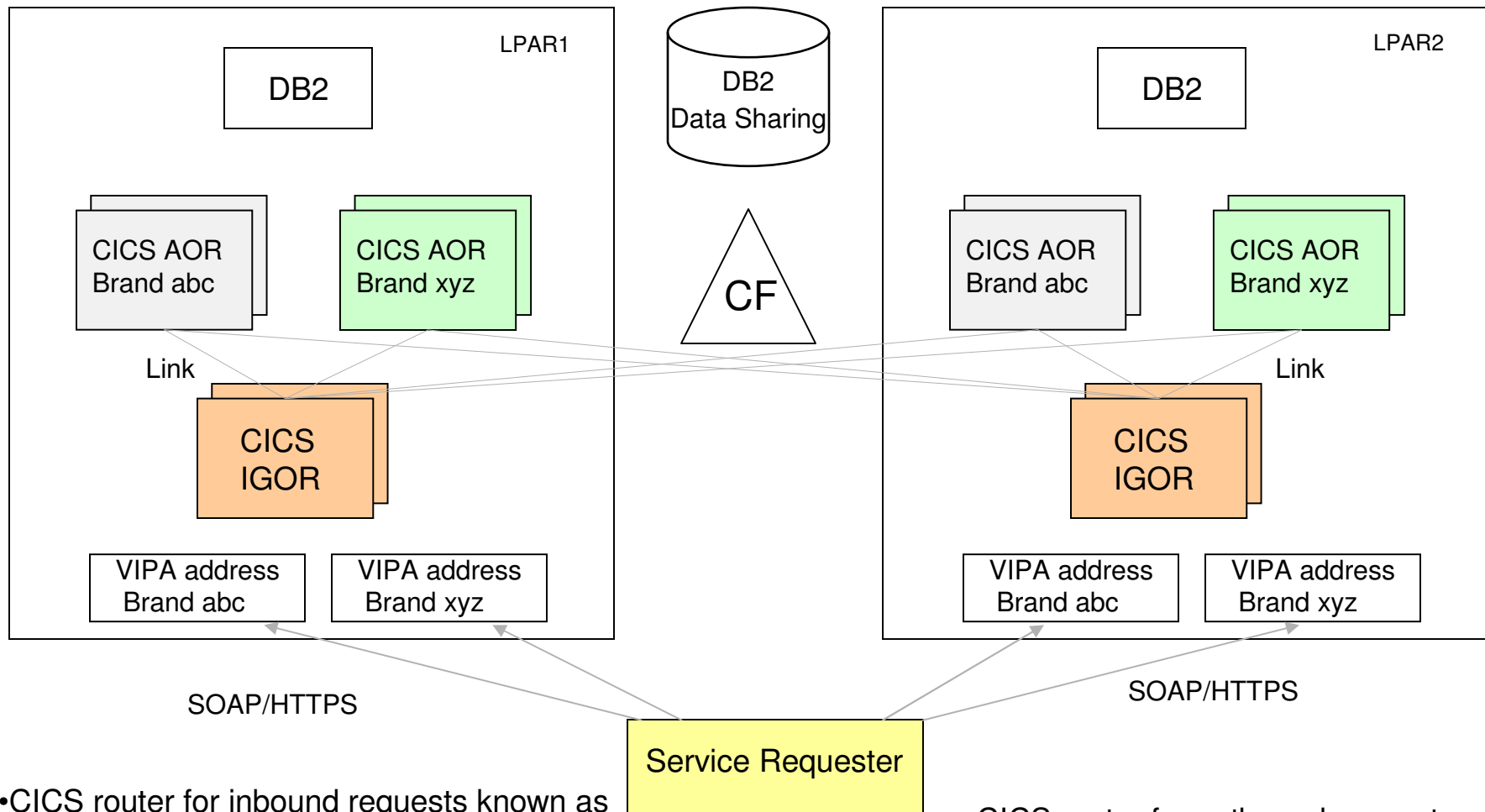
- Web Application in Production = 90
- Capacity in CPU's = 600

A large European Retail Banking Solution

A Service Oriented Architecture that fully exploits the QoS of zSeries!



Typical customer deployment: QoS customer benchmark of an SOA solution with Web services



- CICS router for inbound requests known as 'Inbound Gateway Owning Region' (**IGOR**)
- IGOR runs CICS **wrapper** program ('meet in the middle' approach)

- CICS router for outbound requests known as 'Outbound Gateway Owning Region' (**OGOR**)

Summary of strategic options for extending CICS to a Service Oriented Architecture (SOA)

- CICS applications can be full participants in an SOA environment
- To achieve the best reuse of your CICS programs and to support multiple access options, architect clear and concise business logic interfaces of the right granularity for the solution
- CICS and other tools help transform and aggregate your programs to provide the right level of granularity for your service requesters
- CICS provides a broad choice of access options based on TCP/IP, open standard connectivity architectures, and transport mechanisms
- In an SOA environment these are
 - Web services, CICS Transaction Gateway, and WebSphere MQ
- Business considerations and solution technical requirements should be compared with the capabilities of each access option to select the most appropriate
- Architects, CICS application developers, and CICS system programmers will find it easy take advantage of new access options and make existing applications available for reuse

Comparison and recommendation

Architecture/transport	Capabilities	Security to zSeries	Transactional scope	Interface	Coupling
1. Web services	Inbound and outbound Synchronous (HTTP) Asynchronous (WMQ)	Web services SSL User ID + password	Web service Sync on return	COMMAREA CONTAINER	Low
2. JCA	Inbound only Synchronous Asynchronous	SSL User ID + password Thread identity	Local Global Sync on return	COMMAREA CONTAINER	Medium
3. WebSphere MQ	Inbound and outbound Asynchronous Assured delivery	SSL User ID + password	Sync on return	COMMAREA WebSphere MQ API	Low when used with ESB or WS. Medium if WMQ only.

Architecture/transport	Description	Positioning	Recommendation
1. Web services	Comprehensive W3C standards for messaging over the Web supporting SoA to and from CICS	Industry-wide open standard integration technology that includes CICS connectivity. Improving QoS, features and performance	<i>Establish plans to transform CICS apps so they can participate in a SOA pattern with Web services</i>
2. JCA	Lightweight J2EE standard for calling CICS and other EIS's	Widely adopted precision CICS connectivity with highest qualities of service today	<i>Continue to exploit JCA and CICS TG and use within an SOA and ESB</i>
3. WebSphere MQ	Comprehensive industry standard for assured messaging	Widely adopted B2B integration technology that includes CICS connectivity	<i>Continue to exploit WebSphere MQ for optimised messaging and flowing Web services</i>

References

- White papers

www.ibm.com/software/htp/cics/tserver/v32/library/index.html#wpapers

- Options for integrating CICS applications in an SOA
- Deploying CICS Web services to preserve IT investments in the banking industry
- Increase the value of CICS applications with WebSphere MQ
- Integrating WebSphere Application Server and CICS using the CICS Transaction Gateway

- IBM Redbooks

www.redbooks.ibm.com/cgi-bin/searchsite.cgi?query=cics

- Architecting Access to CICS within an SOA (SG24-5466-05)
- Implementing CICS Web Services (SG24-7206-02)
- CICS Transaction Gateway for z/OS (SG24-7161-00)

- Product information

- IBM CICS www.ibm.com/software/htp/cics
- IBM CICS Transaction Gateway www.ibm.com/software/htp/cics/ctg/
- IBM WebSphere MQ www.ibm.com/software/integration/wmq

Thank You for Joining Us today!

Go to www.ibm.com/software/systemz to:

- ▶ Replay this teleconference
- ▶ Replay previously broadcast teleconferences
- ▶ Register for upcoming events