Keeping track of the animals

Progress towards redundant services in dCache

Gerd Behrmann
On behalf of the project team

10th dCache User Workshop

norden

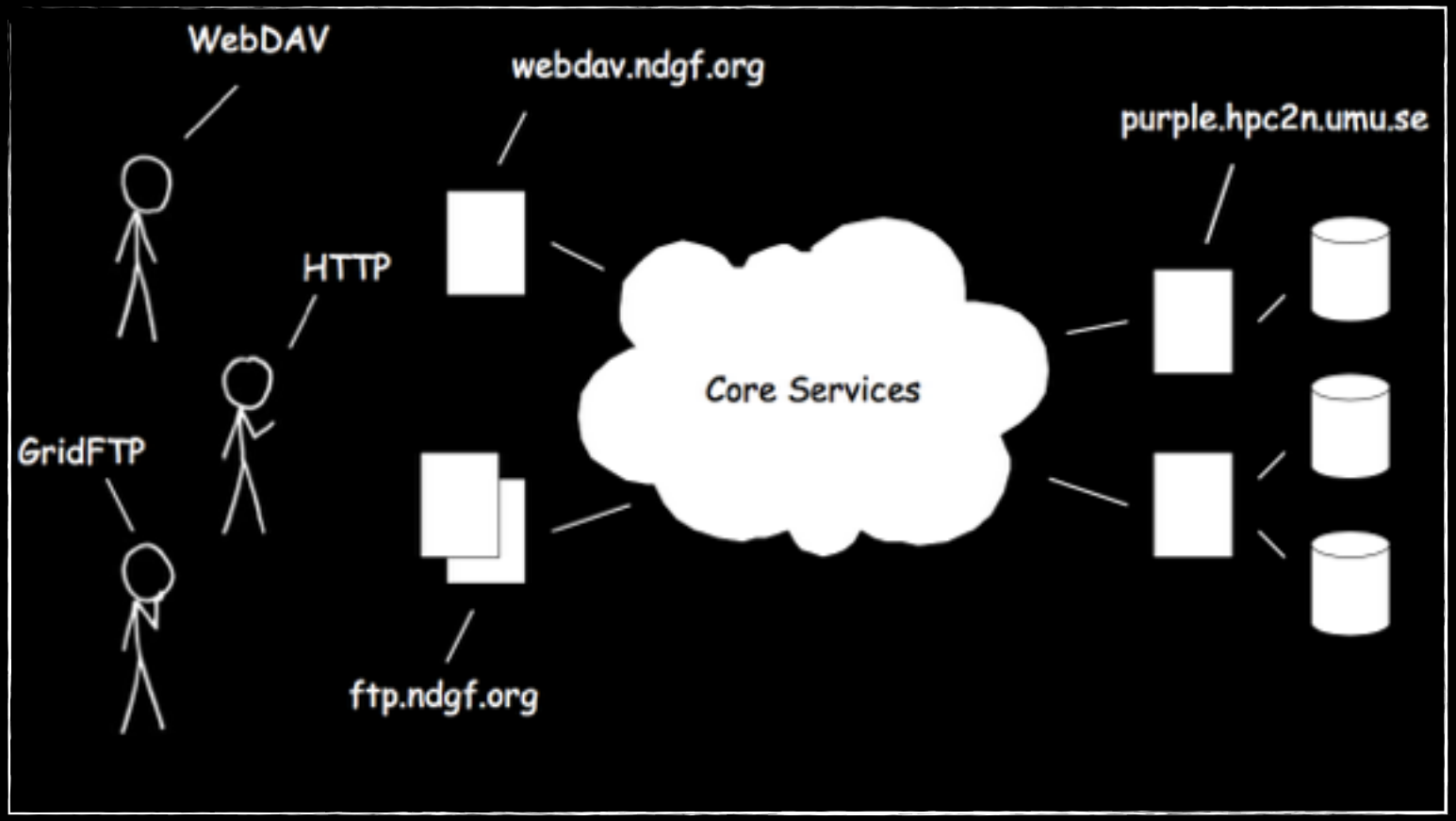
NordForsk

neic

Nordic e-Infrastructure Collaboration

What to remember from this talk

- Make a standalone Apache ZooKeeper installation.
- dCache can have redundant services.

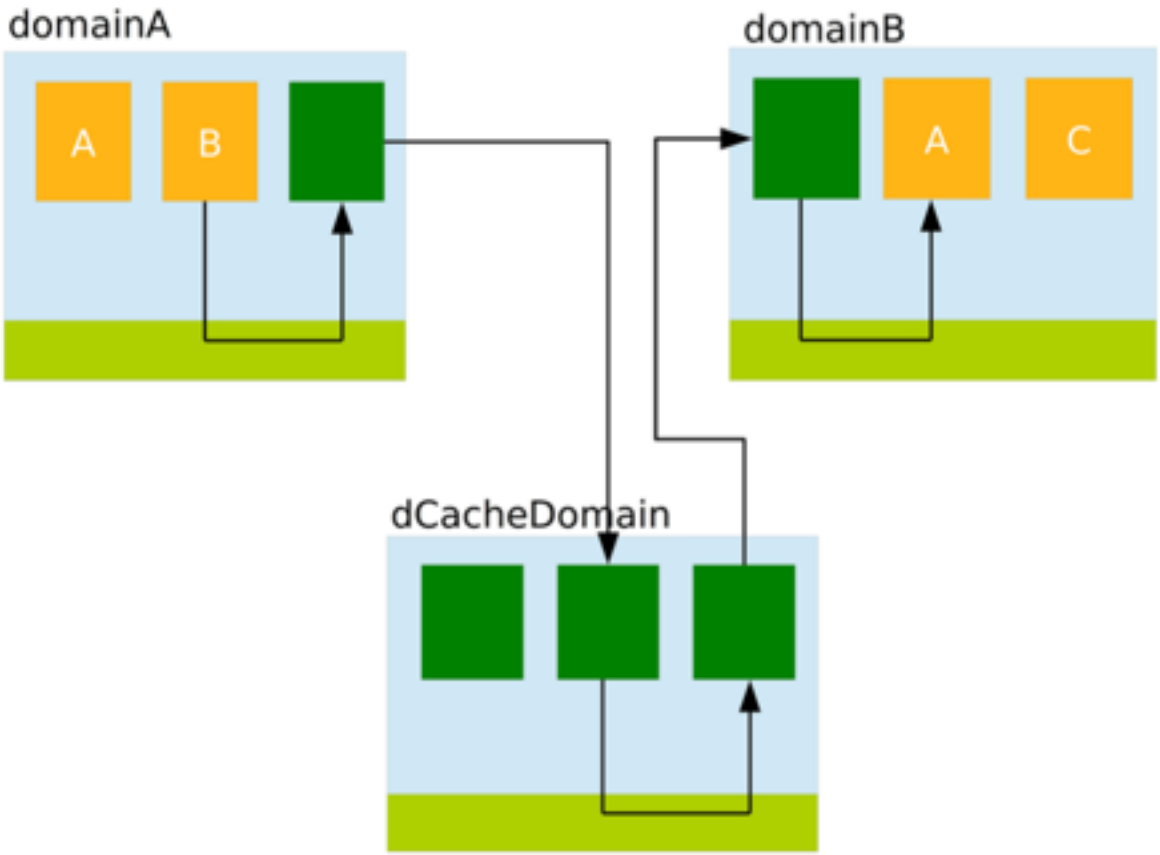

WebDAV
webdav.ndgf.org
purple.hpc2n.umu.se
HTTP
Core Services
GridFTP
ftp.ndgf.org

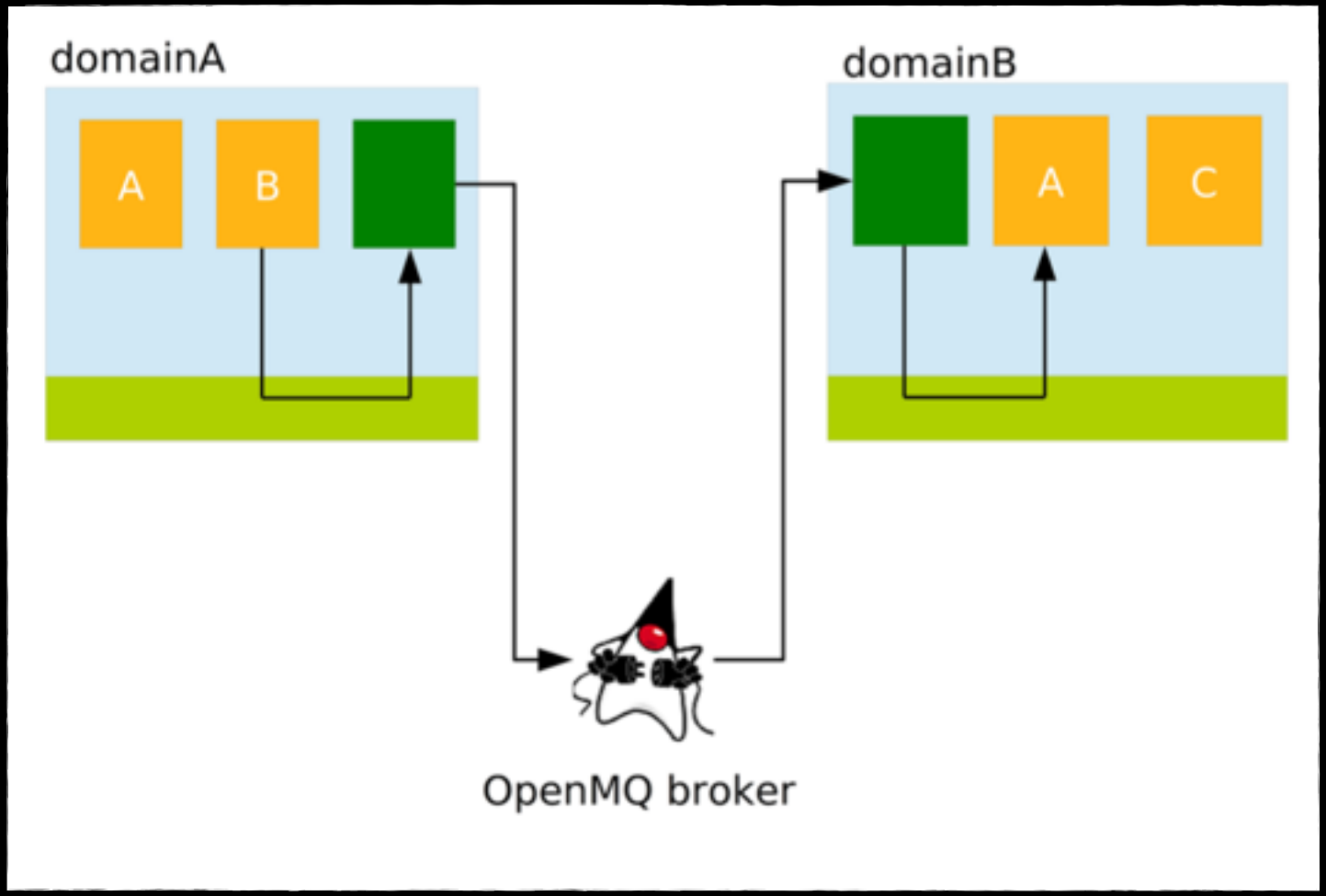
Use cases for redundancy in dCache

- Planned maintenance and rolling upgrades
- Crashes
- Horizontal scaling

NORDUnet
Nordic infrastructure for Research & Education

Default is a star topology




domainA
A
B
domainB
A
C
OpenMQ broker

dCache services

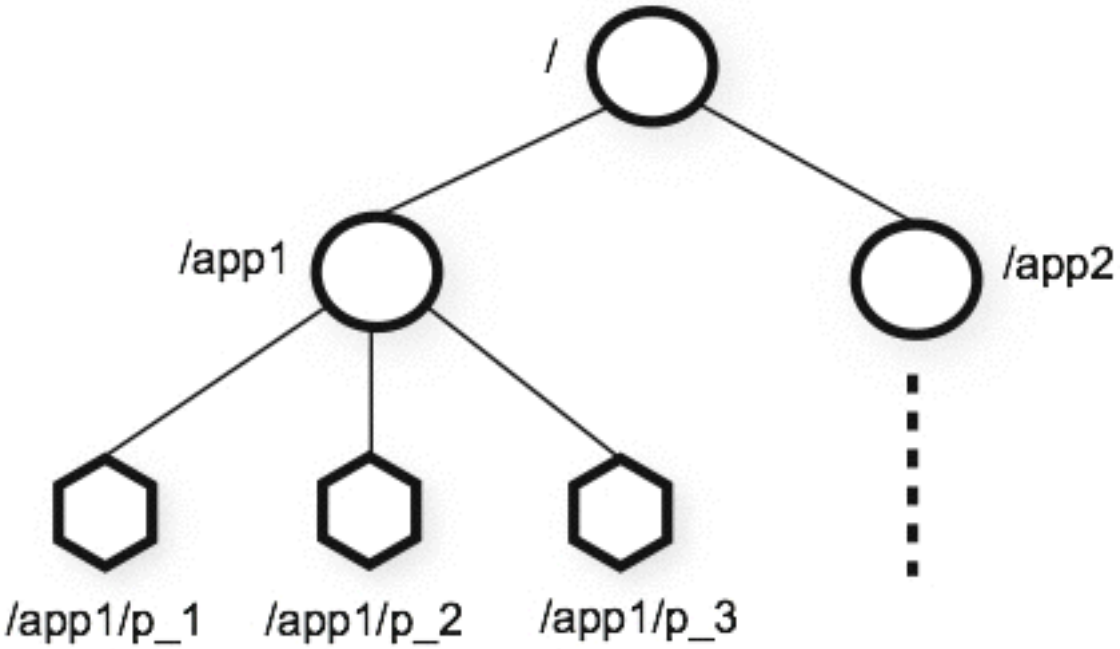
dCacheDomain

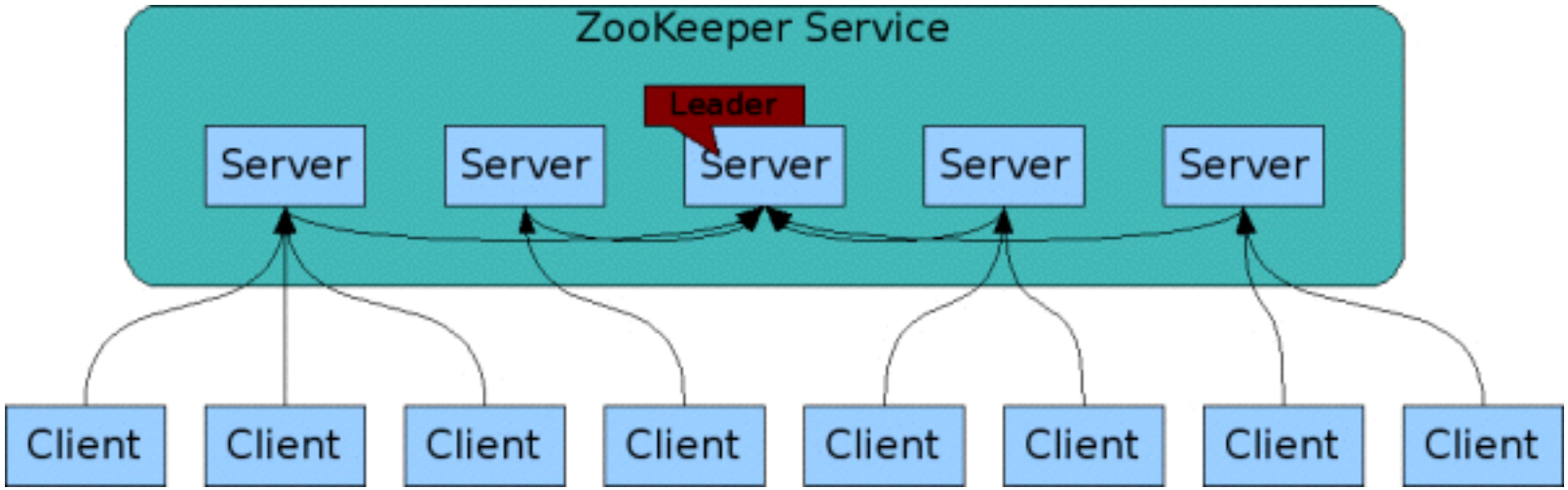
Location … Daemon

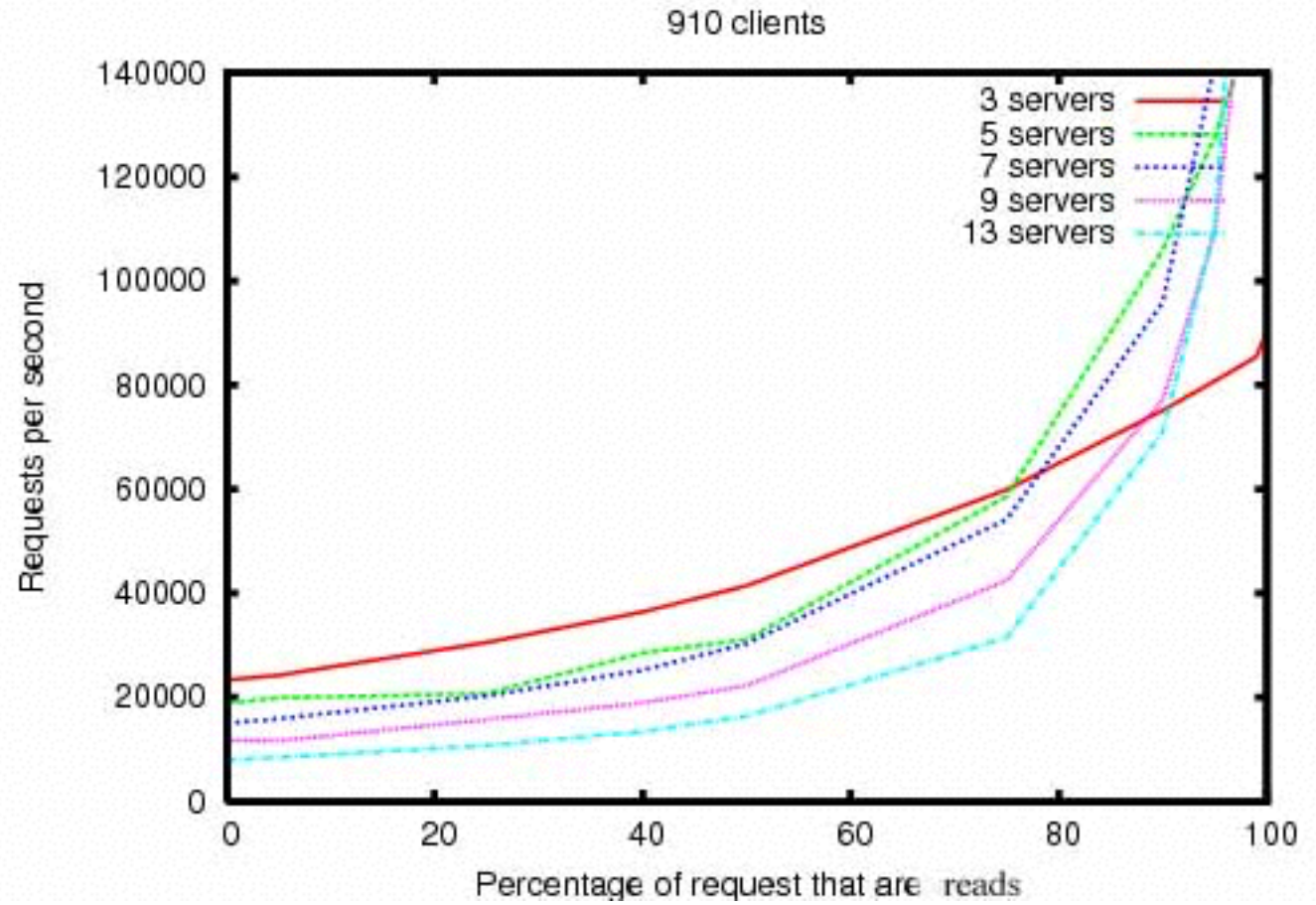
Apache ZooKeeper

Apache ZooKeeper

ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications.

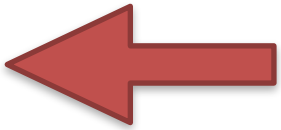

/
/app1
/app2
/app1/p_1
/app1/p_2
/app1/p_3


ZooKeeper Service
Leader
Server
Server
Server
Server
Server
Client
Client
Client
Client
Client
Client
Client
Client


910 clients
140000
120000
100000
80000
60000
40000
20000
0
Requests per second
0
20
40
60
80
100
Percentage of request that are reads
3 servers
5 servers
7 servers
9 servers
13 servers

```
#  ---- ZooKeeper connection string
#
#    Usually a comma separated list of host:port pairs.
#
dcache.zookeeper.connection = ${dcache.broker.host}:${dcache.zookeeper.net.port}
```

```
[dCacheDomain/admin]
admin.paths.history=${system-test.home}/var/admin/history

[dCacheDomain/zookeeper]

[dCacheDomain/alarms]

[dCacheDomain/topo]
```



```
[zookeeperDomain]
[zookeeperDomain/zookeeper]

[dCacheDomain]
```

https://zookeeper.apache.org/doc/trunk/zookeeperAdmin.html

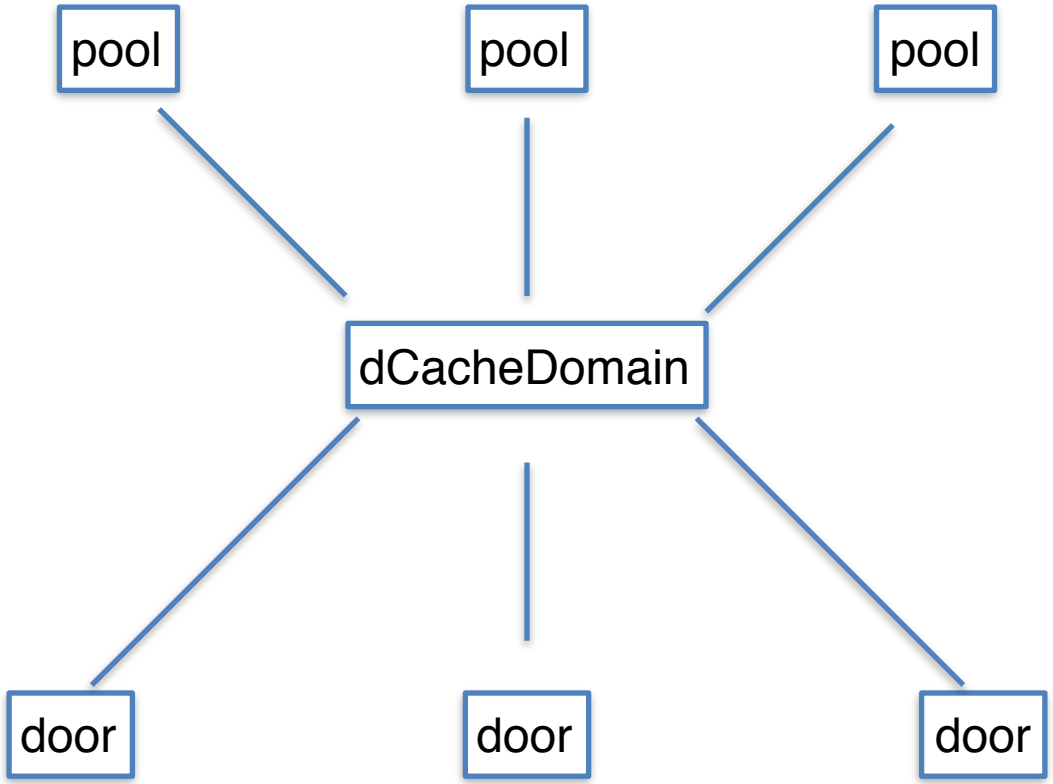
```
[Gerds-MacBook-Pro] (System@dCacheDomain) admin > \h zk get
NAME
       zk get -- get zookeeper node data

SYNOPSIS
       zk get path
```

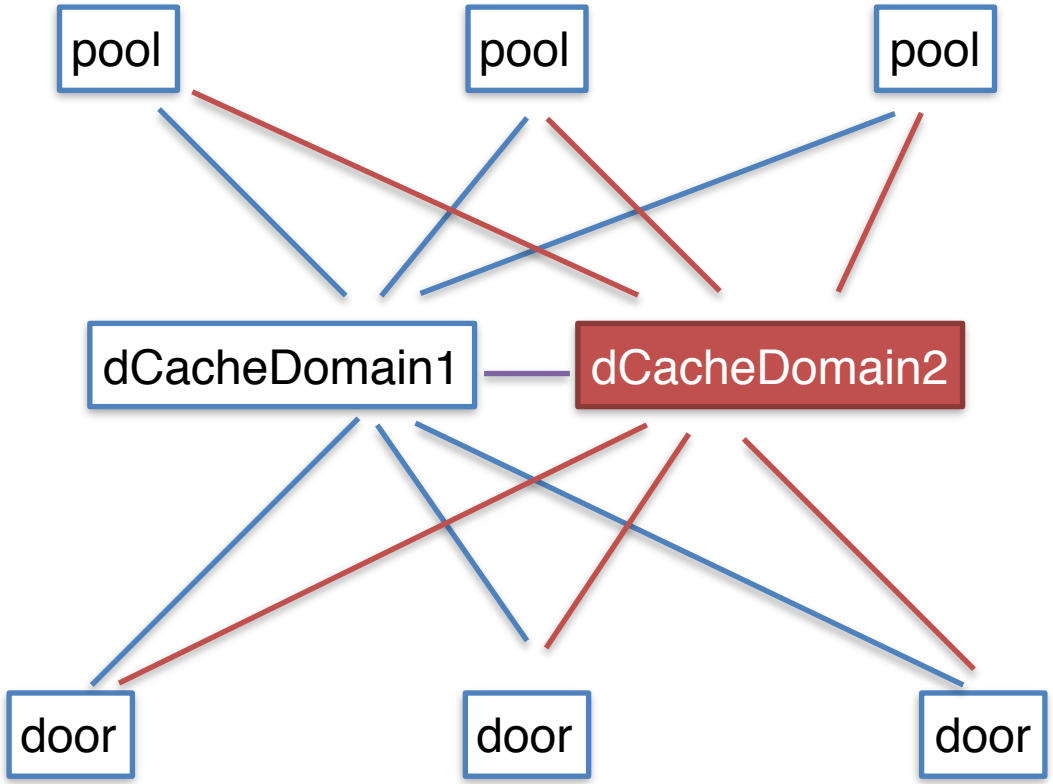
```
[Gerds-MacBook-Pro] (System@dCacheDomain) admin > \h zk ls
NAME
       zk ls -- list zookeeper node

SYNOPSIS
       zk ls [path]

ARGUMENTS
       path
              Defaults to /.
```

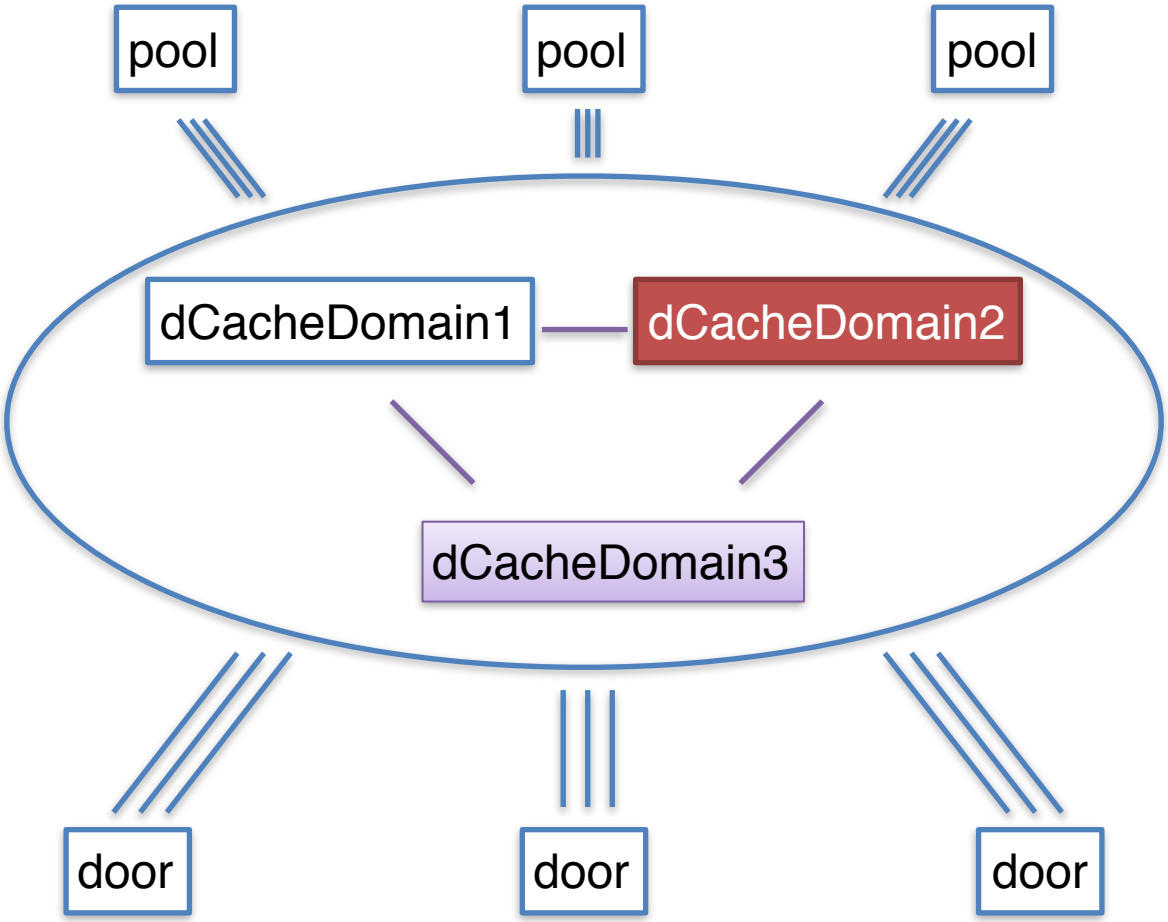
```
[Gerds-MacBook-Pro] (System@dCacheDomain) admin > zk ls
dcache
zookeeper
[Gerds-MacBook-Pro] (System@dCacheDomain) admin > zk ls /dcache/lm/cores
hub2
hub1
[Gerds-MacBook-Pro] (System@dCacheDomain) admin > zk get /dcache/lm/cores/hub1
192.168.0.10:11111
[Gerds-MacBook-Pro] (System@dCacheDomain) admin > zk get /dcache/lm/cores/hub2
192.168.0.10:11112
```

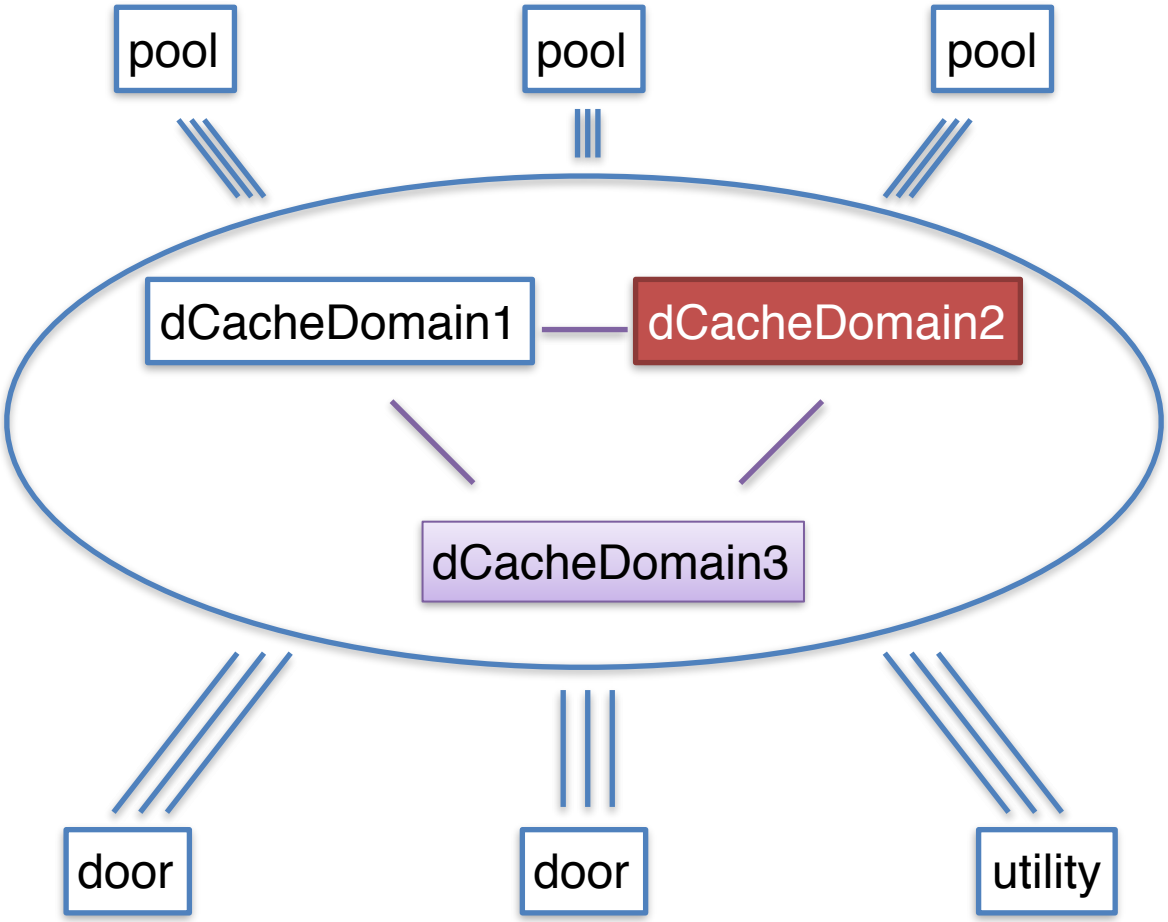
```
$ zkCli
Connecting to localhost:2181
Welcome to ZooKeeper!
JLine support is enabled

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[zk: localhost:2181(CONNECTED) 0] ls /dcache/lm/cores
[hub2, hub1]
[zk: localhost:2181(CONNECTED) 1] get /dcache/lm/cores/hub1
192.168.0.10:11111
cZxid = 0x3a8
ctime = Thu Apr 07 08:46:38 CEST 2016
mZxid = 0x3a8
mtime = Thu Apr 07 08:46:38 CEST 2016
pZxid = 0x3a8
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x153ed45953f0058
dataLength = 18
numChildren = 0
```

```
#  ---- Which message broker implementation to use
#
#   Selects between various message brokers. The message broker
#   determines how dCache domains communicate with each other. Valid
#   values are:
#
#   'none' no broker connection is establish. This is used for single
#   domain deployments.
#
#   'cells' is the classic cells based system. It relies on a central
#   broker domain that all domains connect to. By default that domain
#   is the dCacheDomain.
#
#   'core' domains are message brokers that forward messages on behalf
#   of satellite domains. All domains connect to all core domains.
#   There may be more than one core domain.
#
#   'satellite' domains connect to all broker domains.
#
(not-for-services,one-of?cells|none|core|satellite)\
dcache.broker.scheme = cells
```


pool
pool
pool
dCacheDomain
door
door
door


pool
pool
pool
dCacheDomain1
dCacheDomain2
door
door
door


pool
pool
pool
dCacheDomain1
dCacheDomain2
dCacheDomain3
door
door
door


pool
pool
pool
dCacheDomain1
dCacheDomain2
dCacheDomain3
door
door
utility

dCache services

Multi-pathing

Apache ZooKeeper

Redundant services

- Fully qualified cell addresses (domain name must be unique)
  System@dCacheDomain

- Wellknown cell addresses (cell name must be unique)
  PnfsManager

- Topics (publish-subscribe)
  LoginBrokerTopic

- Named Queues

  Multiple writers, multiple readers, but each message is only delivered to one of them.

Demo

Services

- admin
- alarms
- billing
- cleaner
- gplazma
- hoppingmanager
- httpd
- info
- pinmanager
- pnfsmanager
- poolmanager
- replica
- resilience
- spacemanager
- statistics
- topo
- transfermanagers

Named queues

dC vices

Multi-pathing

dC main

Location Daemon

Apache ZooKeeper

PostgreSQL redundancy

- Master - slave works out of the box.
- repmgr - replication manager for PostgreSQL allows for easy fail-over (http://www.repmgr.org).
- PgBouncer allows for transparent (to the client) fail-over between master and slave (https://pgbouncer.github.io).
- Recent versions of the JDBC driver actually have integrated support for connection fail-over from master to slave.

From some talk in 2011

Future Developme

@NDGF, storage focussed

- Self-service portal
  - Fine grained authorization
  - Web interface for file management
  - Short-term password generator for non-X509 authentication
  - Integration with federated identity providers
- Move to FHS compliant packaging (& OSGi?)
- Decoupled, redundant and secure messaging
- More flexible data placement and movement

What to remember from this talk

- Make a standalone Apache ZooKeeper installation.
  - Essential for reliable behaviour when dCache domains restart.
  - Makes upgrading easier.

- dCache can have redundant services.
  - 2.16 will have redundant cell communication;
  - and some services can be redundant;
  - others will follow in subsequent releases.