

ActiveMessaging

Dead Simple MQ Support for Rails

Cliff Moon
Chariot Solutions



Messaging Basics

- Queues
 - Point to point model
 - Guaranteed delivery
 - One consumer per message
- Topics
 - Publish and subscribe model
 - Delivery not guaranteed
 - Messages get delivered to all subscribed consumers



Why messaging?

- Messaging services available in many languages
- Loosely coupled systems
- Talking to mainframes
- Silly level of scalability
- Mature solutions already abound



a13g vs. BackgroundRb

- a13g
 - Message oriented
 - Needs a messaging broker with stomp support
 - Can talk with any language or platform that has a stomp library
 - Scales to any number of servers
- BackgroundRb
 - Job/process oriented
 - No dependency on outside broker
 - Only works with ruby
 - Limited scalability
 - Starts and stops processes on demand



Supported Protocols/Brokers

- Brokers
 - ActiveMQ
 - WebsphereMQ
 - Amazon SQS
- Protocols
 - JMS via StompConnect
 - Stomp



The simple case

- Code demo!
 - rails a13g_simple_demo
 - sudo gem install daemons (if you don't have it already)
 - sudo gem install stomp
 - script/plugin install
<http://activemessaging.googlecode.com/svn/trunk/plugins/activemessaging>
 - script/generate processor secretary
 - script/generate controller UpsGuy send



Correlation ID

- More Code demos!
 - Request/response use case
 - uses selector subscription header



Java, we never talk anymore

- Yet More Code demos!
 - Java MDB using spring
 - AMQ handles stomp -> JMS bridge
 - Stomp library needs a patch: remove content-length
 - Stomp -> JMS headers are impedance mismatched



Fair thee well, XML

- The last code demo.
 - Java MDB using spring
 - Uses Jyaml library to parse yaml into a Pojo
 - Date/time gotcha: seconds -> milliseconds



Get Involved!

- Don't just consume. Publish.
 - <http://groups.google.com/group/activemessaging-discuss>
 - Write an adapter to your favorite MQ broker
 - jRuby support project
 - More transparent support for correlation ID

