Iterim Report: Package for Controlling Delivery of Messages

Samuel Cherinet and Robert Pate

November 2017

---

1   Goal and Use Case

"Design and implement a package that allows a programmer to specify and control delivery of messages in a distributed program."

As an application developer, I need different instances of my distributed application to communicate with each other. I want to import a package into my code that lets me do that seamlessly with a few commands.

2   Literature and Other Solutions

Middleware Trends and Market Leaders 2011 is a paper by A. Dworak, F. Ehm, W. Sliwinski, and M. Sobczak for CERN that analyzes the options for replacing the aging middleware for their 4000 servers and 80,000 devices.

ZeroMQ is "distributed messaging" with many features such as multiple languages, multiple platforms, multiple transport protocols, various patters, open source, and good documentation. The CERN paper rated it the highest and it's had continual development since then.

YAMI4 is a "messaging solution for distributed systems" and positions itself as a "competitor" to ZeroMQ.

AMQP stands for "The Advanced Message Queuing Protocol" which is also a good description of it. It became an ISO standard in 2014 and has various implementations by 3rd parties.

3   Functions and Restrictions

3.1   Restrictions

1. Only supports JAVA

2. Only supports TCP

3. Intended for distributed architecture

4. Only support asynchronous messaging

3.2 Some Example Functions

1. bindTo("string") ip address and port

2. Server s = new Server("ip/port string") // takes optional options

3. s.send("Hello")

4. receive - tell the server what to expect

4 Questions and Next Steps

1. Review AMQP to see if it's feasible to build an implementation in a few weeks

2. Decide what level of message ordering we're going to support

3. Review the ZeroMQ and YAMI4 semantics for ideas

4. Outline all the public objects and methods of the package

5. Construct the minimum functioning package

6. Test the package by replacing messaging code from HW2

7. Iterate and improve