Open Message Queue
mq.dev.java.net

Jason Huang
Technical Consultant
Sun Microsystems, Inc.

Objective

Understand basic of JMS API and OpenMQ products

Agenda

- Introduction to JMS
- What's OpenMQ
- Features of OpenMQ
- Demo

What is Java Message Service?

- A Java API for Message Oriented Middleware(MOM)
  - JMS is a specification developed under the Java Community Process as JSR 914.
  - http://www.jcp.org/en/jsr/detail?id=914

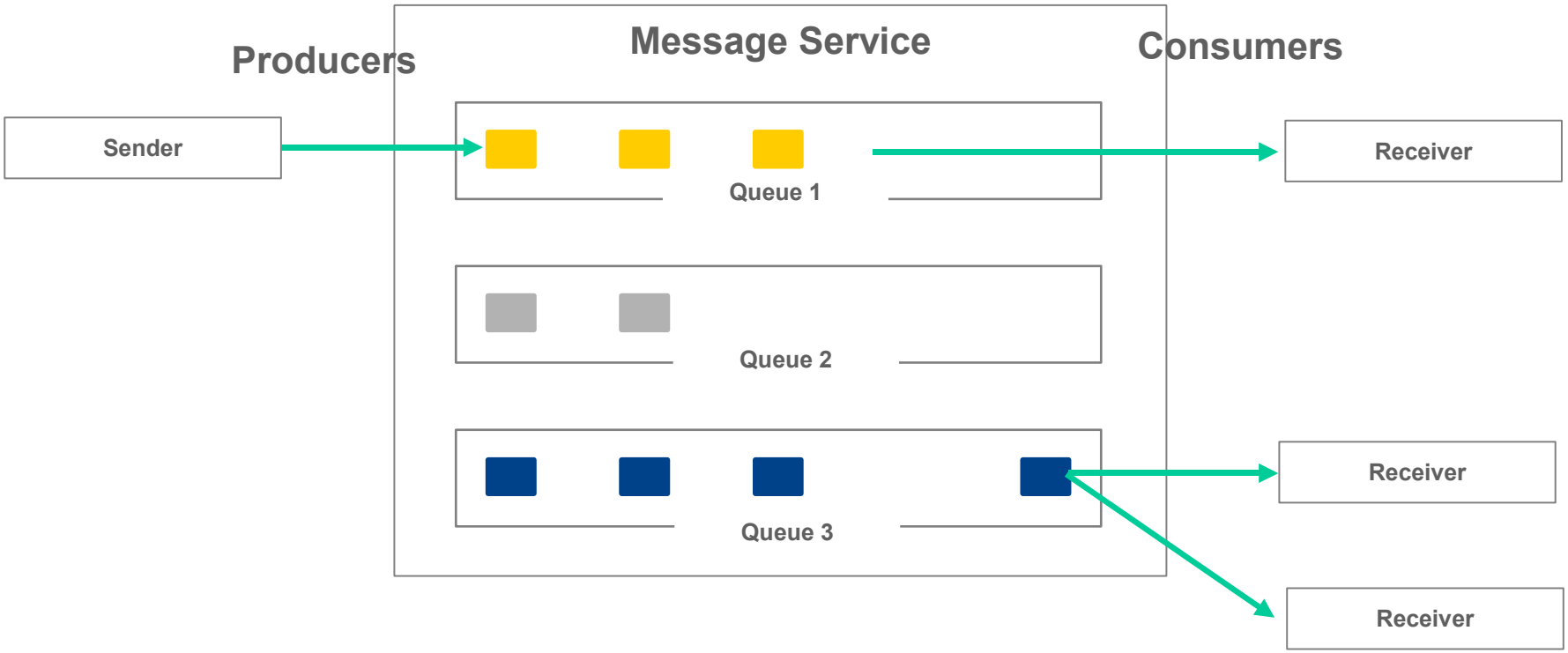
What Is Java Message Service?

- The Java 2 Platform, Enterprise Edition (J2EE™ platform) specification for MOM products
- Defines provider-neutral APIs and administered objects that allow client applications to be portable across Java Message Service providers
- Is defined as part of the J2EE 1.3 and later specification

What Is Java Message Service (continued)?

- Supports two different messaging models: point-to-point and publish-subscribe
- Supports asynchronous messaging and message-driven beans (MDBs) on a J2EE application server
- Allows providers to implement additional provider-specific features
- Provides the ability to tune applications for performance and reliability

Point to Point Messaging

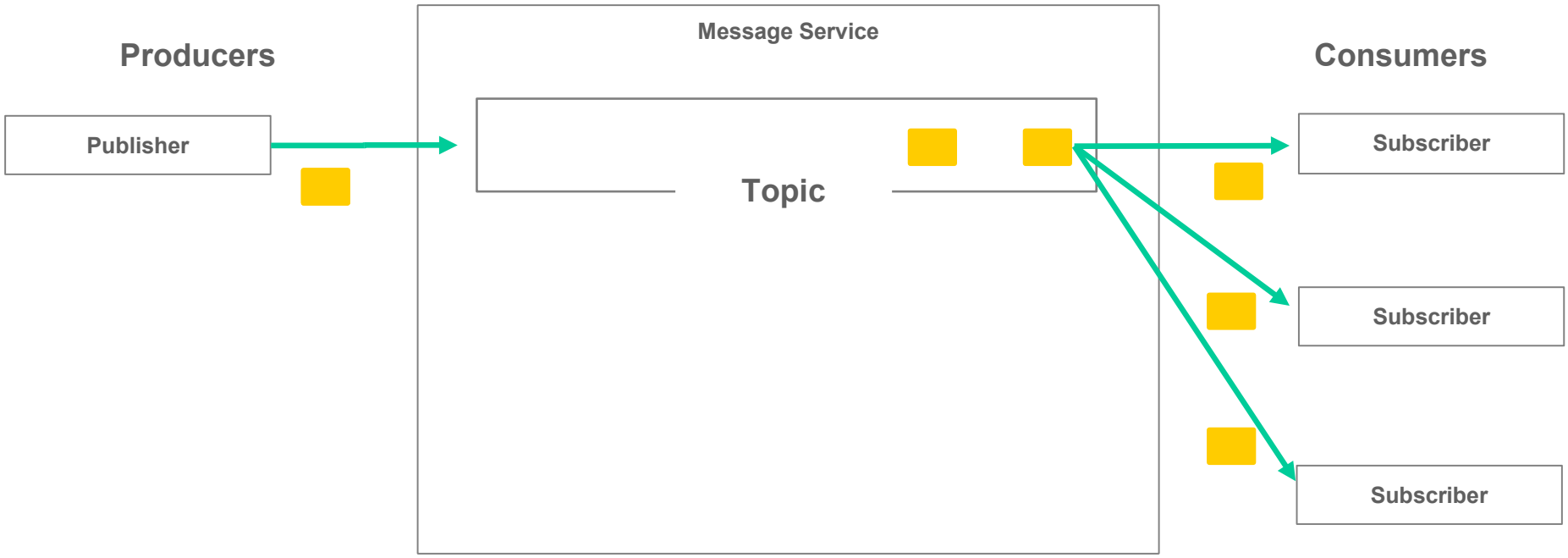


Messages from producers delivered to consumers

Held in Queue

Delivered, in order – *Guaranteed*

Publish and Subscribe Messaging



Publishers are independent from subscribers
Message expiration is configured
Subscriptions may be “durable”

JMS Terminology

- Destinations
  - The target intermediary of all messages
  - Maybe temporary (useful in Request / Response)
- Producers, Consumers
  - Clients produce messages (i.e. place them on a Queue or Topic); and Consume messages (i.e. remove them from a queue or topic)
- Connections
  - Abstract client to broker context
- Sessions
  - Abstract the message context

Java Message Service Concepts

The following concepts are essential to understanding the Java Message Service:

- Administered Objects
  - Connection Factory
  - Physical Destinations
- Connections
- Sessions
- Producers and Consumers
- Durable Subscribers

Administered Objects

- Are abstract objects for connections factories or destinations
- Are stored in a name service through the Java Naming and Directory Interface™ (JNDI) API
- Can be pre-configured with provider-specific properties
- Provide a neutral way for clients to specify which Java Message Service to connect to, and which destination to send or receive messages from
-

Connection Factory

- Create a connection between JMS Client and JMS provider
- various configuration parameters.

Physical Destinations

- A destination:
  - Is a target for messages
  - Is accessed either explicitly by name using the JMS API or loaded from an administered object
- A topic destination is used for Publish/Subscribe delivery.
- A queue destination is a target for point-to-point delivery.
- A temporary destination (topic or queue) can be created to receive replies for messages.

Connections

- Connection:
  - A connection represents a single pipe into a Java Message Service.
  - In MQ, a connection represents a single stream of messages to the broker.
  - A connection is obtained from a connection factory.

Sessions

- Sessions are a context for producing and consuming messages.
  - Messages from different producers are sent in order.
  - Messages to different consumers are received in order
  - They are single-threaded.

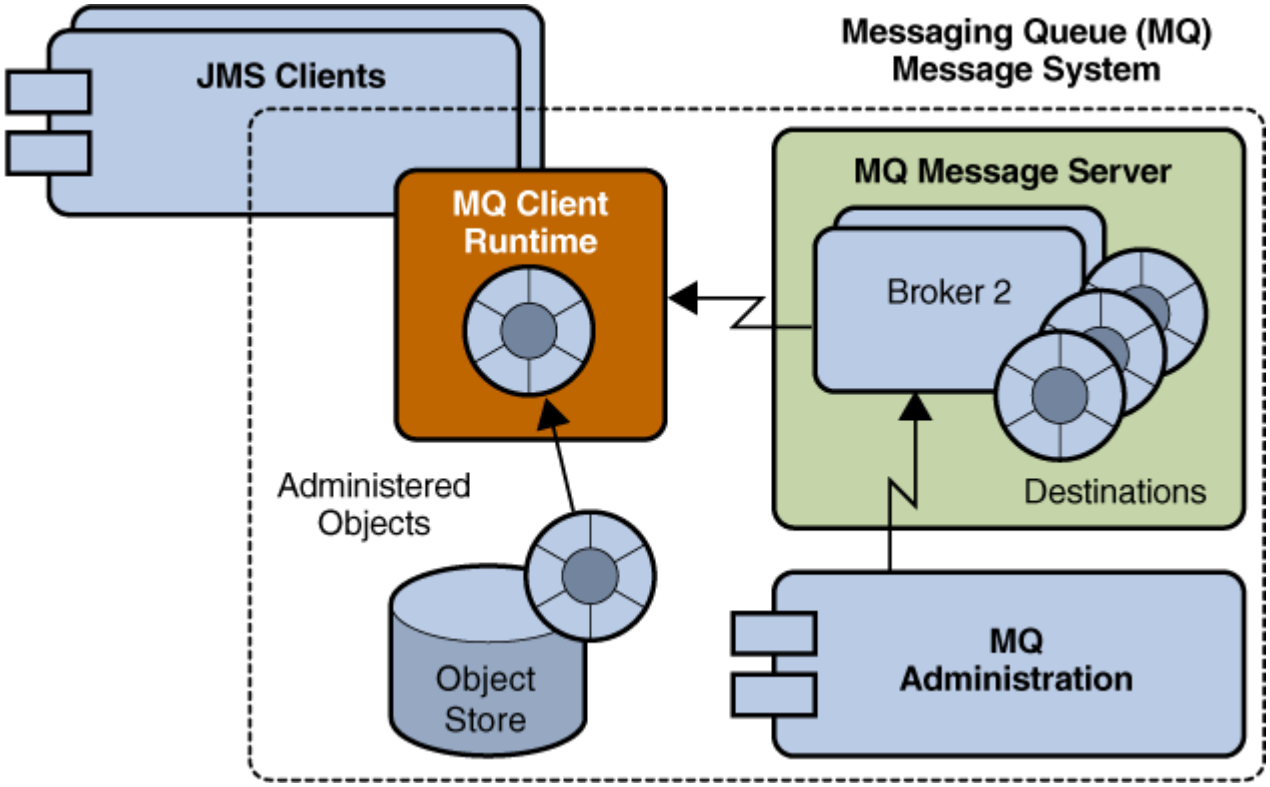
Producers and Consumers

- Producers send messages:
  - Publishers send messages to topics.
  - Senders send messages to queues.
- Consumers receive messages:
  - Subscribers receive messages from topics.
  - Receivers receive messages from queues.
- QueueBrowsers
  - provide an interface that allow Java Message Service clients to examine the contents of a queue.

Durable Subscribers

- Durable Subscribers are a subset of subscribers.
  - With normal subscribers, messages sent while the subscriber is down are lost.
  - Durable subscribers are able to retrieve messages sent while the subscriber was down.

MQ Basic Architecture (continued)



MQ Basic Architecture

- MQ has three major components:
  - The Java Message Service server
    - called the broker
  - The client implementation:
    - A Java API
    - A C API
    - A resource adapter
  - Administration tools:
    - Command-line tools
    - Graphical User Interface (GUI)
    - Java Management Extensions (JMX™) API that provides programmable administrative monitoring and control

Other Messaging Services

- There are different ways to implement distributed computing architecture. Some common implementation methods include:
  - Remote Method Invocation (RMI) − Allows one application to run a procedure on another application
  - Common Object Request Broker Architecture (CORBA) − Allows retrieval of an object from a remote application using a standard protocol (IIOP)
  - Simple Object Access Protocol (SOAP) − Lets applications exchange eXtensible Markup Language (XML) formatted data over Hypertext Transfer Protocol (HTTP)

Agenda

- Introduction to JMS
- What's OpenMQ
- Features of OpenMQ
- Demo

What's OpenMQ!

- Complete stand-alone messaging server
- Fully supports JMS
- Provides many additional features (beyond JMS)

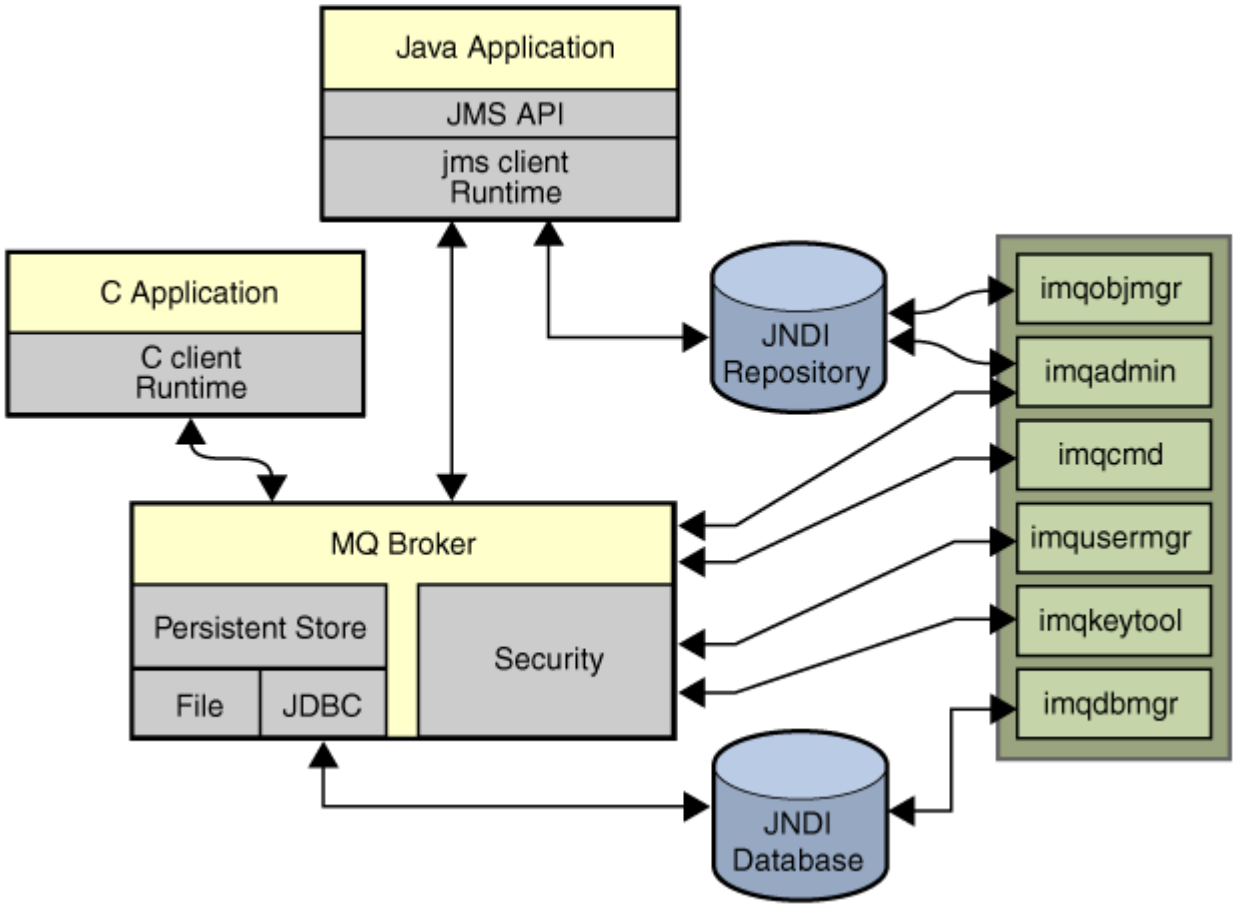
Open Message Queue

- Member of GlassFish project community
  - http://mq.dev.java.net
  - The JMS Provider distributed with GlassFish
- Community version of Sun Java System Message Queue
  - No feature differences between commercial and community versions
- Complete Open source;
  - Stable and Promoted builds
- Community Process
  - feedback, commentary, updates

Roadmap and License

- Stable binaries (with source) from each release
  - Since 4.0 / GlassFish V1
  - Now(2007/11) 4.1 / GlassFish V2
- Complete Source Code
  - Dual license (CDDL or GPL), same as GlassFish

System block diagram



Well organized and easy to browse source

Agenda

- Introduction to JMS
- What's OpenMQ
- Features of OpenMQ
- Demo

OpenMQ 4.1 Features

- High Cluster Availability
  - Data Availability
  - Service Availability
- JAAS Support
- JES Monitoring Framework Support
- Transaction Management
- Fixed Ports for C Client Connections

OpenMQ 4.0 Features

- Broker Administration
  - Ability to quiesce a broker or shut it down at a specific time
- JMX API Support
- Client Runtime Logging
- Connection Event Notification
- Tighter integration when run with Sun Java System Application Server (Application Server)

OpenMQ Features (continued)

- Conventional Cluster Support
  - Service reliability
  - Distributed broker cluster
  - Master Broker mechanism
- Dead Message Queue
  - which is a repository for messages that cannot be delivered, including expired messages
- Message store
  - Both file- and JDBC-based message stores (used for guaranteed delivery)

OpenMQ Features (continued)

- JCA Resource Adapter
  - to allow MQ to be plugged into application servers
- Message Body Compression
- Destination controls
  - Message count limits
  - Limits on number of consumers
- No Acknowledge mode
  - to improve performance when reliability is not important

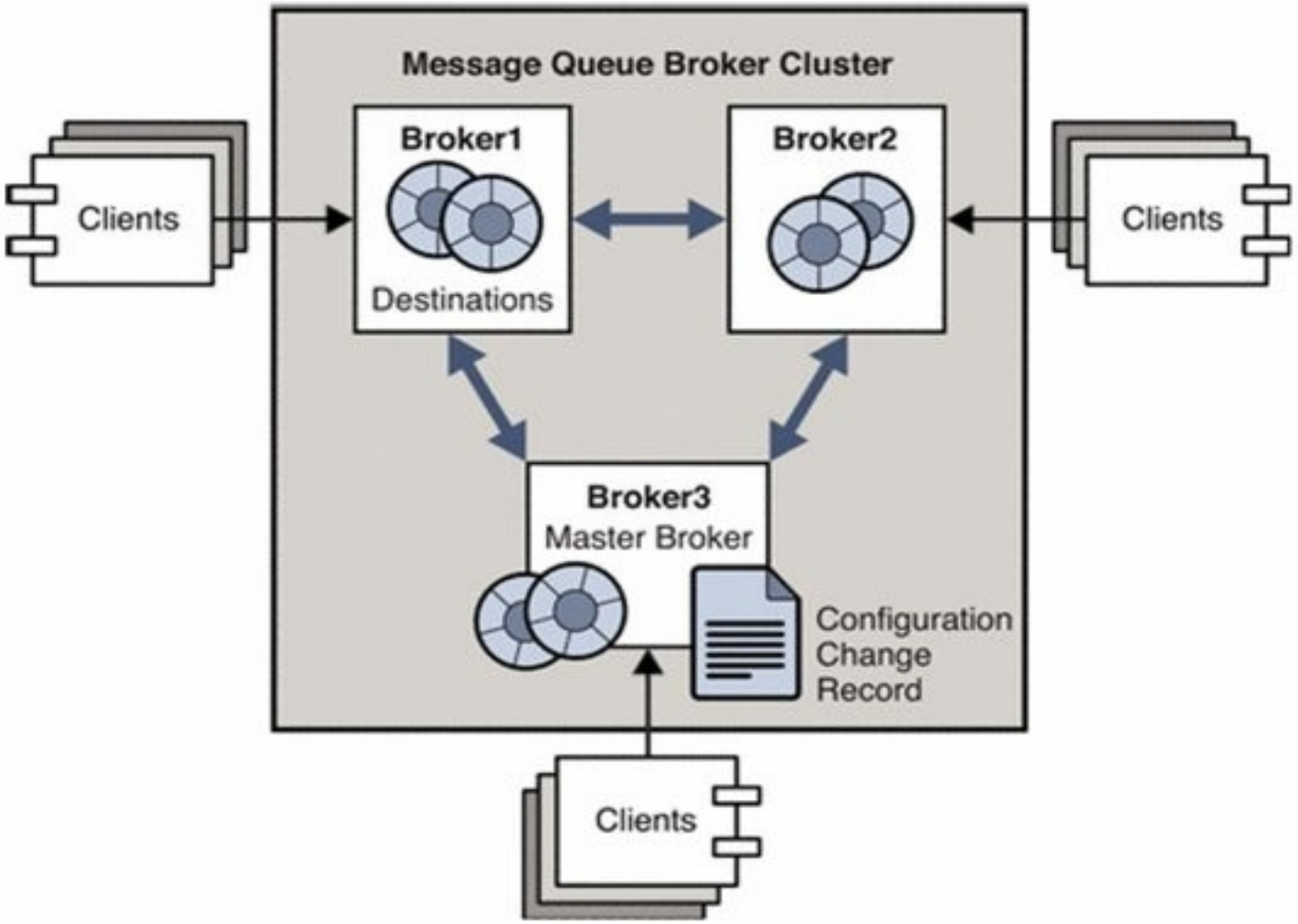
OpenMQ Features (Contd.)

- GUI installer or zip, file-based install
  - GUI installer based on Open Installer
  - Interactive and script based install supported
- Other APIs
  - C-client API
  - SOAP / HTTP
- Administration Tools
  - Command line
  - GUI administration tools
  - JMX Monitoring
- JCA 1.5 compliant resource adapter

Agenda

- Introduction to JMS
- What's OpenMQ
- Features of OpenMQ
- Demo

OpenMQ Cluster Demo



Key points for Demo

- Create a common cluster.properties file
  - imq.cluster.brokerlist=host:port,host1:port
  - imq.cluster.masterbroker=host:port
- Modify configuration file of each broker
  - $broker_instance_root/props/config.properties
  - imq.cluster.url=file:////var/cluster.properties

Call for Action

Want To Participate?

- Join the community
  - http://mq.dev.java.net
- Download the latest build
  - https://mq.dev.java.net/downloads.html
  - GlassFish or Open MQ
- Give us feedback on features
- Find a bug?
  - Use Issuetracker at mq.dev.java.net

Want More Information?

- Complete documentation is available
  - **http://docs.sun.com/app/docs/coll/1307.3**
- Free Online training courses are available
- IE-mail us at
  - users@mq.dev.java.net
- Forum at
  - http://forum.java.sun.com/forum.jspa?forumID=

Thank you!

Jason.Huang@Sun.Com