

Ceph | Gluster | Swift

Similarities and Differences






Who, What, Why?






Projects, brief history and Community

- Open Source
- Software defined storage
- Commodity hardware
- No vendor lock-in
- Massively scalable
 - CERN, Facebook, Rackspace
- Vibrant Community

| | | | |
|-----------------|--|--|--|
| |  |  |  |
| Started | 2007 | 2005 | 2010 |
| Language | C++ | C | Python |



Storage Types

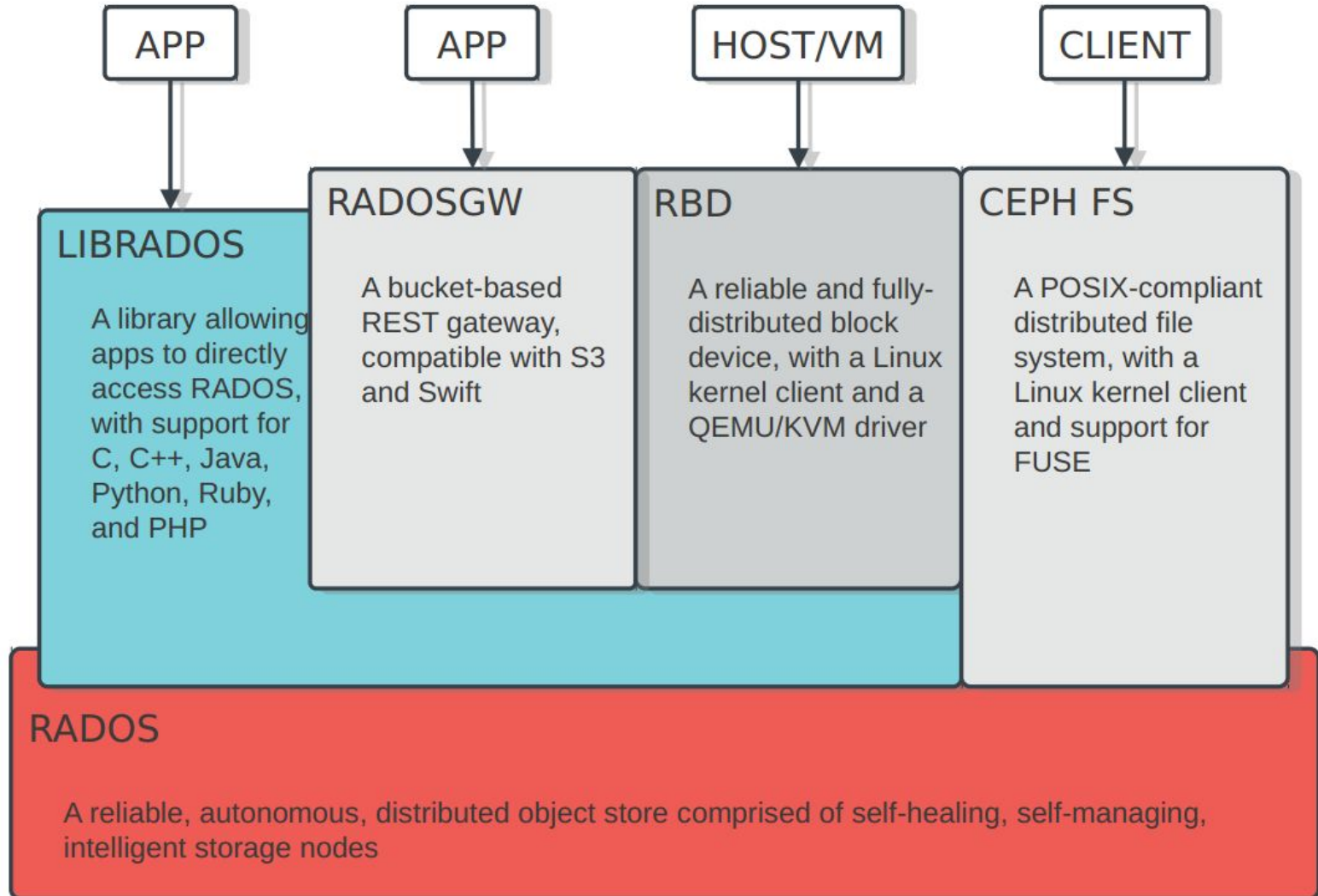
| |  |  |  |
|---------------|---|---|---|
| BLOCK | ✓ | ✓ | ✗ |
| FILE | ✓ | ✓ | ✗ |
| OBJECT | ✓ | ✓ | ✓ |



Architecture



Ceph Architecture

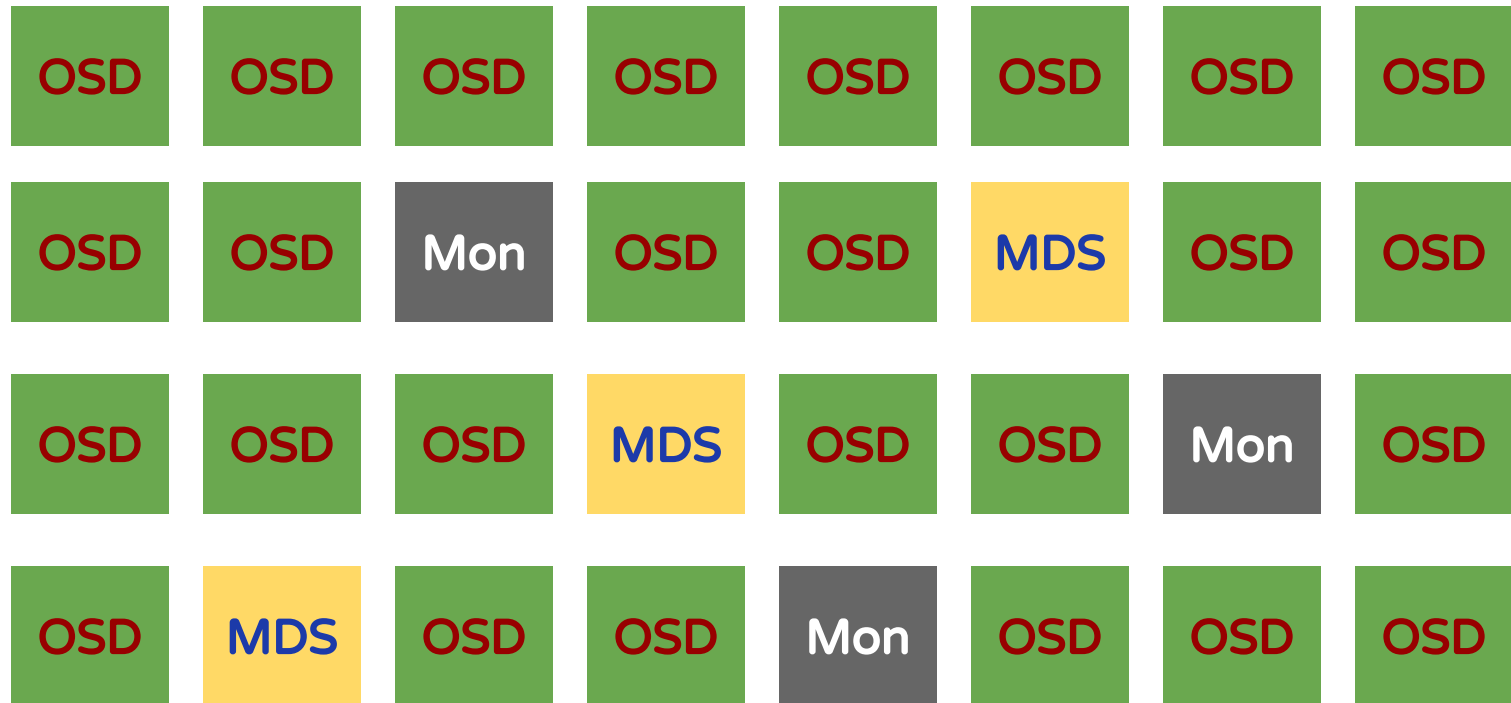


OSDs

- 10s to 1000s
- One per disk
- Serves objects to clients
- Peer replication

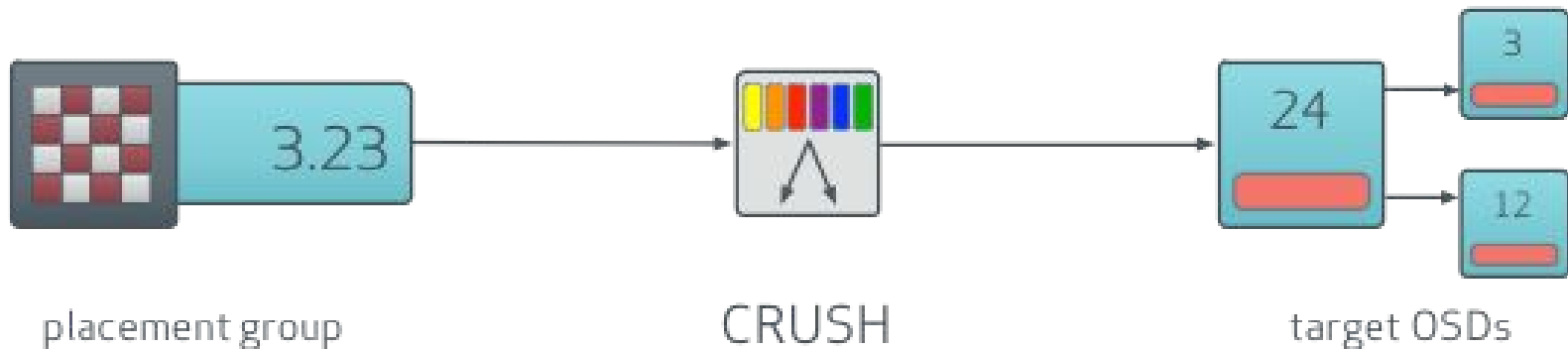
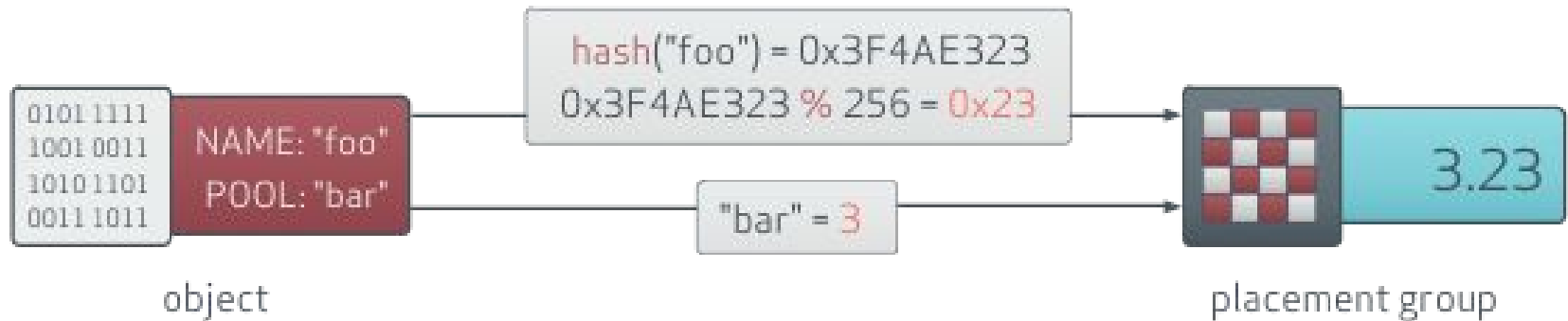
Monitors

- Maintain cluster membership and state
- Consensus for decision making
- Small, odd number

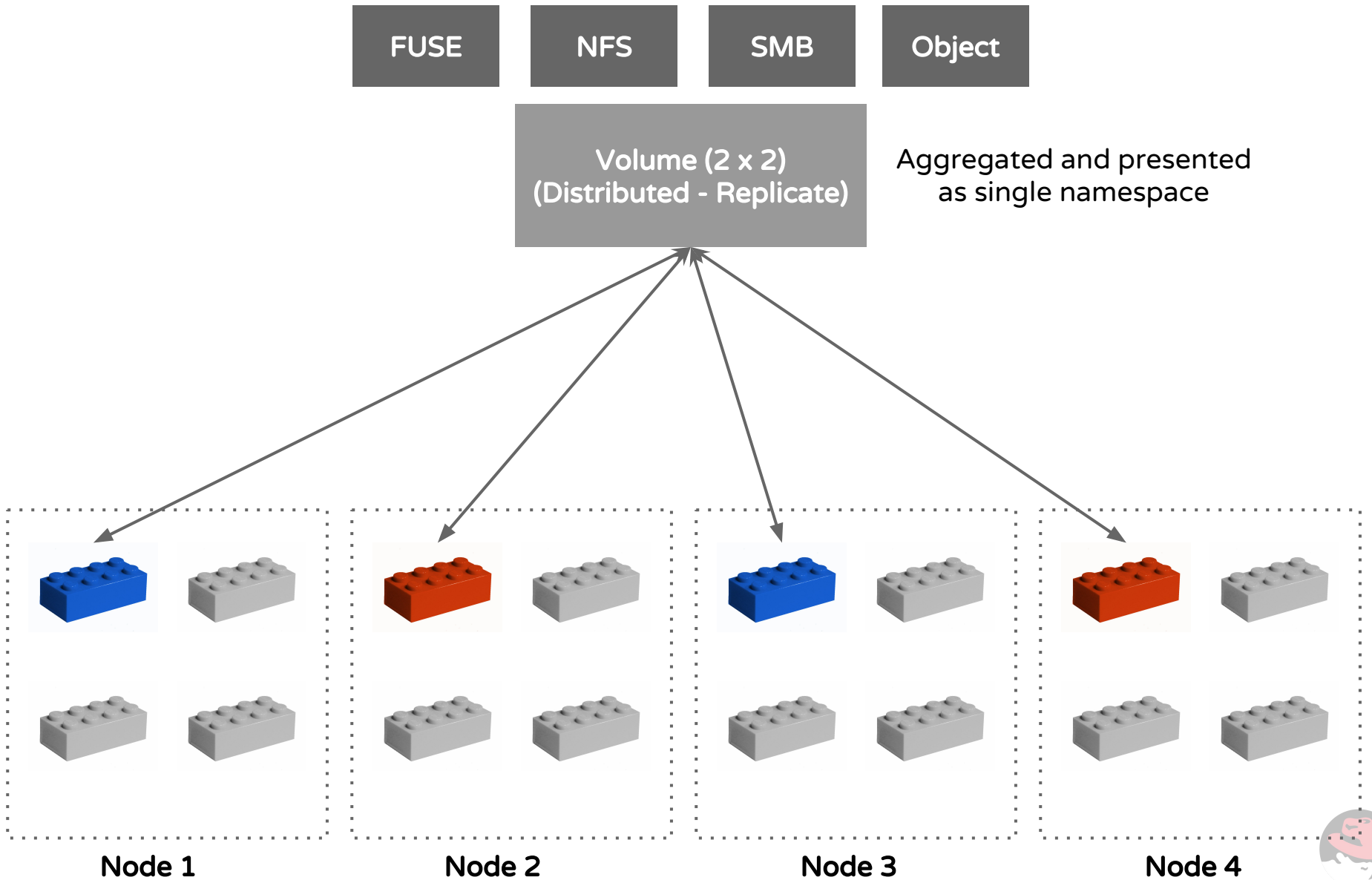


Distribution and Replication in Ceph - CRUSH

- Pools are logical groups
- Pools are made up of PGs
- PGs mapped to OSDs
- Rule based configuration
- Pseudo-random placement
- Repeatable and deterministic

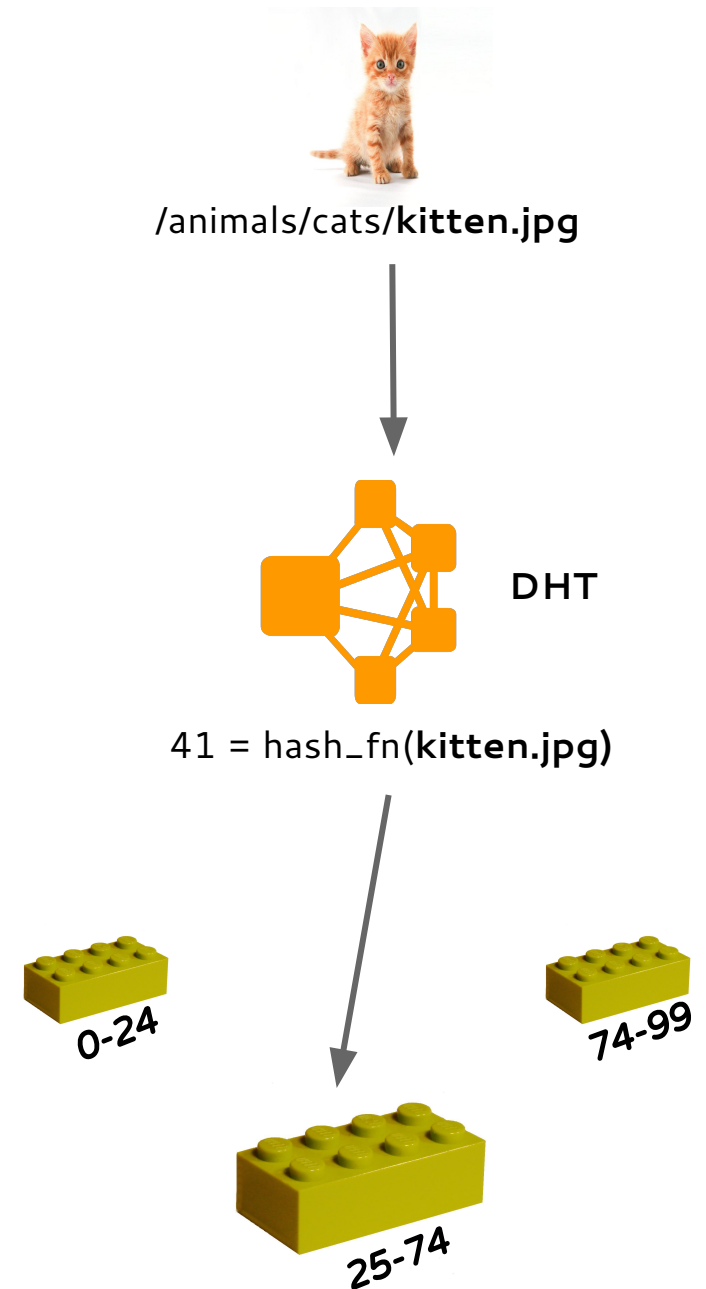


Gluster Architecture



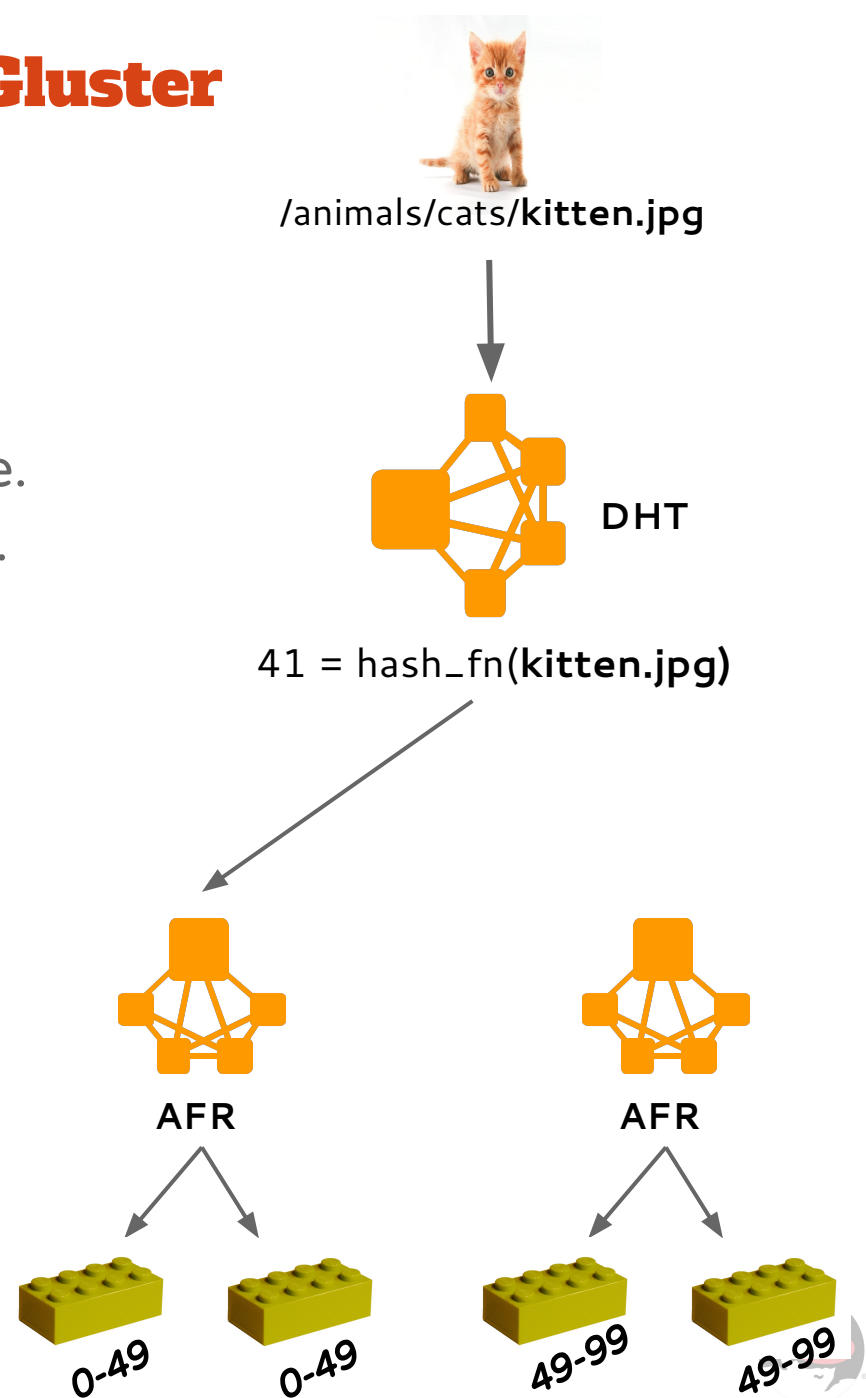
Distribution in Gluster

- No central metadata server (No SPOF).
- Hash space divided into N ranges mapped to N bricks.
- Directories are created on all bricks.
- Hash ranges assigned to directories.
- Renames are special.
- Rebalance moves data.

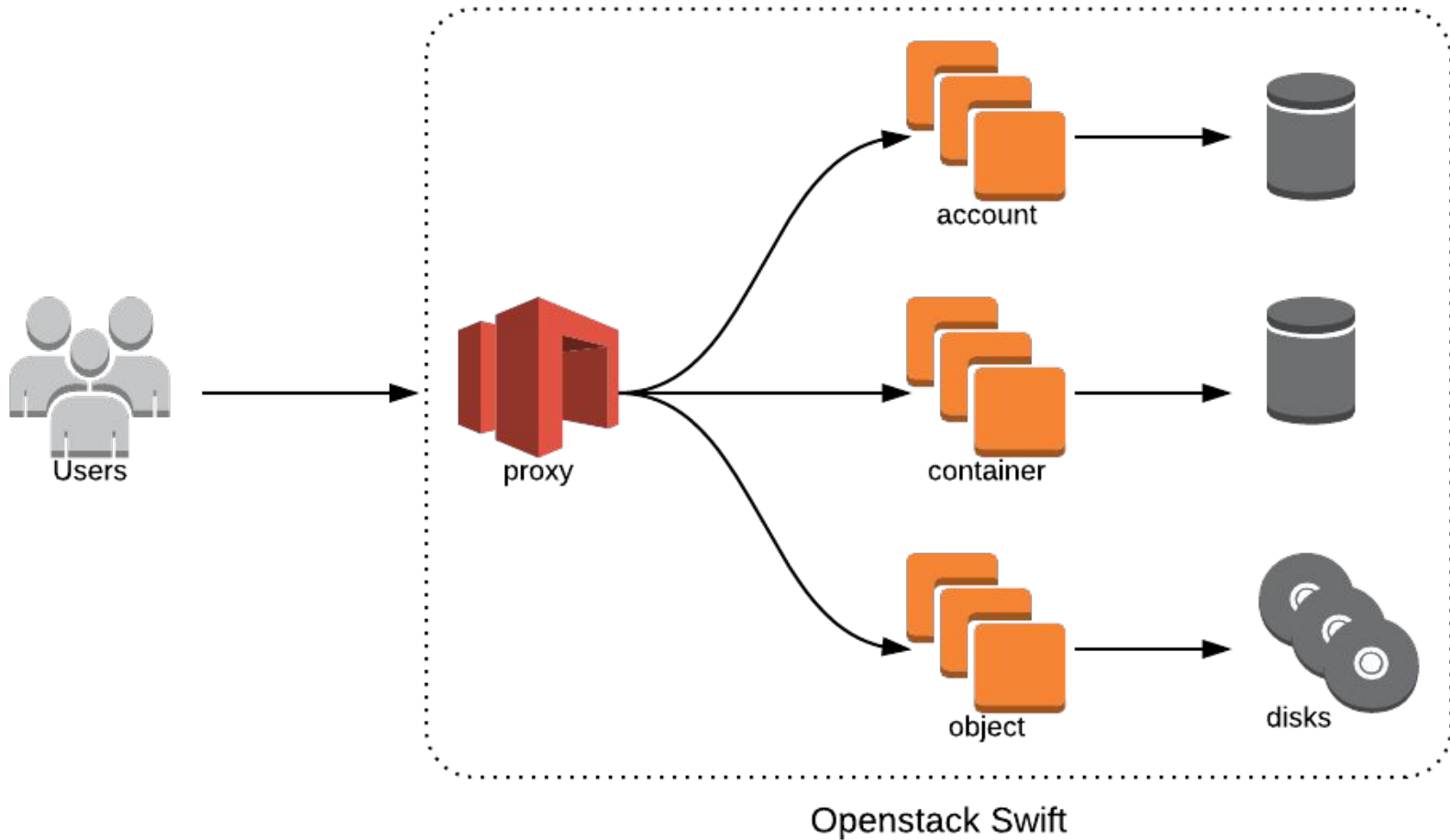


Distribution + Replication in Gluster

- Replication is synchronous.
- Provides high availability on failure.
- Self-healing (automatic file repair).
- Optionally enforce quorum.
- Follows a transaction model.

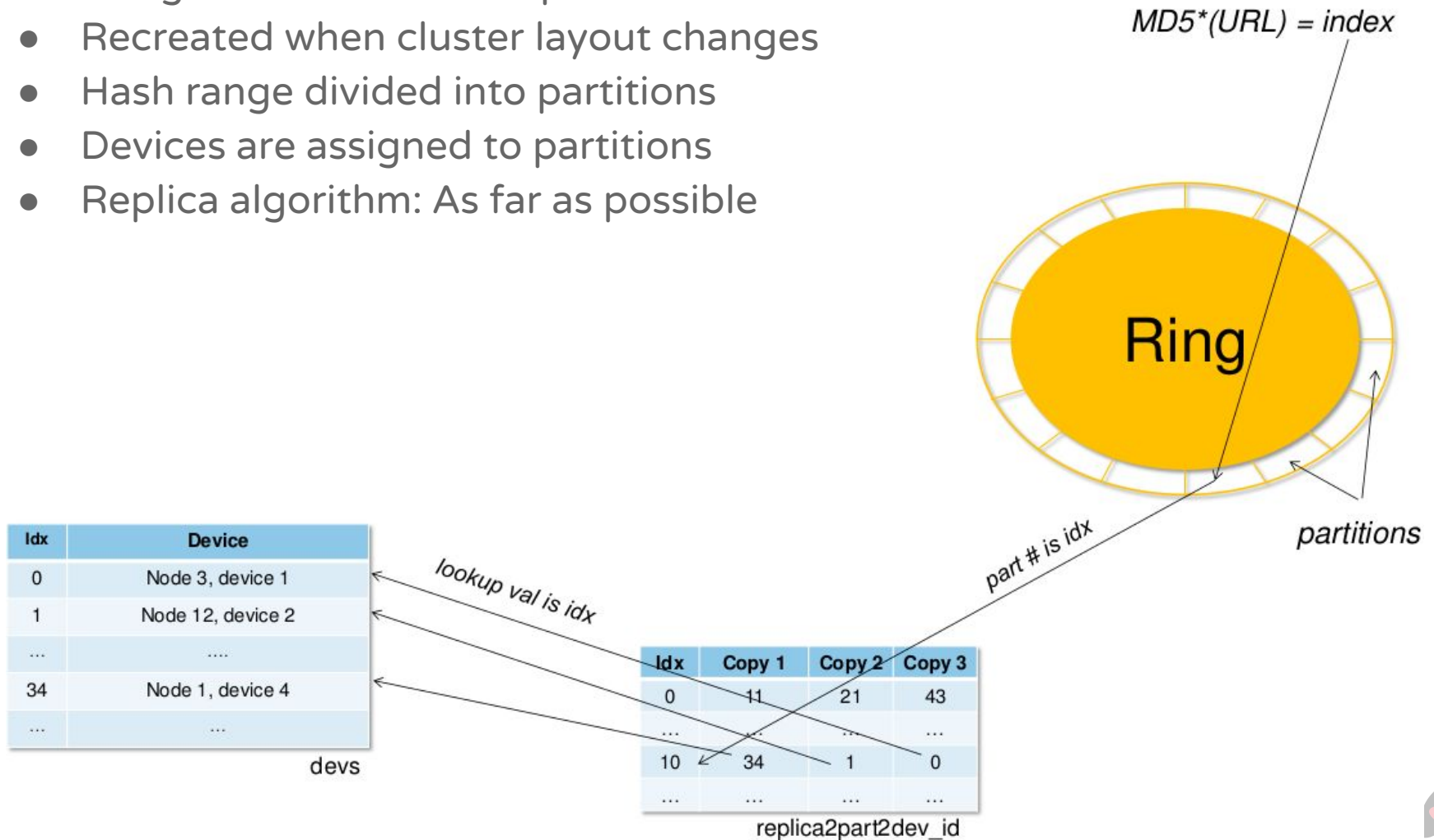


Swift Architecture



Distribution and Replication in Swift

- “Ring files” = Cluster map
- Recreated when cluster layout changes
- Hash range divided into partitions
- Devices are assigned to partitions
- Replica algorithm: As far as possible



Similarities and Differences



Storage Nodes



Communicate directly
with ceph client.
(socket)



OSD
(Object
Server
Daemon)

FS

← xfs
(fs with xattr
support)

BLOCK



Communicate directly
with glusterfs client.
(socket)



glusterfsd

FS

← xfs
(fs with xattr
support)

BLOCK



Communicate with
swift proxy server.
(HTTP)



swift
object
server




FS

← xfs
(fs with xattr
support)

BLOCK



Differences in Redundancy and Rebalance

| |  |  |  |
|---|---|---|---|
| Redundancy type (Replication and EC) and redundancy factor granularity | Pool | Volume | Container |
| Replica placement into failure domains | Managed by CRUSH | Manual effort by Admin ^[1] | Managed by Rings |
| Rebalance migrates | Placement Groups | Individual Files | Partitions |



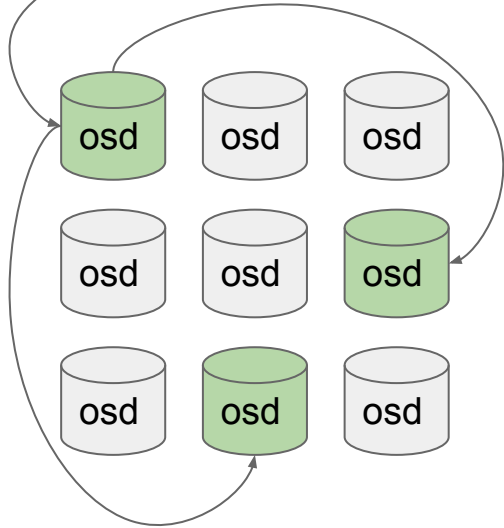
[1] - Luis Pabon will save us
<https://github.com/heketi/heketi>



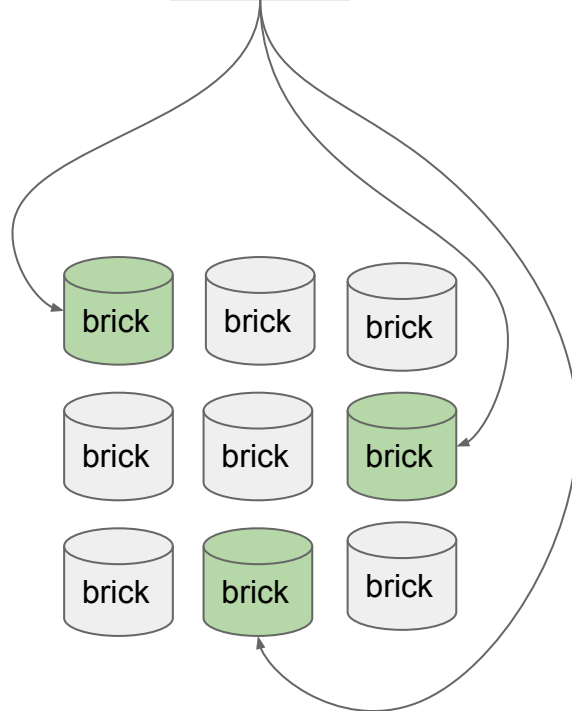
Replication



client RBD
librados

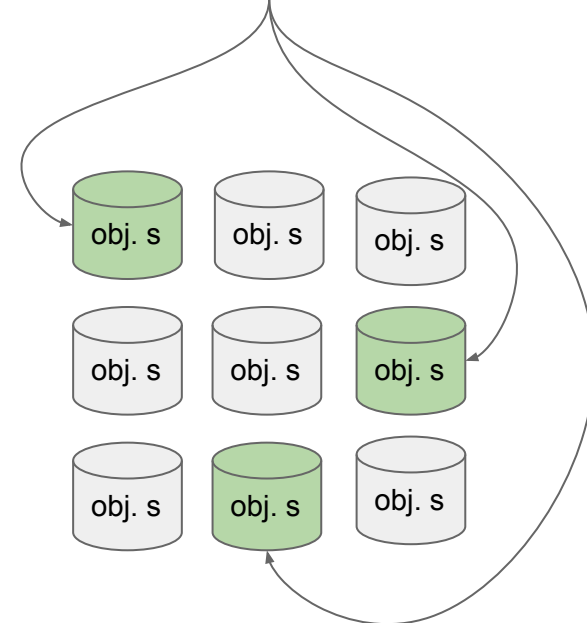


client Fuse
libgfapi

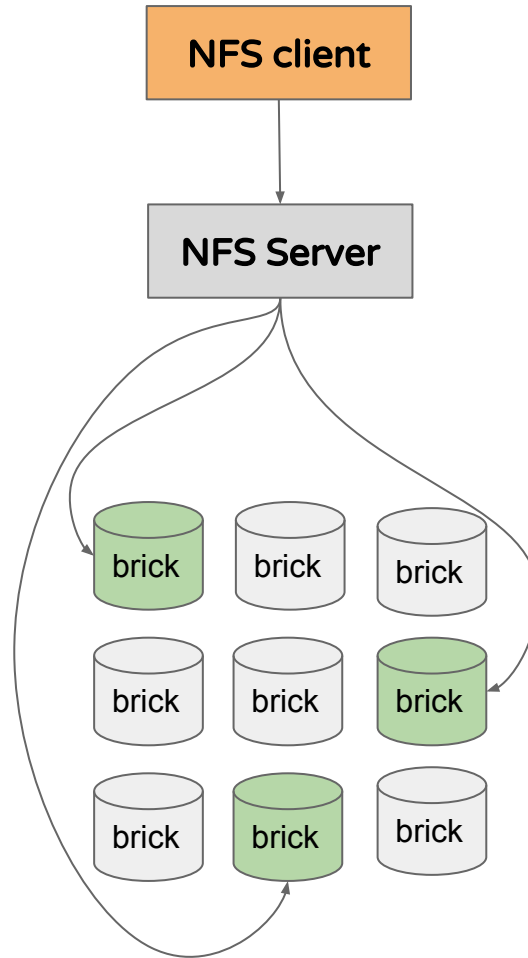
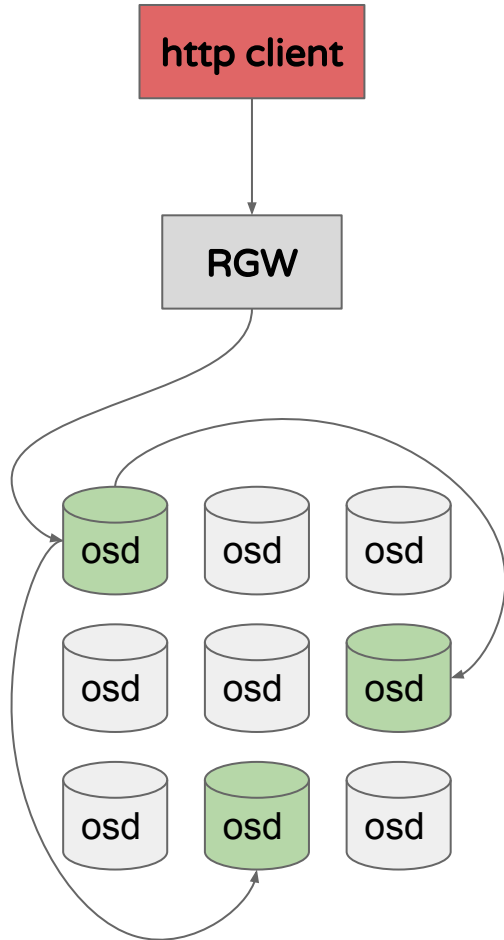


http client

proxy



Replication



Where's my data ?



```
# rados put -p testpool kitten.jpg kitten.jpg  
# ceph osd map testpool kitten.jpg  
  osdmap e14 pool 'testpool' (3) object 'kitten.jpg' -> pg 3.9e17671a (3.2) -> up [2,1] acting [2,1]
```

```
/var/lib/ceph/osd/ceph-2/current/3.2_head/kitten.jpg__head_9E17671A__3
```

```
# cd /mnt/gluster-vol  
# touch animals/cat/kitten.jpg
```



```
/export/brick1/animals/cat/kitten.jpg
```




```
# curl -X PUT http://example.com:8080/v1/AUTH_test/animals/cat/kitten.jpg
```



```
/mnt/sdb1/objects/778/69f/c2b307d78b6c419c0c1b76d91c08c69f/1412628708.01757.data
```



Feature Parity

| |  |  |  |
|--------------------------|---|---|---|
| Quota | Pool, bucket and user quota | Volume, Directory and Inode Count | Account and Container quota |
| Tiering | yes | yes | no |
| Geo-replication | active-passive* | active-passive | active-active |
| Erasur Coding | yes | yes | yes |
| Bit-rot detection | yes | yes | yes |



Thank You



IRC:
#ceph-devel
#gluster-dev
#openstack-swift

