



1) User types in command to his IRC client.

2) Command is sent to IRC server.

3) We have a custom IRCd that can forward user-defined commands using RabbitMQ. Users should be able to add additional command support.

4) Command is forwarded to appropriate slave message queues (SMQs) using RabbitMQ's 'topic' system, see <http://www.rabbitmq.com/tutorials/tutorial-five-java.html>. The user defines what topics a command encompasses and which slaves handle which topics.

5) Slaves get message from queue and do work.

6) If a result is to be returned, the slave sends the result back into the master message queue (MMQ).

7) Master sends results back to client.

8) User gets results.

\* Note, each slave and master will need to be running the RabbitMQ server software.

\* Note, this architecture supports a single master node. If there are going to be multiple master nodes, it will need to be updated to support synchronization and acknowledgements between nodes.