

# Diving into Elasticsearch Discovery

Shop directly from people around the world.



Alana Little of makepienotwar  
California, United States



123  
items

**1.4M**

Active Sellers

**20.8M**

Active Buyers

as of Mar 31, 2015



# Search Infrastructure at Etsy



Unsharded Solr Master/Slave

Hand-sharded Solr Master/Slave

Elasticsearch

Our largest indexes are on Elasticsearch, ~ 1TB.









“One winds on the distaff what the other spins” (both spread gossip)  
by Pieter Bruegel the Elder





?

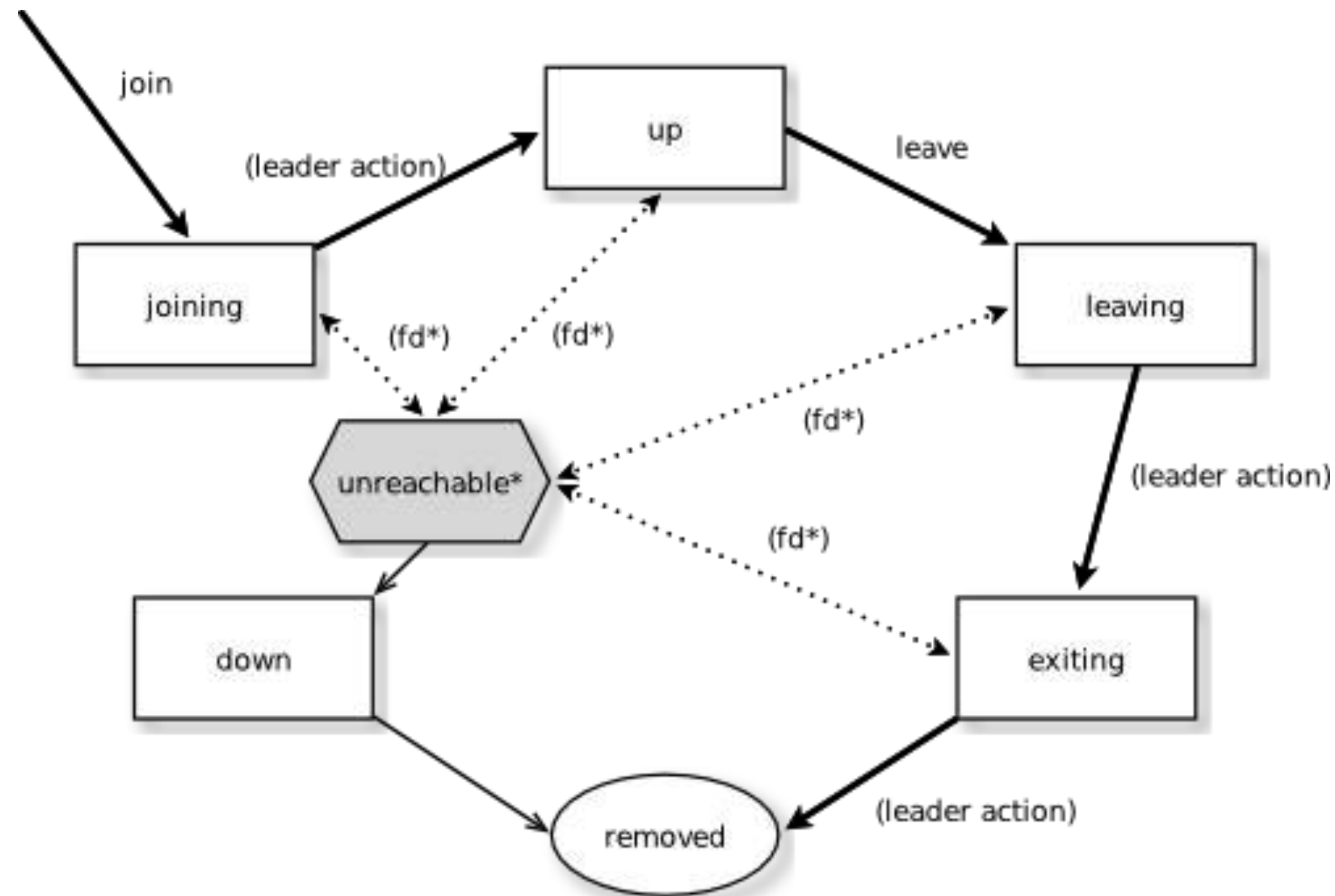


```
public class DiscoveryModule extends AbstractModule
```

## Pluggable

api backwards-compatibility not guaranteed

# eskka: elasticsearch discovery using akka cluster



Akka Cluster state diagram for member states

```
private volatile ClusterState clusterState; *
```



```
{  
  cluster_name: "flop-elastic1",  
  version: 208774,  
  master_node: "1PPf9tUcQhu7df91k7-W8Q",  
  blocks: { },  
+ nodes: {...},  
+ metadata: {...},  
+ routing_table: {...},  
+ routing_nodes: {...},  
  allocations: [ ]  
}
```

transient state

```
nodes: {
  - bb271oNcR6O5Y1AEdLRZuw: {
    name: "search77-es1",
    transport_address: "inet[/172.31.240.88:8301]",
    - attributes: {
      host: "search77.ny4.etsy.com"
    }
  },
  - Z11tgVcSQG-2NWfV8Rc61g: {
    name: "search79-es2",
    transport_address: "inet[/172.31.240.90:8302]",
    - attributes: {
      host: "search79.ny4.etsy.com"
    }
  },
}
```



persistent state

```
metadata: {  
  + templates: {...},  
  - indices: {  
    - apps-1432720801: {  
      state: "open",  
      + settings: {...},  
      + mappings: {...},  
      + aliases: [...]  
    },  
  },  
}
```

```

routing_table: {
  - indices: {
    - apps-1421668801: {
      - shards: {
        - 0: [
          - {
            state: "STARTED",
            primary: true,
            node: "hLsAK6reQeeRrdnHxm0HLA",
            relocating_node: null,
            shard: 0,
            index: "apps-1421668801"
          },
          - {
            state: "STARTED",
            primary: false,
            node: "kp_AIfIJRFOnZihcpqzBFg",
            relocating_node: null,
            shard: 0,
            index: "apps-1421668801"
          }
        ]
      }
    }
  },
}

```

transient state

node discovery

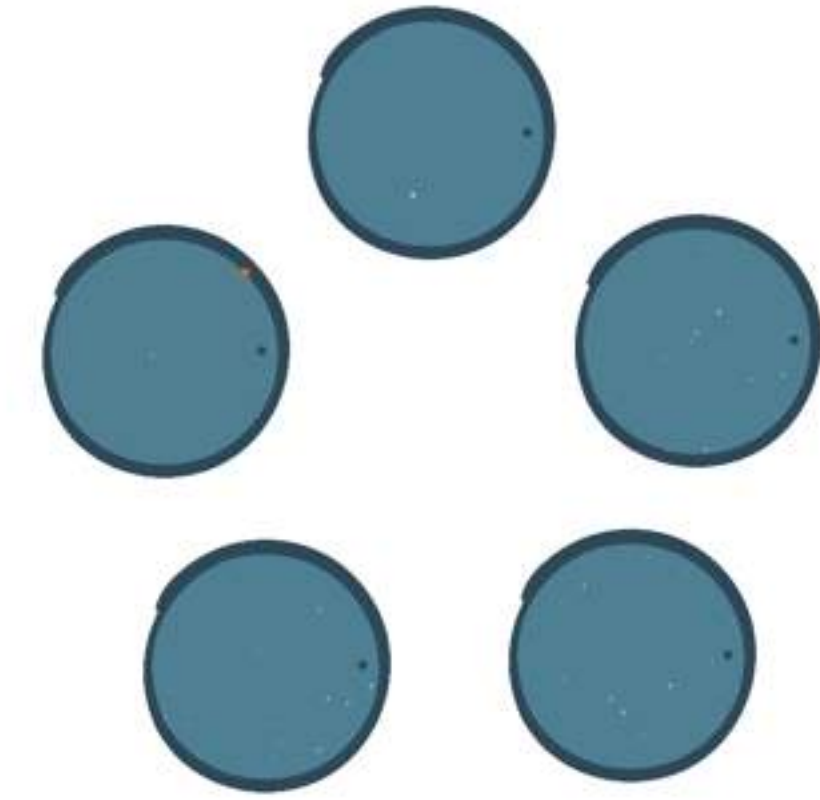
leader election

state publishing

failure detection / handling

**zen & eskka: properties**





**node discovery**

master election

state publishing

failure detection / handling

# node discovery

## zen

Unicast mode: static list of 'gossip routers'

Multicast mode: multicast address

Batching of state updates from membership changes (in recent releases)

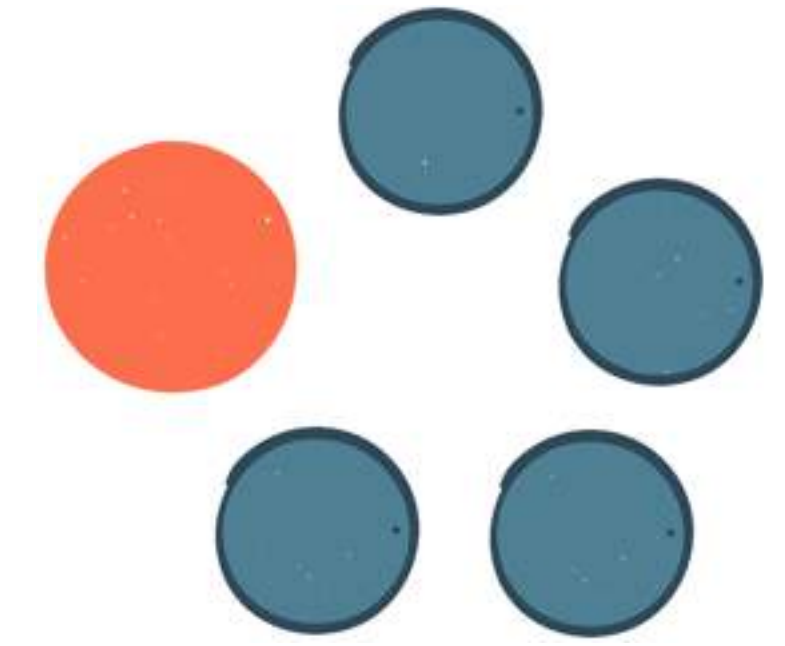
```
-Des.discovery.zen.ping.unicast.hosts
```

## eskka

Static list of seed nodes: 'contact points for new nodes joining the cluster'

Batching of state updates a result of membership changes

```
-Des.discovery.eskka.seed_nodes
```



node discovery

**leader election**

state publishing

failure detection / handling

# leader election

## zen

Master-eligible node with lowest node ID

```
RANDOM_UUID_GENERATOR.getBase64UUID();
```

Handling of edge cases improved in ES 1.4  
(#2488)

## eskka

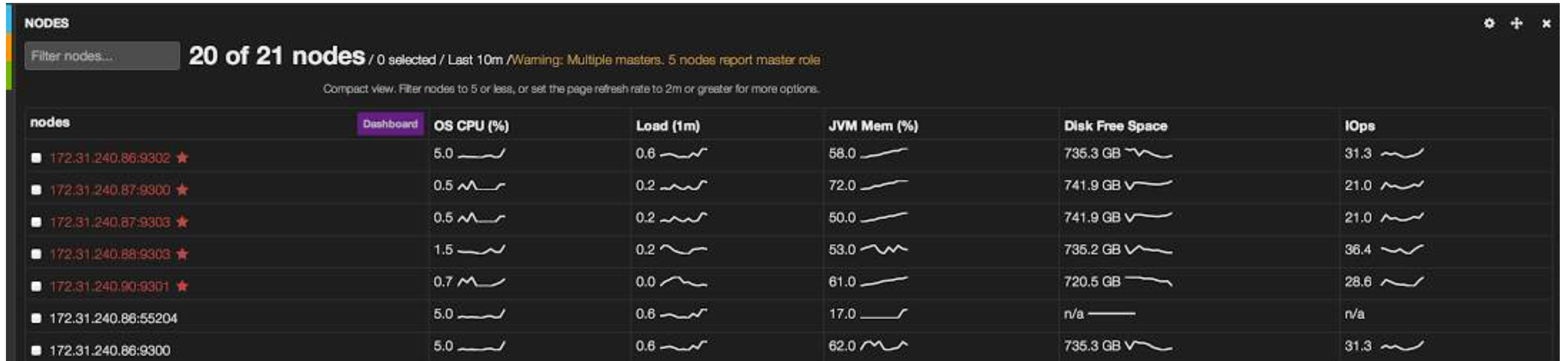
Akka 'Cluster Singleton' -

Oldest master-eligible cluster member

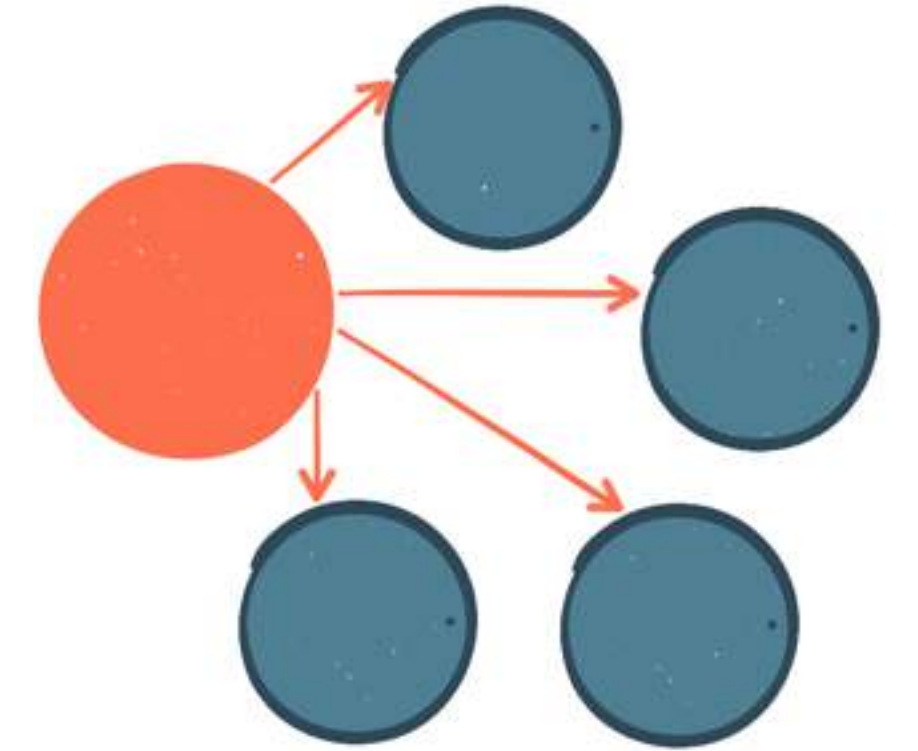
Edge cases around fail-over handled with  
timeouts.



?!



ES 1.2 with Zen, *minimum\_master\_nodes* configured correctly, meant to use unicast discovery but multicast was not turned off.



node discovery

leader election

**state publishing**

failure detection / handling

# state publishing

## zen

Internal ES transport

Serialized & compressed

Block upto

'discovery.zen.publish\_timeout' (30s default)

but no consequence to timeout

## eskka

Akka Remoting

Serialized, compressed & chunked

Asynchronous

v2.0.0

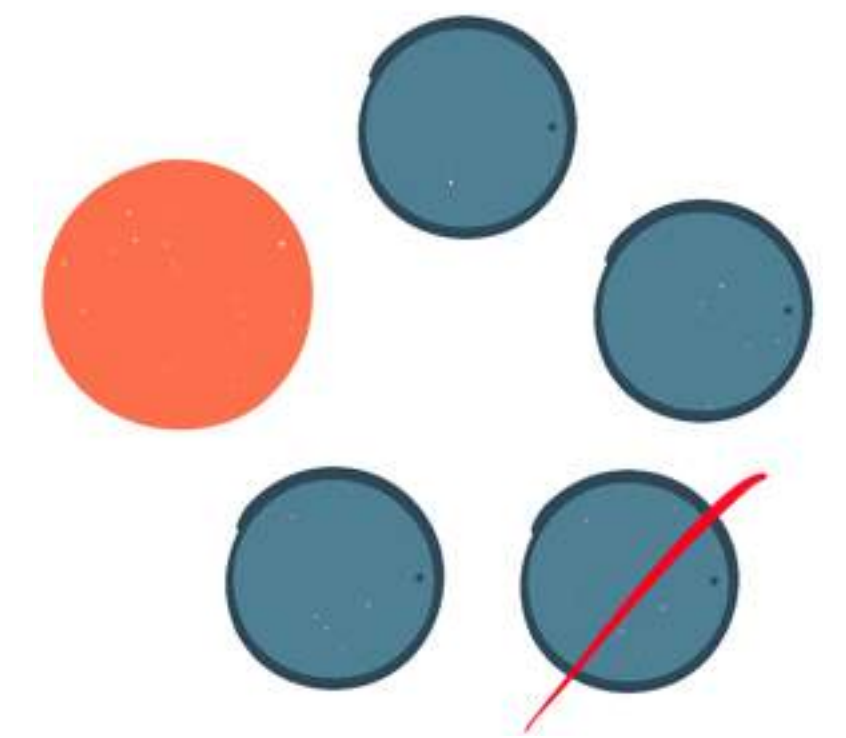
# Only send diff of cluster state instead of full cluster state

#6295

 **Closed**

**bluelu** opened this issue on May 23, 2014 · 10 comments





node discovery

leader election

state publishing

**failure detection / handling**

# failure detection

## zen

Master monitors all nodes with pings, all other nodes monitor master with pings.

Knobs around retries and timeouts.

## eskka

All nodes partake in monitoring heartbeats.

Knobs for failure certainty\* and acceptable heartbeat pause time.

Quorum of seed nodes decides availability of unreachable node.

\* Phi Accrual Failure Detector

# minority partitions

zen

minimum\_master\_nodes constraint violated

=> we are on minority partition

eskka

Quorum of seed nodes unreachable

=> we are on minority partition

# failure handling

Failure detection is **Best Guess**.

Once decided:

- if minority partition, either block all operations (`no_master_block=all`) or write operations only (`no_master_block=write`)
- remove suspect from cluster
- fail-over master if required

**node discovery**

**leader election**


**state publishing**

**failure detection / handling**



**Solid ES Discovery  $\neq$  Jepsen Win**

# [Indexing] A network partition can cause in flight documents to be lost #7572

 **bleskes** opened this issue on Sep 3, 2014 · 12 comments

What Jepsen tests: an acknowledged write won't be lost, particularly under partition.

This has more to do with replication semantics, e.g.

- What guarantees are implied when you receive an acknowledgment
- How a primary is selected from the replicas of a shard

If you evaluate ES Discovery as a distributed store,  
ClusterState is the only document.

# ClusterState update safety

```
[sbhushan@search75]~% curl -XDELETE localhost:8200/_template/marvel  
{"acknowledged":true}
```

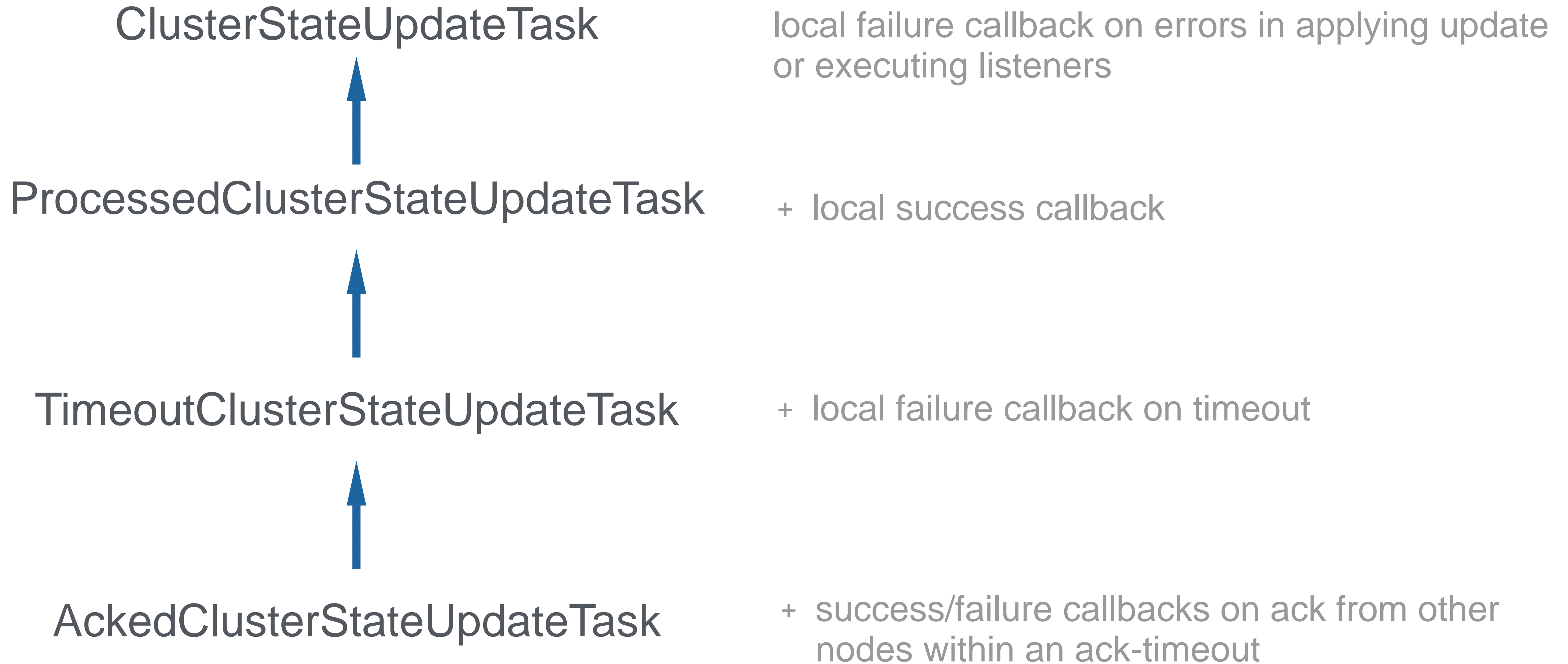
**UpdateTask :: ClusterState -> ClusterState**

Asynchronously applied from single thread by InternalClusterService



← → ↻ 🏠 search36:8200/\_cluster/pending\_tasks

```
{
  - tasks: [
    - {
      insert_order: 617,
      priority: "NORMAL",
      source: "indices_store",
      executing: false,
      time_in_queue_millis: 10,
      time_in_queue: "10ms"
    },
    - {
      insert_order: 618,
      priority: "NORMAL",
      source: "indices_store",
      executing: false,
      time_in_queue_millis: 10,
      time_in_queue: "10ms"
    },
  ],
}
```



**NOT** `[sbhushan@search75]~% curl -XDELETE localhost:8200/_template/marvel {"acknowledged":true}`

```
clusterService.submitStateUpdateTask("remove-index-template [" + request.name + "]",  
    Priority.URGENT, new TimeoutClusterStateUpdateTask() {
```

(most metadata update requests *do* use AckedClusterStateUpdateTask)

System overall seems workable.

Ability to replace Elasticsearch Discovery is awesome.

Doc replication semantics need work!

thank you

shikhar@etsy.com

 @shikhrr