iBus™ MessageServer 4.5

The Challenge

Large and small organizations are under intense competitive pressure brought about by the influences of globalization and Internet technology. Applications covering the automation of business processes through to full Enterprise Application Integration Systems (EAI) with on-line connections to both customers and suppliers need to be developed at Web speed.

The Solution

The Softwired iBus™ architecture, built within the framework specified by Sun Microsystems Java Message Service (JMS) standard, offers all the necessary features to develop and deploy messaging applications. The iBus//MessageServer abstracts the details of network programming while allowing complete control of the messaging system and design flexibility for the application. Application developers can focus their efforts on content instead of the complex mechanics of application-to-application messaging and information dissemination. The iBus//MessageServer speeds up the development process, saving both cost and critical time to market.

The iBus™ supports multiple transport protocols, all the JMS programming paradigms, and a fully scalable deployment environment. The iBus//MessageServer is specifically designed to complement an existing infrastructure. It fits perfectly into standard programming APIs including the Web, wireless protocols and more traditional development environments such as C, C++, CORBA and EJB.

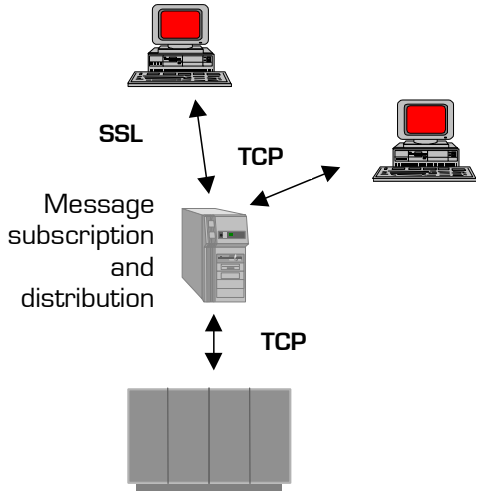
Standards Compatible

JMS from Sun Microsystems is the preferred messaging standard for the Java developer. It provides a standard set of 'easy to use' programming abstractions for developing messaging based applications. The iBus//MessageServer fully supports the JMS open standard, ensuring freedom of choice in the selection of platforms, software vendors and off-the-shelf applications.

Softwired's iBus™ JMS, **written in 100% pure Java**, is highly portable, easy to install, extensible for new protocols and devices and most importantly a 'write once', 'test and run everywhere' development system. JMS is a member of the fastest growing group of enterprise APIs, jointly known as the "Java 2 Enterprise Edition", or J2EE. This allows for truly seamless integration of the iBus//MessageServer with current Enterprise Java Beans (EJB) application servers.

Architecture

The iBus//MessageServer is a highly scalable and network centric message hub. Producers and consumers communicate with the server through protocols such as TCP/IP or SSL. The server takes care of forwarding messages from producers to consumers, of message persistency, security and access control.



iBus//MessageServer 4.5 Key Features:

XA Distributed Transactions
In compliance with the new J2EE 1.3 standard Chapter 8, XA Connection – Consumer, iBus//MessageServer 4.5 enables distributed XA transactions via an implementation of JTA XAResource.

Ping
Ping is a plugable architecture that actively controls the connection in the event of client server inactivity.

Slow Subscriber Detection
Allows for the detection and handling of slow subscribers thus eliminating unnecessary system slowdowns.

Java Message Service (JMS) Conformance
Publish/Subscribe: iBus//MessageServer multicasts messages to any number of recipients and implements the JMS publish/subscribe communication model for asynchronous delivery of messages to topics with any number of consumers.

Point-to-Point: Asynchronous delivery of messages to the queue of a given recipient. Optionally, high-speed in-memory queues are supported for applications that trade speed against 100% message delivery guarantee (such as user-interface driven messaging applications).

Access Control: Access to JMS queues and topics can be controlled on a per-user level. In addition to standard implementation support for the JMS userid/password combination and access control lists, you can provide your own Access Controller implementation and integrate it seamlessly into the message server runtime. This approach also supports client SSL certificates as a basis for access control decisions. Integration into an existing Enterprise PKI (public key infrastructure) is thus as easy as implementing a simple Java interface.

Easy Administration via Graphical User Interface (GUI)
The iBus//MessageServer features a graphical administration application to setup users / groups, server configuration, and to browse and administer queues, topics and messages. For easy integration with an existing managing infrastructure, all configuration operations are also accessible programmatically through a documented API.

XML Integration
The iBus//MessageServer provides support for applications using XML to represent their data structures. Applications can send and receive XML documents directly, without converting them into the available JMS message types first.

The producer can send an XML document represented as XML text or DOM document. The consumer retrieves the XML document either as XML text or DOM document independent of how the document was initially produced (DOM or text). The producer and consumer can ask iBus™ to validate messages for correctness.

End-to-End, Guaranteed Delivery
It is critical, particularly in message queues, that every message reaches the consumer exactly once, and only once. No message must be lost, and no message may be duplicated. The built-in transactional database of Softwired's iBus//MessageServer guarantees one-time delivery (exactly once) even in the case of power loss, server hardware failure or other 'catastrophic' events. Optionally, an external Oracle® Database can be used to store messages.

Messaging over the Web via SSL
For messaging applications that require privacy, iBus//MessageServer Business 4.5 offers the SSL transport protocol with full strength cryptography.

Extensible Protocol Support
The iBus//MessageServer supports messaging across a range of protocols out-of-the-box: TCP and SSL. But the iBus//MessageServer doesn't stop here. Its patent-pending "protocol plug-in" architecture supports every transport protocol, quality-of-service and delivery guarantee imaginable. Examples include "raw Ethernet" for embedded devices, wireless protocols such as pager and mobile phone standards, or satellite downlinks.

Extended Request/Reply
Developers can use the JMS request/reply API as a replacement for RMI or CORBA. iBus//MessageServer enhances this API in three key areas to allow greater flexibility: Timeouts, multiple replies and fault-tolerance.

Requirements
The iBus//MessageServer is fully tested and supported on Sun Solaris, Windows 2000, NT, and Linux and is known to work on AIX, HP UX, IRIX and AS/400. Development and client deployment can be done using Java 1.1.6 upwards. The server side requires a Java 2 compatible virtual machine.

Contact Information
Softwired Inc.
www.softwired-inc.com, info@softwired-inc.com
Phone +41 1 445 23 70, fax +41 1 445 23 72
Technoparkstrasse 1, CH-8005 Zurich, SWITZERLAND

Softwired
Next stop: ubiquity