

Group Organisation

For 1.1 onwards we're implementing a range of *Group Organisation* protocols to help arrange nodes into groups. To get an idea of the progress we're making, check out the [javadocs](#).

Organising Groups

There are a few different ways in which we want to arrange nodes into groups...

Groups

The basic idea is we want to arrange nodes into logical groups. Each group acts as a single logical entity.

For example to implement the master/slave type High Availability (HA) model, we may wish to have just 1 single group with a *master* node and a collection of one or more *slave nodes*. If the master goes down or becomes unavailable or overworked; we can failover to a slave.

The default GroupModel supports this behaviour and can handle one or more groups. So there could be several groups (with a single master and multiple slaves) for different instances of a service. You can configure the GroupModel with a maximum number of groups - so if you wish you can restrict the number of groups.

Buddy Groups

The organisation policy for buddy groups is often a little different. Generally we want each node in a cluster to be a master node, in its own group - but for other nodes in the cluster to be buddies to act as backups.

(We may wish to restrict some nodes from being master nodes, so there is an optional `masterNodeFilter` we can use to restrict those nodes on the `BuddyGroupModel` being master nodes).

So the `BuddyGroupModel` will create a group for every possible node joining and then assign buddy groups from other members in the cluster.

Using Groups

To use groups, the leader is elected by the cluster then it decides, based on the current node state, what groups to create / destroy and how to arrange the cluster. As nodes come and go the leader decides what to do.

If the leader fails, a new leader is elected and the process starts again.

When the leader decides that a node must join or leave a particular group, it communicates with the node in question. Depending on the application; some kind of *node synchronization* is typically required - such as state transfer from buddy nodes. When that is complete (or if it fails) then the node should announce its membership status to other nodes - then the leader can decide what to do next.