Messaging

This is something I haven’t cracked yet. Three implementations down and nothing has really stuck.

There are a variety of things that keep cropping up:

* Need to be able to do a simple Client Server model
* Need to be able to connect multiple clients to a hub and have them exchange messages
* Need to be able to subscribe to something and receive updates on that subscription
* Need to be able to throw pojo’s down the wire without any config faff
* If the other side doesn’t know how to decode the message nothing bad happens (with a handler mechanism if they want to log the fact)
* Using the interface approach – I want to use the same interface remotely as I would locally, and not really know the implementation is remote
* Adding listeners - same as creating a subscription in the classic messaging mode
* Could map to tuple-space idea of putting objects into the registry
  + New objects can notify listeners based on filters
  + Objects can be listened to for changes
  + A remote version of MVC style observable objects used in the front ends
  + Objects can be pojos or interfaces for interactions
* Must be possible to vary the encoding without causing backwards compatibility issues
* Must be possible to extend messages with legacy clients still decoding the bits they understand

We definitely don’t want to be writing the tcp/ip bits – especially in java. So we should be using Netty probably – this seems to be well regarded. It’d be nice for that to be well hidden under the covers.

