AMQP (Advanced Message Queueing Protocol)

Pranta Das
pranta.das@cisco.com
Senior Technical Leader,
Cisco Systems, Inc., Founding Member, AMQP Working Group

**13th International Workshop on High Performance Transaction Systems (HPTS),
October 25th – 28th, 2009, http://www.hpts.ws**

Genesis of AMQP

Protocol born out of frustration

- Conceived by JPMorgan in 2006
- Goal was to make Message Oriented Middleware (MOM) pervasive, practical, useful, interoperable
- Brought together users and vendors to solve the problem

MOM needs to be *everywhere* to be useful

- dominant solutions are proprietary
  - too expensive for everyday use (Cloud-scale)
  - they don't interoperate at the wire-protocol level although API-level compatibility (like JMS) exists.
- has resulted in lots of ad-hoc home-brew

Answer to current Middleware Hell

- 100's of applications
- 10,000's of links
- every connection different
- massive waste of effort

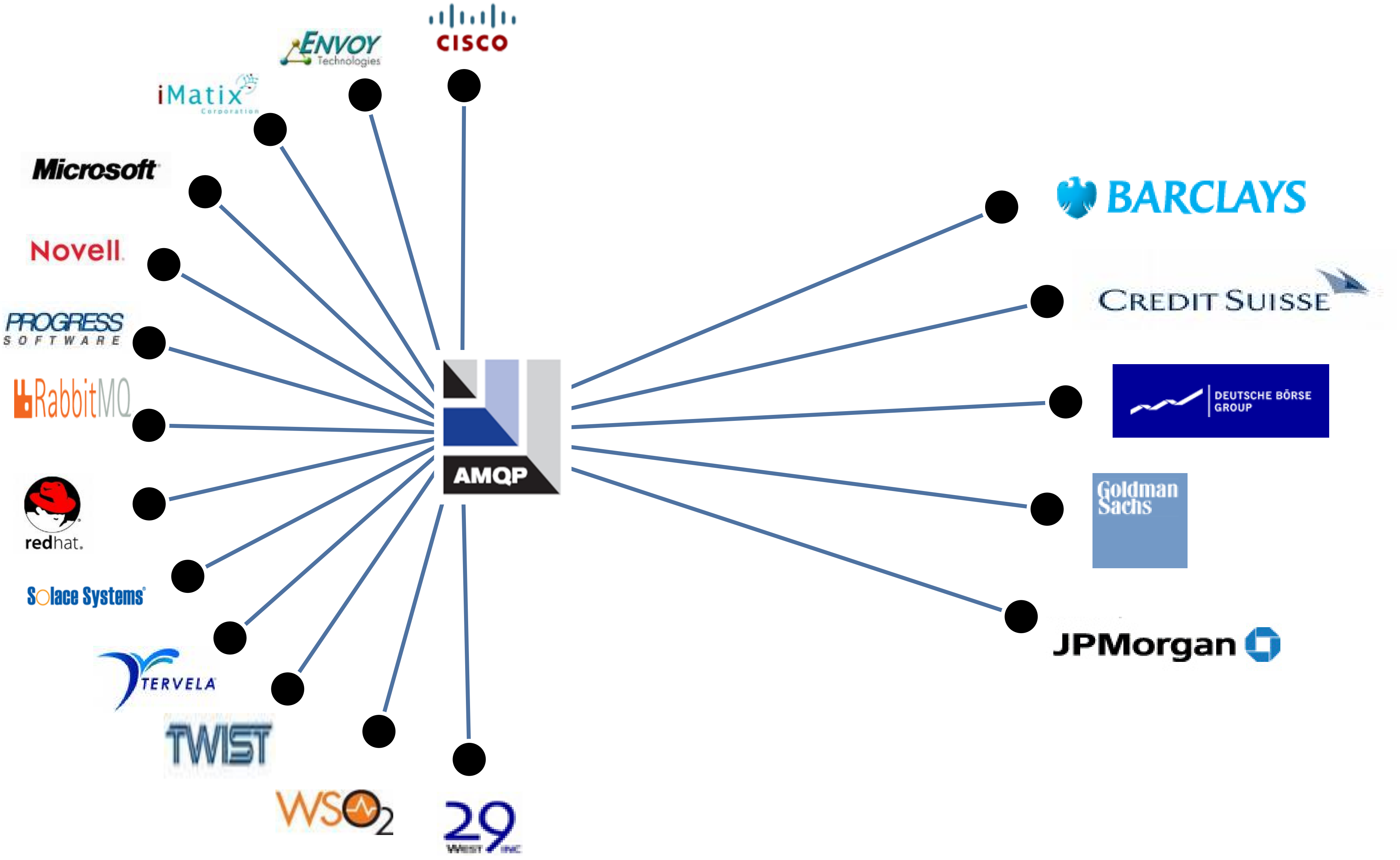
The Internet's missing standard

Open Ubiquitous Internet Standard Protocols		
Reliability ⇨ ⇩**Synchronicity**	Unreliable	Reliable
Synchronous	HTTP	IIOP
Asynchronous	SMTP	**? (AMQP)**

- *Why has no one done this before?*

http://www.amqp.org

AMQP Working Group vendor and user firms

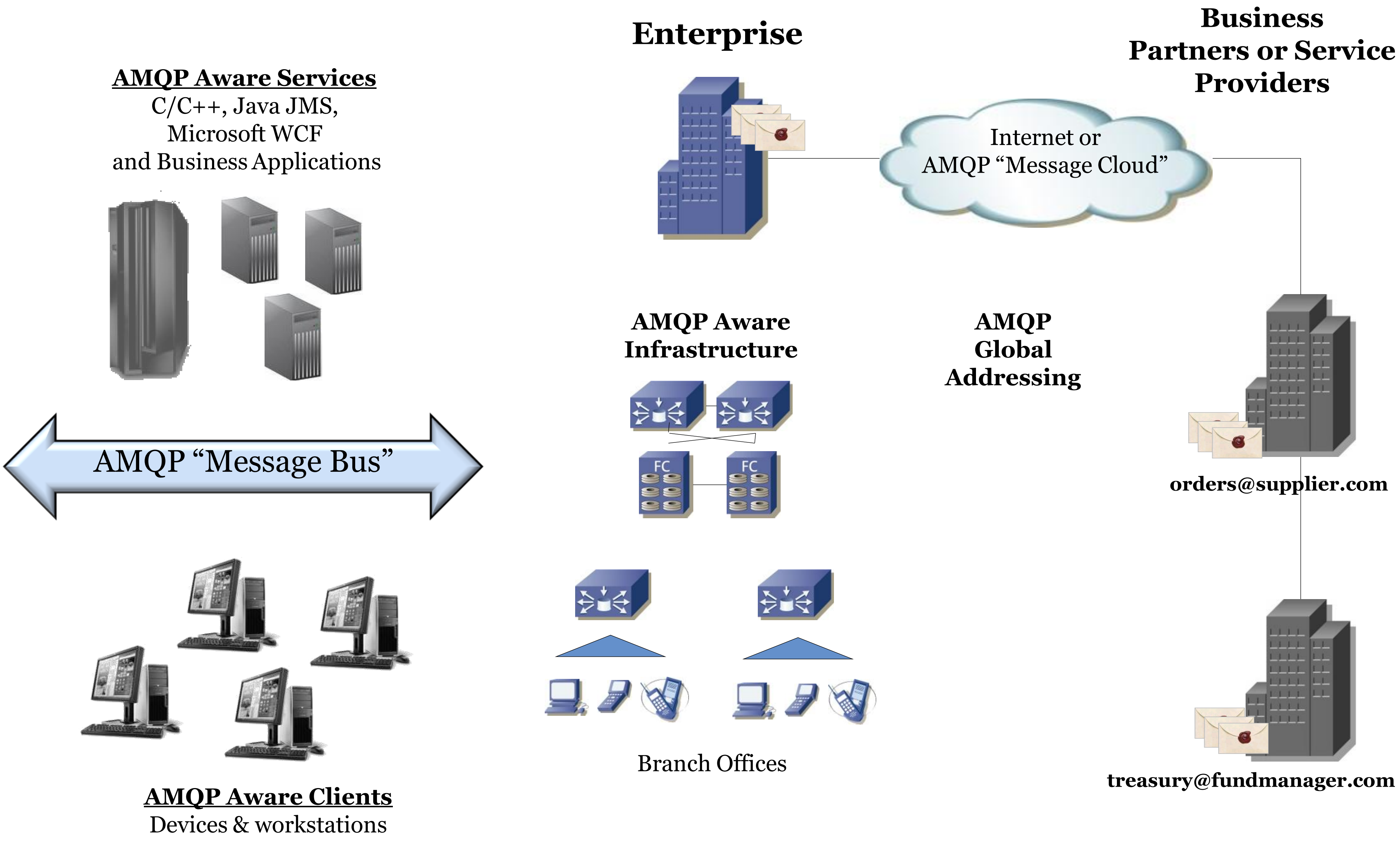


What is AMQP?

AMQP is a Message Oriented Middleware (MOM) protocol that:

- Transfers application data units from senders to receivers – layer 7
- Sets an expectation that the message transfer is via trusted intermediaries:
  - An expectation that messages will be delivered unchanged
  - An expectation of security
  - Applications can be separated by (large amounts) of space and time
- Is abstract from the underlying technology
  - Physical network limits should be hidden (message size, node location)
  - Technology concerns should be hidden (platform, language, OS)
- The intermediaries offer various delivery options, as defined by either the sender or the receiver (s)
- The intermediaries provided various defined qualities of service for the sender and the receiver (s)
- Provides stability and backwards compatibility (10yrs+)

AMQP Vision



Enterprise

Business Partners or Service Providers

AMQP Aware Services
C/C++, Java JMS,
Microsoft WCF
and Business Applications

Internet or
AMQP "Message Cloud"

AMQP "Message Bus"

AMQP Aware Infrastructure

AMQP Global Addressing

orders@supplier.com

Branch Offices

treasury@fundmanager.com

AMQP Aware Clients
Devices & workstations

Elements of an AMQP Overlay Network

Broker

- Applications Connect to a Broker to participate in the AMQP network
- The Connection is used to establish a Session
  - Sessions provide state between Connections, establish identity, ease failover
- Connections are further subdivide into Channels
  - Multiple threads of control within an Application can share one Connection

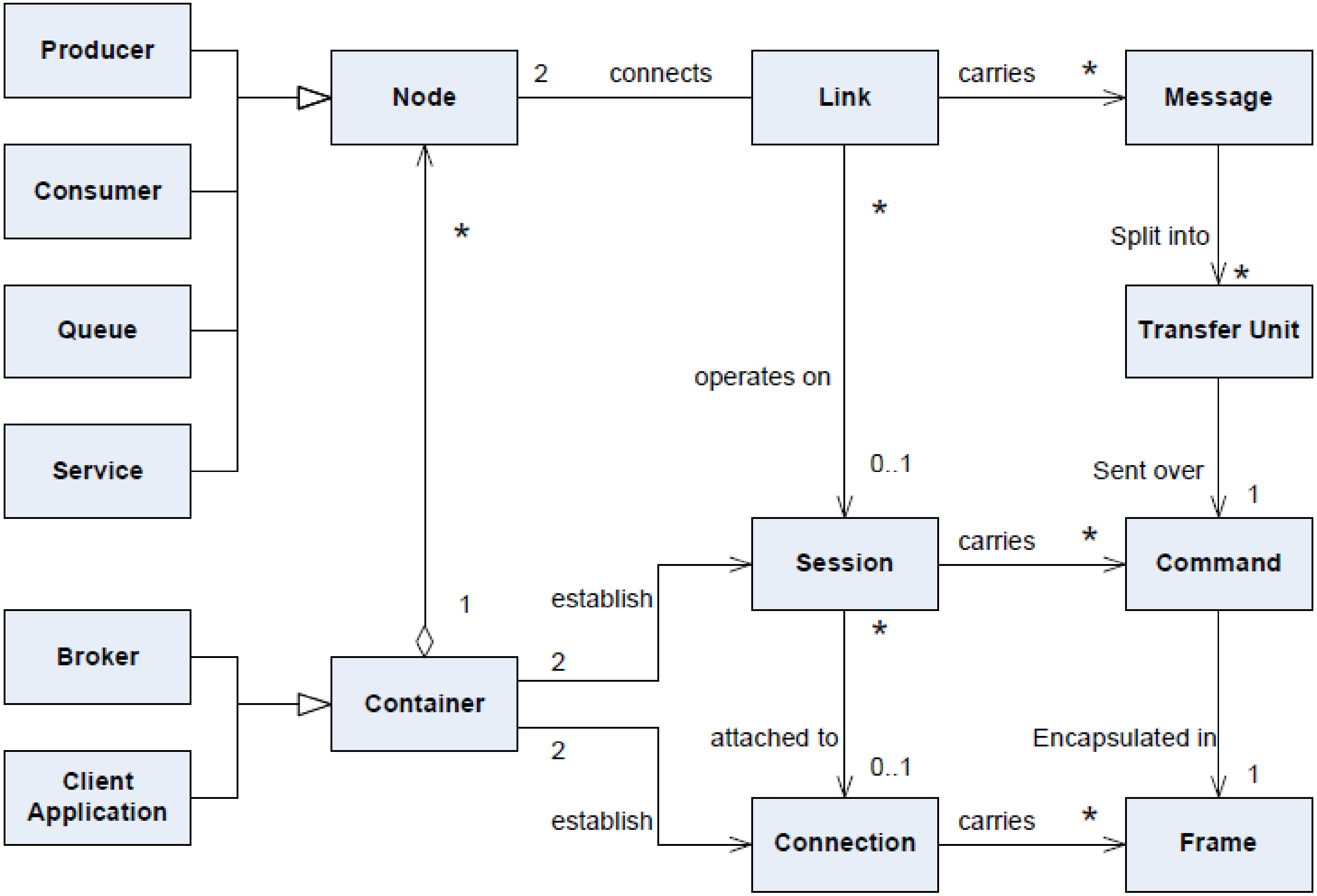
Queues

- Applications logic interacts ONLY with Queues
- Queues have well known Names == Addressable
- Applications do not need to know how messages get in/out of Queues
- Queues can be smart, they are an extension point
- Applications will assign implied semantics to Queues (e.g. “StockOrderQueue”)

Links

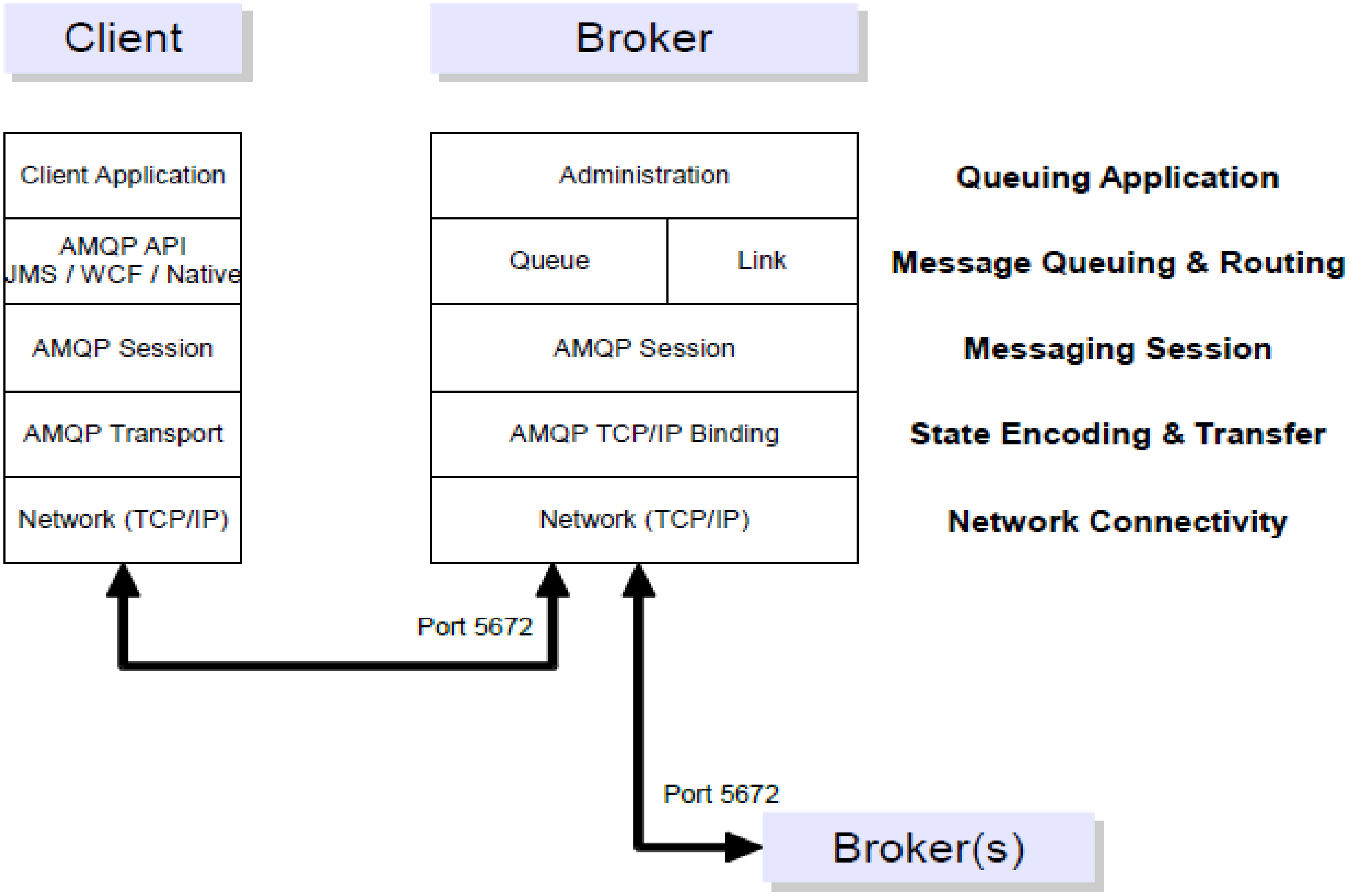
- Links move Messages between Queues and/or Applications
- Contain Routing and Predicate Evaluation Logic – similar to Complex Event Processing

AMQP 1.0 Protocol Information Model



http://www.amqp.org

AMQP Peer-to-Peer Layering



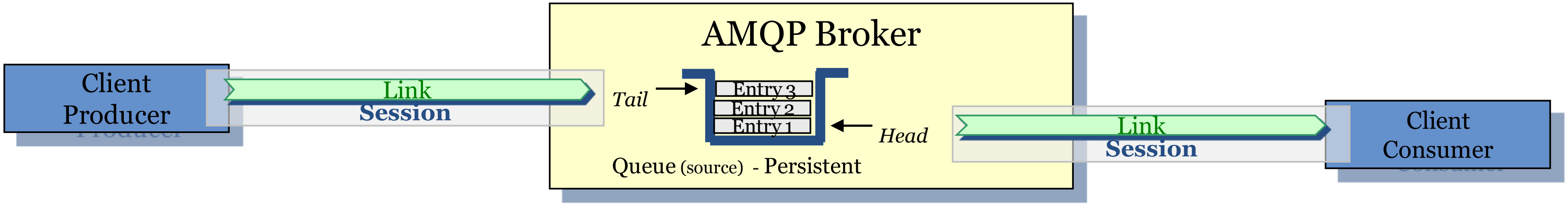
Client	Broker	
Client Application	Administration	Queuing Application
AMQP API JMS / WCF / Native	Queue Link	Message Queuing & Routing
AMQP Session	AMQP Session	Messaging Session
AMQP Transport	AMQP TCP/IP Binding	State Encoding & Transfer
Network (TCP/IP)	Network (TCP/IP)	Network Connectivity

Port 5672

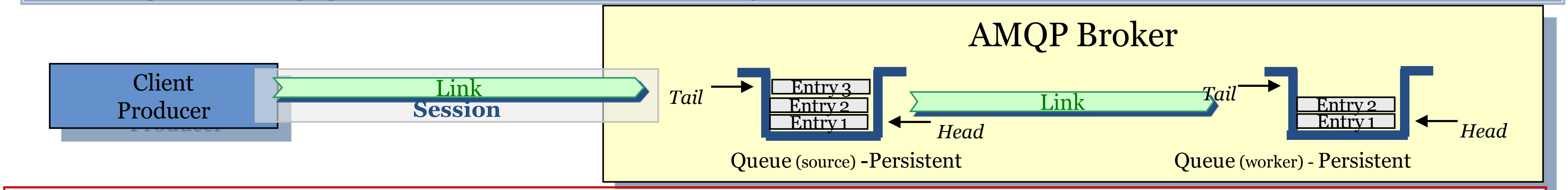
Port 5672

Broker(s)

AMQP Usage Patterns – Point-to-Point

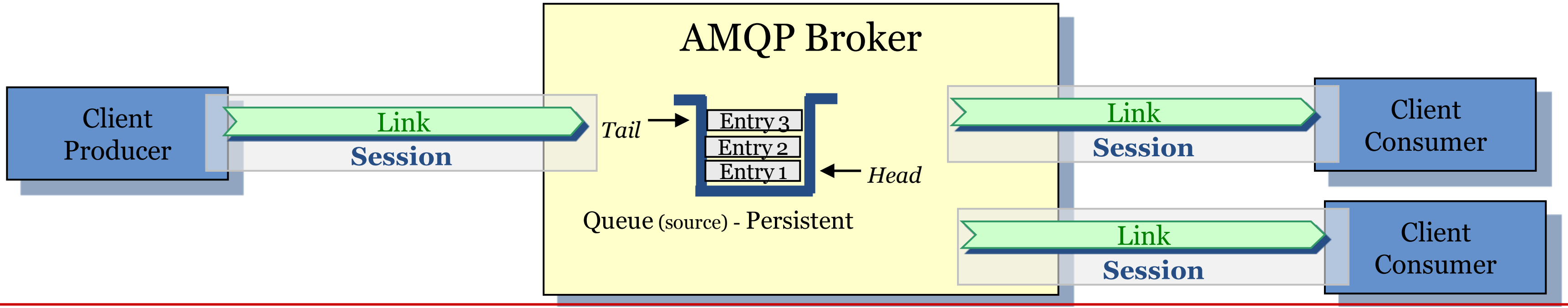


1. Point-to-point Queue Delivery: Only “Source” queue is required and can be read directly by consumer over Link (i.e. dedicated consumer Worker queue and bridging between Source and Worker unnecessary).



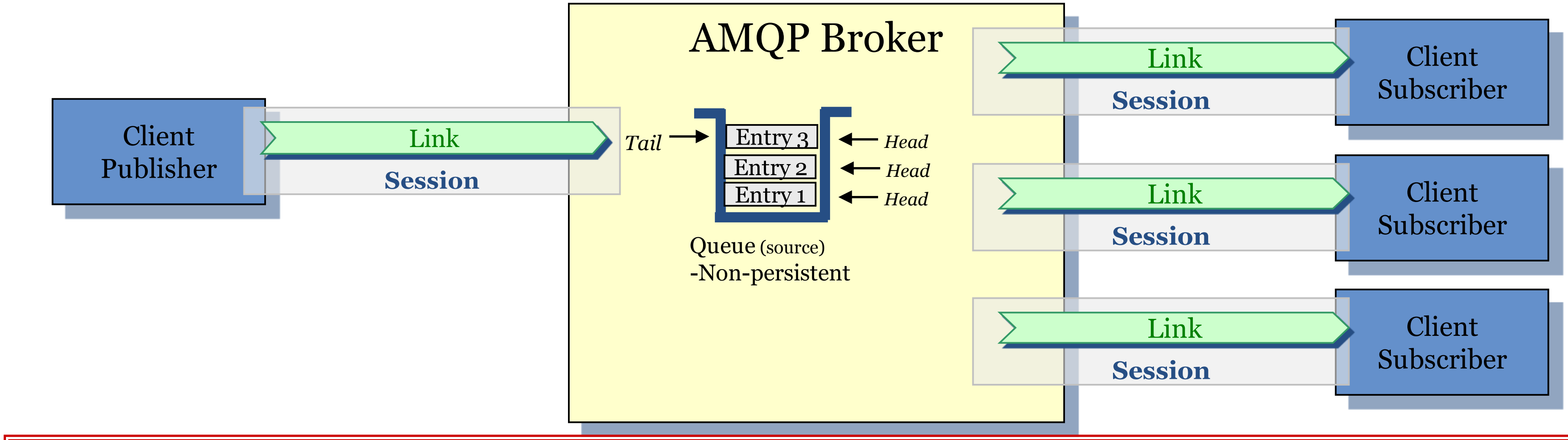
2. Abstracted Point-to-point Queue:

- One Queue performs the role of holding the “Well Known” name for the outside world. All messages are automatically forward on to the real worker queue. Allows internal topology to change without the outside world seeing (this PO Box)

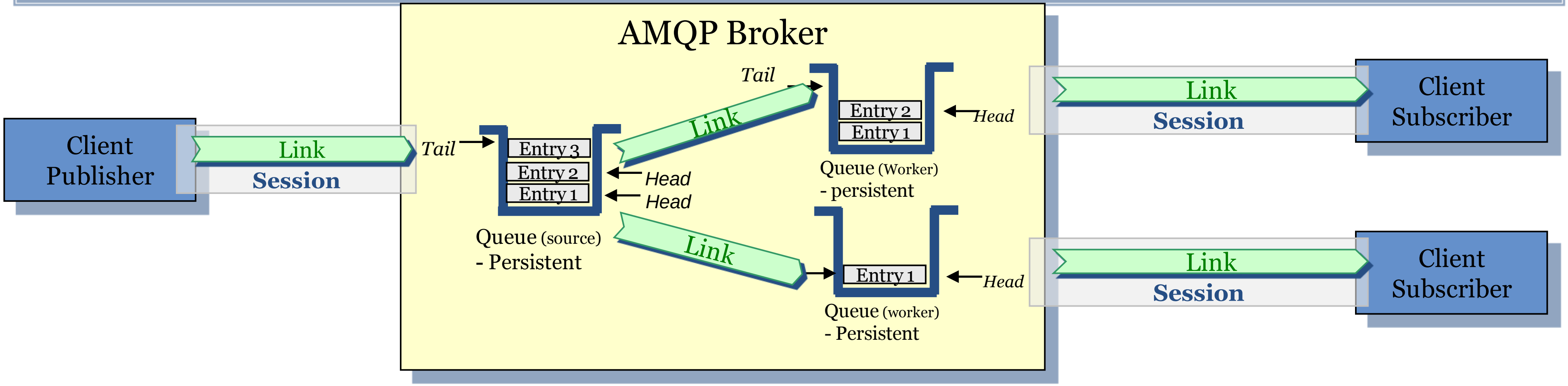


3. Load-Balanced Point-to-point Queue Delivery: Each message will be delivered to a separate consumer, based on some load-balancing scheme, such as round-robin, etc.

AMQP Usage Patterns – Publish-Subscribe



4. Dynamic (non-persistent) Pub/Sub Delivery: Messages are "garbage collected" in an implementation specific manner.



5. Durable (persistent) Pub/Sub Delivery: Messages are reliably delivered to each durable subscriber.

http://www.amqp.org

Prominent AMQP Broker Implementations

- Apache Qpid: Open Source AMQP Messaging (http://qpid.apache.org/)
  - Java and C++ implementations
- OpenAMQ (http://www.openamq.org/)
  - C++ implementation
- RabbitMQ (http://www.rabbitmq.com/)
  - Erlang implementation
- Red Hat's MRG (Messaging, Realtime & Grid) variant of the Qpid implementation (http://www.redhat.com/mrg/messaging/)
  - C++ implementation

Some interesting links

- AMQP website:
  http://www.amqp.org
- Wireshark dissector for AMQP
  http://wiki.wireshark.org/AMQP
- Design and Evaluation of Benchmarks for Financial Applications using AMQP over InfiniBand (*Hari Subramoni, et.al., Department of Computer Science and Engineering, The Ohio State University)*
  http://www.cse.ohio-state.edu/~narravul/papers/subramoni_whpcf08.pdf
- Intel brief: AMQP: Secure business messaging for the future
  http://www.twiststandards.org/index.php?option=com_docman&task=doc_download&gid=70&&Itemid=90
- CloudMQ: Message -queuing as a Service (MaaS)
  http://www.cloudmq.com/

Questions?

http://www.amqp.org