

# Web Enabling Architectures for e-server iSeries

John Stroh and Randy Ruhlow



## *Web Enabling Architectures for e-server iSeries*



- ▲ Speaker Introduction
- ▲ General Architectural Approaches
- ▲ Customer Solutions
  - ▶ Landsafe Credit, Inc.
  - ▶ Mohawk Industries
  - ▶ Renaissance Cruises
- ▲ More About iSeries Custom Technology Center
- ▲ Reference Material

# Basic Web Enablement concepts



- ▲ There are many iSeries web application models with
  - ▶ various runtime execution environments
  - ▶ different advantages and disadvantages
  - ▶ diverse levels of system services
  - ▶ assorted business and technical requirements
  
- ▲ Thin-client, distributed n-tier architectures
  - ▶ at least two physical tiers (client and server)
  - ▶ three logical tiers (presentation, logic, data)
  
- ▲ Communication mechanism between tiers
  - ▶ presentation -- logic      HTTP(S), JSP/HTML, XML
  - ▶ logic -- data                database drivers, stored procedures, data queues, messaging service, sockets, program call (PCML), distributed object call (RMI), JNI, EJB
  
- ▲ Design considerations
  - ▶ portability
  - ▶ reuse of legacy code (if not logic)
  - ▶ current skill set
  
- ▲ Key decisions will be in the definition of the legacy integration layer

## General Web Architectural Considerations

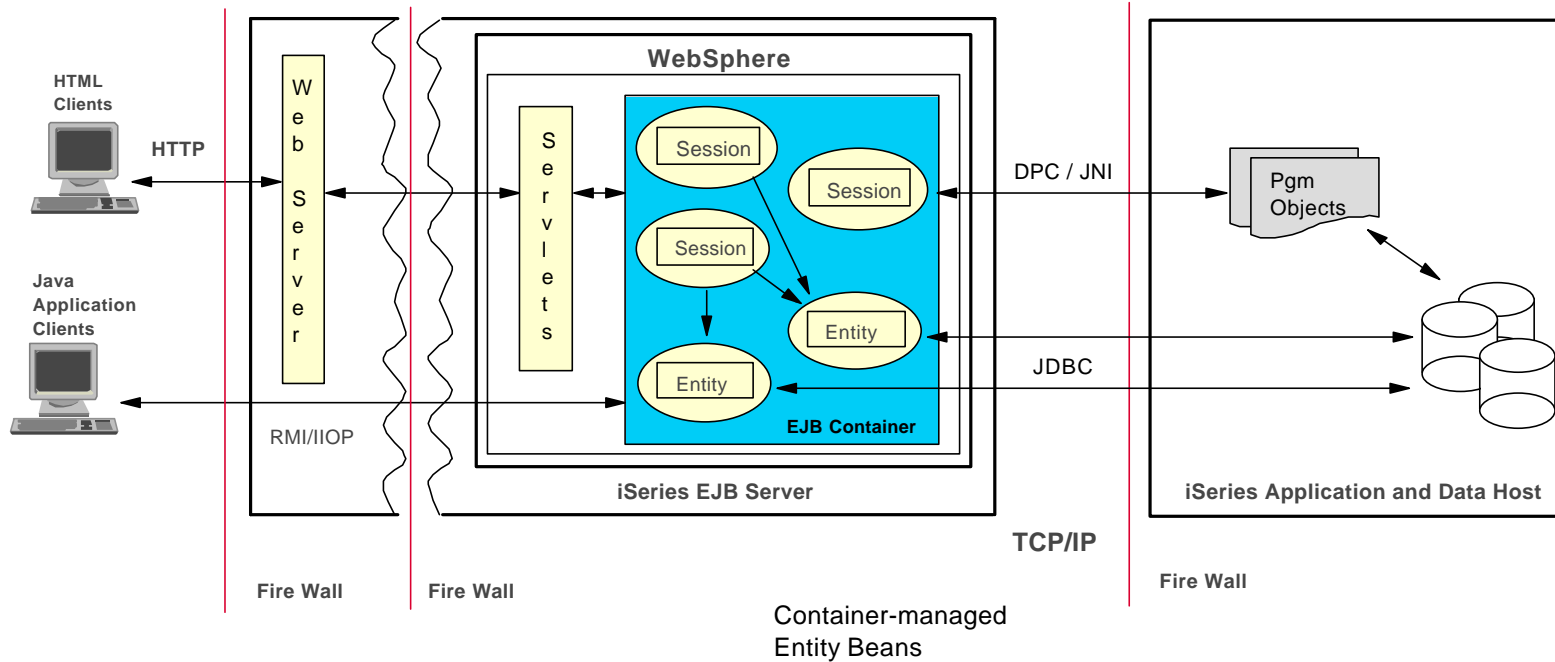


- ▲ Separate presentation code from core business logic.
- ▲ Create a "batch" or API interface to your application functions.
- ▲ Building blocks -- set of callable, stateless services devoid of screen I/O.
- ▲ ILE constructs are often used to modernize and modularize code.
- ▲ These back-end services are now available to various requesters
  - ▶ 5250 client
  - ▶ Internet browser
  - ▶ Java client
- ▲ Consistent enforcement of business rules, ease of maintenance.

# General Architectural Model - EJB



## iSeries e-business / web application model using EJB



## General Architectural Model - EJB Overview

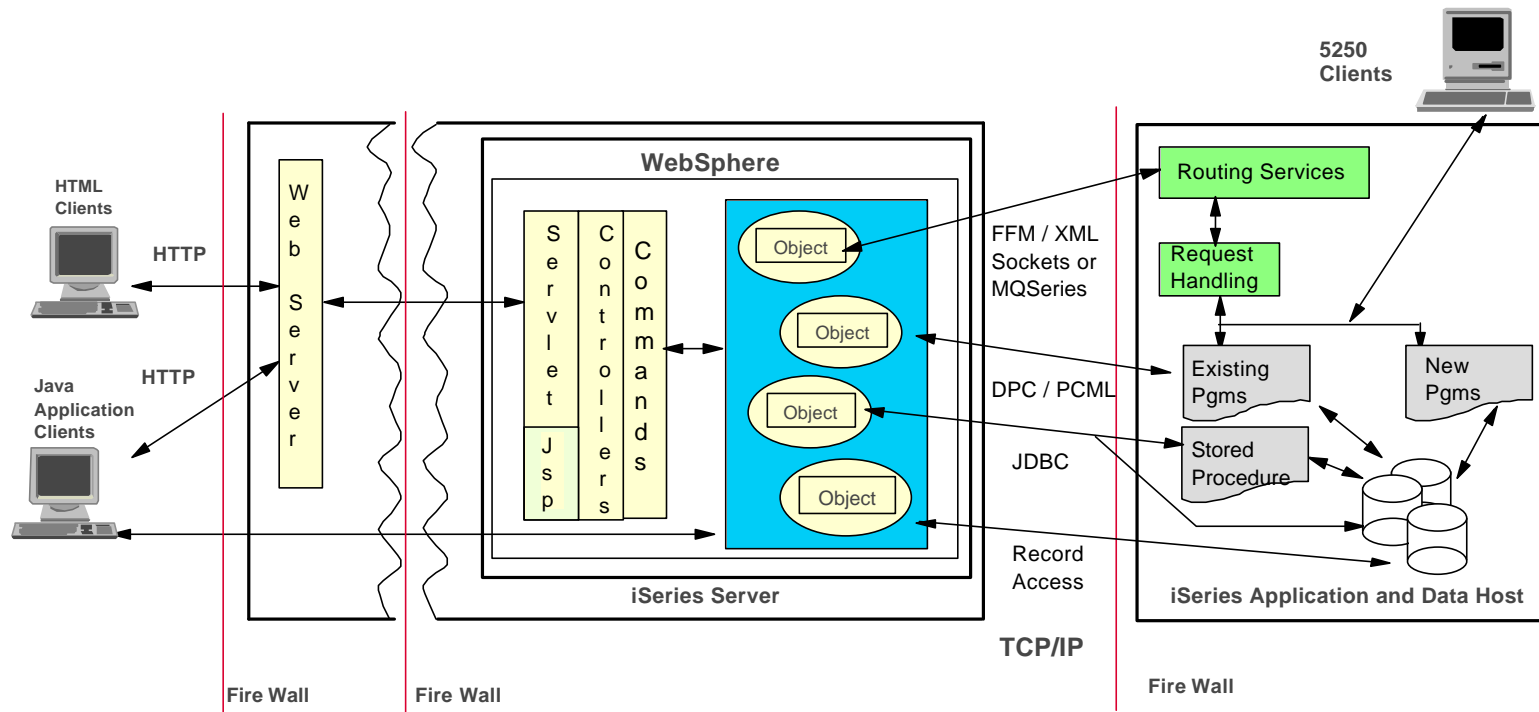


- ▲ EJB Standard
  - Infrastructure (plumbing and wiring) is provided
  - EJB components handle details of accessing platform services
  
- ▲ Session beans represent process services.
  
- ▲ Entity beans represent data services.
  - container-managed persistent
  - bean-managed persistent
  
- ▲ A container-managed persistent entity bean is automatically persisted by the EJB container/server.
  
- ▲ A bean-managed persistent entity bean must be persisted by the component developer.
  
- ▲ EJB and legacy transaction processing can work together.

# General Architectural Model



## iSeries e-business / web application model (multiple host access techniques)



# General Architectural Model Overview



- ▲ Distributed n-tier architecture
- ▲ Web serving with Java Servlets or EJB components
- ▲ Legacy Integration Layer Options:
  - AS/400 Toolbox for Java
    - ✓ Distributed Program Call (DPC)
    - ✓ Program Call Markup Language (PCML)
    - ✓ Record level access
    - ✓ Data Queues
  - Java Database Connectivity (JDBC)
  - Stored Procedures (External HLL and SQL)
  - Java Native Interface (JNI)
  - File Format Messaging (FFM)
  - Extensible Markup Language (XML)
  - Sockets
  - MQSeries

# FFM and GPSS



## ▲ File Format Messaging (FFM)

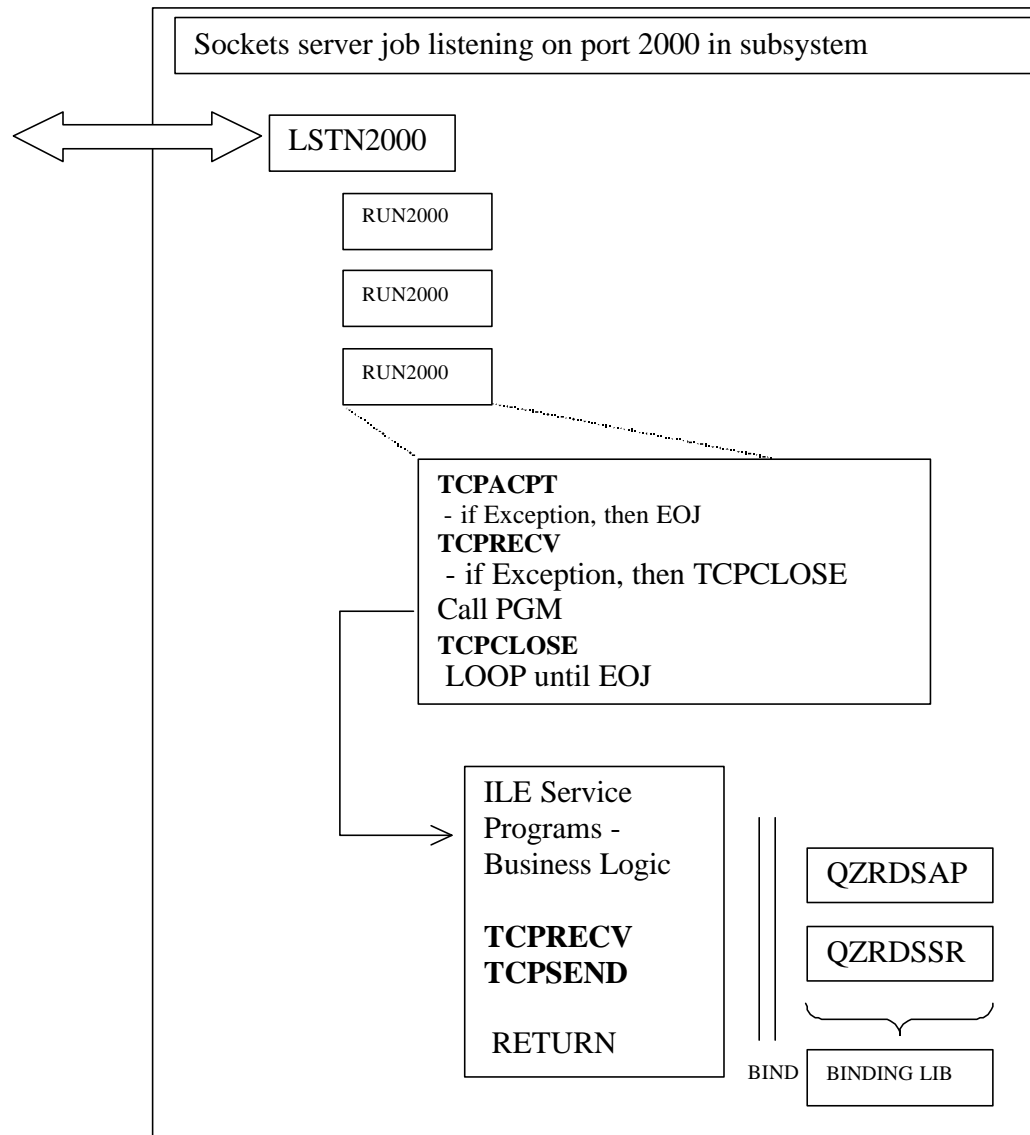
- ▶ Messaging system used over sockets or MQ Series.
- ▶ Uses a standard physical-file (created with DDS or SQL) to define the message structure.
- ▶ The PF does not contain any records. Its purpose is only to define the record format.
- ▶ A Java class generator (AS/400 pgm) is run against the PF to create the application objects (message records) and data access objects (set and get methods).
- ▶ Java classes are generated and stored in the IFS, then exported to the run-time server.
- ▶ On the legacy system, OPM or ILE programs use the same PF and record formats to construct and manipulate buffer structures.

## ▲ General Purpose Socket Server (GPSS)

- ▶ A set of APIs and programs that make it very easy to develop a TCP/IP socket server with languages such as RPG and COBOL.
- ▶ Includes subsystem, listener jobs, and run jobs.
- ▶ Handles all low-level socket communications.
- ▶ All EBCDIC to ASCII and data type conversions are handled automatically.
- ▶ Your responsibility is to develop the request handler, i.e. dispatcher job (sample provided).

📄 Refer to January 2001 News/400 article: "With GPSS and FFM, connecting Java processes to applications written in legacy HLLs is a breeze."

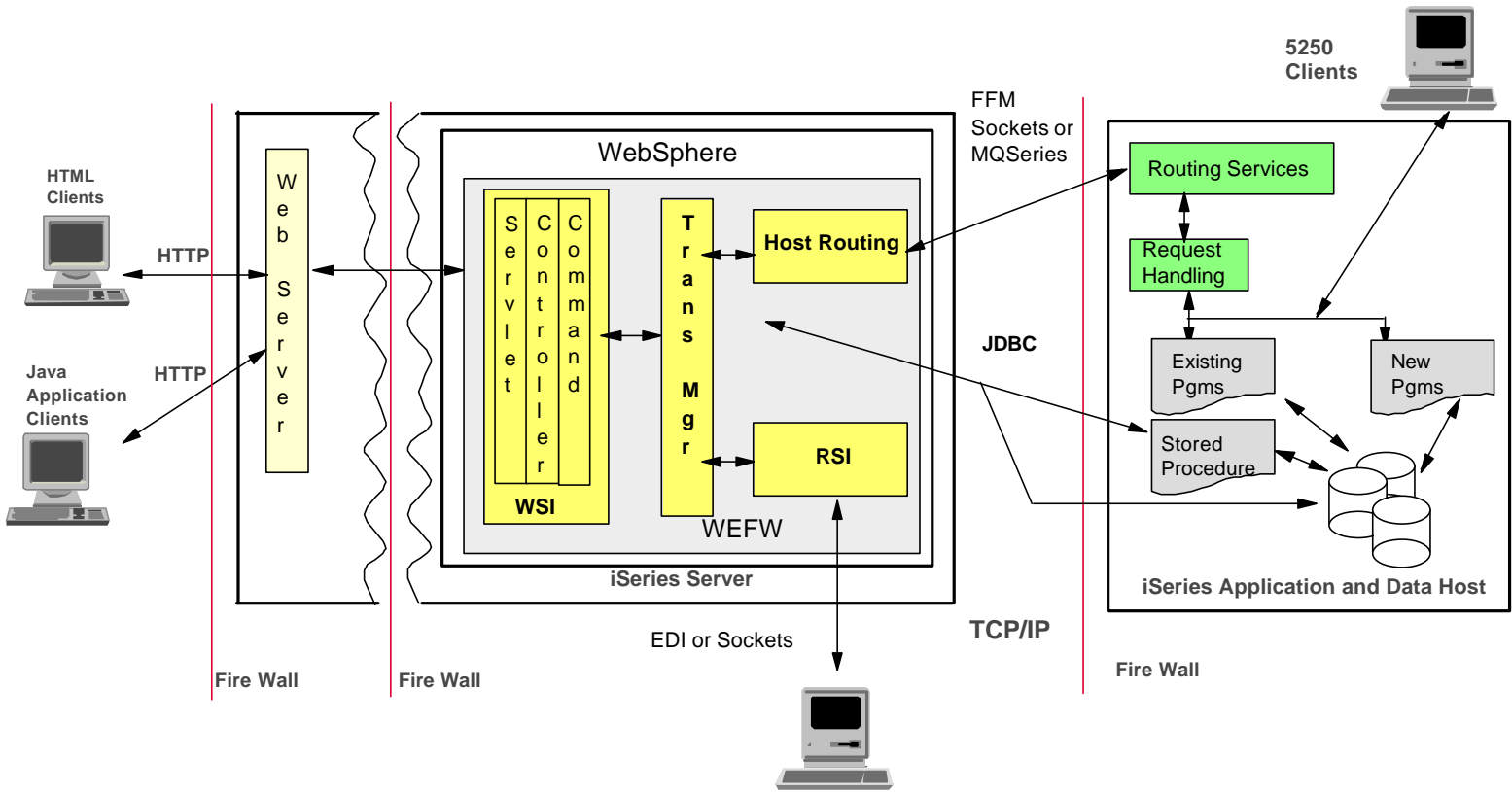
# HLL program and GPSS APIs



# CTC Web Enablement Framework



## iSeries e-business / web application model using CTC WEFW



# CTC Web Enablement Framework Overview



- ▲ Distributed n-tier architecture
- ▲ Web serving with Java Servlets
- ▲ CTC Web Enablement Framework Pre-build Components
  - Presentation Services
  - Transaction Services
  - Host Routing Services
  - Remote Integration Services
- ▲ Legacy Integration Layer Options:
  - JDBC
  - Stored Procedures
  - FFM / XML
  - Sockets
  - MQSeries

## CTC Web Enablement Framework Components



- ▲ Consists of four components
  1. Web Server Integration (WSI)
  2. Transaction Manager
  3. Host Routing
  4. Remote Services Integration (RSI)
  
- ▲ The WSI layer provides a custom servlet and supporting objects to automatically call the proper controller object and correctly process the incoming request. One controller per command or one controller per process.
  
- ▲ The Transaction Manager receives the requests and passes the proper transaction and application objects. Transaction objects identify the commands and use application objects that know how to process the transaction and fulfill the request.
  
- ▲ The Host Routing layer provides the underlying communications and messaging protocol with the host server. It uses FFM with sockets or MQSeries.
  
- ▲ The RSI provides access to host services for EDI or socket connections.

## *Customer Solutions*



- ▲ Landsafe Credit, Inc.
  - ▶ Java server application with web and Electronic Data Interchange interfaces
  
- ▲ Mohawk Carpets
  - ▶ Java servlet application with web interface (Intranet)
  
- ▲ Renaissance Cruises
  - ▶ Java servlet application with web interface (Internet) and MQSeries

## *Landsafe Introduction*



- ▲ LandSafe Credit, Inc. division of Countrywide Credit Industries, Inc.
- ▲ Based in the Los Angeles area of California USA
- ▲ Provide credit reports to Countrywide branch offices and other lending institutions
- ▲ Known for providing revolutionary processes in the mortgage lending industry
- ▲ Developed the Merged Credit Report concept

## *Landsafe Current Process*



- ▲ The current process was slow, cumbersome, and error-prone.
  - ▶ Entered in Landsafe system and via EDI to a third party vendor
  - ▶ Third party vendor submitted request to Equifax, Experian and Trans Union credit bureaus
  - ▶ Credit information was collected and manually merged
  - ▶ Final report was further refined and send back to the requester
  - ▶ 2 - 5 day process, sometimes longer
  
- ▲ X12 Message Standards Used:
  - ▶ Mortgage Credit Report Transaction Set 200
  - ▶ Application Advice Transaction Set 824
  - ▶ Mortgage Credit Report Order Transaction Set 833
  - ▶ Text Message Transaction Set 864
  
- ▲ At each stage of processing, the data had to be stored before and after any manipulation for close auditing and legal purposes.
  
- ▲ Data Centric processing. Data is the core asset of Landsafe.

## *Landsafe Business Requirements*



- ▲ Eliminate dependency on third party processing
- ▲ Automate the manual merging and manipulation of data
- ▲ More accurate and efficient Merged Credit Report
- ▲ Immediate calculation of "score cards" (credit risk indicator)
- ▲ Create a web user interface for accessibility to branches from the Internet
- ▲ Continue Electronic Data Interchange interface into the application
- ▲ Provide 24 x 7 365 days availability
- ▲ Create a foundation for future web applications and pervasive computing devices
- ▲ C++ or Java based

## *Landsafe Technologies and Tools Employed*



- ▲ Java
- ▲ IBM HTTP Server
- ▲ VisualAge for Java
- ▲ Sockets
- ▲ HTML, JavaScript
- ▲ Java DataBase Connectivity
- ▲ ILE C for CGI program
- ▲ DB2 UDB for AS/400

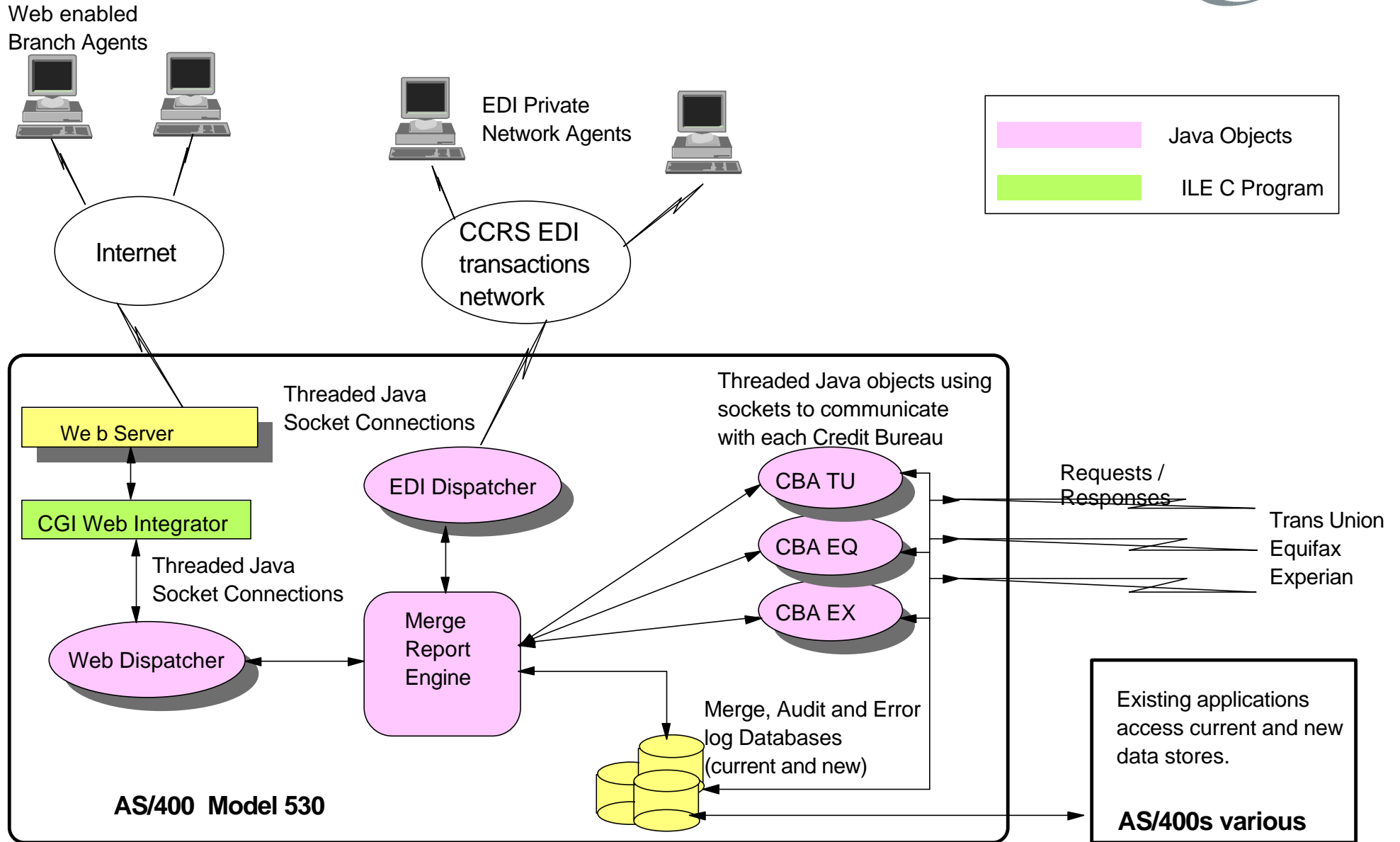
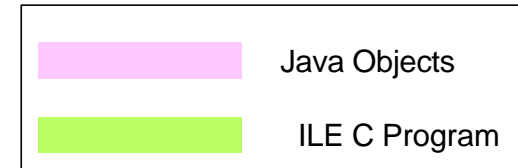
# Landsafe Solution: Simple Application Diagram



Web enabled  
Branch Agents



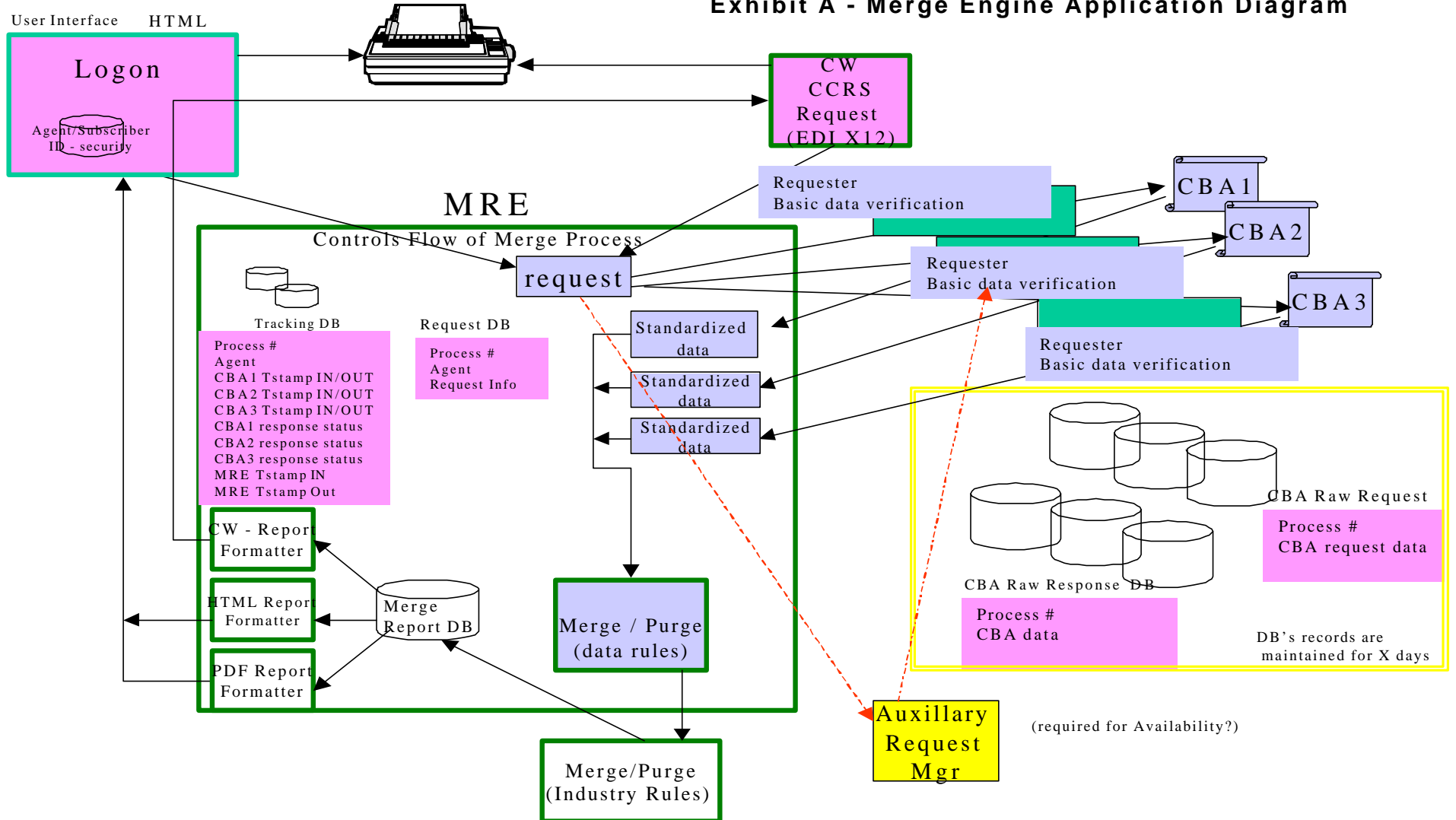
EDI Private  
Network Agents



# Landsafe Solution: Detail Application Diagram



Exhibit A - Merge Engine Application Diagram



# Landsafe Summary



- ▲ Development effort took 6 months
  
- ▲ One ILE C - CGI program
  - ▶ Servlet support just GA'd at time of design - too high of risk
  - ▶ CGI is passive processing, just provides hook into Java application
  - ▶ HTML presentation generated by Java classes delivered via CGI and Web Server
  
- ▲ Application written in 100% native Java - no AS/400 specific classes used
  
- ▲ Over 250 classes used to implement application
  
- ▲ Custom code
  - ▶ Database connection pooling and monitoring
  - ▶ Socket connection caching and monitoring
  
- ▲ Average response time well within customer's requirements
  - ▶ One request spawns 3 threads to send and receive responses from each credit bureau
  - ▶ Each response can range from 40k -200k of information
  - ▶ Application standardizes each response, removes duplicate information, merges the data
  - ▶ Avg. response time: 6 -10 seconds
  - ▶ Estimated Return On Investment (ROI): 3 - 4 months
  - ▶ Target: 40,000 request a day

## *Mohawk Carpet Introduction*



- ▲ Mohawk Industries
- ▲ Based out of Calhoun, Georgia USA
- ▲ One of the largest floor covering manufacturers in the world

## *Mohawk Carpet Current Process*



- ▲ 5250 RPG application used by customer service center
  
- ▲ Retailers call Mohawk
  - ▶ Product Availability
  - ▶ Product Reservation
  - ▶ Order Placement
  - ▶ Order Status
  
- ▲ All activity handled by phone, mail and fax statements

# Mohawk Carpet Business Requirements



- ▲ To Web Enable existing RPG 5250 applications for
  - ▶ Order Placement
  - ▶ Order Status
  - ▶ Product Availability
  - ▶ Product Reservation
  
- ▲ Portability was not a requirement.
  
- ▲ Wanted to reused existing RPG code.
  
- ▲ Improve customer service through automation and availability.
  
- ▲ Develop the system in time for industry trade show
  - ▶ Competitor promised to demonstrate similar web functionality

## *Mohawk Technologies and Tools Employed*

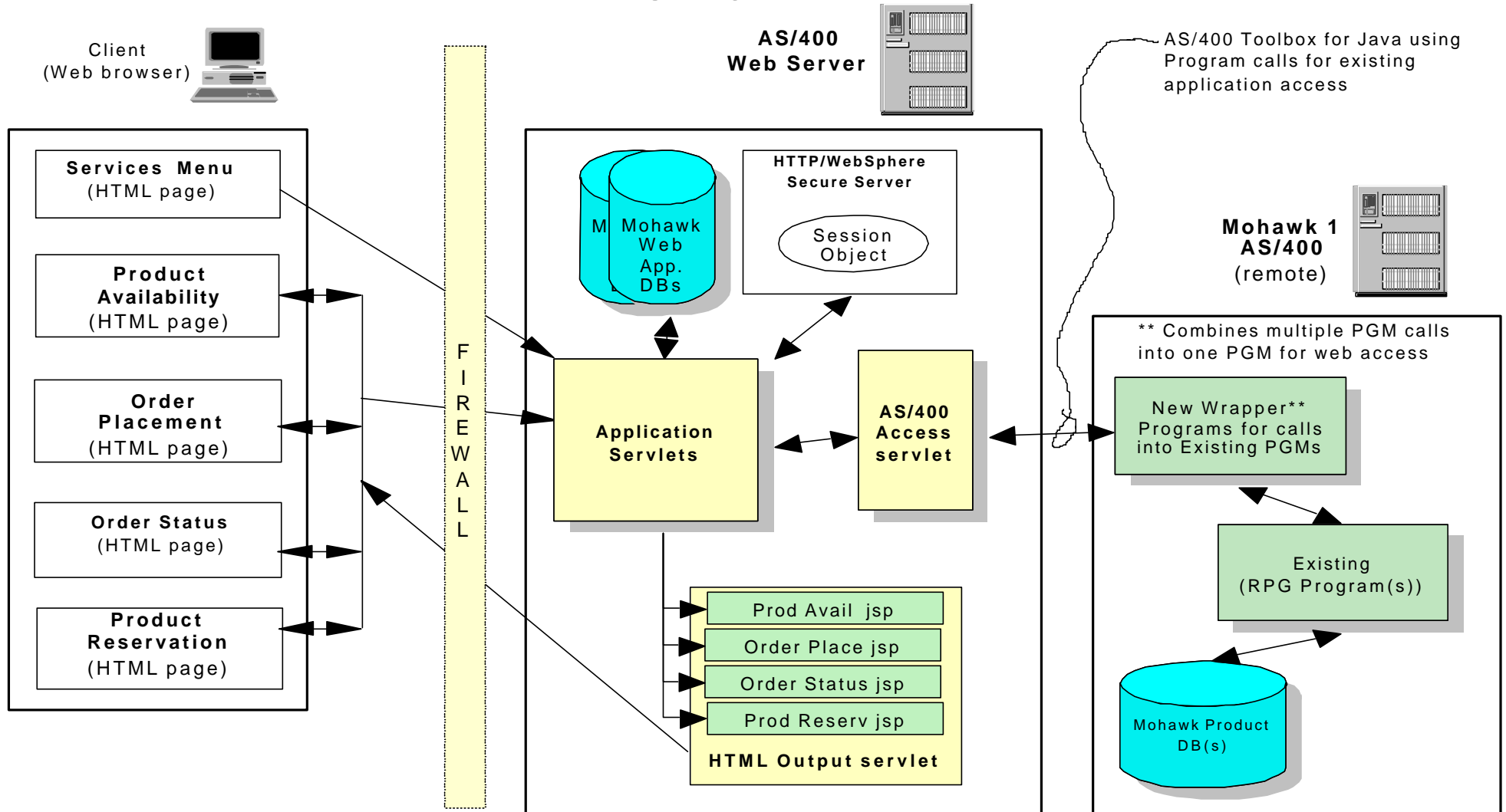


- ▲ Java
- ▲ IBM Web Server
- ▲ VisualAge for Java
- ▲ WebSphere Application Server
- ▲ Servlets
- ▲ JSPs, JavaScript, HTML
- ▲ Java DataBase Connectivity
- ▲ AS/400 Toolbox for Java
- ▲ Program Calls
- ▲ DB2 UDB for AS/400

# Mohawk Carpet Solution Diagram



## Mohawk



## *Mohawk Carpet Summary*



- ▲ Development effort took two months
  - ▶ Java and JSP development effort
  - ▶ Customer did all ILE programming efforts
  
- ▲ Minimal coding effort to integrate existing applications
  - ▶ Existing code base already modularized
  - ▶ Created ILE RPG wrapper pgms to minimize program calls from the application server
  
- ▲ Delivered LIVE, not demo application at the trade industry show
  - ▶ Actually registered users to the application during the show
  - ▶ 3000 additional customers registered the following month
  
- ▲ Customer happy to reuse all existing code base

## *Renaissance Cruises Introduction*



- ▲ Travel Industry company based in Ft. Lauderdale Florida USA
- ▲ A leading cruise liner in the travel industry.
- ▲ Passenger Cruise Lines and tour packages.
- ▲ Utilizes both direct marketing and travel agencies for customers.

## *Renaissance Cruises Current Process*



- ▲ Customer Service Center structure using 5250 COBOL applications with some internal web enabled applications provided by Host On Demand
  
- ▲ Travel Agents and Customers call Customer Service Center
  - ▶ To make travel queries
  - ▶ To book or cancel travel plans
  - ▶ To solve problems and collect all other travel information required
  
- ▲ Correspondence done by Phone, Fax, e-mail, postal mail
  
- ▲ Domino servers currently used for e-mail
  
- ▲ Basic web presence (no bookings) by NT servers

## *Renaissance Cruises Business Requirements*



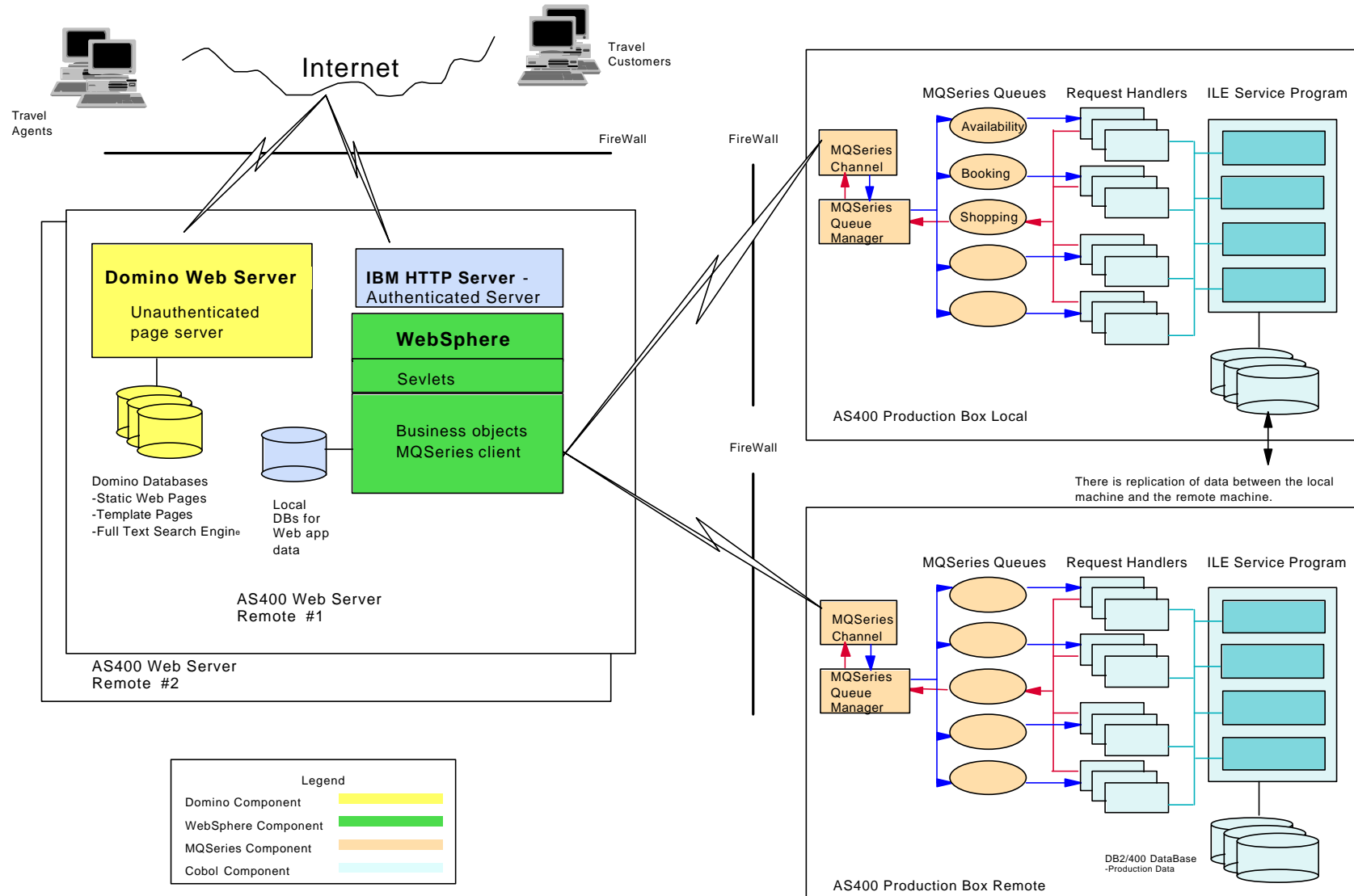
- ▲ Be the first Cruise Lines to offer bookings over the web
- ▲ Utilize as much of the existing COBOL applications as possible
- ▲ Two second response time between Application Server and Host/Data server
  - ▶ This would ensure adequate response time to the end web user
- ▲ Secure Transactions for bookings
- ▲ Force a process flow throughout the booking process
- ▲ 24/7 365 days availability
- ▲ Provide a foundation for future enhancements and applications

## *Renaissance Technologies and Tools Employed*



- ▲ Java
- ▲ IBM Web Server
- ▲ VisualAge for Java
- ▲ WebSphere Application Server
- ▲ Servlets
- ▲ JSPs, JavaScript, HTML
- ▲ Java DataBase Connectivity
- ▲ MQSeries
- ▲ Domino Web Server
- ▲ DB2 UDB for AS/400
- ▲ CTC Web Enablement Framework
- ▲ CTC Fixed Format Messaging

# Renaissance Cruises Solution Diagram



# Renaissance Cruises Summary



- ▲ Created Integration Layer on Host data server
  - ▶ Consists of Request Handler ILE COBOL pgms to direct external requests to existing ILE COBOL pgms
  - ▶ Request Handlers communicate externally via MQSeries Queues
  - ▶ CTC's Fixed Format Messaging protocol and MQ API used to transmit messages and data
  
- ▲ CTC Web Enabling Framework provided
  - ▶ Middle tier transaction routing and processing
  - ▶ Presentation layer delivery and request routing
  
- ▲ Subsecond response times between application server and host data server
  
- ▲ System is averaging 35 transactions a minute during peak loads of 200 - 500 users
  
- ▲ First cruise line to offer bookings on the web without human intervention
  
- ▲ Developed a process flow to enforce proper sequence
  
- ▲ 24/7 365 days availability through dual sites and data mirroring

## Other Web Enablement Technologies

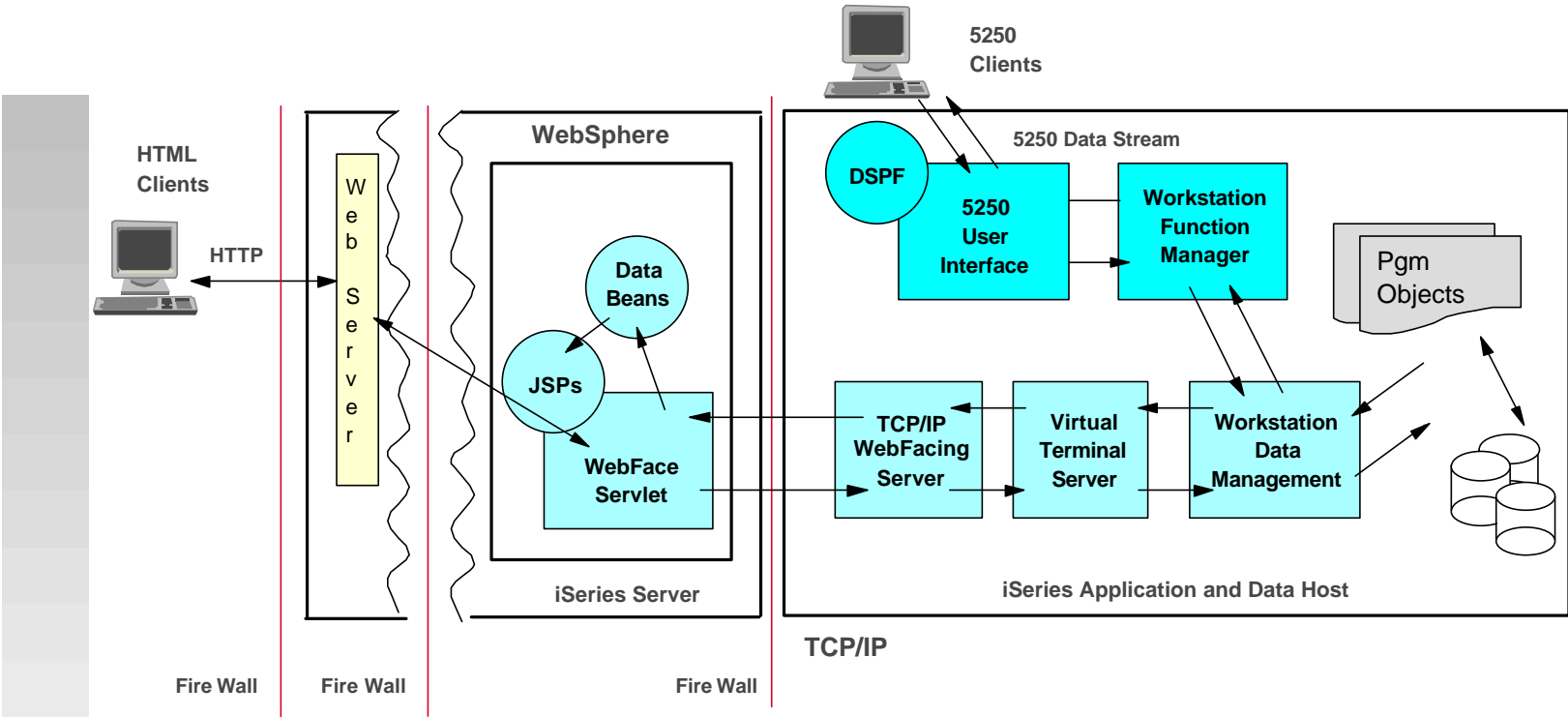


- ▲ Host On-Demand
  - Internet-to-host connectivity requiring only a browser
  - Java applet technology
  
- ▲ Host Publisher
  - Integration Objects and JSPs built using Host Publisher Studio
  - ILE language environment
  - Java servlet technology
  
- ▲ WebSphere Commerce Suite for AS/400
  - e-commerce software, Net.Commerce, Net.Data
  - WebSphere Plug-in and Java servlet engine
  
- ▲ Lotus Domino for AS/400
  
- ▲ ILE-RPG CGI Toolkit
  - CGI technique using IBM-supplied ILE service programs
  - <http://www.easy400.ibm.it/cgidev2/start>

# WebFacing Tool



## iSeries e-business / web application model using WebFacing



## WebFacing Tool Overview



- ▲ Designed to convert existing iSeries programs to run on the web with little or no modification to the original application.
  
- ▲ The WebFacing Tool consists of two parts:
  1. Development component -- Display File (DSPF) DDS to JavaServer Page (JSP) conversion tool
  2. Run-time environment -- WebSphere Application Server and iSeries server components which support a browser interface by intercepting 5250 data stream
  
- ▲ The user interface is converted once (at development time).
  
- ▲ Performance improvement over conventional "screen-scraping" which converts 5250 data stream to HTML on the fly (at run time).
  
- ▲ The JSPs preserve the look and feel of the 5250 application.
  
- ▲ Browser operational and navigational behavior is consistent with the 5250 application.
  
- ▲ Dual user interface. Application programs run in the interactive subsystem.

# *iSeries Custom Technology Center*



***Extension of iSeries development laboratory***

***Assist customers, business partners, ISVs in developing and enhancing iSeries business applications***

***Specialize in integrating legacy application with web technologies***

***Developed reusable software components***

## ***Skill Set:***

- WebSphere/Java (Configuration, Servlets, JSP, EJB, Applets)
- WebSphere Commerce Suite
- Web (HTTP, HTML, XML, CGI)
- Database (Constraints, Stored Procedures, Triggers, Performance, SQL, Propagation/Replication)
- Lotus Domino
- Reusable software (GPSS, FFM, Retail Application Interface Suite)
- Languages (Java, ILE, OPM) and Connectivity (Sockets, MQ Series)
- Development tools (Visual Age for Java, WebSphere Studio, Rational Rose, ErWin)

## *iSeries Custom Technology Center Contacts*



### EMEA Geography

General EMEA email: [CTCEMEA@fr.ibm.com](mailto:CTCEMEA@fr.ibm.com)

Eric Aquaronne 33 (0)4 9211 5791 [aquaronn@fr.ibm.com](mailto:aquaronn@fr.ibm.com)

John Stroh 33 (0)4 9211 5325 [jstroh@us.ibm.com](mailto:jstroh@us.ibm.com)

### Americas Geography

Mark Even 507-253-1313 [even@us.ibm.com](mailto:even@us.ibm.com)

Pete Cornell 507-253-4955 [pcornell@us.ibm.com](mailto:pcornell@us.ibm.com)

Ray Harney 507-253-0920 [harney@us.ibm.com](mailto:harney@us.ibm.com)

# References



## ▲ iSeries Java Reference accounts web page

- ▶ <http://www.as400.ibm.com/developer/java/solutions/jjem.html>

## ▲ Landsafe Case Study

- ▶ <http://www.as400.ibm.com/developer/java/solutions/landsafe.html>

## ▲ Renaissance Cruises website

- ▶ <http://www.renaissancecruises.com>

## ▲ Mohawk Industries Case Study

- ▶ <http://www2.software.ibm.com/casestudies/swcs.nsf/customername/82D231BAB00E2C2887256954003CB510>

## ▲ Mannheim Case Study - another CTC project

- ▶ <http://www.as400.ibm.com/developer/java/solutions/manheim.html>
- ▶ <http://www.manheim.com>

# Appendix 1 - CTC WEFW Overview



Custom Technology Center's (CTC) Web Enablement Framework (WEFW) is a custom set of Java software components which provide web enabling services for existing applications and data residing on AS/400 - iSeries systems. Java based WEFW components provide a platform neutral mechanism for integration with AS/400 - iSeries systems. WEFW includes a customized communication messaging protocol, Fixed Format Messaging (FFM) that is DDS based and can be used over sockets or MQSeries. The FFM protocol is not exclusive to WEFW and data access can be obtained by various other technologies such as Java Database Connectivity (JDBC), Java Stored Procedures and Remote Procedure Calls (RPC).

The main components of the framework are:

## **Web Services Integration (WSI)**

Provides a presentation layer set of components that allow the developer(s) of the web application to concentrate on writing the application and not so much on the coding efforts required to integrate with HTTP and the web. Included with this component are custom servlets and supporting objects designed to automatically call the proper Controller objects that correctly processes incoming requests. WSI can be loosely or tightly coupled to the Transaction Management Component depending on the Servlet used as the entry point into the application. A set of external configuration files are used to map the requests to the proper Controller. This component also provides a caching mechanism of the Controller objects so that object creation and heap size do not impact the performance of the application.

## **Transaction Management (TransMgmt)**

Provides a built in mechanism for moving requests received from the WSI component into the proper Transaction and Application objects. Transaction objects identify the units of work and use Application objects to process the transactions with existing applications or data on remote application and data servers. This component may use but is not limited to the Host Routing component to communicate and pass message data to and from the remote AS/400 - iSeries systems. Other data access methods and technologies such as JDBC or RPCs can be utilized with extensions to the Transaction or Application objects. External configuration files provide the mapping of Transactions to Applications and Applications to the Host Routing mechanism. This component provides a pooling mechanism for the most commonly used resources for better application performance.

# Appendix 1 - CTC WEFW Overview



## **Host Routing**

Provides the underlying communications and messaging protocol with remote AS/400 - iSeries systems. It provides either Sockets or MQSeries communications and incorporates the FFM messaging protocol developed by CTC. The combination of FFM and Sockets or MQSeries provides a performance optimized messaging system specifically designed to receive and pass messages to AS/400 - iSeries systems. To accommodate communications on the remote AS/400 - iSeries systems, an external CTC Socket Server or MQSeries FFM asset is installed and incorporated in the web enablement solution. FFM uses DDS files to provide a common definition point for the development of the messages. The DDS files define the structure of the messages and is used to build the buffer structures in the ILE/OPM programs and to generate Java classes used in the Host Routing component. External configuration files are used to define the characteristics of the communications required for the application. The Host Routing objects are pooled for optimum performance and reuse of the system communication resources.

## **Remote Services Integration (RSI)**

Provides other than web access to the same application code used for web enabling existing AS/400 - iSeries applications and data. This is done using the Host Routing components to provide either Sockets or MQSeries connectivity to the existing application code base. It is FFM base and provides the same benefits of the Host Routing component except it is not using the HTTP protocol and the HTTP Web Server for connectivity into the entry point of the application.

## Appendix 2 - CTC General Purpose Socket Server



### AS/400 TCP/IP General Purpose Socket Server II

The AS/400 TCP/IP General Purpose Socket Server (GPSS) offers performance, reliability and especially ease of implementation. This server has been in use since V3R1 and will migrate to future OS/400 releases. The GPSS is packaged as a standalone library with everything needed to get started. A complete Programming Guide is supplied. Support and consulting services are also available. The GPSS optionally includes File Format Messaging (FFM) capabilities and utilities to create associated Java classes for TCP/IP clients. All of the server's source code is supplied (ILE-C, CL, etc.). Finally, we offer a restricted version for initial evaluation purposes.

The server is comprised of two parts; a framework which handles normal TCP/IP server overhead functions and the middle-ware which provides the server's data exchange interfaces.

The server's framework specifically provides simple server startup and shutdown through OS/400 commands. It will listen and accept connections on any TCP/IP Port. Connection handling, and job submission are managed within this framework. Multiple instances of the GPSS may be in operation on unique TCP/IP Ports. There is a capability to select any specific local IP interface and the GPSS will handle both Transaction orientated and Session orientated server implementations.

The GPSS requires a user written server plug-in program, in any AS/400 high level language. This plug-in program must handle TCP/IP data traffic and the business logic. This program interfaces to the server's middle-ware through Socket APIs. Samples of such programs are supplied in RPG, COBOL and C. The server supports OPM and bound (ILE) calls to the Socket APIs, however ILE calls are highly recommended.

The Sockets implementation pitfalls that are commonly experienced by those that are not familiar with them, are all handled within the GPSS middle-ware. The plug-in program's Socket I/O looks similar to DB2 file or data queue I/O. A rich set of send and receive API options is provided that will handle most methods of data exchange over TCP/IP. ASCII/EBCDIC translation, is provided under the covers or is available through the middle-ware APIs. Also included is a powerful internal trace capability, a client IP Address retrieve capability, the Socket APIs that can be used to implement an AS/400 TCP/IP client application, and more.

## Appendix 2 - CTC General Purpose Socket Server



### Terms:

- OS/400 - Compatible with V3R1 and above.
- Y2K ready.
- Enterprise wide license.
- Source code shipped.
- Code is "as is" Type II material.
- A Programming Guide and operation manual is included.
- Support and consulting services available on an hourly billable basis.

### Available on the following AS/400 Media:

- E-mail attachment (Preferred and the fastest delivery method).
- CD-ROM.
- Tape media of your choice.

**Pricing:** Flexible plans through IBM Global Services or IBM Business Partners.  
Contact the Custom Technology Center.

**Contact:** IBM iSeries Custom Technology Center in La Gaude, France

Eric Aquaronne  
33 (0)4 92 11 57 91  
aquaronne@fr.ibm.com

John Stroh  
33 (0)4 92 11 53 25  
jstroh@us.ibm.com

**Contact:** IBM iSeries Custom Technology Center in Rochester Minnesota.

Pete Cornell  
507-253-4955  
Pcornell@us.ibm.com

Mark Even  
507-253-1313  
even@us.ibm.com

Ray Harney  
507-253-0920  
harney@us.ibm.com