OOI CYBERINFRASTRUCTURE
NSF OCEAN OBSERVATORIES INITIATIVE

OOI – CyberInfrastructure

Common Operating Infrastructure (COI)

Technology Evaluation - AMQP

WORKING DRAFT

June 2008

NSF · Joint Oceanographic Institutions · UCSD

Document owner: OOI CI Architecture & Design Team

Document History

Date	Version	By	Description of Changes
2008-02-13	0.1	Emilia Farcas	Added an AMQP Overview, AMQP Domain Models, and some context for the Communication Infrastructure technologies.

Table of Contents

OOI - CyberInfrastructure

Common Operating Infrastructure (COI)

Technology Evaluation - AMQP

1 Context

Interoperability There are several dimensions of interoperability. For example, JMS defines API-level interoperability, whereas TCP and IP protocols define binary-level interoperability. However, there is a need also for message-level interoperability. Web services address the core interoperability issue by using standard communication protocols (HTTP/SOAP), data marshaling (XML), and interface description (WSDL) technologies. For event notification systems, the Web-services specifications WS-Eventing [1] and WS-Notification [2] define publish-subscribe interfaces between clients and brokers, but do not address the interoperability among notification brokers from different organizations. JMS attempts to mask the differences between publish-subscribe and store-and-forward systems, but it is exclusively for Java applications and provides only an interface standard; JMS implementations provide their own protocols. The Advanced Message Queuing Protocol (AMQP) [4] aims to fill in the interoperability gap in current Web services and message specification for asynchronous messaging. Section 2 presents an overview and domain models for AMQP.

Performance XML messages provide both semantic and structural information on the data, but require more bandwidth and computational-intensive parsing. Binary protocols are more efficient. In general, content-based message filtering requires more processing time than topic-based filtering, but content-based XML filtering is much more expensive.

Scalability Can be achieved via the separation of concerns via a layered architecture, for example. Moreover, by separating message distribution/routing from message filtering/matching, each one can be scaled separately.

2 AMQP

AMQP [4] defines a standards-based messaging infrastructure for queue-based messaging. AMQP specifies both a binary-wire protocol and a model that messaging brokers need to implement. The protocol can be implemented on top of network transports such as TCP and can be used in different programming environment and operating systems. AMQP targets application domains with high demands on reliability, performance, and publish-subscribe capabilities. AMQP encompasses and goes beyond JMS semantics.

AMQP is split into three layers: transport, session, and model. The model layer specifies the routing and queuing services. The session layer provides reliable transport, synchronization, and error handling. The transport layer is a binary protocol that provides framing, channel multiplexing, failure detection, and data representation.

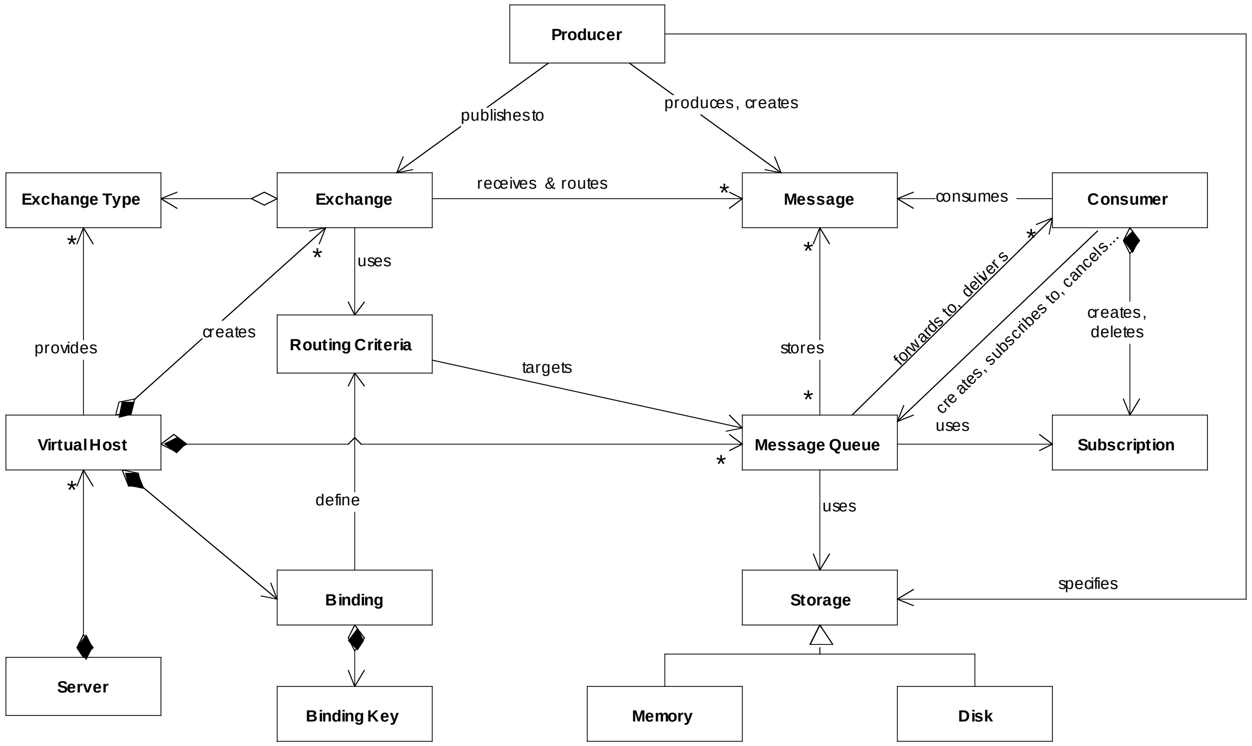

Figure 1 Domain model of the AMQP Model layer

The AMQP Model (Figure 1) decouples producer and consumer applications via three main concepts: exchanges, message queues, and bindings. A middleware server can provide several virtual hosts, which have to implement the components of the AMQP Model. Producers publish messages to exchanges, and consumers get messages from message queues. Exchanges accept messages, examine them, and route them to the appropriate queues. The binding between exchanges and message queues defines the routing criteria; thus, exchanges abstract different middleware delivery models. AMQP supports, among others, direct point-to-point, store-and-forward, and publish-subscribe message delivery. On one hand, the producers choose which exchange should route their messages. On the other hand, the consumer subscribe to message queues. The bindings are the arguments passed to the exchanges for instructing the exchange what messages to route into the queues, but it is not always clear which entity should provide the binding information.

AMQP messages are self-contained and long-lived, and AMQP does not impose restrictions on their size. The binding key specifies the matching criteria, and the routing can be done based on message headers (routing key or other properties) or message content. Exchanges do not store messages and can duplicate messages to several queues. Queues store messages in memory or on disk as requested by the consumer, and can also search and reorder messages. AMQP aims at providing reliable, pervasive, fast, and secure shared access to a distributed network of message queues [3].

Sessions (see Figure 2) are interactions between AMQP peers that provide reliability: guaranteed command execution, recovery from network failure, and reconciliation of state when peers fail. After network failures, the session layer is responsible with replaying the appropriate commands, without duplicating delivery; this involves a negotiation between the peers. To achieve reliability, the session maintains its state while the session is detached.

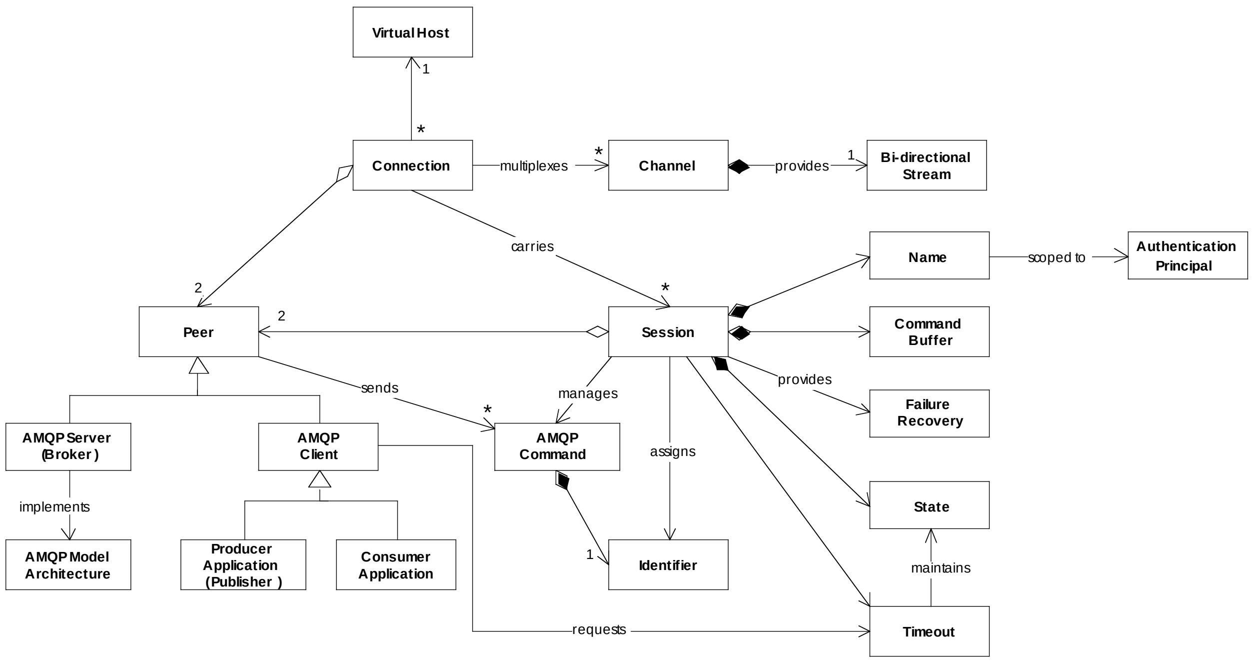

Figure 3 Session Layer

AMQP supports multiple virtual networks within the same physical network (see Figure 3). All virtual hosts in a server share the same authentication scheme, but the authorization can be different. All channels within a connection work with the same virtual host.

AMQP assumes a stream-based protocol such as TCP underneath. The transport layer transmits sequential frames over channels, and supports multiple channels on one connection. The framing is depicted in Figure 4. AMQP has a three-level structure, with an Assembly encoding commands and data content, and Frames being the unit sent on the network. Assemblies and Segments have no size limit, but Frames are limited by the underlying transport mechanism.

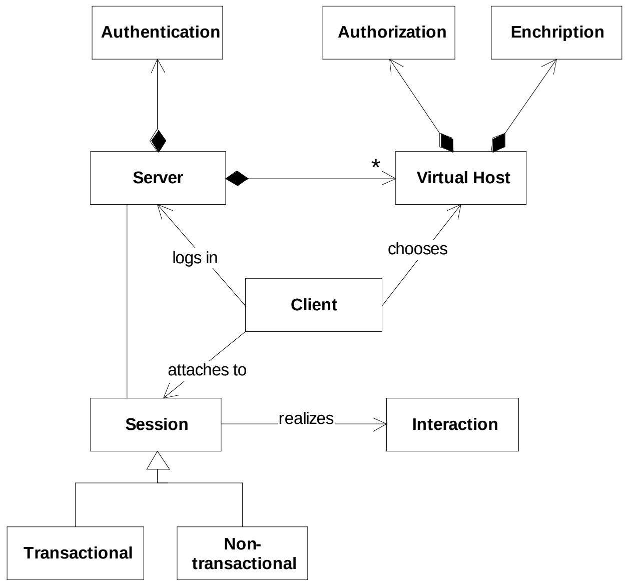

Figure 2 Virtual Hosts

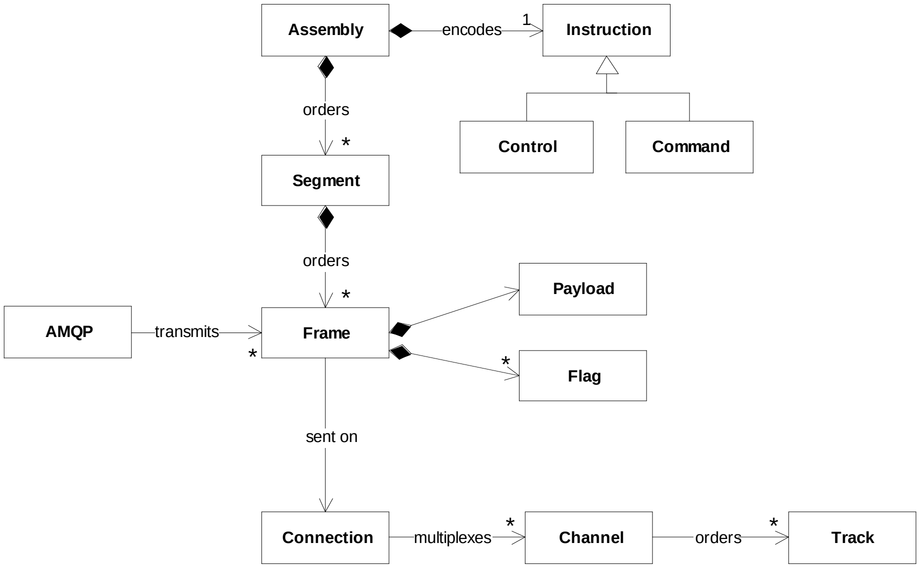

Figure 4 Transport layer

3 Appendices

3.1 Abbreviations

Abbreviation	Meaning
AMQP	Advanced Message Queuing Protocol

3.2 References

Reference	Citation
[1]	D. Box et al. Web Services Eventing. Available from: http://ftpna2.bea.com/pub/downloads/WS-Eventing.pdf.
[2]	OASIS. WS-Notification (v1.2). Available from: http://docs.oasis-open.org/wsn/2004/06/.
[3]	Pieter Hintjens, AMQ Background - Background to the AMQ Project. Available from: http://www.openamq.org/doc_background.txt_flat.html
[4]	AMQP, Advanced Message Queuing Protocol specification, version 0-10.