# Extending Your Mainframe for More Business Value

Extend Connectivity With a Mainframe Communications Backbone

## Business Challenge

Our payments business is a key source of revenue, but it is too costly to maintain the connections

Service Oriented Finance CIO

A Communications Backbone can solve this problem

IBM

# Providing Application-to-Application Connectivity Can Be Complicated

- System Platforms

- Programming Models

Asynchronous Messaging | Synchronous RPC | Publish/ Subscribe

- Programming Languages

RPG | COBOL | .net | Java

- Transport Protocols

Web Services | WebSphere MQ | JMS | FTP | TCP/IP Multicast | HTTP | SMTP

- Standards & Message Formats

ACORD | HIPAA | ebXML | COBOL Copybook | SWIFT | EDI-X.12 | Word/Excel/PDF | Custom Formats | XML | IFX | AL3 | EDI-FACT | HL77

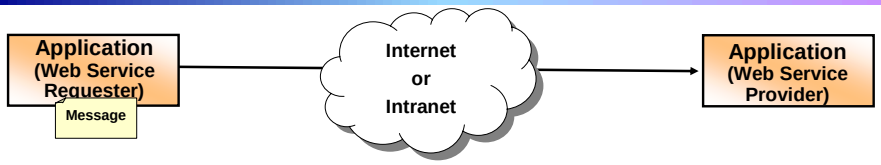
- Error Recovery



# How to Provide Application-to-Application Connectivity

- Installed environments are very diverse
  - No single technology can provide the all of the required power and flexibility
- Use a combination of middleware technologies as needed
  1. **Web Services**
     Standards-based, heterogeneous, Internet-based exchanges
  2. **Asynchronous Messaging**
     Adds reliability, assured delivery, application de-coupling
  3. **Mediation Broker**
     Adds services to transform and enrich information as it flows from one application to another
- Implementations of these technologies is known as an **Enterprise Service Bus**

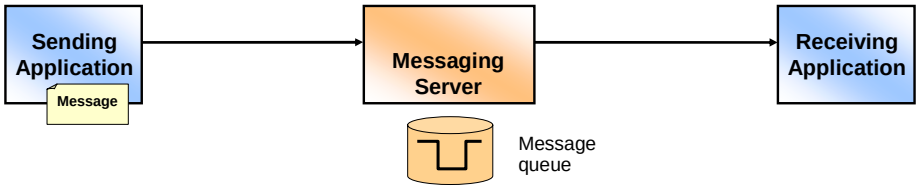
# Web Services Provide Simple Point-to-Point Connectivity




- Advantages
  - Almost every platform supported
  - Standards-based, works across the internet
- But there are considerations…
  - The requester and provider must be running at the same time
  - No infrastructure for managing overall web services
- Mainframe supports web services via WebSphere Application Server, CICS, and IMS SOAP Gateway



# Message Queues Provide Greater Flexibility with Asynchronous Messaging



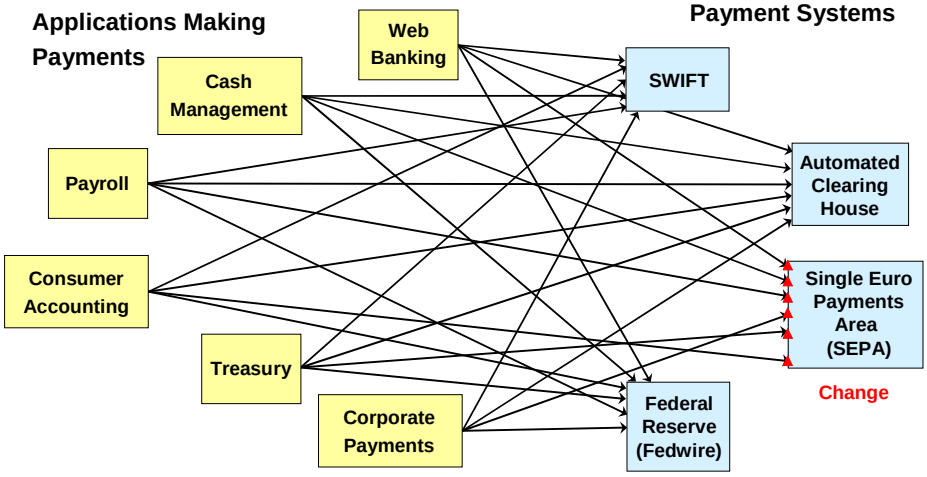

- Sender and receiver do not need to run at same time
  - Put and get messages from queues
- Reliable, assured delivery
- Sender and receiver can process messages at different rates
- Message servers can be networked together
  - Messages automatically arrive at named destination queue
- Mainframe supports messaging via WebSphere MQ and WebSphere Application Server (JMS)

# Connect Applications Point-to-Point with WebSphere MQ

- Connects to virtually everything
  - Over 80 platform configurations
  - Uses IBM Message Queuing Interface (MQI), Java Message Service (JMS), or SOAP/JMS
  - Bridges Web 2.0 AJAX client applications to the WebSphere MQ queues using RESTful interfaces
- Very simple API (put/get) for all main programming languages: C++, C#, Visual Basic, .NET, COBOL, Java
- The de facto standard for asynchronous messaging
  - 42% of z/OS customers have WebSphere MQ
  - 90% of the Fortune 100 businesses have WebSphere MQ
  - 60% of the Fortune 500 businesses have WebSphere MQ
  - 66% of North American and European banks
  - Banking clients move transactions worth $35 trillion/day
  - Government clients move 675+ million messages/day

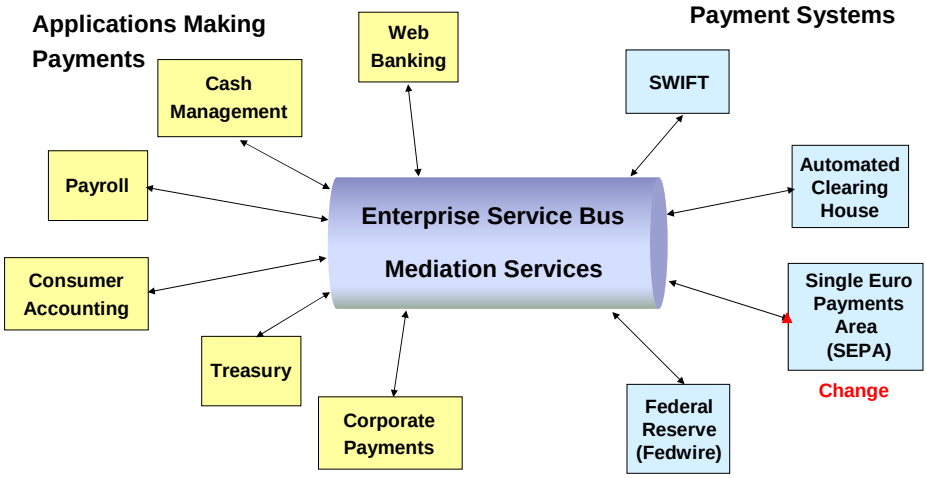


# However, Point-to-Point Connectivity Can Be Costly to Maintain




- Services are tightly coupled to one another
- One change requires many other changes

## An Enterprise Service Bus Reduces Costs By Providing Centralized Mediation Services




- A change requires only one change in the ESB mediation services
- Services can be created and maintained independently



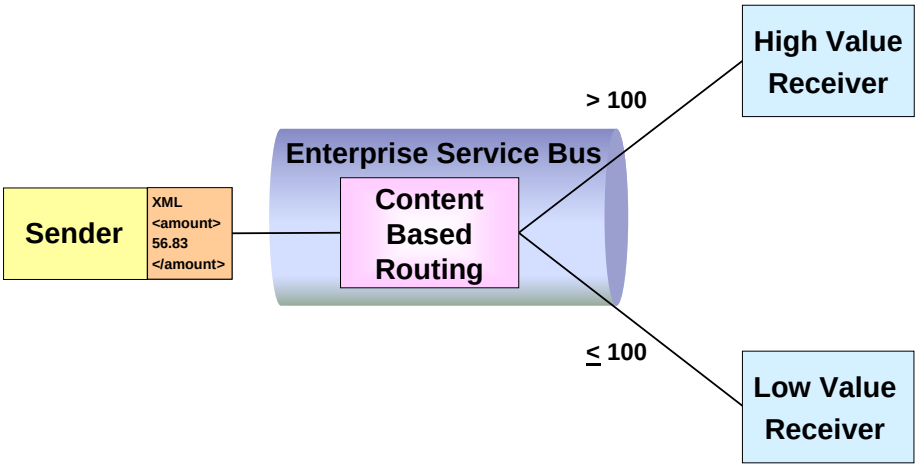
## Case Study – Analysis Showed Benefit of Using WebSphere Message Broker for Enterprise Integration

- The ESB on z/OS solution offered these benefits over the custom point-to-point connection option over the 5-year period:
  - 62% reduction in solution build cost
  - 73% reduction in on-going code maintenance of the integration solution
  - 42% reduction in infrastructure administration
- For an investment of $2.5M in WebSphere software, the company would realize a benefit of **$165M** over a 5-year period
  - Resulting in an ROI of **6,715%**

Source: High-level analysis for a large U.S. Health Insurance Company using IBM's Business Value Assessment (BVA) model, 2006

# Mediation Service: Content-Based Routing

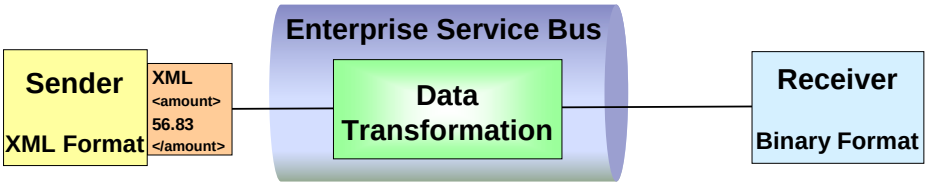
**Example: Route payment based on payment amount**






# Mediation Service: Data Transformation

**Example: Transform XML to binary format**

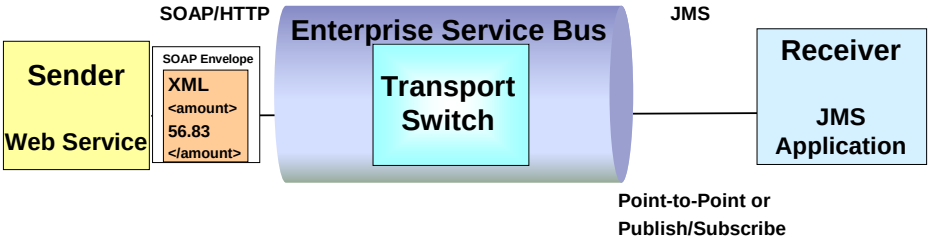



Other Common Transformations

- One XML schema to another XML schema
- Industry specific transformations, e.g., IFX to SWIFT

# Mediation Service: Transport Switching

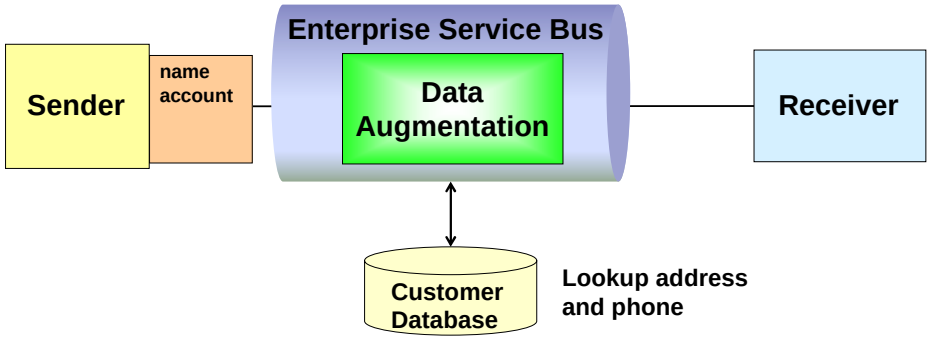
**Example: Switch from SOAP/HTTP to a JMS message**





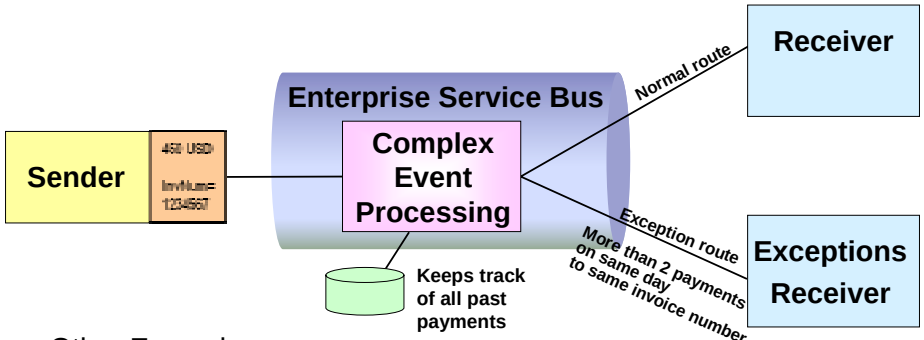

# Mediation Service: Data Augmentation

**Example: Add customer information from an external database**



# Complex Event Processing

**Example: Fraud detection and alerting**

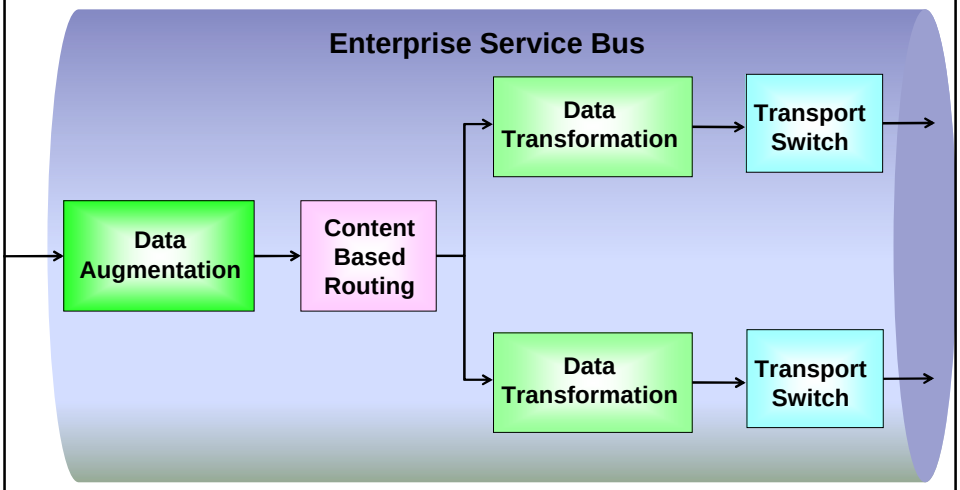



Other Examples

- Enforcement of regulatory constraints
- Periodically report aggregate payments
- Service level agreement monitoring and notification

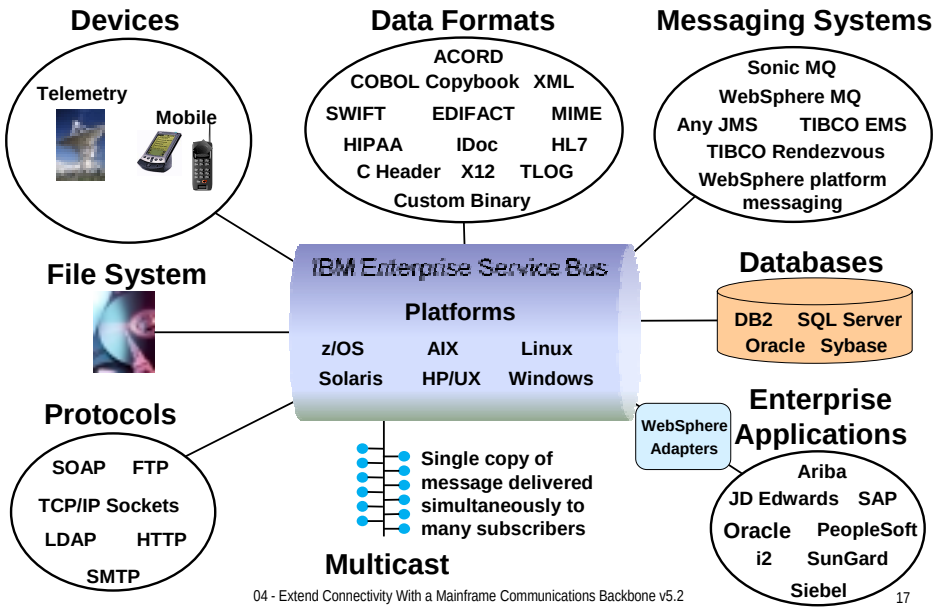


# Combine Mediation Services Together To Meet Connectivity Requirements



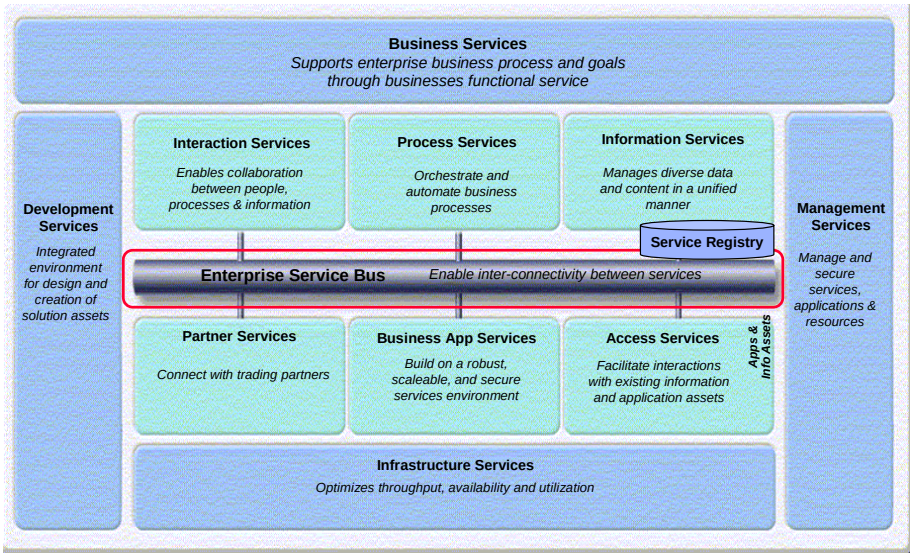

- Combine mediation services in any order
- Construct mediation flow to connect services

# IBM Enterprise Service Bus Connects Almost Anything to Anything






# An Enterprise Service Bus Serves as the Communications Backbone to Connect All Services

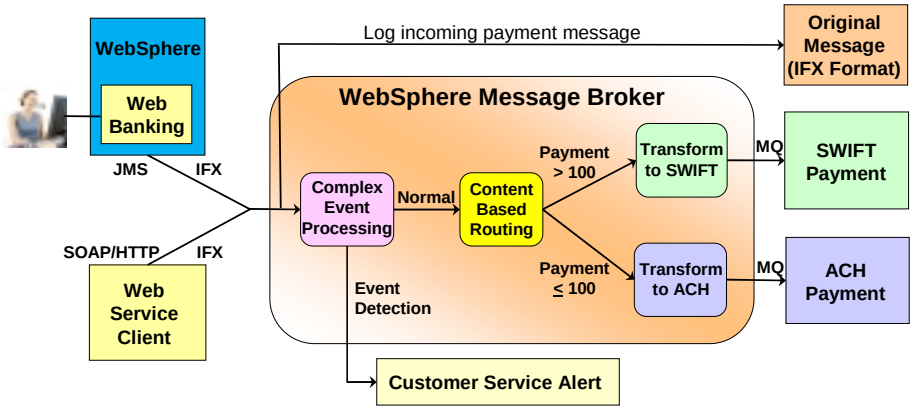


# Implementing Your Enterprise Service Bus Depends Upon Your Requirements

| | Web Services and Mediation | Extend Reach and Speed |
|---|---|---|
| | **WebSphere ESB (Runs on z/OS)** | **WebSphere Message Broker (Runs on z/OS)** |
| Built on WebSphere Application Server | ✓ | |
| Wide Range of Platforms | ✓ | ✓ |
| Web Services (SOAP/HTTP) | ✓ | ✓ |
| Content-Based Routing & Transformation | ✓ | ✓ |
| Transport Switching & Database Support | ✓ | ✓ |
| Adapters for Enterprise Applications | ✓ | ✓ |
| XML Data Format | ✓ | ✓ |
| Non-XML Data Formats | | ✓ |
| Complex Event Processing | | ✓ |
| Content-Based Publish/Subscribe | | ✓ |
| Mobile and Telemetry Devices | | ✓ |
| Multicast | | ✓ |
| Third Party Messaging Systems | | ✓ |

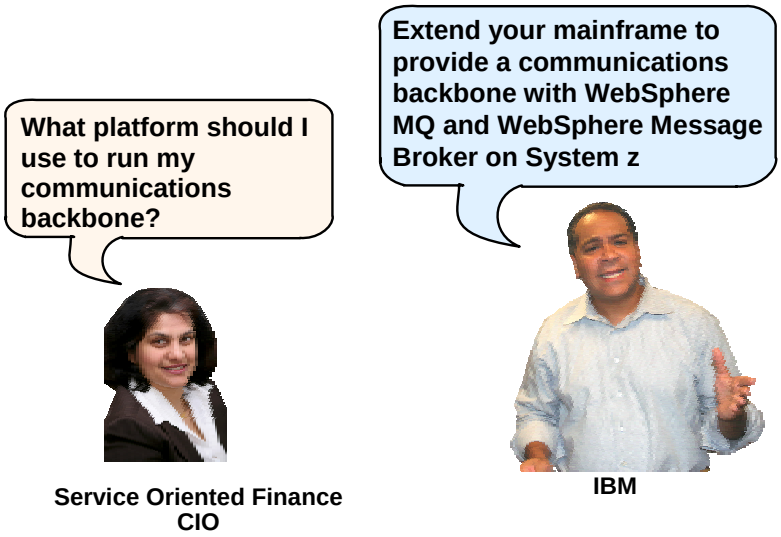


# DEMO: Using WebSphere Message Broker For Payments




- Web banking payments routed to payment system based on amount
- Transformation from IFX to SWIFT and ACH formats
- 3rd payment on same invoice number on same day creates customer service alert
- Payments are processed exactly the same for a web service client

# Run Your Communications Backbone on the Mainframe

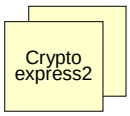




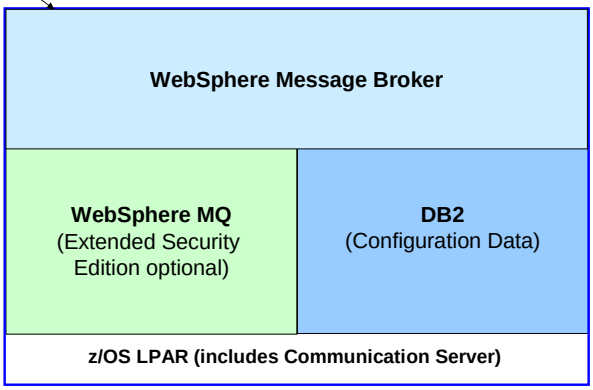

# Mainframe Extension Solution – Communications Backbone

WebSphere Message Broker Includes three components installed in one LPAR with z/OS

**WebSphere Message Broker Developer Toolkit Windows or Linux**




**Optional hardware cryptography assist**



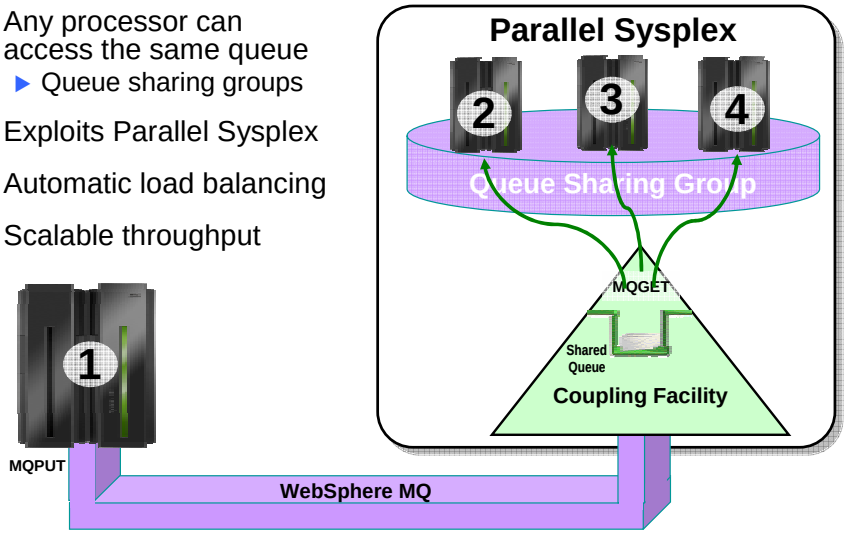
# Communications Backbone Exploits z/OS Capabilities

- Exploits sysplex clustering to provide true 24X7 operations
  - WebSphere MQ takes advantage of Parallel Sysplex to enable MQ shared queues
- Leverage System z hardware advantages
  - Huge I/O bandwidth (z10 InfiniBand - 6 GBps)
  - Hipersocket in-memory networking eliminates latency
  - Unmatched hardware reliability
  - Crypto Cards accelerate encryption
- RACF security
- Disaster recovery via GDPS
- Capacity upgrade on-demand for unexpected peaks



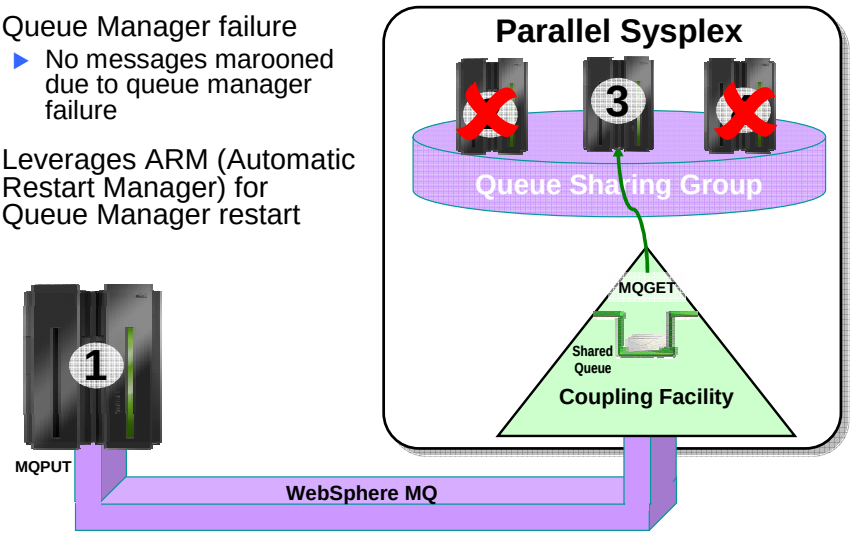
# WebSphere MQ Shared Queues on z/OS

- Any processor can access the same queue
  - Queue sharing groups
- Exploits Parallel Sysplex
- Automatic load balancing
- Scalable throughput



# Shared Queues Enable High Availability

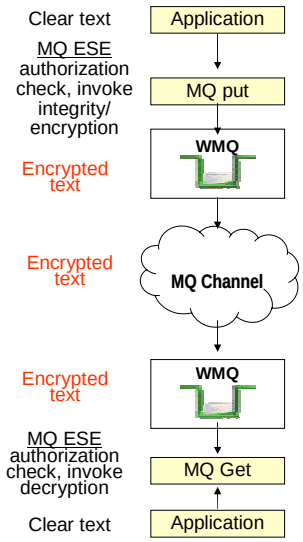
- Queue Manager failure
  - No messages marooned due to queue manager failure
- Leverages ARM (Automatic Restart Manager) for Queue Manager restart





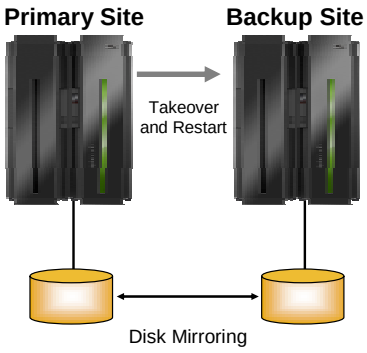

# WebSphere MQ Extended Security Edition for z/OS V6 Enhances Security and Compliance

- Protects message data end-to-end-including when it resides in queues. 3 security levels:
  - None-authorization only
  - Integrity-attaches digital signatures to messages
  - Privacy-encrypt/decrypt
- Exploits System z cryptographic processor
- Simple upgrade on top of WebSphere MQ
  - Intercepts application message before it enters/leaves queues
- Provides key element of solution for Payment Card Industry (PCI) Data Security Standard (DSS)



## A Communications Backbone on System z Benefits From Systematic Disaster Recovery

- Leverages Geographically Dispersed Parallel Sysplex (GDPS) capabilities in case of a data center disaster
  - Capacity backup to support critical workloads
  - Disk mirroring avoids message loss
  - Automated scripts drive automatic failover






## Customer Example

**Background:**

- A prominent Spanish financial institution
- 900 bank branches
- Traditional System z client with a COBOL/CICS/DB2 environment

**Challenge:**

- Required to have Secure Web Internet File Transfer (SWIFT) integration with external partners
- Needed to expose CICS transactions as Web Services without any change in business logic

**Solution: Websphere MQ and WebSphere Message Broker on z/OS**

- Transformed CICS transactions into outbound SWIFT format messages
- Reformatted inbound SWIFT messages in order to be used by host applications
- Enriched the contents of transactions with access to external databases
- Stored audit information in a database for later queries
- Developed Web Services to expose CICS transactions using standard interface
- Implemented solution on a proven and reliable message mediation platform

# Summary

**IBM's communication backbone solution provides the availability, scalability, and security I need**

**Service Oriented Finance CIO**