



# Building Large Scale Distributed Systems with AMQP

Ted Ross  
[tross@apache.org](mailto:tross@apache.org)

# Agenda

- What is AMQP?
  - Why is AMQP important to large distributed enterprises?
  - How is the Apache Community making AMQP a reality?
  - Wrap-up and Questions
- 

# What You will Take Away

- AMQP is More Than Messaging
  - AMQP is Complex and Capable
  - Apache Makes AMQP Accessible and Easy to Use
  - The Apache Software Foundation is on the Cutting Edge of Distributed Computing
- 

# Your Presenter

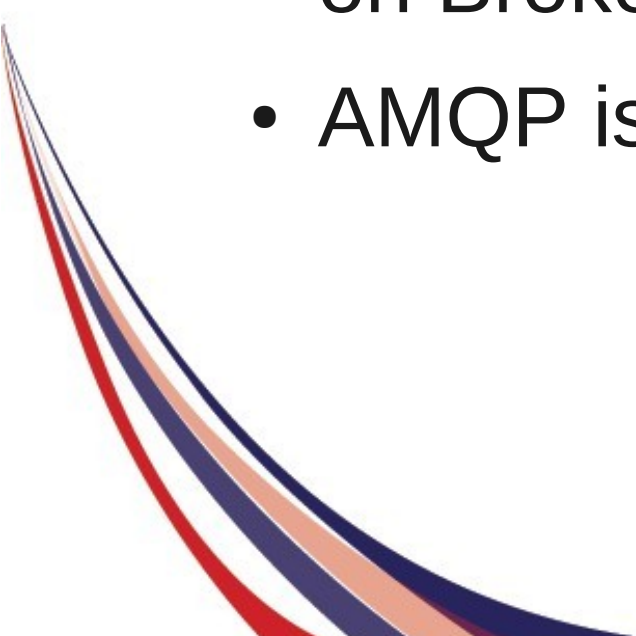
- Ted Ross  
Managing Principal Software Engineer  
Red Hat Inc. (Westford, MA)  
PMC Member and Committer
- Apache Qpid project  
MRG-M product
- Background in Computer Networking and  
Network Security

# What is AMQP

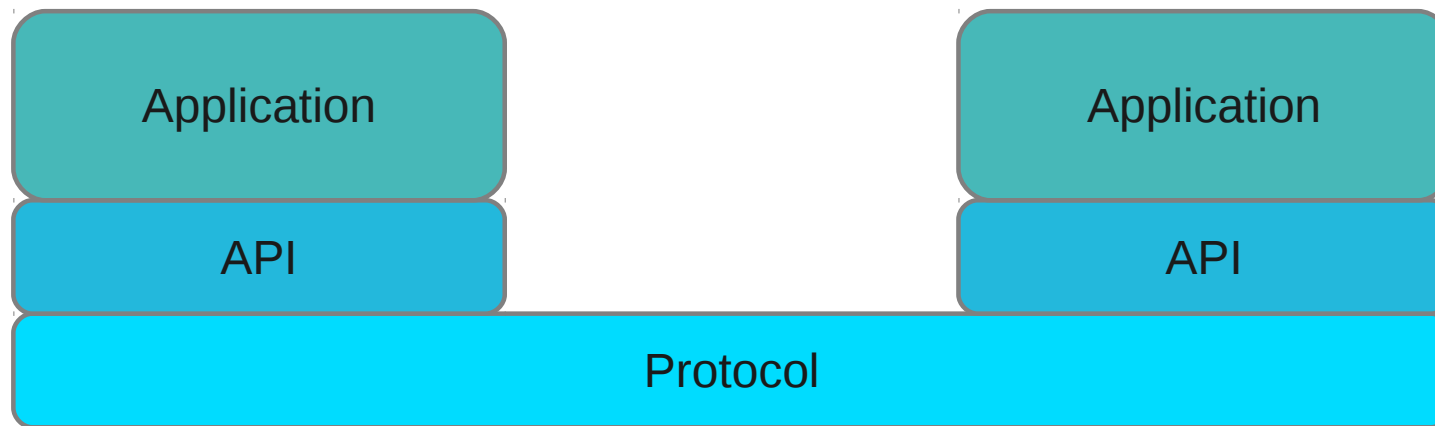
- Advanced Message Queuing Protocol  
“Open Internet Protocol for Business Messaging”  
<http://www.amqp.org>
- Developed
  - By Users and Vendors
  - In Financial Services and Other Industry Verticals
  - To address lock-in from proprietary messaging systems

# AMQP and MOM

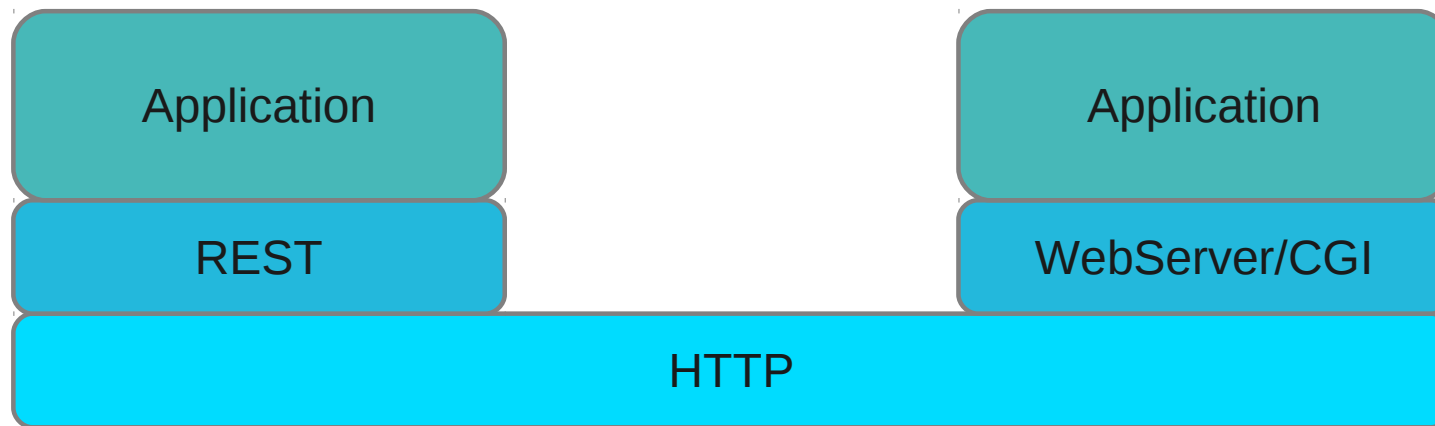
- AMQP Came from the Messaging Oriented Middleware (MOM) World
  - Early Versions based on Client and Broker, Defined Broker Behavior
- AMQP 1.0 is Symmetric and Mostly Silent on Broker Behavior
- AMQP is More than Messaging...



# Protocols and APIs

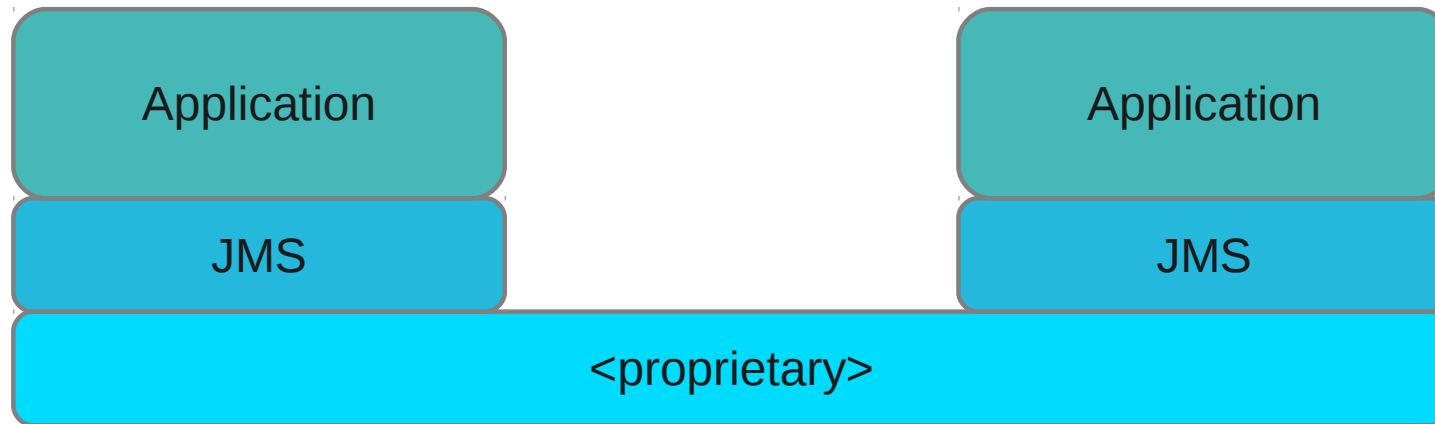


# Protocols and APIs

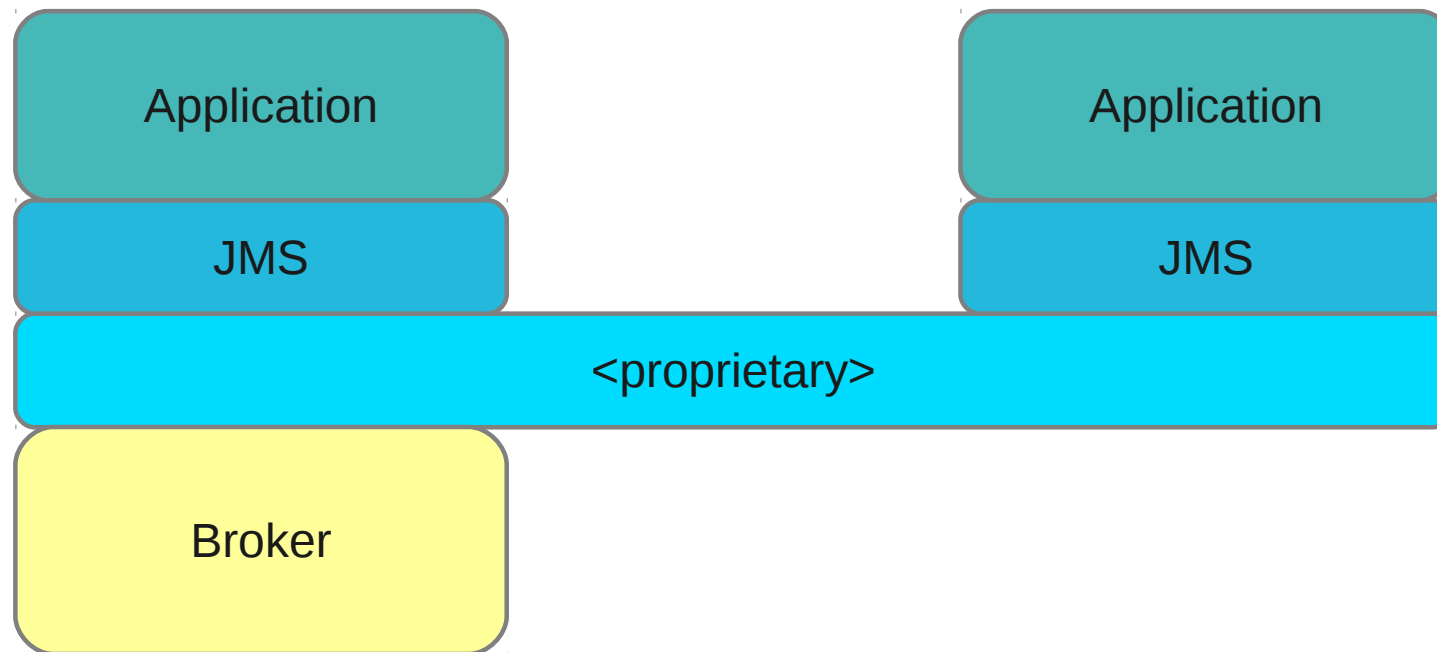




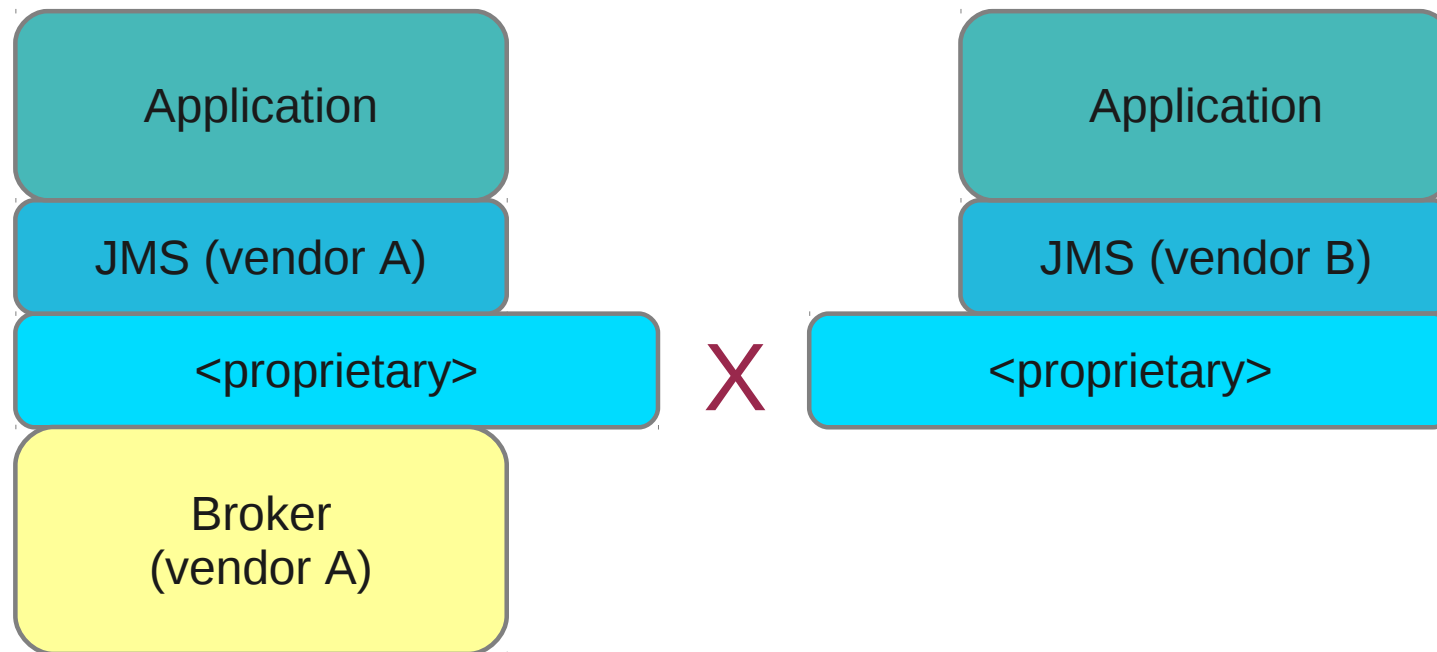
# Protocols and APIs



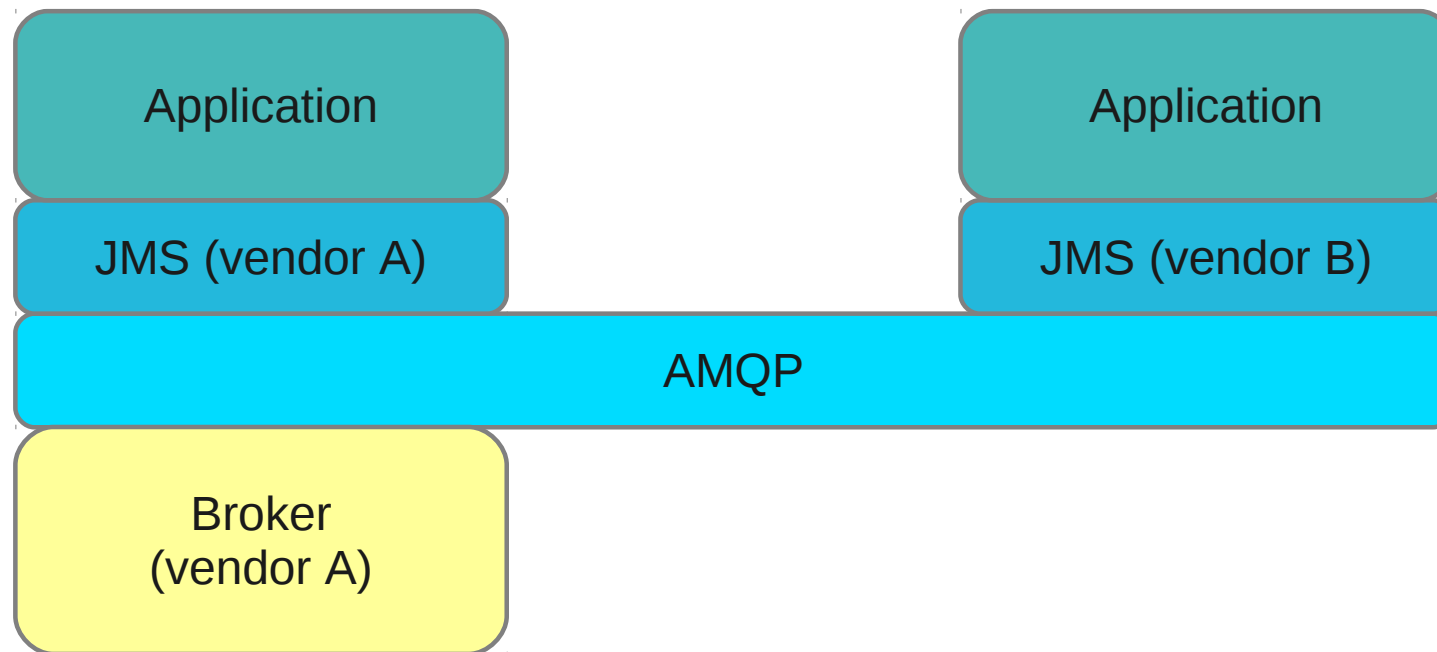
# Protocols and APIs



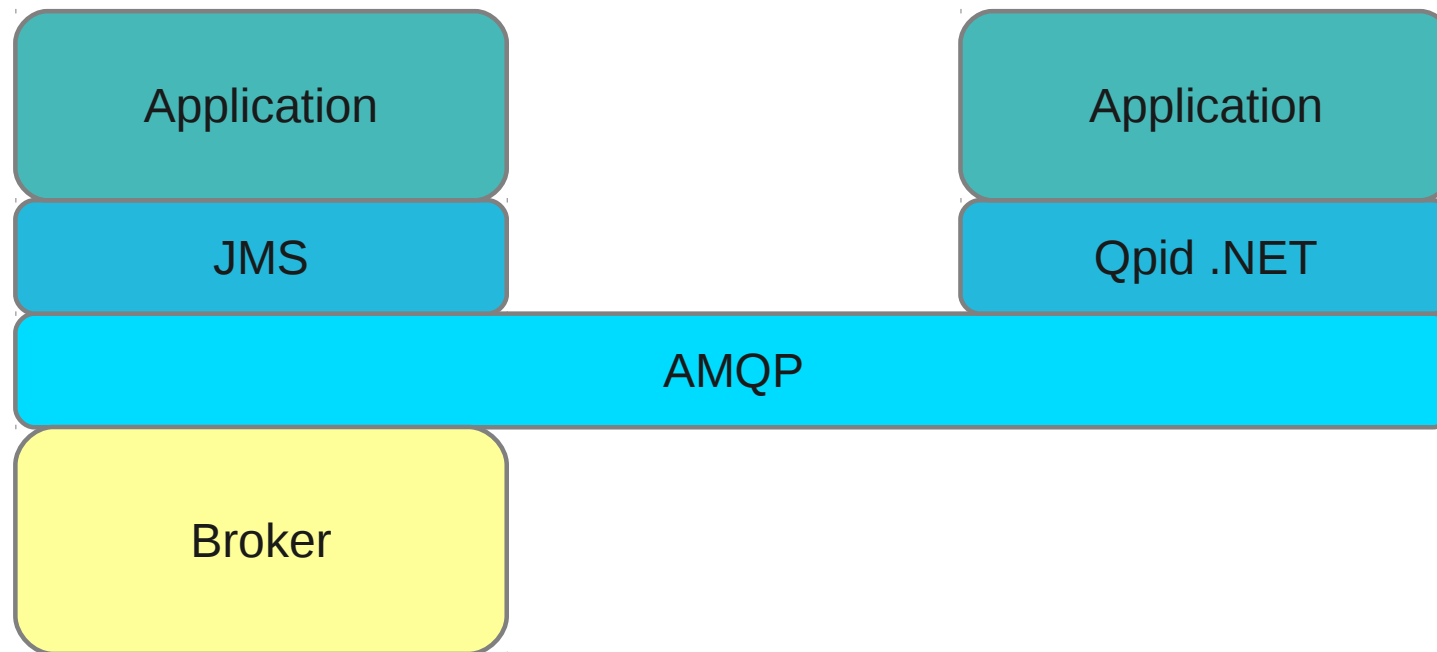
# Protocols and APIs



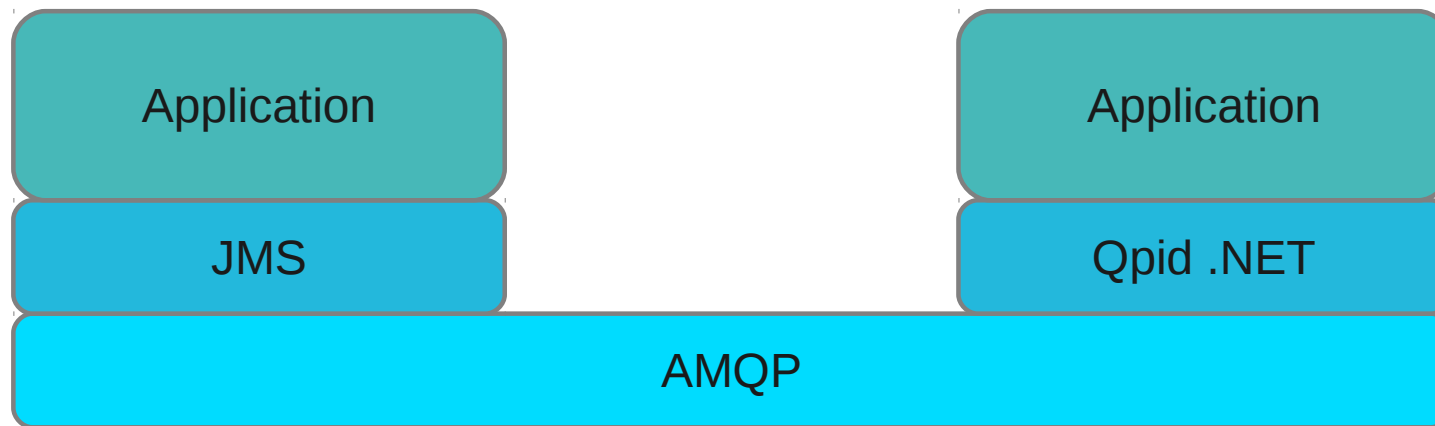
# Protocols and APIs



# Protocols and APIs



# Protocols and APIs

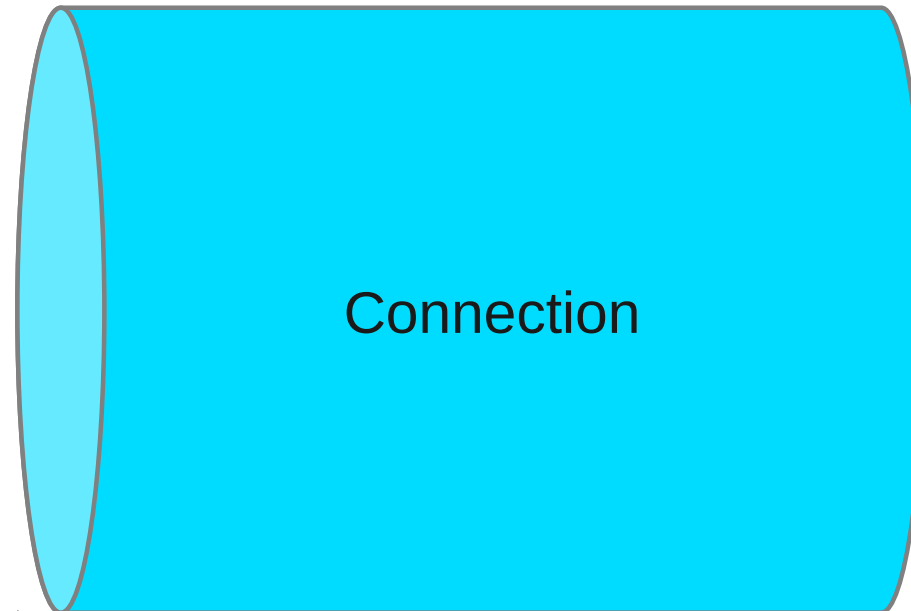


# Protocols to Contrast

- HTTP
- SMTP
- STOMP
- MQTT
- ...

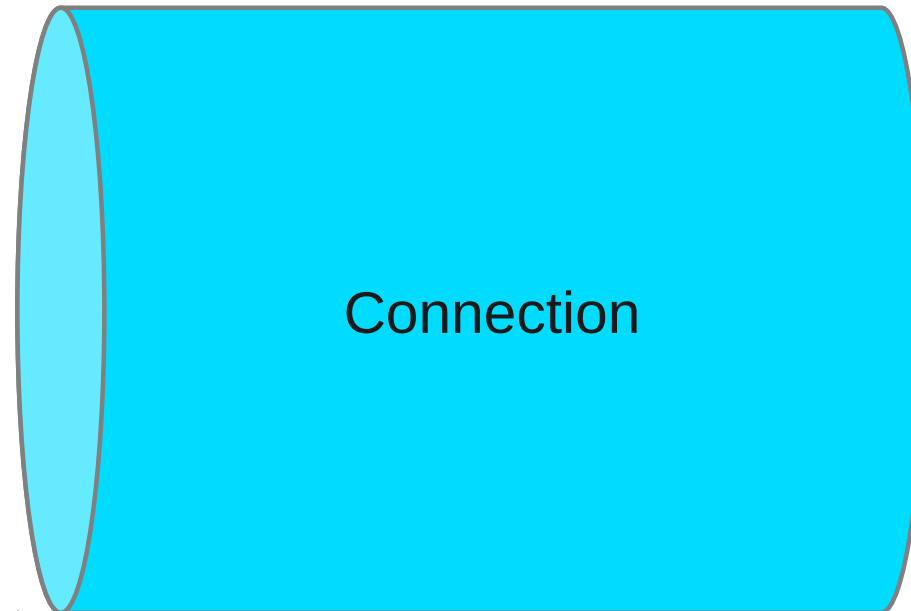


# AMQP Anatomy



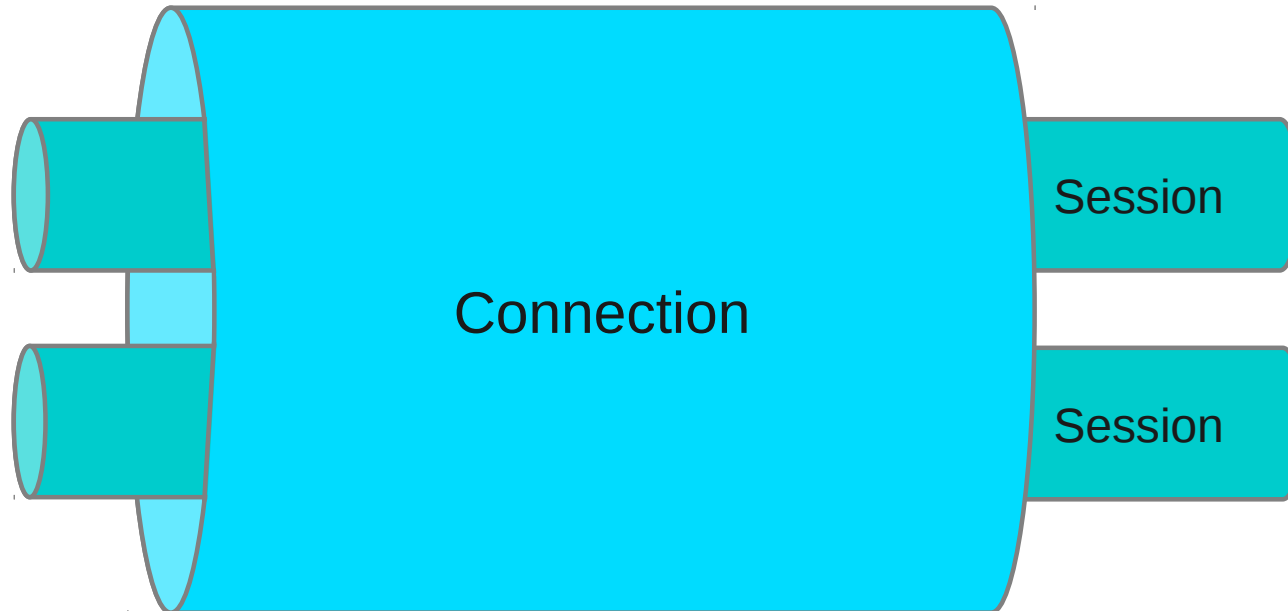


# AMQP Anatomy

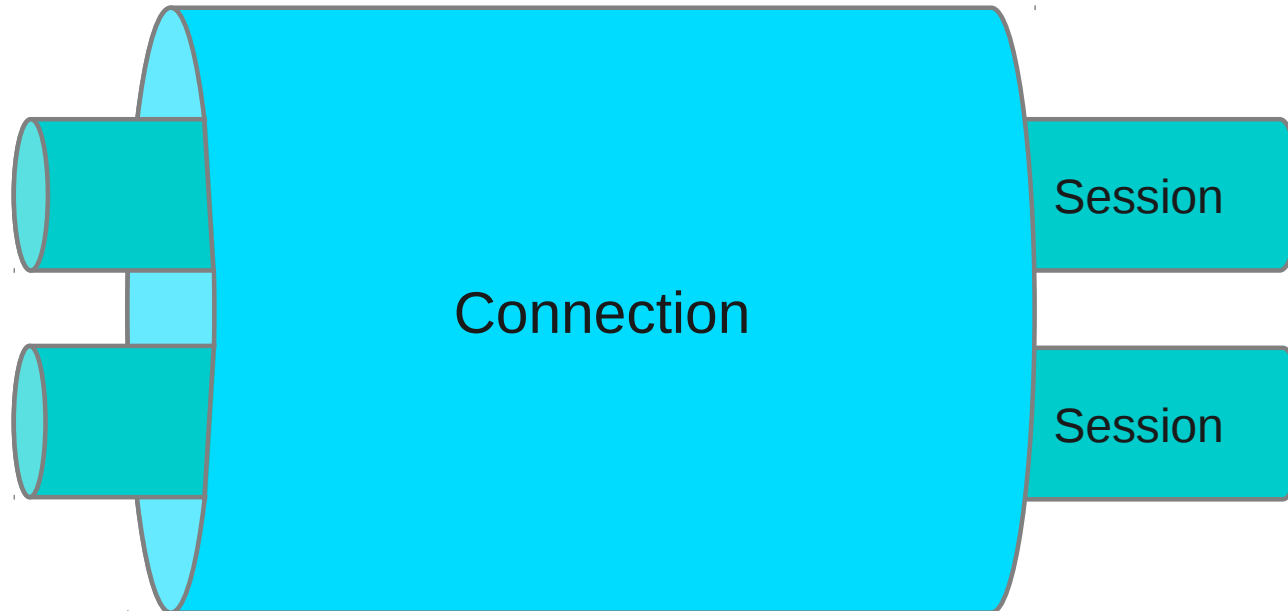


- Reliable Transport Connection
  - TCP, RDMA over Infiniband/10GigE, etc.
  - Transport-Level Security (SSL/TLS, SASL)

# AMQP Anatomy

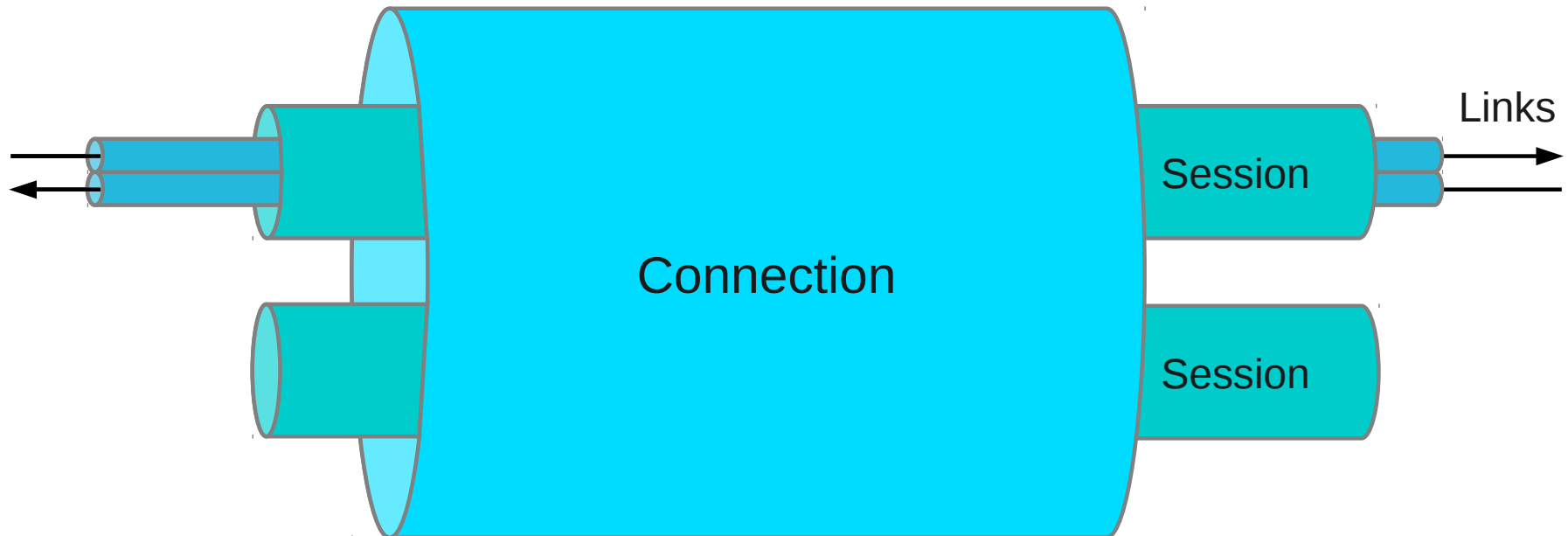


# AMQP Anatomy

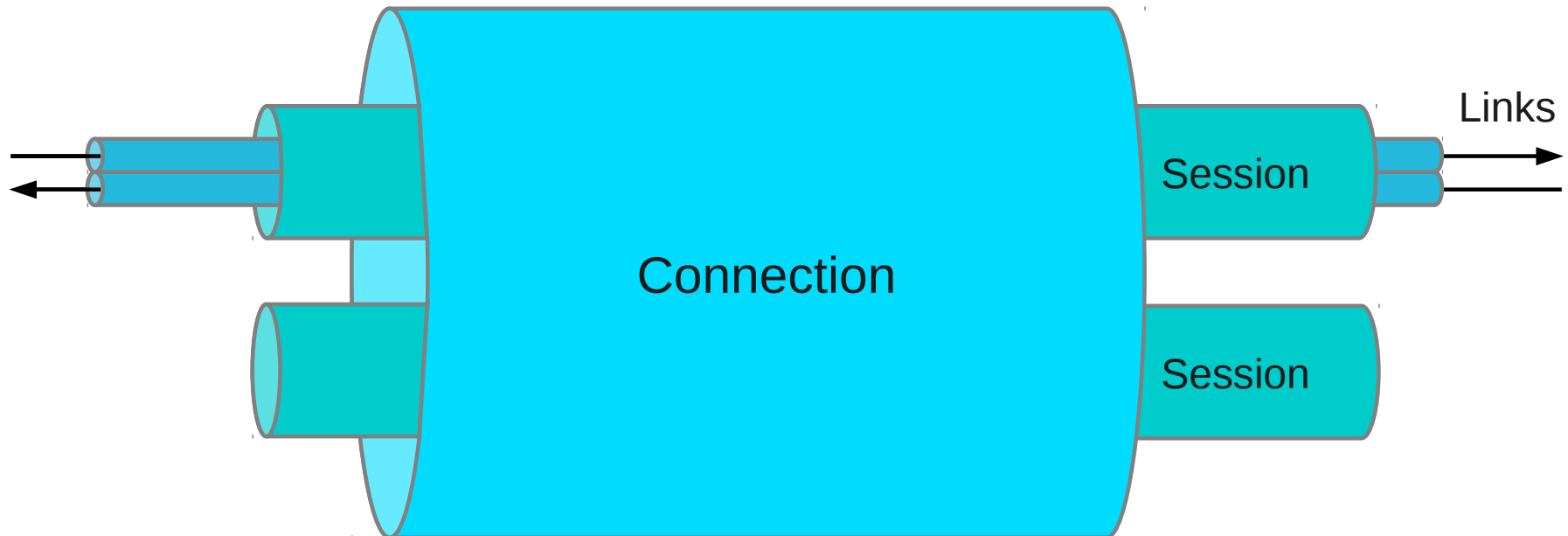


- Session Multiplexing
- Interleaving of Message Flow

# AMQP Anatomy



# AMQP Anatomy

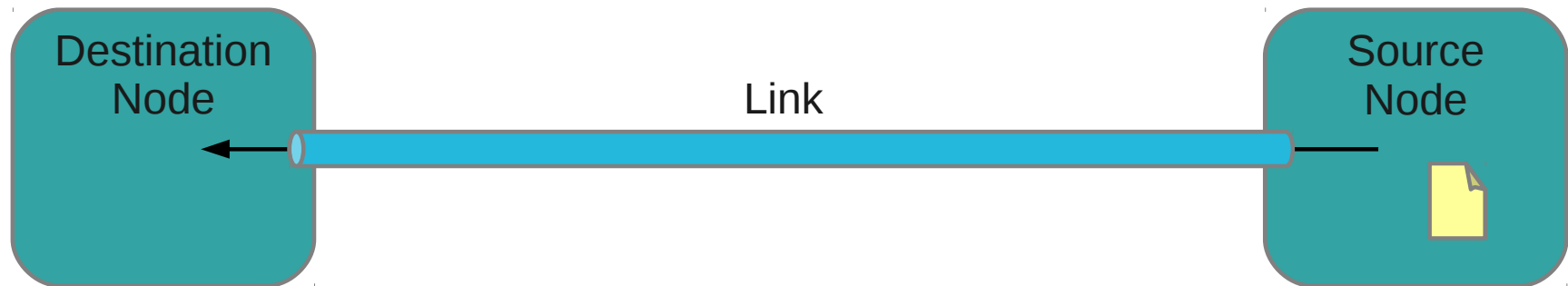


- Full Duplex Message Transfer
- Asynchronous Message Transfer
- Independent Flow Control

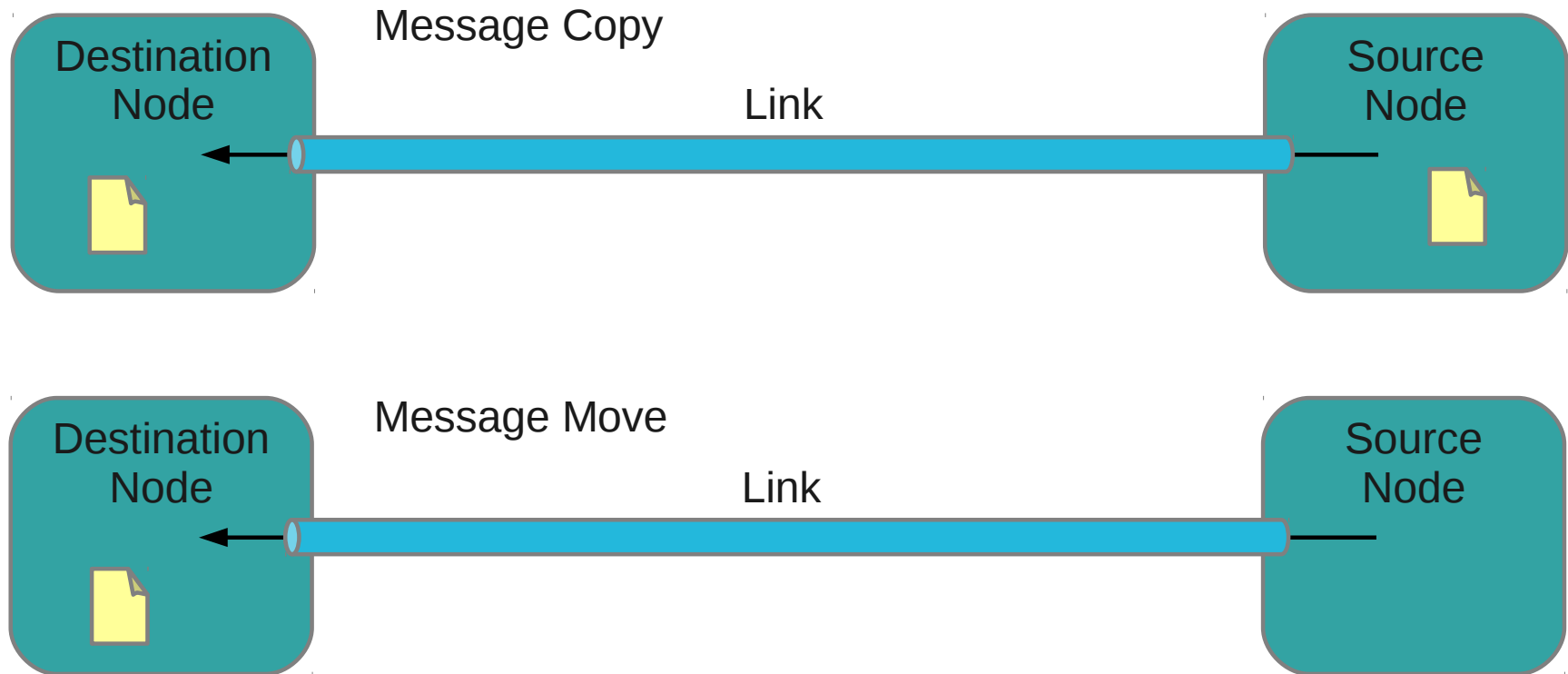
# Message Transfer



# Message Transfer

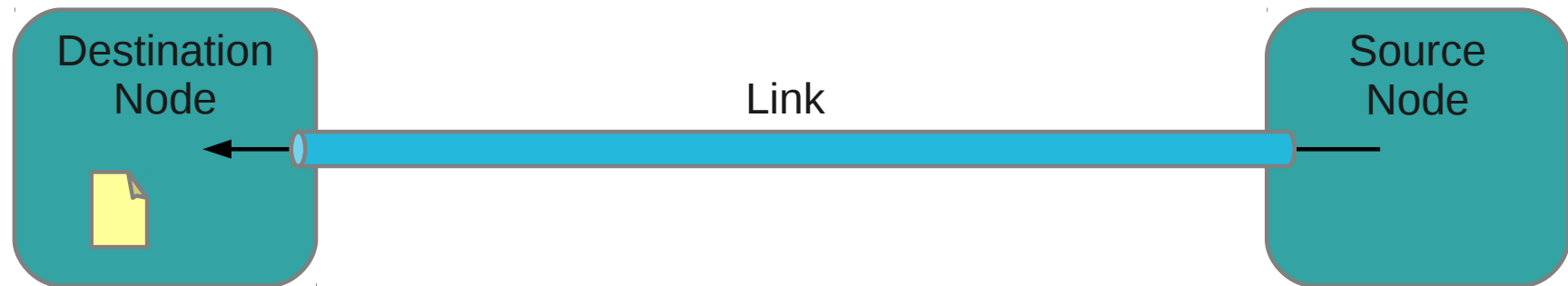


# Message Handoff – Copy/Move



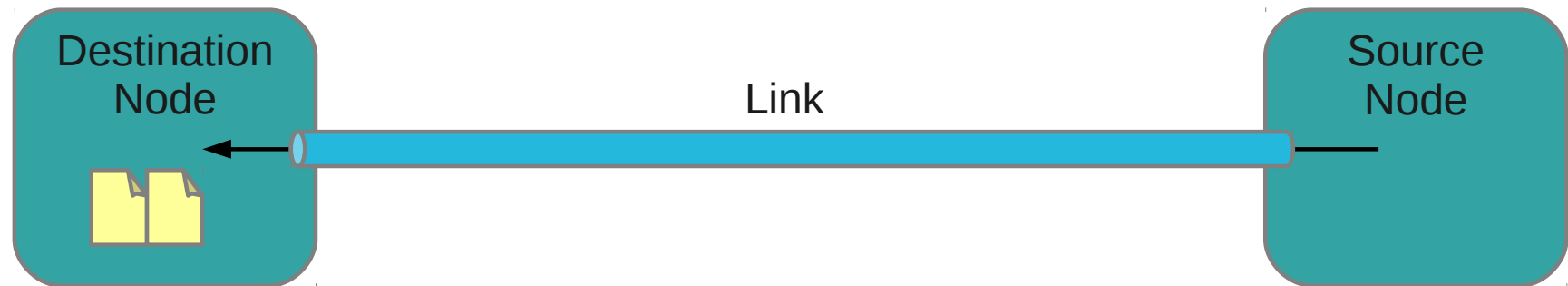


# Settlement and Disposition



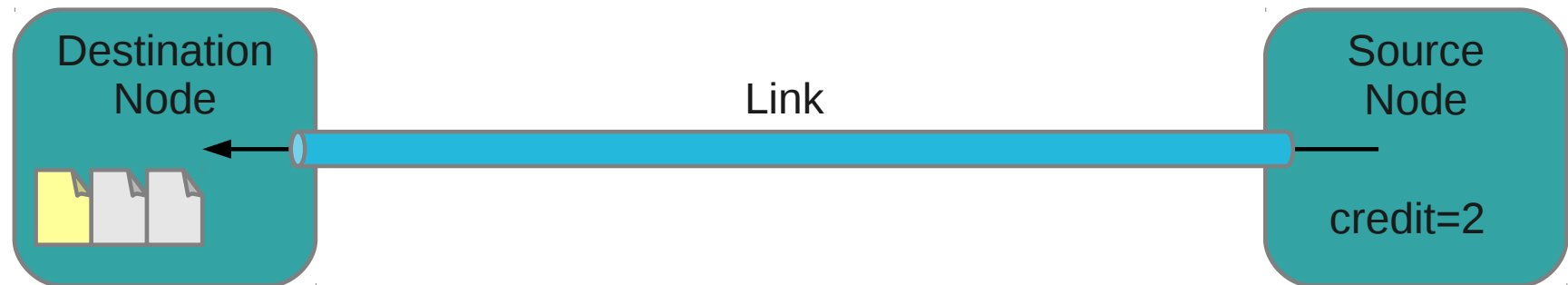
- Pre-Settled (Fire and Forget)
- Settled with Disposition
  - Accepted
  - Rejected
  - Released

# Transactions



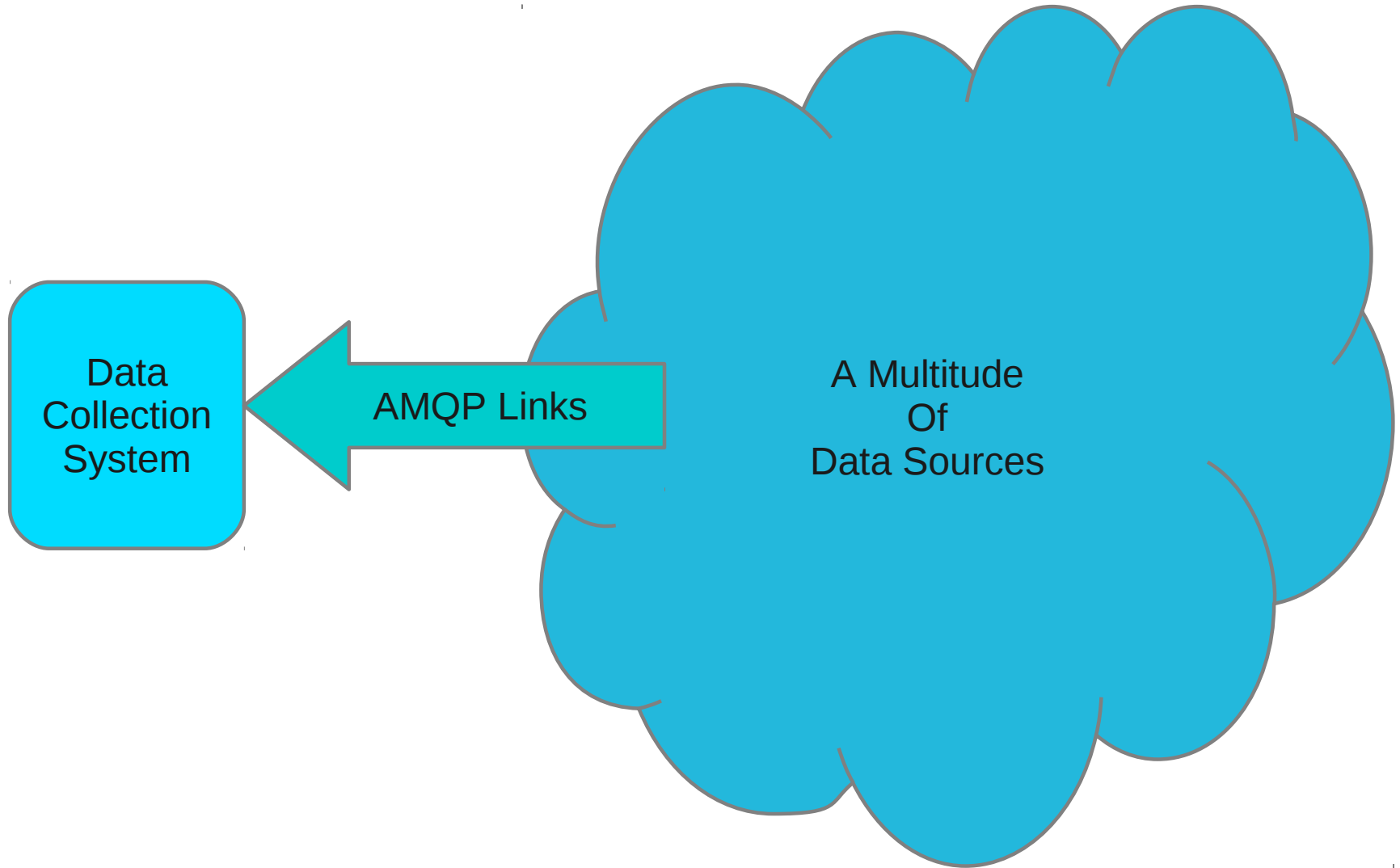
- Local Transaction
- Distributed Transaction

# Message Handoff – Flow Control




- Destination Issues Credit
- Source Sends no more than  $\langle \text{credit} \rangle$  Deliveries
- Flow Control is Independent from Settlement

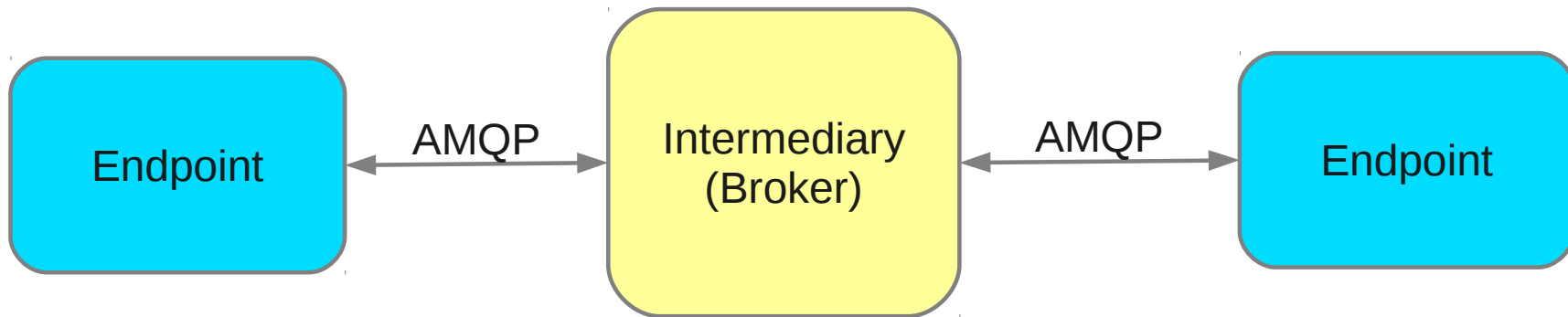
# Flow Control



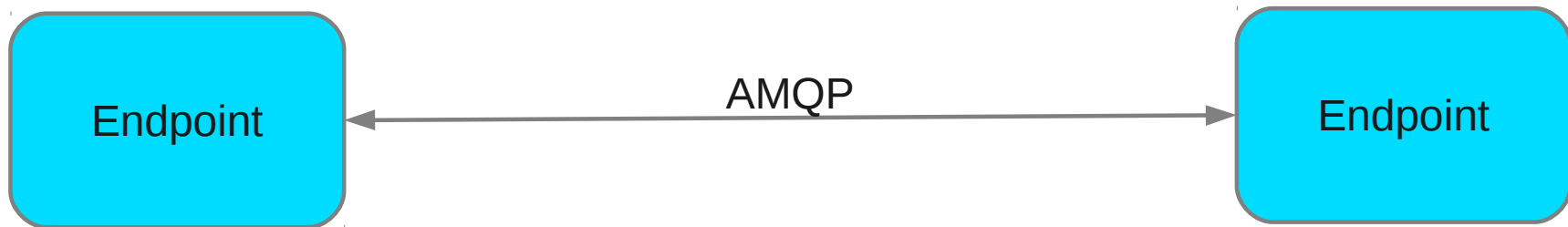
# Summary of Benefits

- Session Multiplexing
  - Full Duplex, Asynchronous Transfer
  - Formal Semantics of Message Hand-Off
  - Data Security
  - Flow Control
  - Serialization of Structured Data
  - Message Metadata
- 

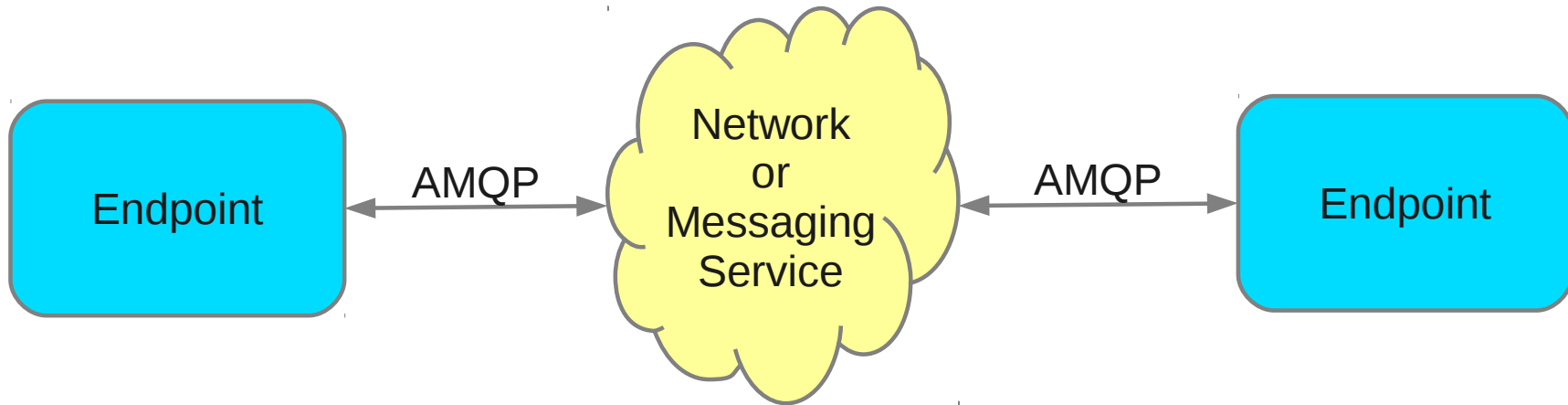
# Topologies - MOM



# Topologies – Point to Point

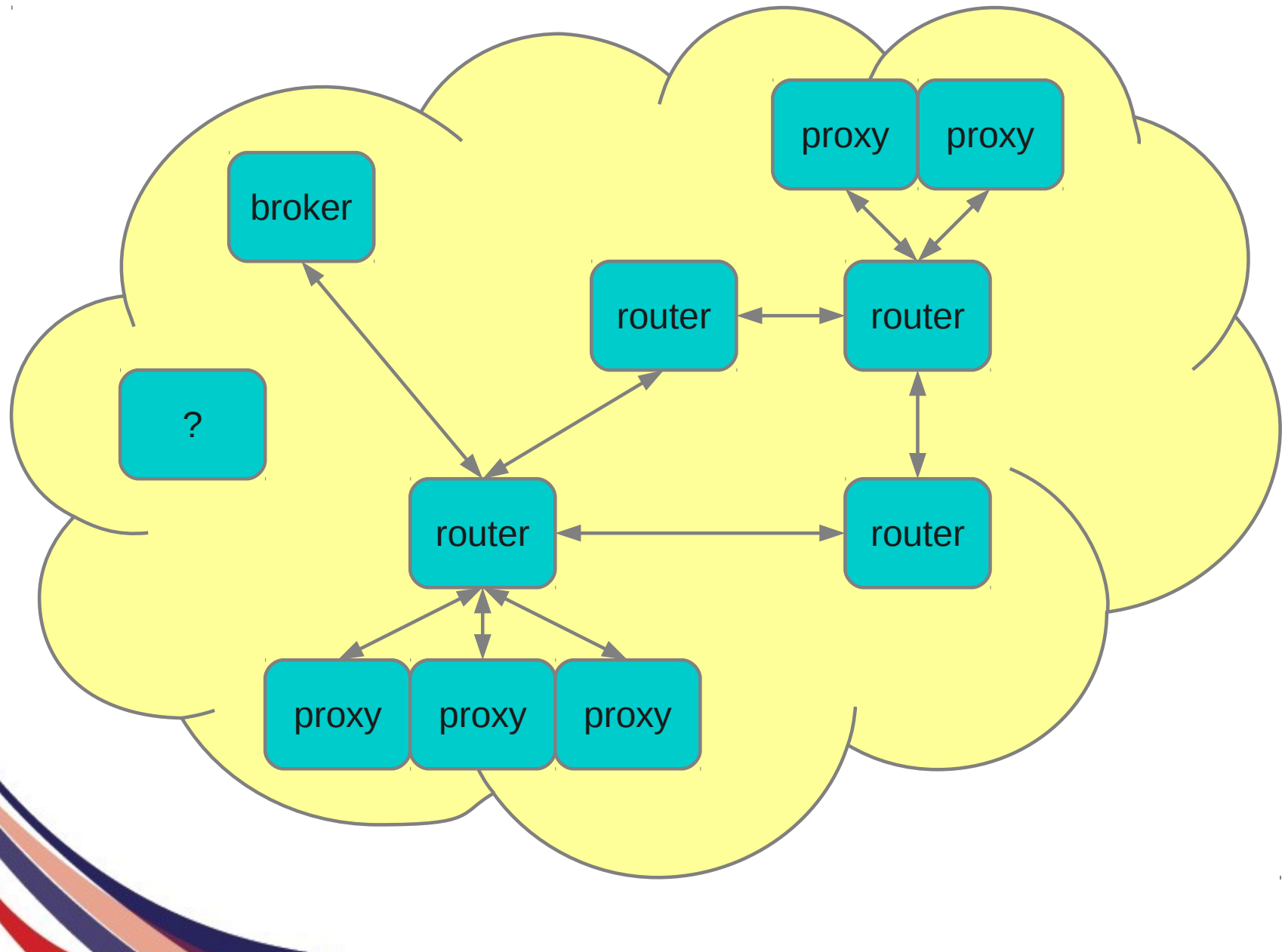


# Topologies

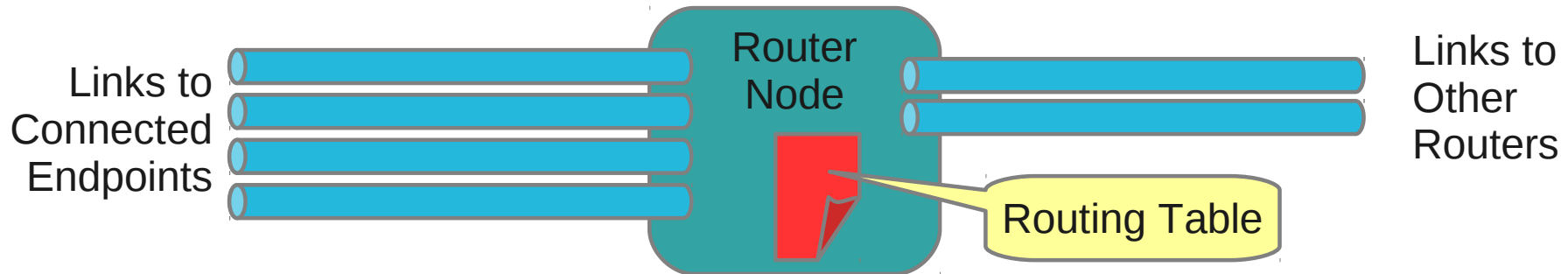




# AMQP Network

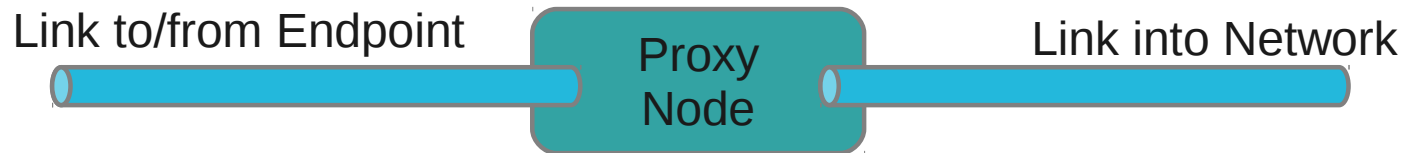


# AMQP Router



- Tracks Locally Connected Endpoints
- Can Assign Temporary Addresses
- Shares Address Information with Other Routers

# AMQP Proxy



- Ties a “public-facing” link to an Internal Link
- Policy Enforcement at Link Setup Time
- Renaming of Link Target/Source may Occur
- Delivery Transfer is Very Simple

# Use Case

- Public Service Delivery over AMQP
  - Hardened Proxies
    - Enforce Access Policy
    - Provide Multi-Tenancy
    - Resist Denial-of-Service
  - Routers
    - Provide Redundancy and Scaling
  - Brokers
    - Provide Queuing, Persistence, etc.

This is Complicated

How Do I Use It?



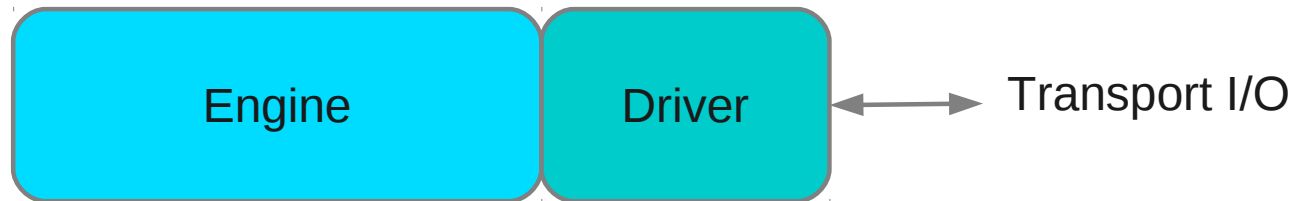
# AMQP in Messaging Systems

- Apache ActiveMQ
  - Multi-protocol Java Message Broker
  - Includes AMQP Transport
- Apache Qpid
  - AMQP Brokers (Native and Java)
  - AMQP Clients
    - Native, Java JMS/JCA, .NET
    - C++, Java, Python, Ruby, Perl, PHP, \*.NET

# AMQP in Your Systems

- Qpid Proton
  - Intended for Embedding and Integration
  - Implementation of the AMQP Protocol
  - C, Java, Javascript
  - Messenger API with script language wrappers
- Qpid Dispatch
  - Event-Driven, Multi-Threaded container for Proton

# Qpid Proton



- Engine

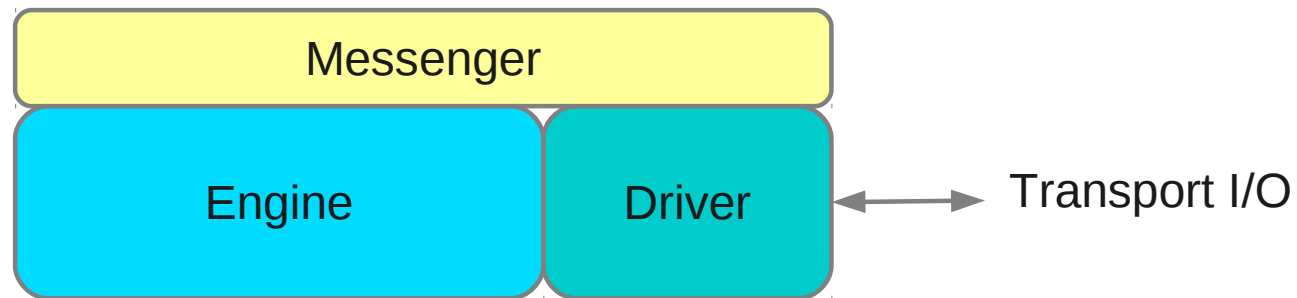
- State Machine
- Non-Blocking API

- Driver

- Provides I/O
- Defines Threading Model

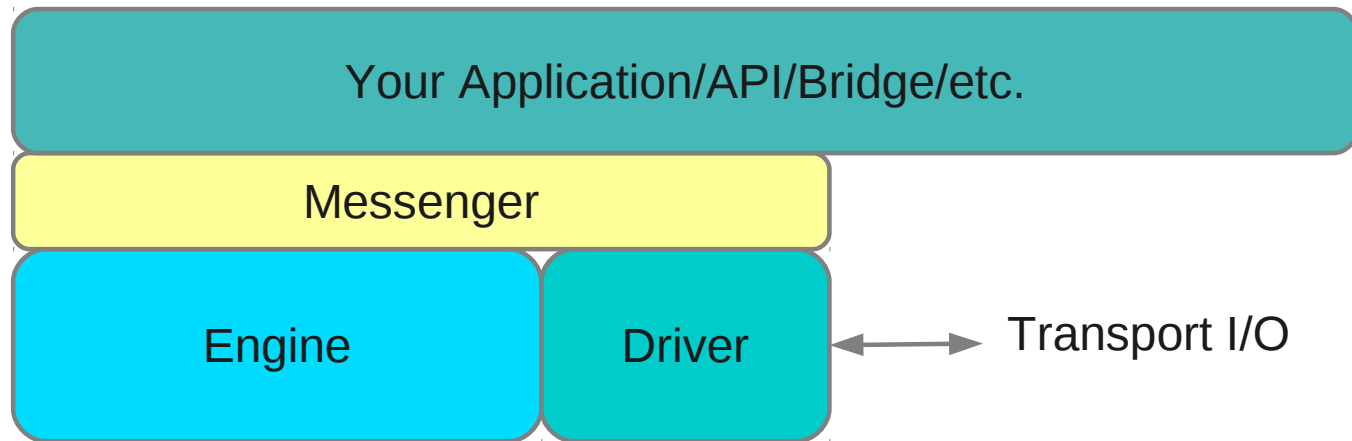


# Qpid Proton

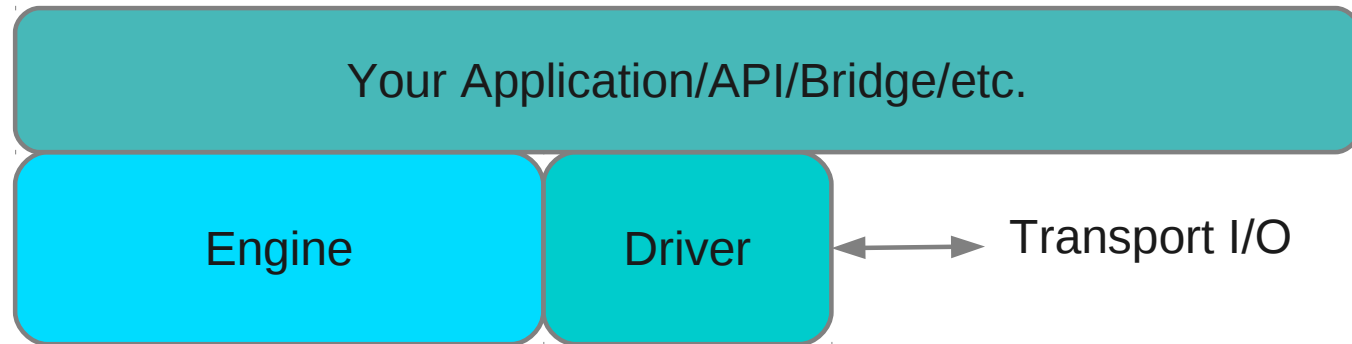


- Proton Messenger
  - Easy to Use Messaging API for Developers
  - Hides many Details of AMQP

# Integrating over Messenger



# Integrating over Engine



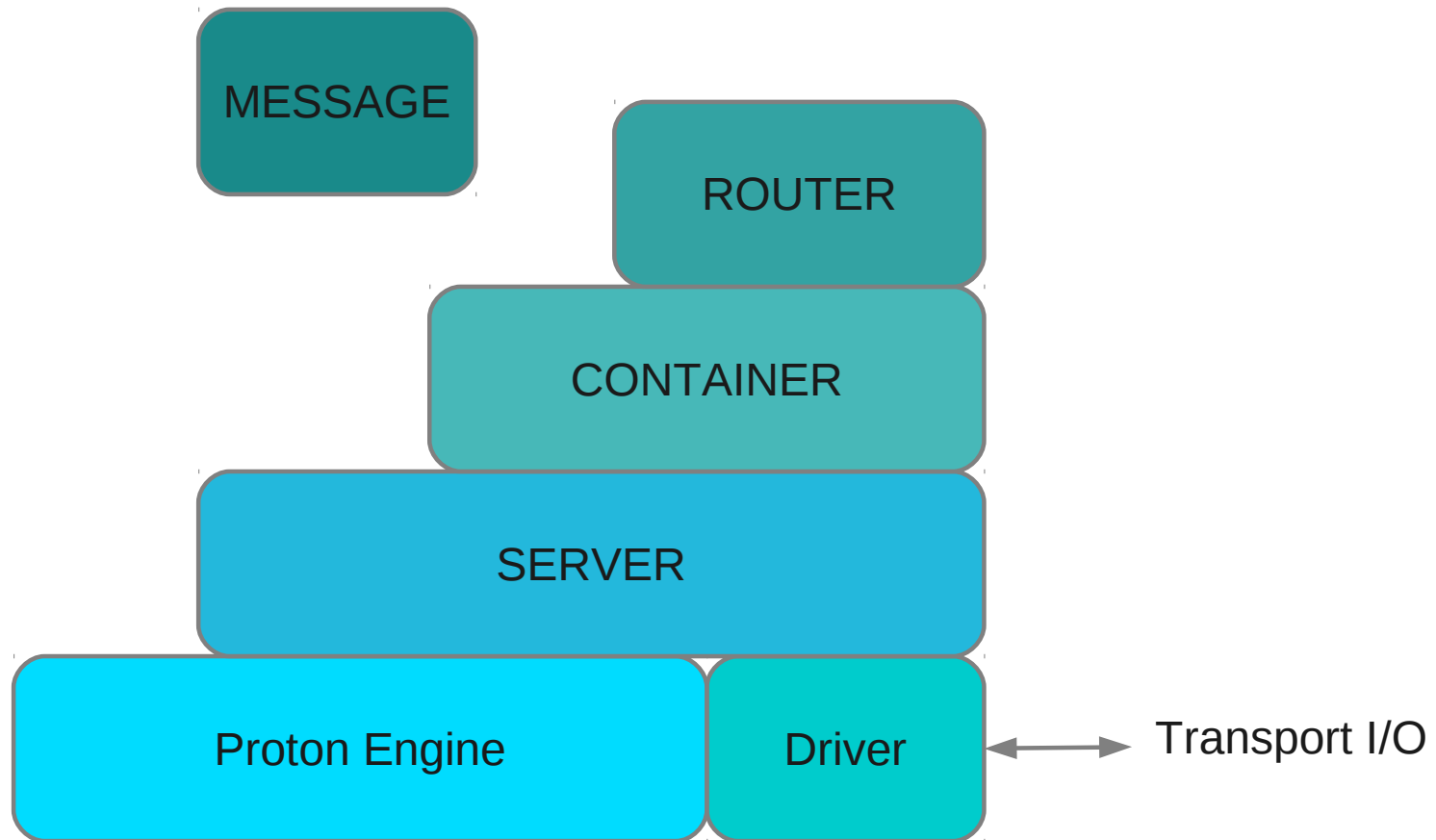
- The “Red Pill”
- When you need control over every aspect of AMQP
- Significantly more complex

# Qpid Dispatch

- Goals
  - Aid in Integration over Proton Engine
  - Provide a framework for development of AMQP infrastructure
  - Simple intermediaries that can fully utilize expensive network infrastructure
- Multi-Threaded Event-Driven Container

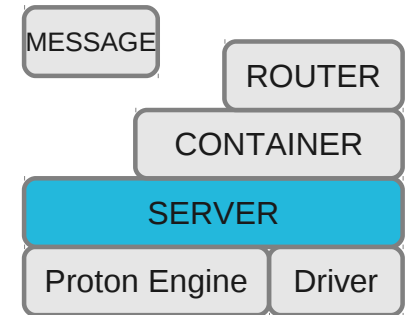


# Qpid Dispatch Architecture



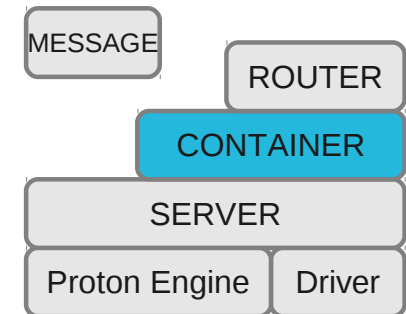
# Qpid Dispatch - Server

- AMQP Connections
  - Listeners
  - Resilient Connectors
- Timers
- Thread Control (quiesce/resume, etc.)
- Management of non-AMQP FDs
- Handling of Signals



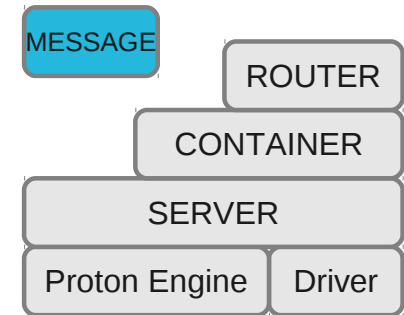
# Qpid Dispatch - Container

- API for Node Creation
- Manages Node Lifecycle
- Handles Links and Deliveries
  - Link attach/detach
  - Delivery inbound/outbound
  - Disposition and Settlement
  - Flow Control



# Qpid Dispatch - Message

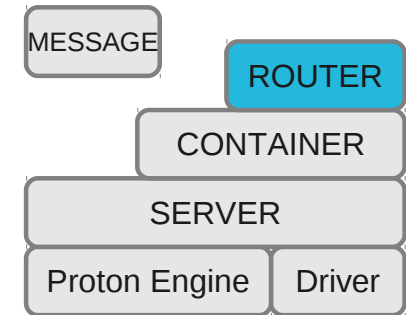
- API for Message Manipulation
  - Fixed-size buffer chaining
  - Access to fields regardless of buffer boundaries
  - Parses message contents only as far as needed
  - Efficiently handles modified annotations





# Qpid Dispatch - Router

- Tracks Consumers by Address
- Forwards Messages from Inbound Link to Outbound Link by Address
- Supports Internal and External Endpoints
- Interacts with Other Routers to Compute Paths across Networks



# Future Work

- Configuration
- Agent for Remote Management
- Proxy Node



# Qpid Dispatch

- Sub-Project of Apache Qpid

svn: qpid/trunk/qpid/extras/dispatch

- Installed Artifacts

include/qpid/dispatch/\*.h

lib/libqpid-dispatch.so

bin/dispatch-router


- Web Site

- Content will appear after the release of Qpid 0.22 (End of March)

# Summary of Apache Projects

<u>Project</u>	<u>Version</u>	<u>Maturity</u>
Apache ActiveMQ	5.8	Mature
Apache Qpid	0.20	Mature
Qpid Proton	0.4	Emerging
Qpid Dispatch	unreleased	New

# What You will Take Away

- AMQP is More Than Messaging
  - AMQP is Complex and Capable
  - Apache Makes AMQP Accessible and Easy to Use
  - The Apache Software Foundation is on the Cutting Edge of Distributed Computing
- 

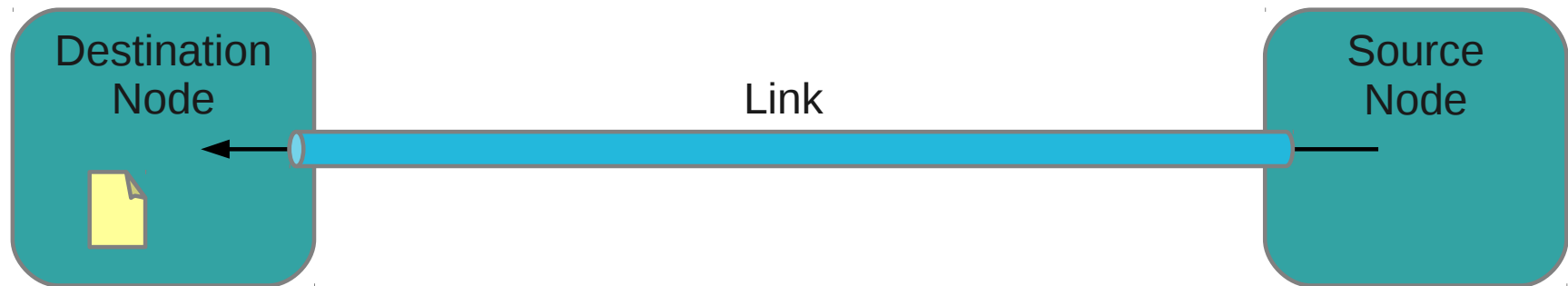
# Questions and Discussion



# Bonus Material



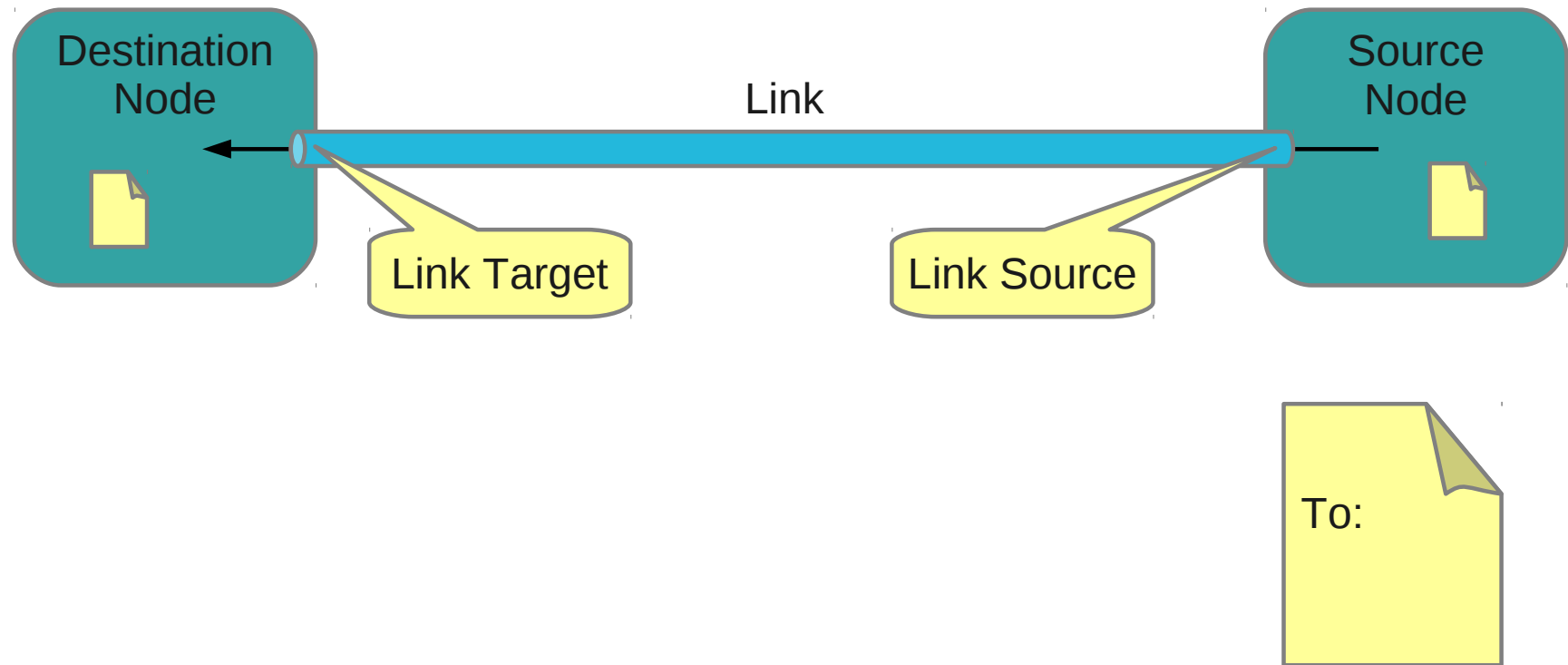
# Message Handoff



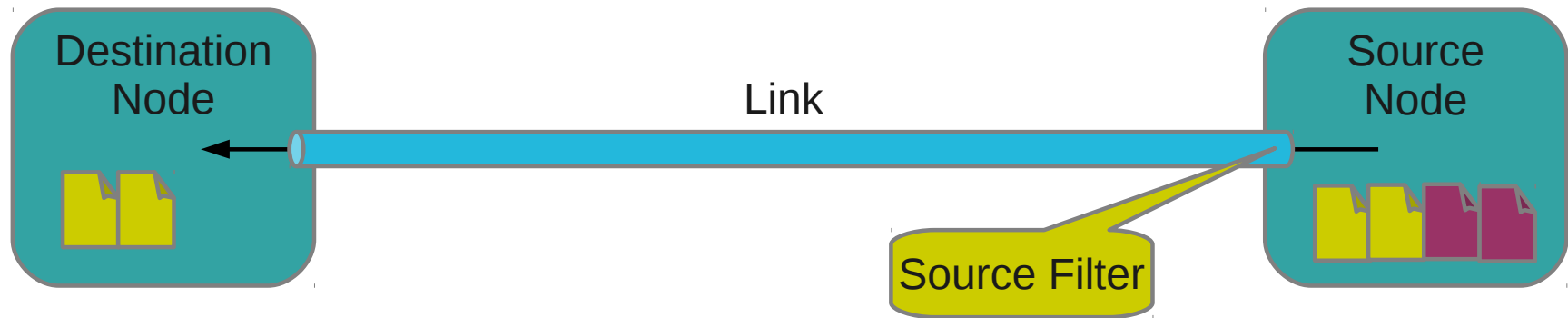
- Message and Delivery are Separate
  - “Letter” and “Envelope”
  - Same message may be delivered to multiple destinations or redelivered to the same destination



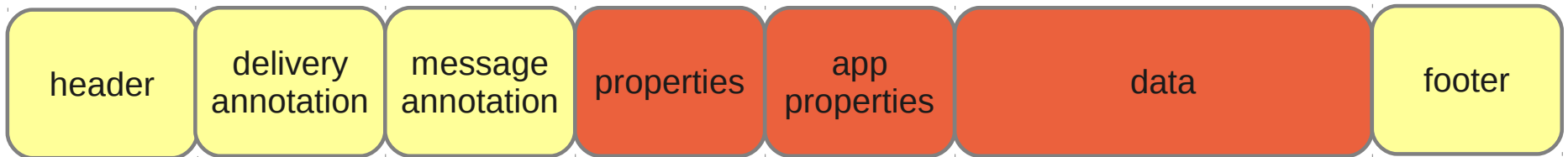
# Link and Message Address



# Source Filters



# Message Structure



- Data may be in any Format including AMQP-formatted data
- Structured Properties and Annotations
- Security
  - Bare Message may be signed/encrypted
  - CRC/Signature in footer for efficiency