



inktank

Inktank

Ceph Distributed Object Storage

MSST Tutorial, May 6, 2013

Agenda

- Introduction to Ceph and Inktank
- Challenges of 21st Century Storage
- Ceph Storage Clusters
- How Ceph Addresses these Challenges
- RBD and CephFS
- Hands-on demo
- Q&A

Hands-on Tutorial Prep

Download VM image

<http://ceph.com/tutorial>

tutorial.img.tar.gz (KVM/Qemu)

tutorial.vdi.gz (Virtualbox, ...)

2GB RAM

Attach 4 additional disks (~8GB each)

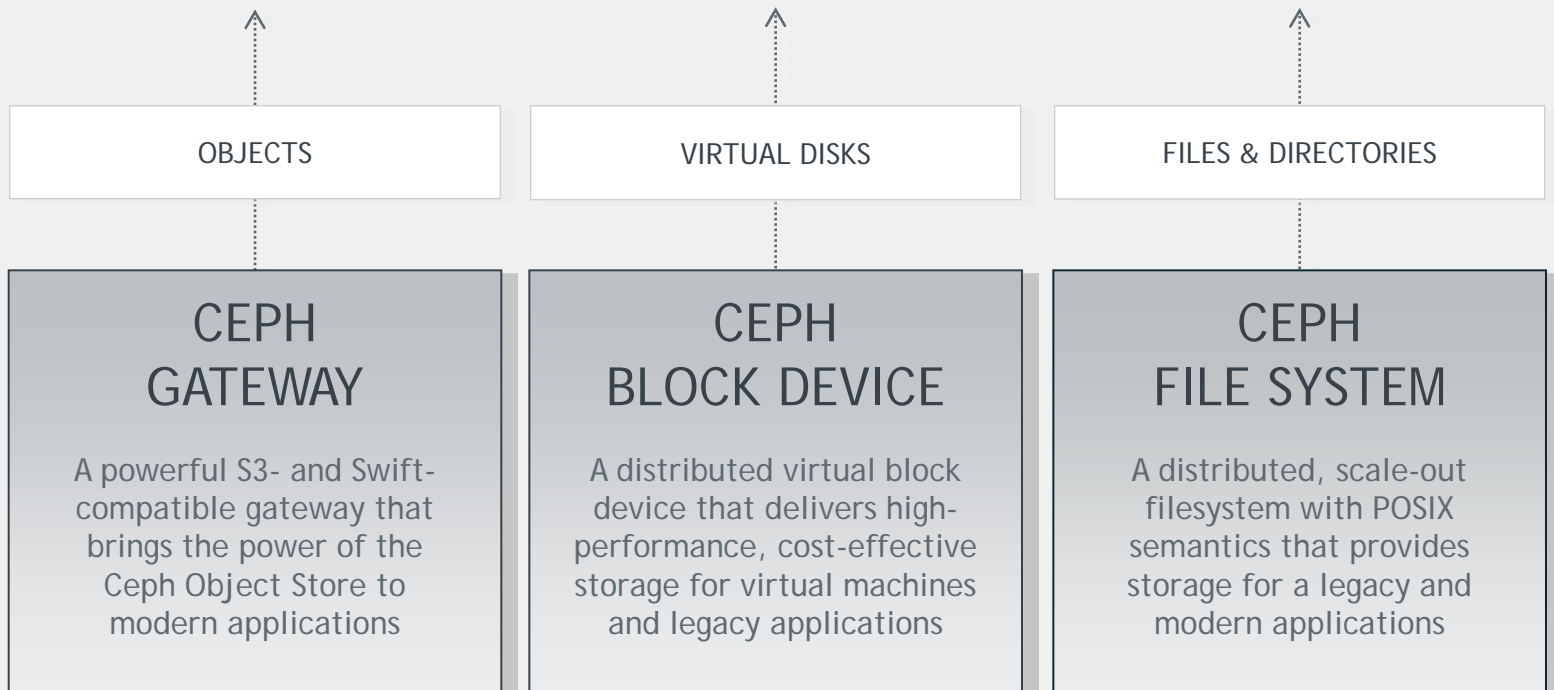


- Distributed unified object, block and file storage platform
- Created by storage experts
- Open source
- In the Linux Kernel
- Integrated into Cloud Platforms




- Company that provides professional services and support for Ceph
- Founded in 2011
- Funded by DreamHost, Mark Shuttleworth, others
- Employs core Ceph developers, including creator and maintainer

Ceph Unified Storage Platform




CEPH STORAGE CLUSTER

A reliable, easy to manage, next-generation distributed object store that provides storage of unstructured data for applications



The Challenges of 21st Century Storage



Performance: making it fast

- direct communication between clients and servers
- no proxies or redirectors


- stripe requests across multiple servers
- large requests - use the bandwidth of multiple servers
- small requests - use the IOPS of multiple servers

- good load distribution
- ensure that all servers are sharing the load
- the key to this is intelligent capacity distribution

- don't make the clients pay for write replication
- this cuts per client throughput in half (or worse)

Reliability: making it last

- data replication
 - configurable, per-pool replication factors
 - automatic failure domain aware placement
 - user-controlled persistence rules
 - support for strong consistency models
-
- no Single Points of Failure
 - configurable to withstand arbitrarily many failures
 - robust “split-brain” protection
 - rolling upgrades and live replacements
-
- prompt and automatic recovery from all failures
 - recovery cannot wait for human intervention
 - continued normal data access during recovery



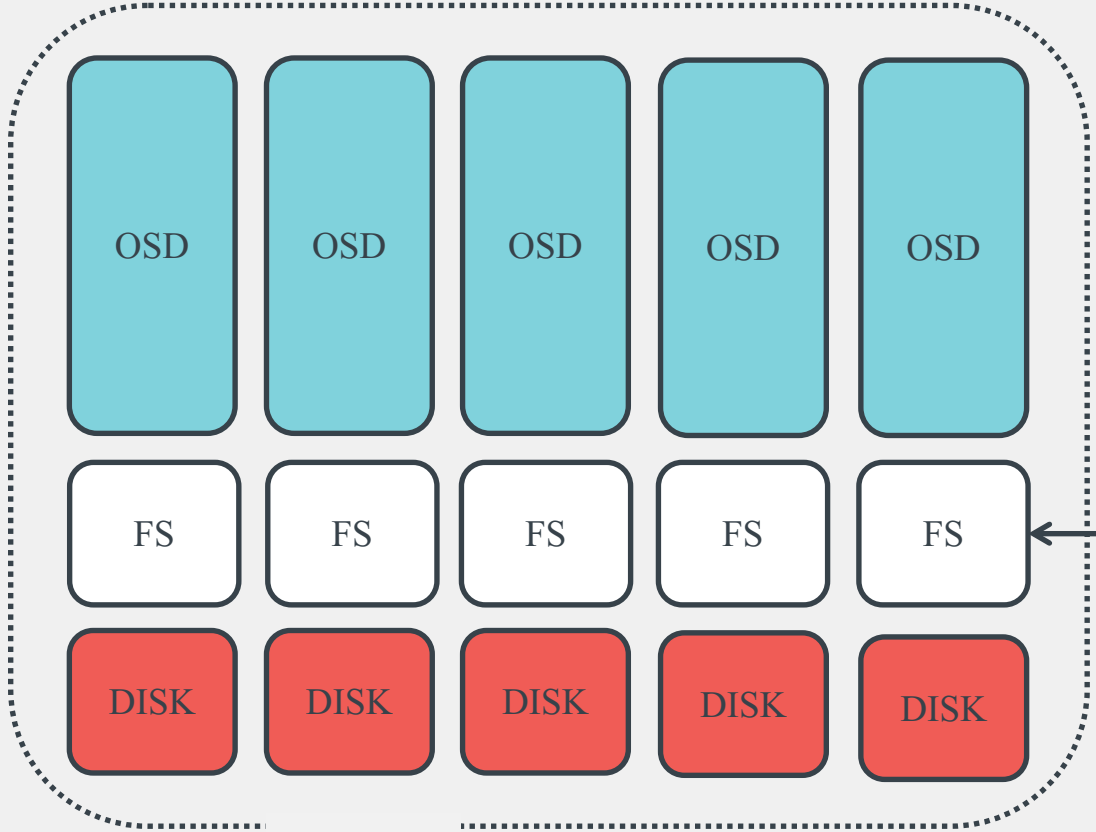
Scalability: petabytes to exabytes

- Parallelism
 - no single controlling or data-directing components
 - all work dynamically partitioned among parallel servers
 - effective work partitioning: no $O(N)$ processes
 - delegate much functionality to intelligent storage devices
- Independence
 - each operations has a single well-known owner
 - owner has complete responsibility for data integrity
 - client data updates do not require distributed services
- Self Managing
 - easy expansion, upgrade and replacement
 - automatic data re-replication after component failure
 - automatic data redistribution after component changes

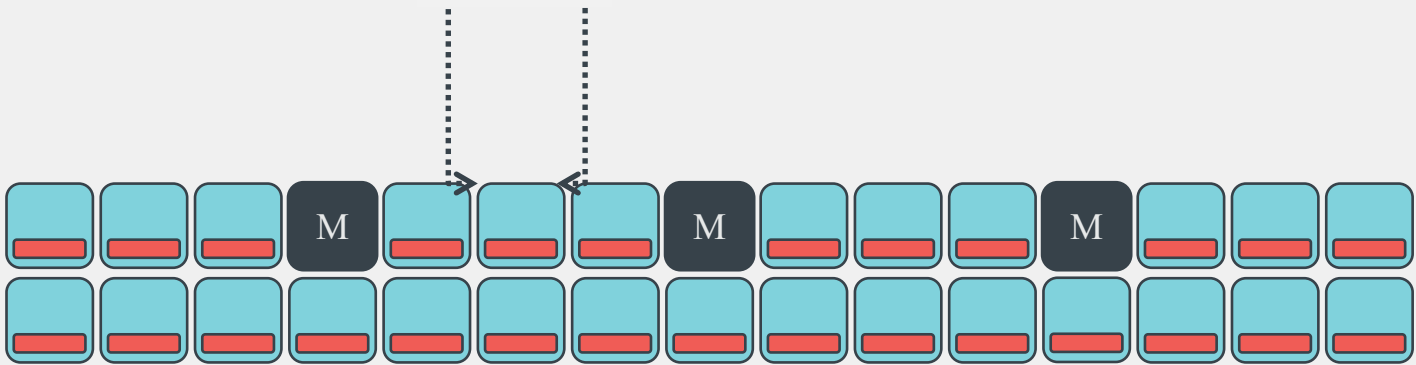


The Ceph Storage Architecture



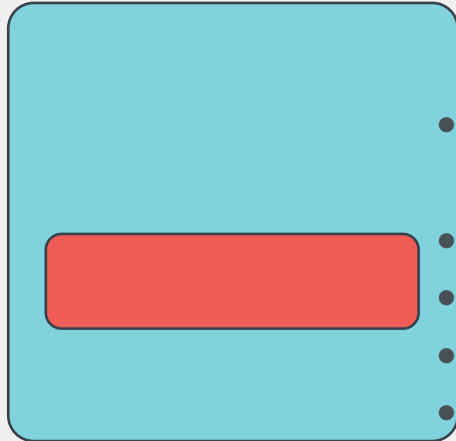


btrfs
xfs
ext4



Ceph Object Storage Daemons

Intelligent Storage Servers

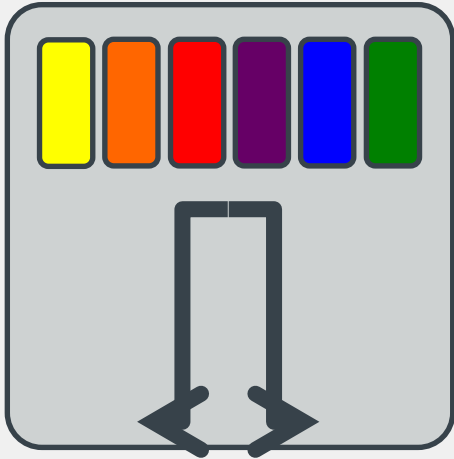


- Serve stored objects to clients
- OSD is primary for some objects
- Responsible for replication
- Responsible for coherency
- Responsible for re-balancing
- Responsible for recovery

- OSD is secondary for some objects
- Under control of primary
- Capable of becoming primary

- Supports extended object classes
- Atomic transactions
- Synchronization and notifications
- Send computation to the data

CRUSH



Pseudo-random placement algorithm

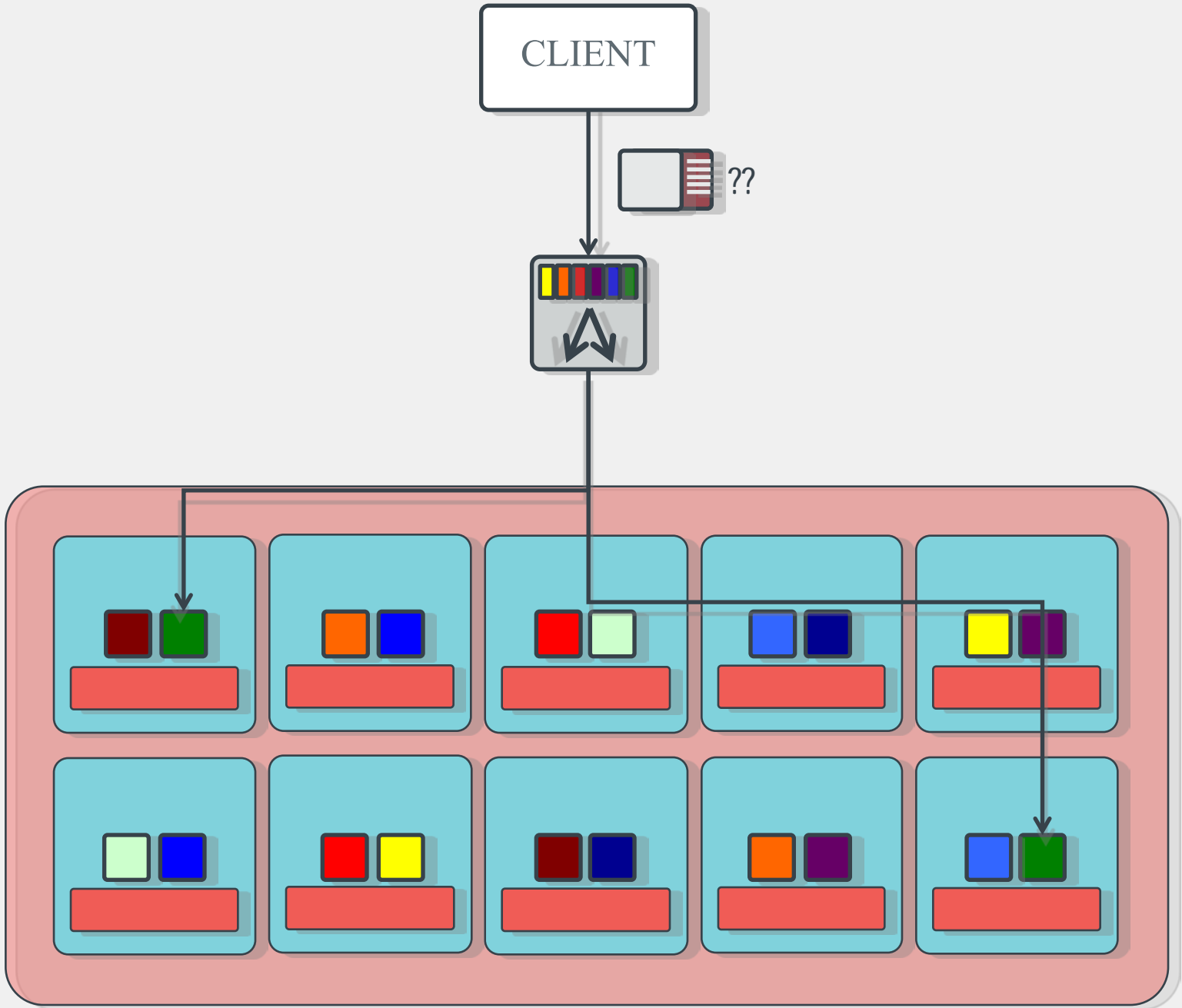
- deterministic function of inputs
- clients can compute data location

Rule-based (per pool) configuration

- desired/required replica count
- affinity/distribution rules
- infrastructure topology
- weighting for each device

Excellent data distribution

- declustered placement
- excellent data re-distribution
- migration proportional to change



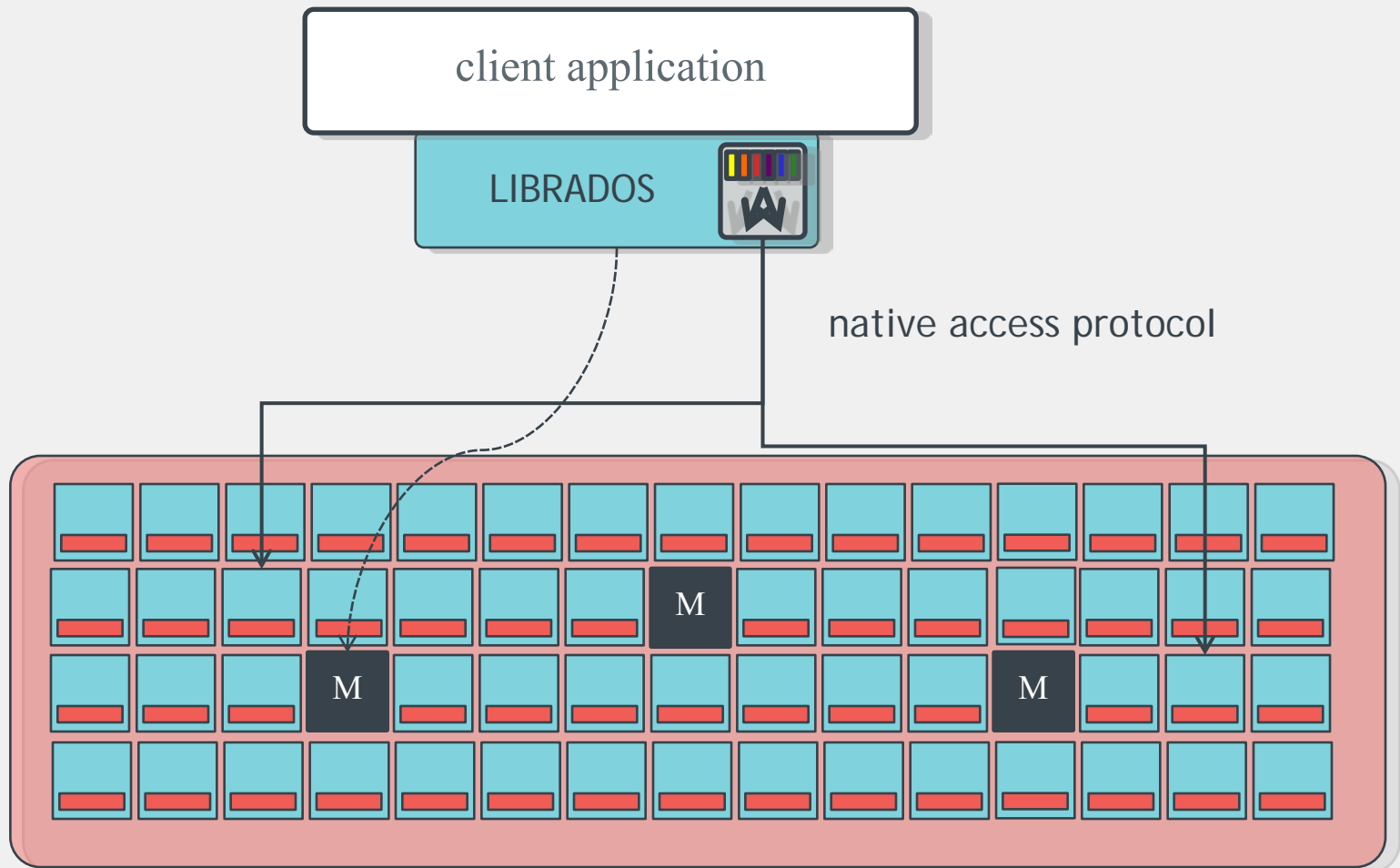
Ceph Monitors



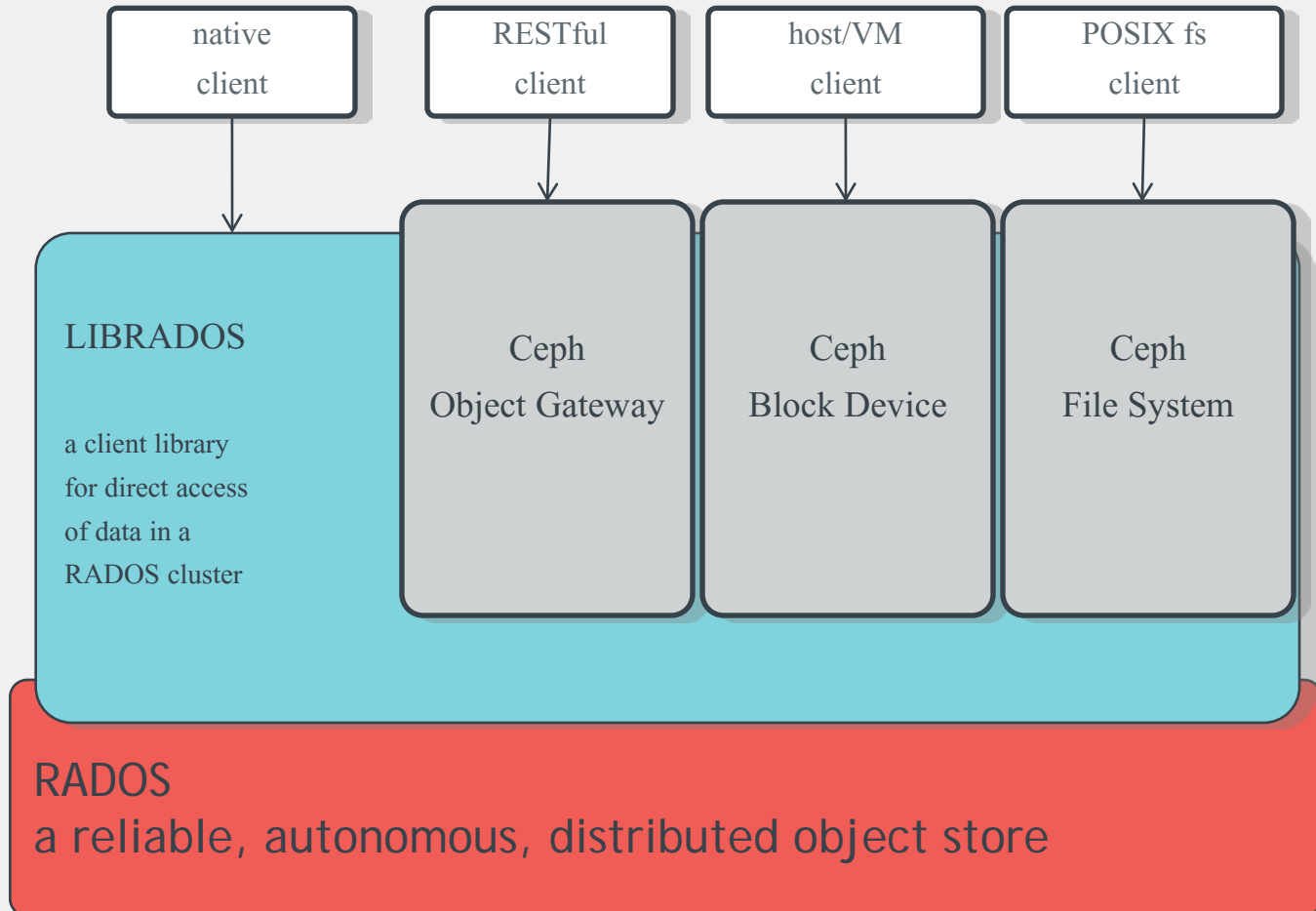
Stewards of the Cluster

- Distributed consensus (Paxos)
- arbiters of cluster state
- odd number required (quorum)
- Maintain/distribute cluster map
- map controls the CRUSH algorithm
- scalable gossip distribution protocol
- Authentication/key servers
- Monitors are **not** in the data path
- clients talk directly to OSDs

Ceph Storage Client Library



Ceph Software Layering

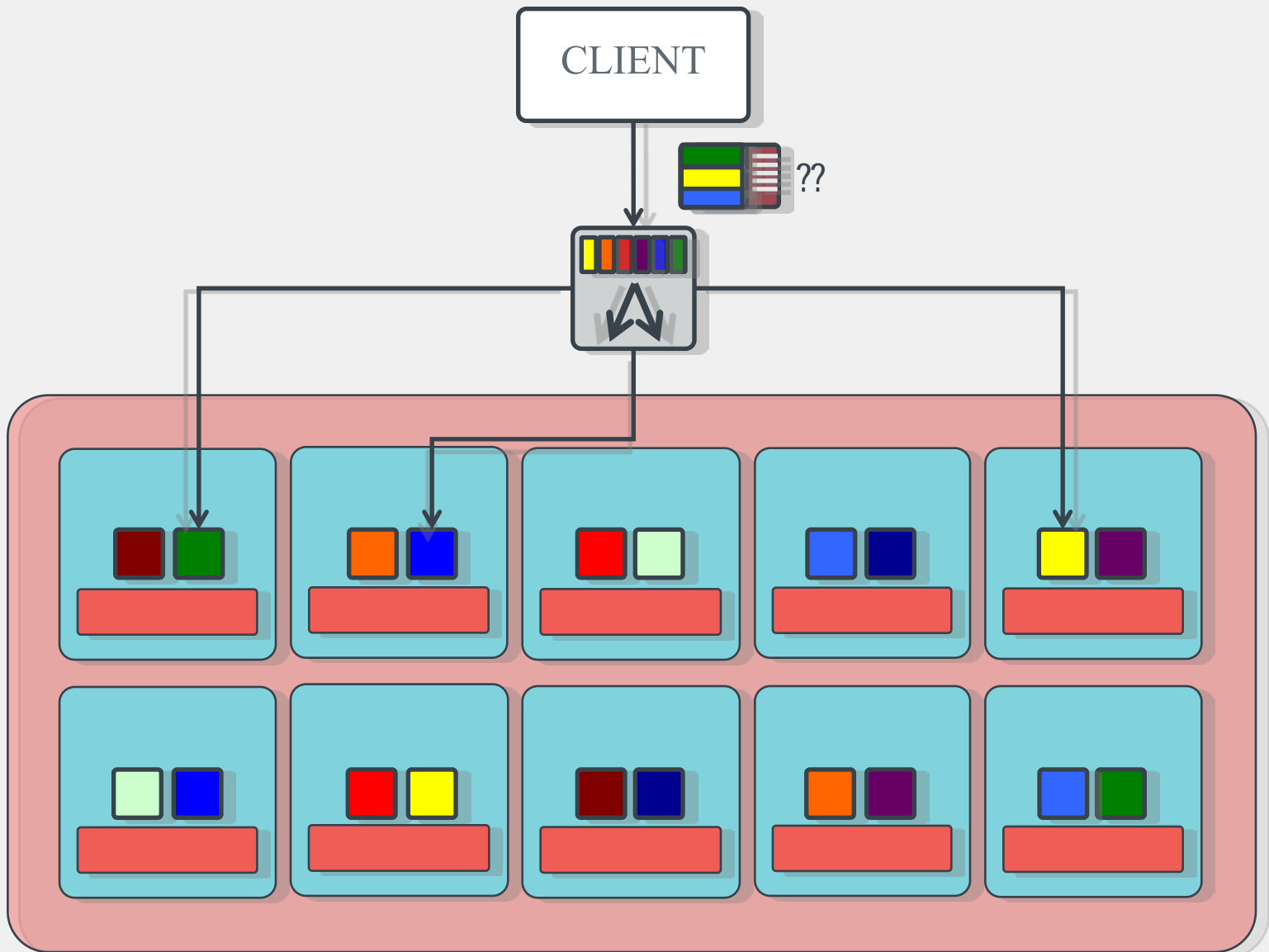




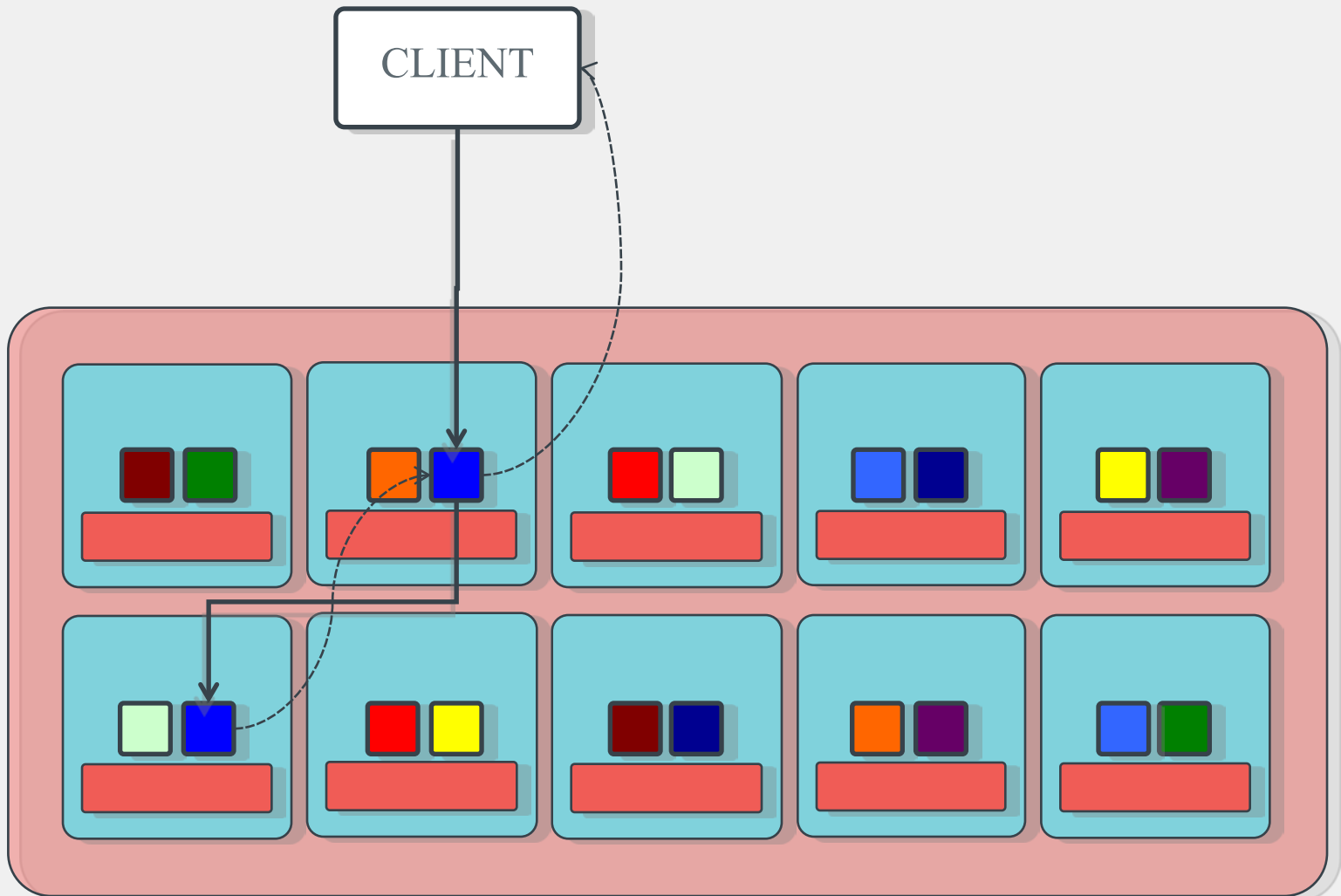
How Ceph Meets these Challenges



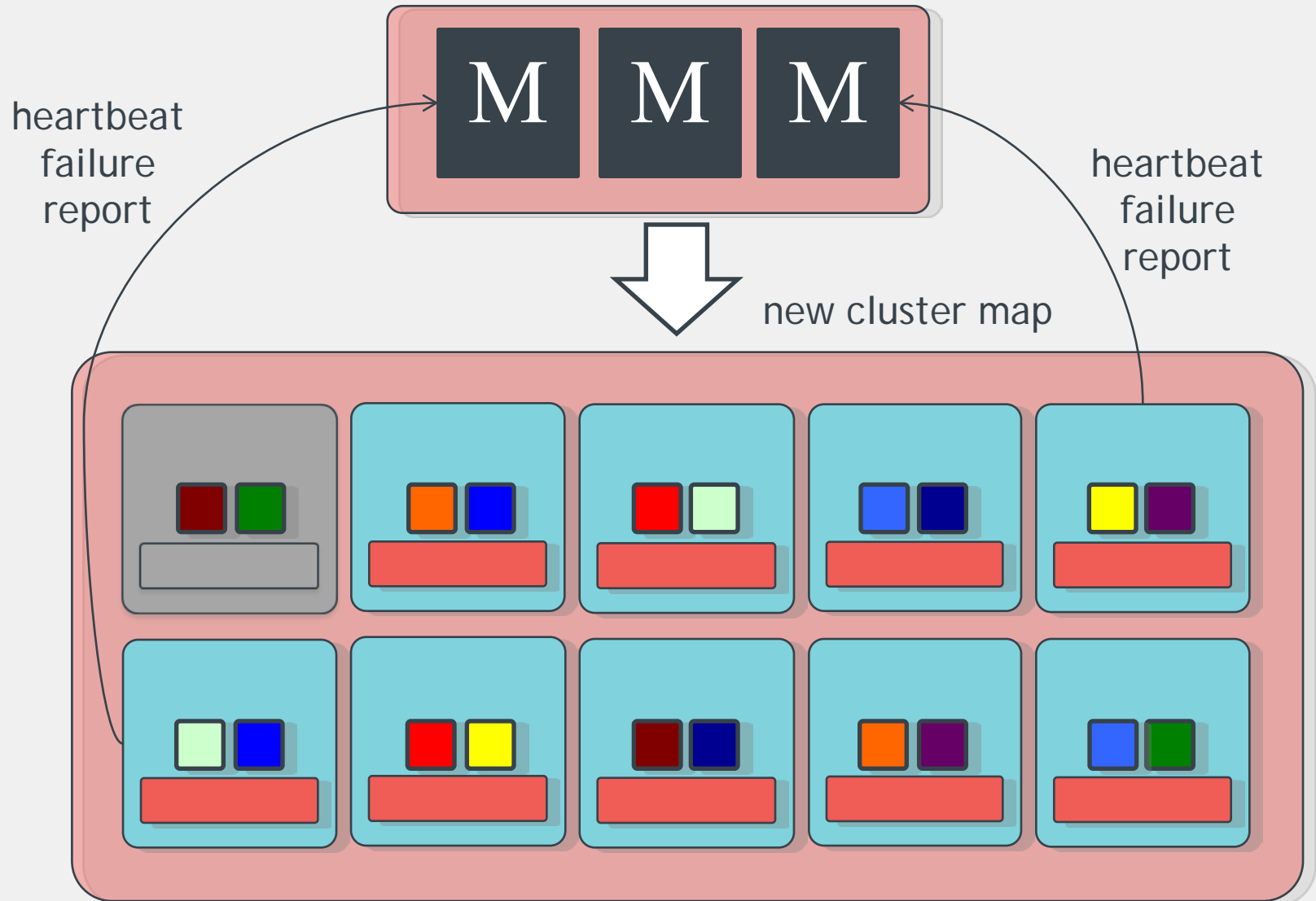
Striped Parallel Client Writes



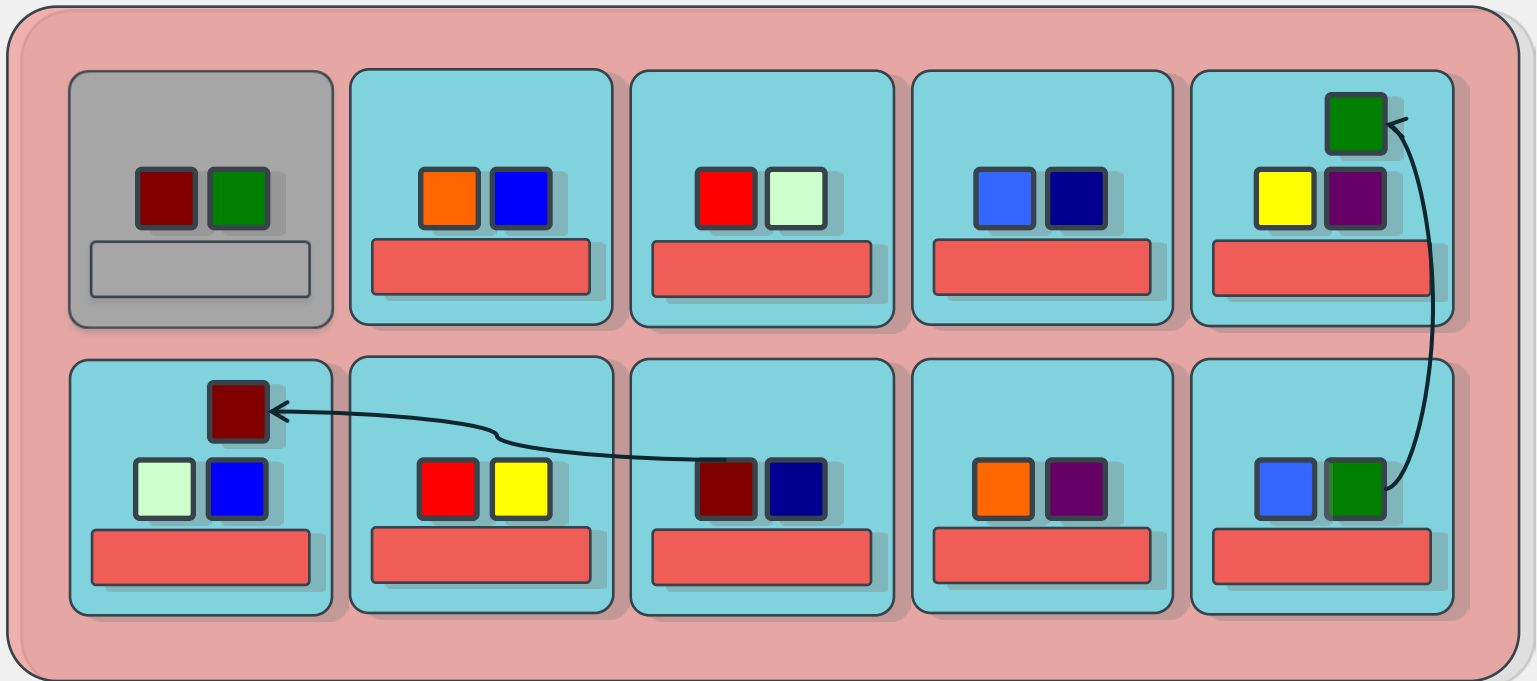
Replication and Acknowledgement



Automatic Failure Detection



Distributed Recovery



Self Managing Storage

- Many common operations require data redistribution
 - adding new storage nodes and volumes
 - retiring old storage nodes and volumes
 - changing replication and placement policies
- All are handled very similarly to the failure case
 - new topology and rules are introduced through a monitor
 - a new cluster map announces the changes
 - OSDs use CRUSH to learn their new responsibilities
 - primary OSDs drive the required data redistribution
- Any component can be replaced at any time
 - no single points of failure
 - multiple failures can be handled (w/sufficient redundancy)
 - this (and protocol interoperability) enable rolling upgrades

APP



APP



HOST/VM



CLIENT



LIBRADOS

A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP

RADOSGW

A bucket-based REST gateway, compatible with S3 and Swift

RBD

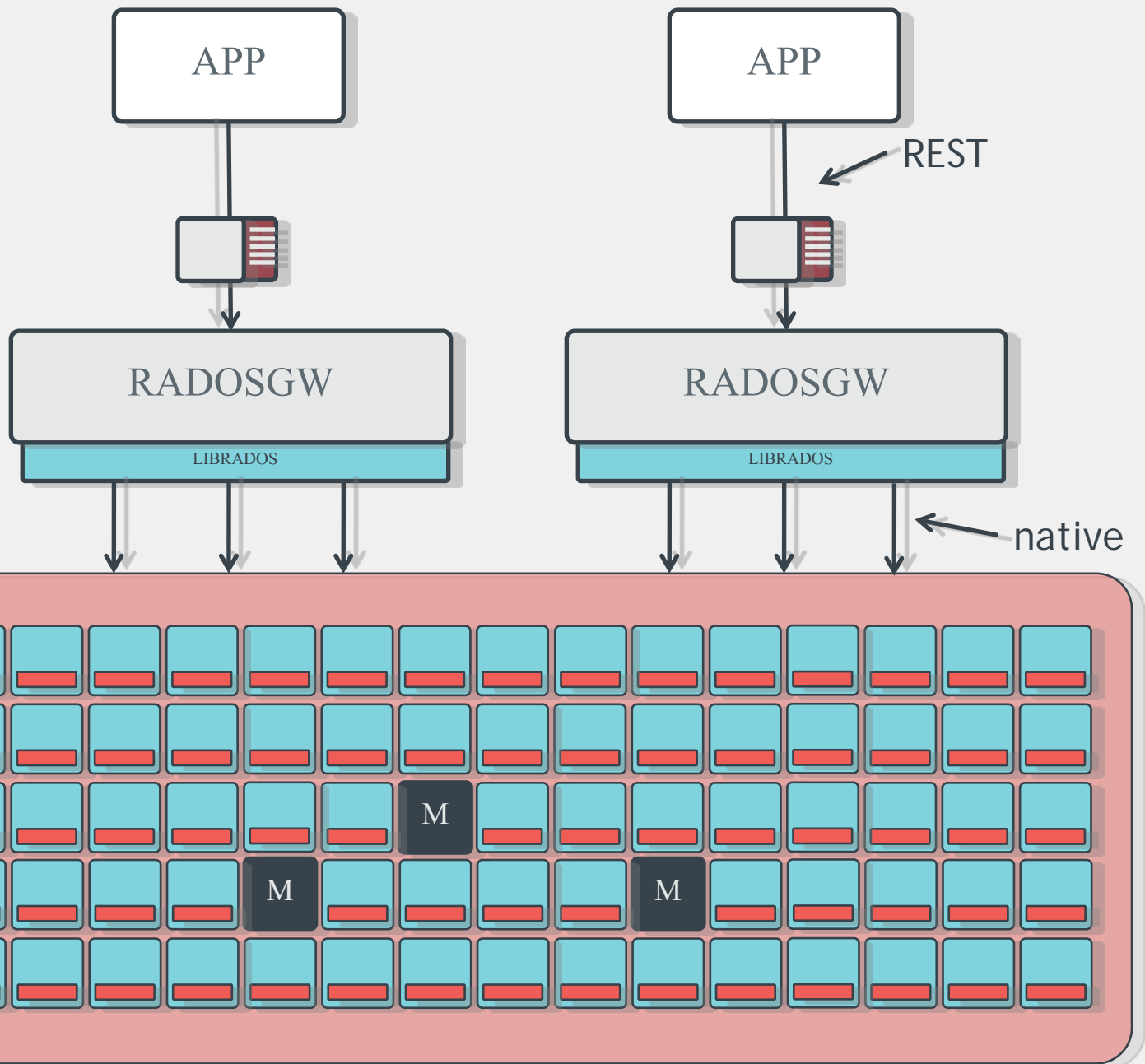
A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver

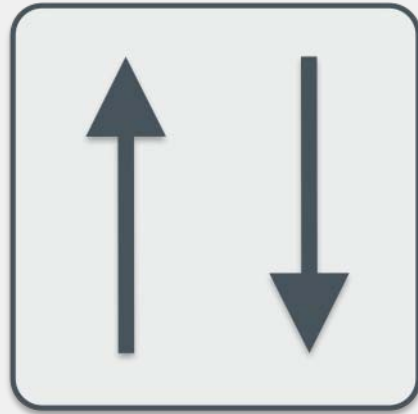
CEPH FS

A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE

RADOS

A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes





RADOS Gateway

REST-based object storage proxy

uses RADOS to store objects

API supports buckets, accounting

usage accounting for billing purposes

compatible with S3, Swift APIs

APP



LIBRADOS

A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP

APP



RADOSGW

A bucket-based REST gateway, compatible with S3 and Swift

HOST/VM



RBD

A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver

CLIENT

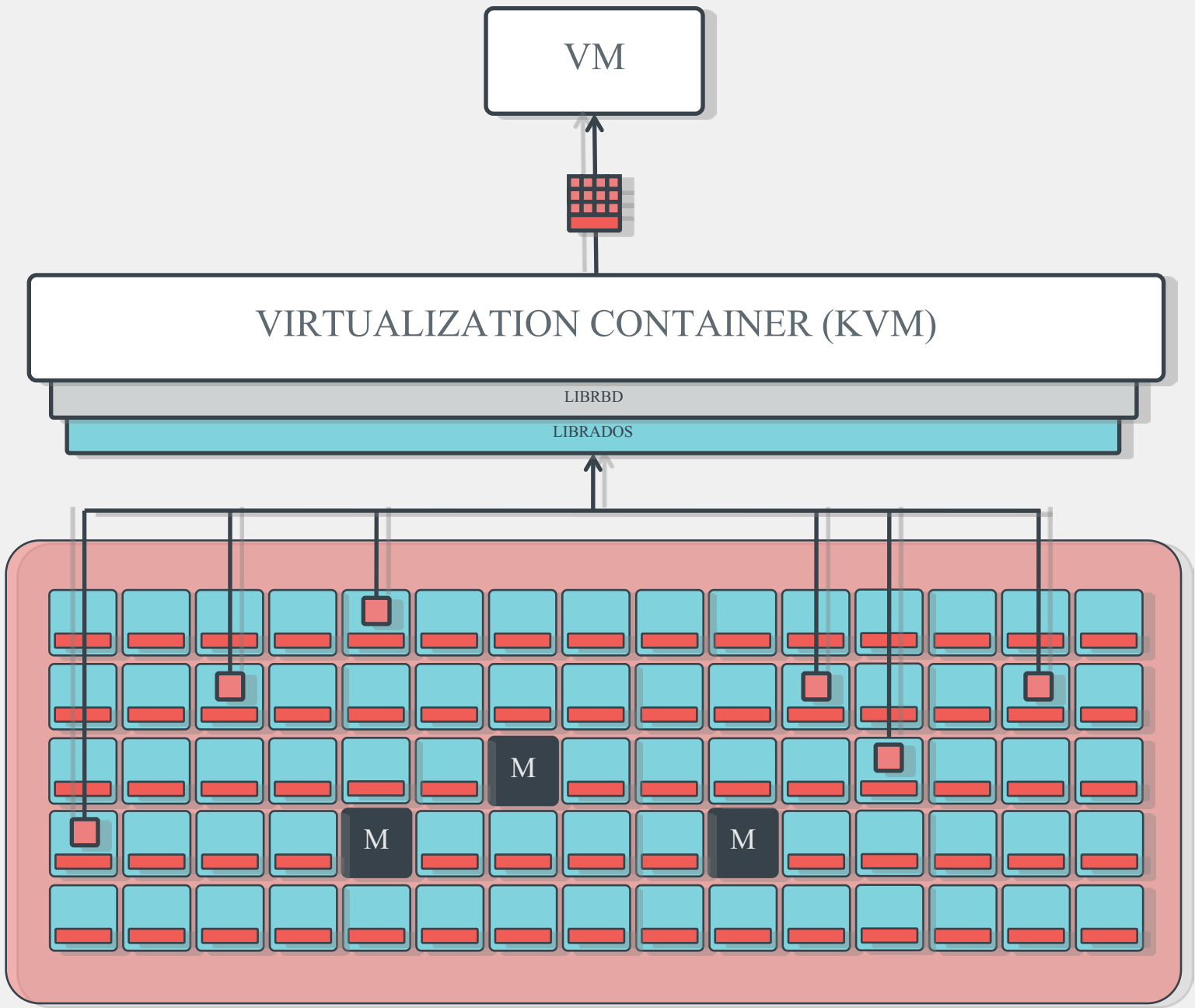


CEPH FS

A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE

RADOS

A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes



VM

VIRTUALIZATION CONTAINER (KVM)

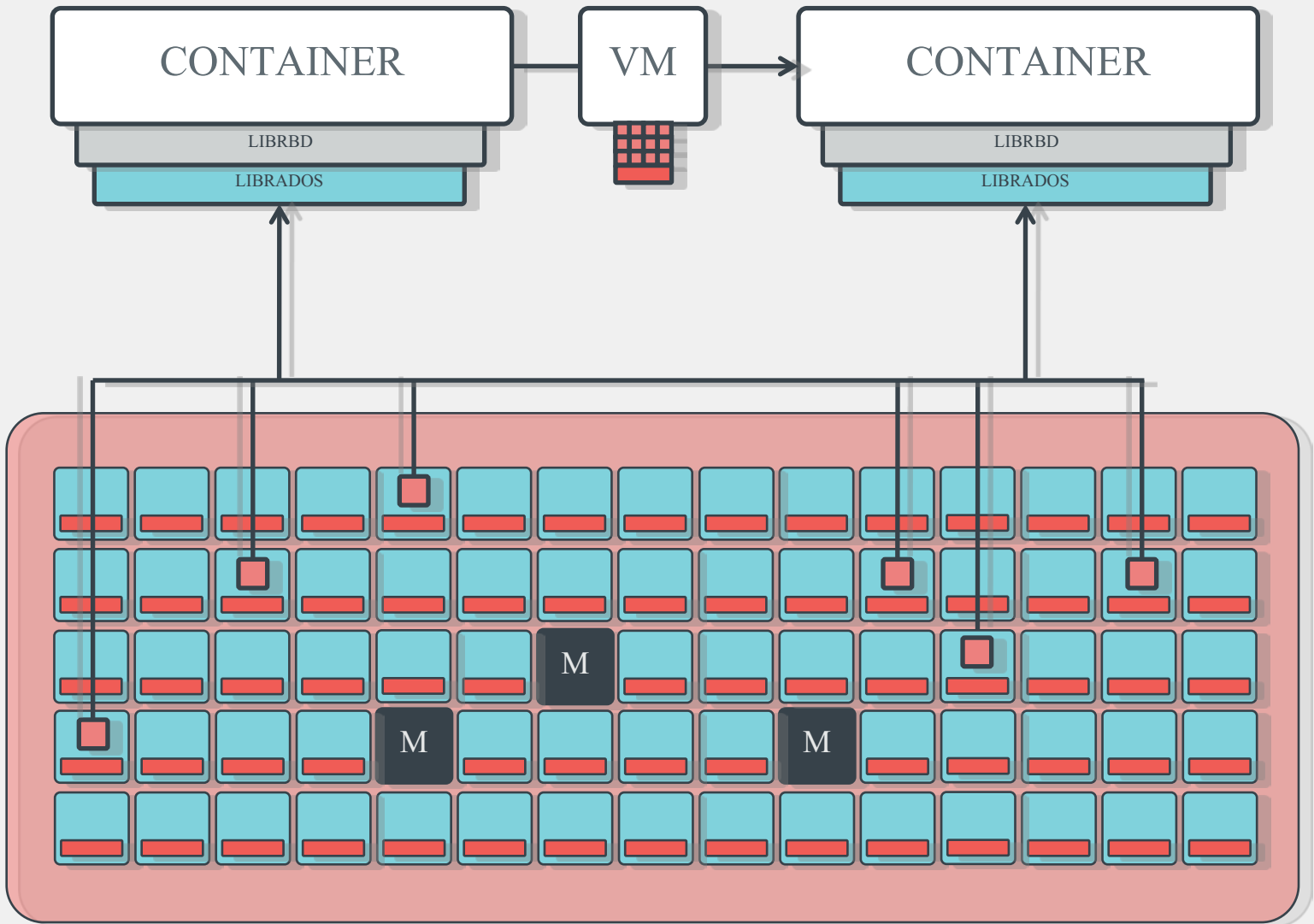
LIBRBD

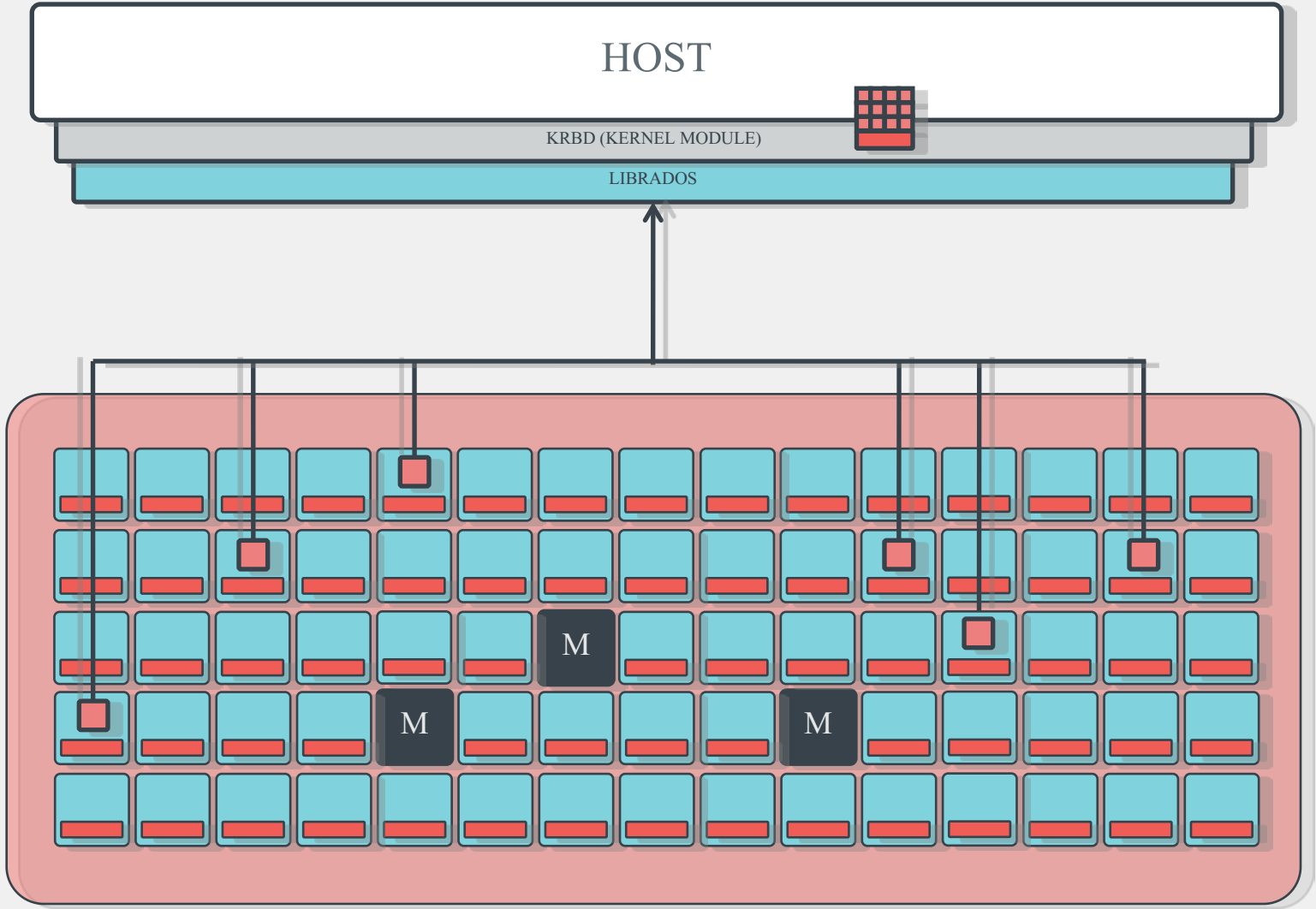
LIBRADOS

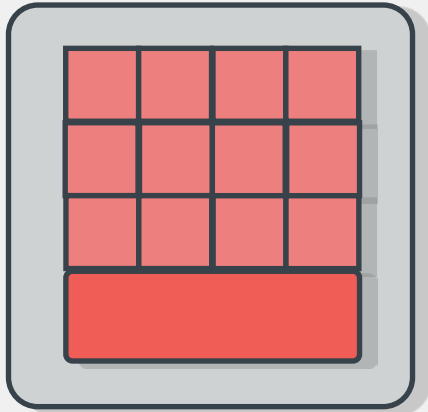
M

M

M







RADOS Block Device

storage of disk images in RADOS

decouple VM from host

images striped across entire
cluster (pool)

snapshots

copy-on-write clones

support in

mainline Linux kernel (2.6.39+)

Qemu/KVM, native Xen coming soon

OpenStack, CloudStack, Nebula, ...

APP



LIBRADOS
A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP

APP



RADOSGW
A bucket-based REST gateway, compatible with S3 and Swift

HOST/VM



RBD
A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver

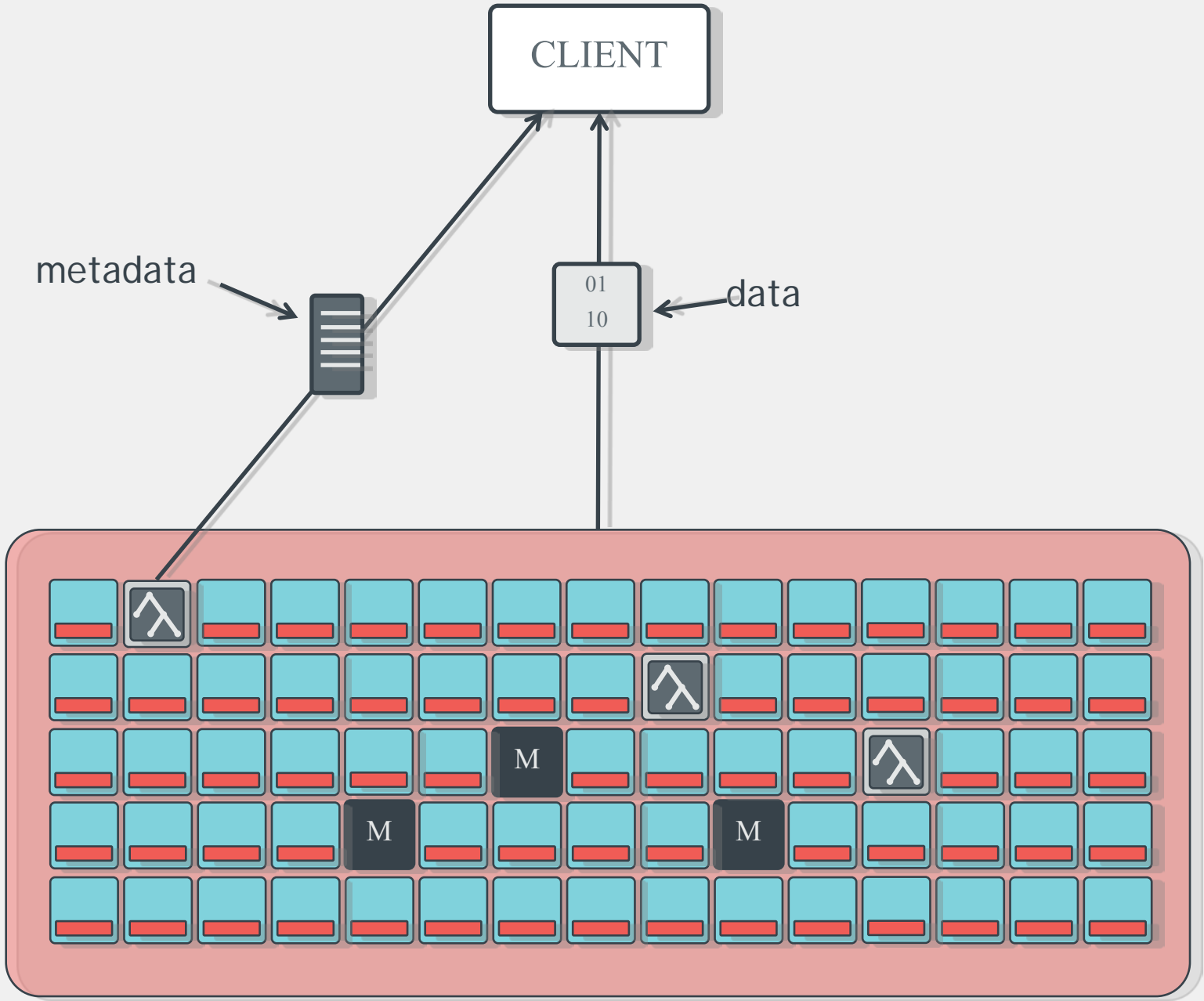
CLIENT

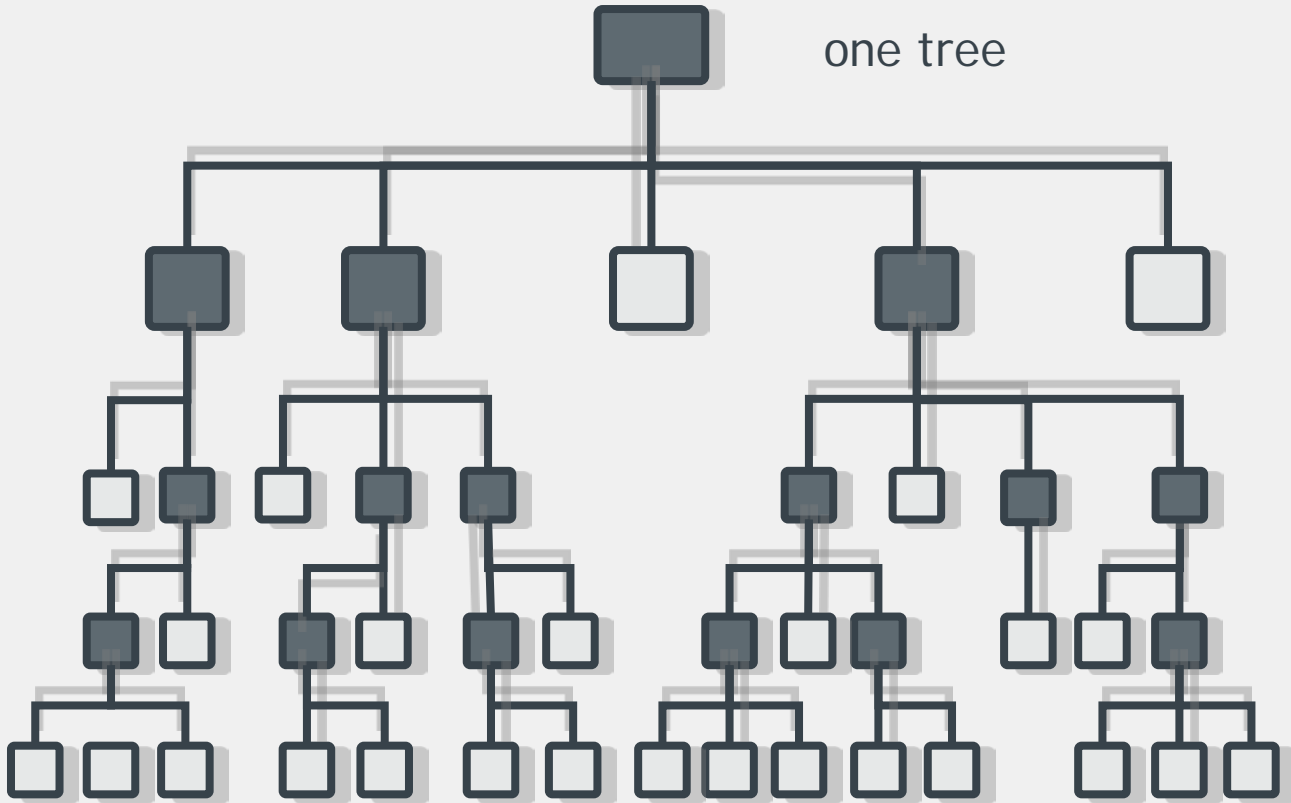


CEPH FS
A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE

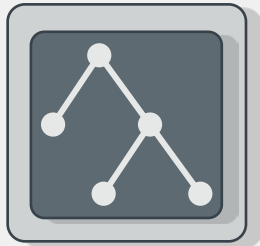
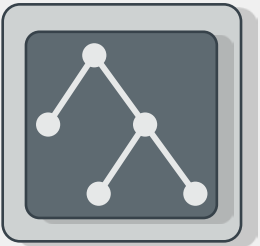
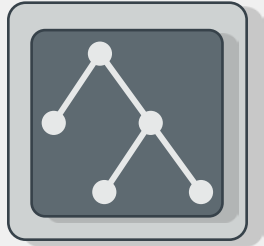
RADOS

A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

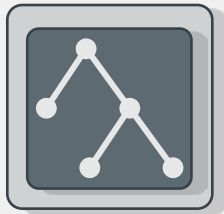
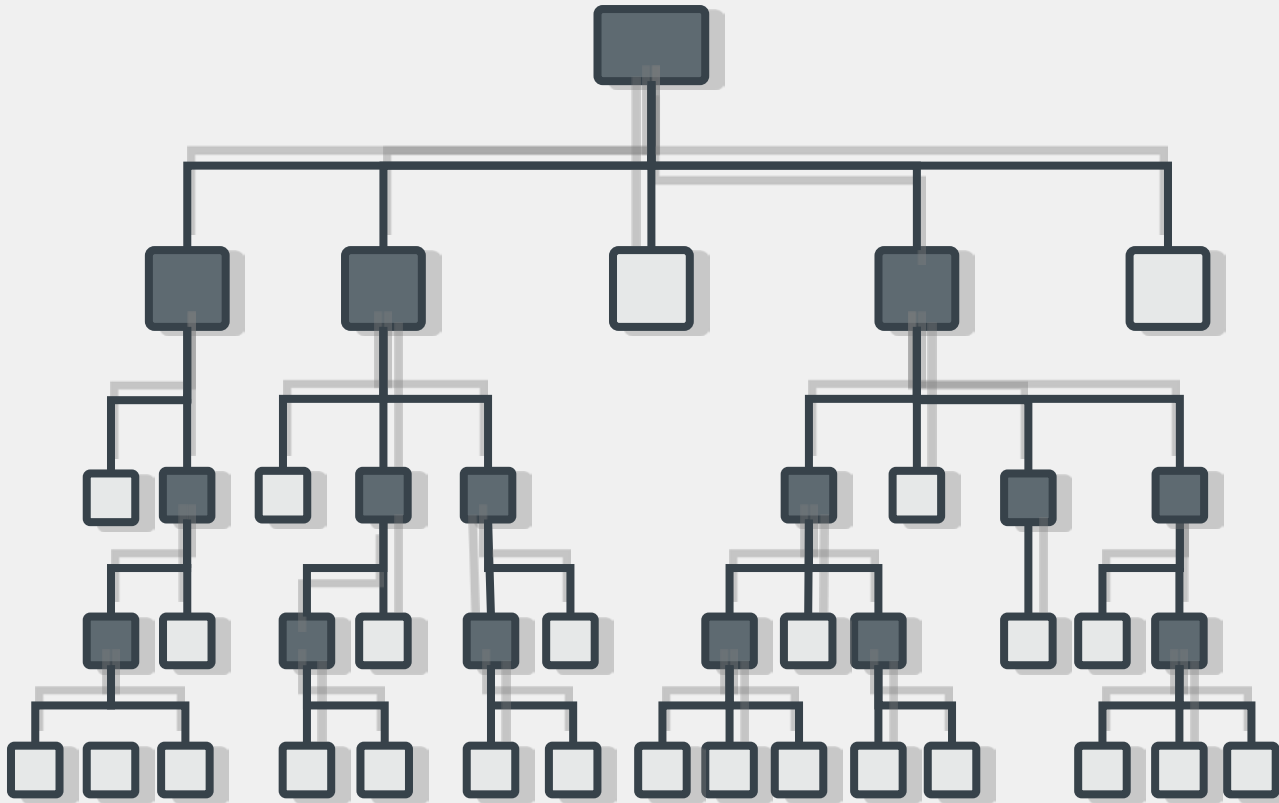


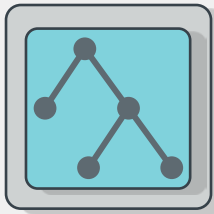
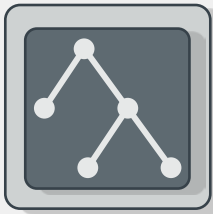
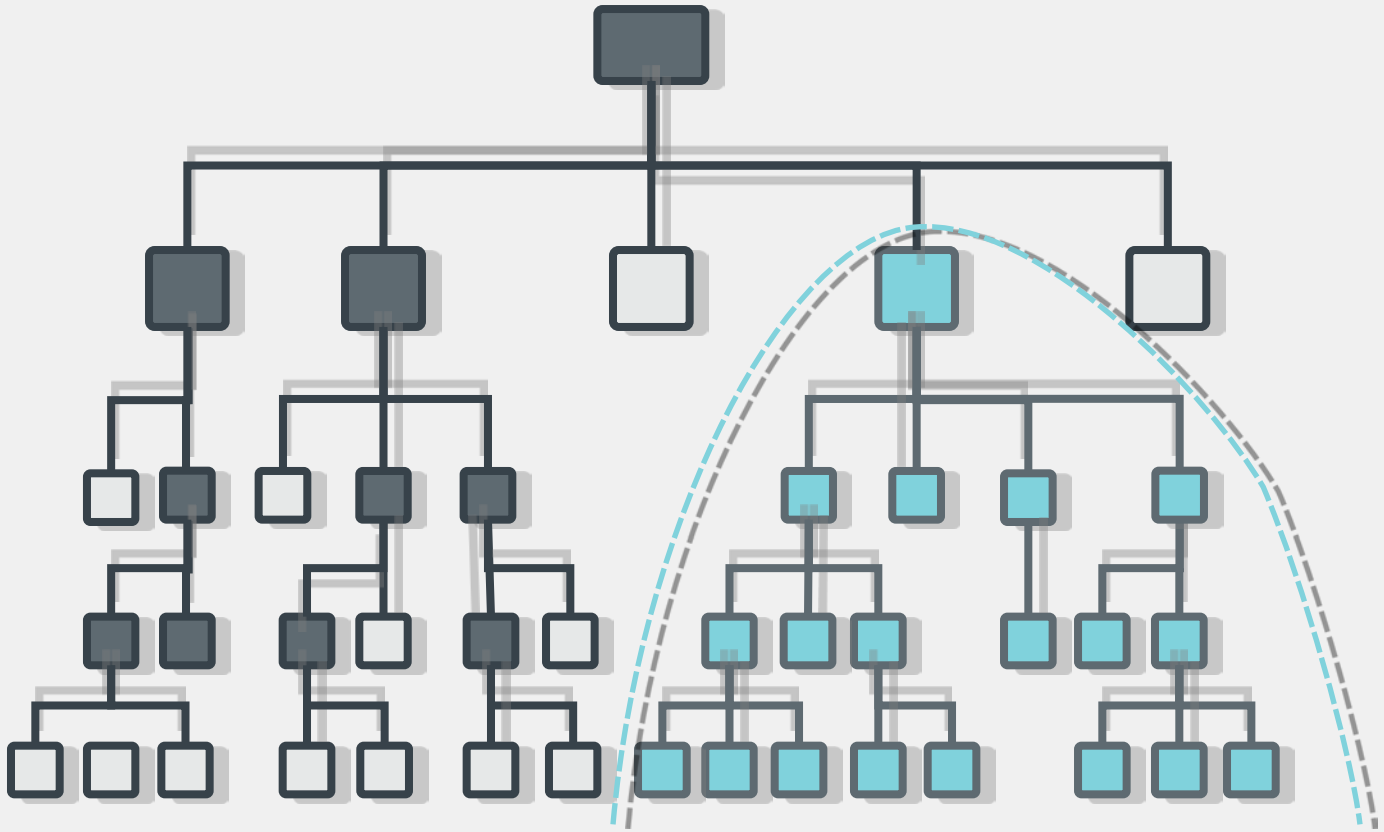


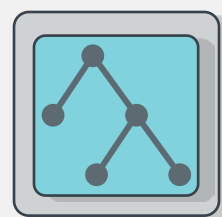
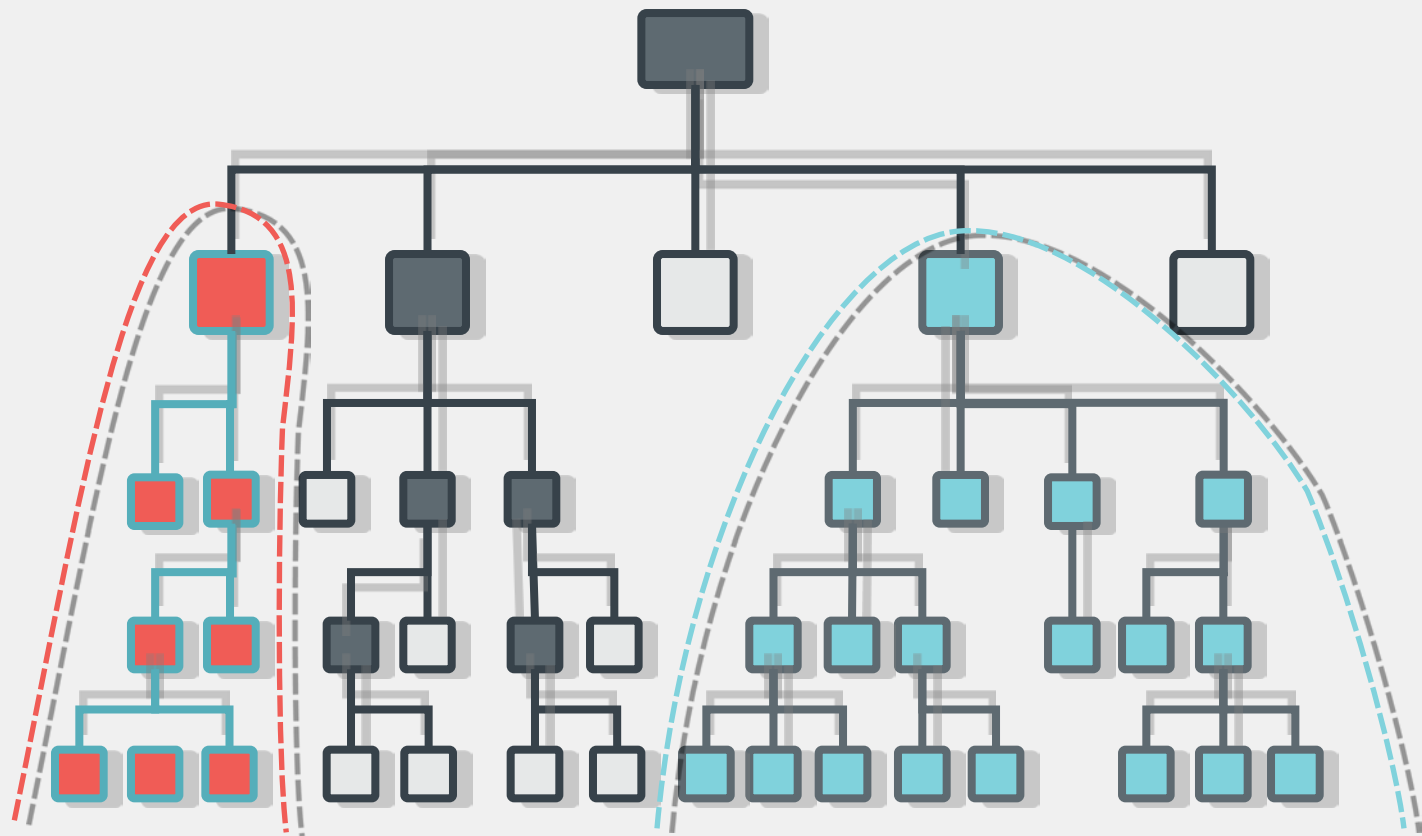
three metadata servers

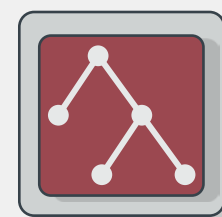
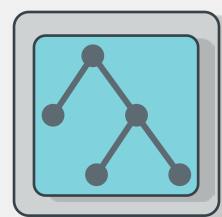
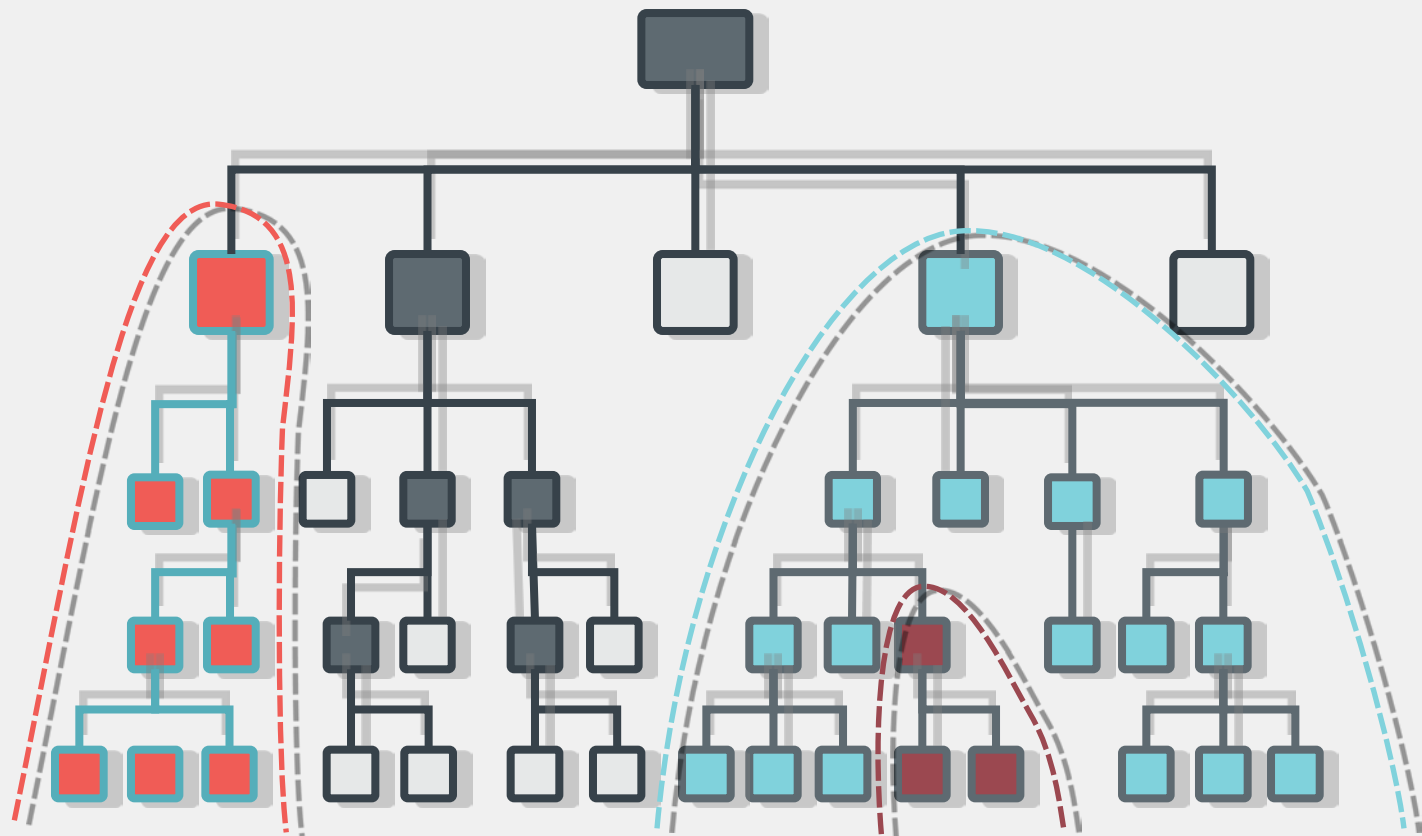


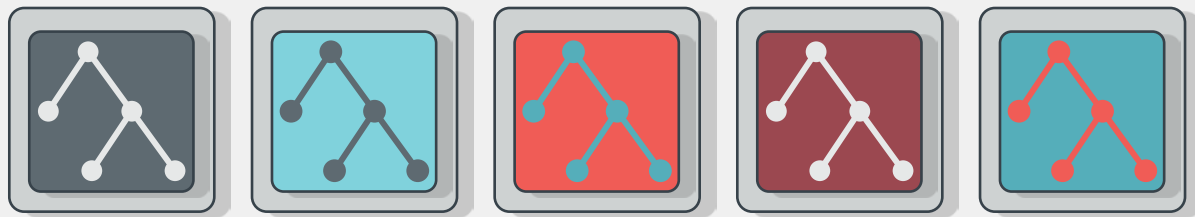
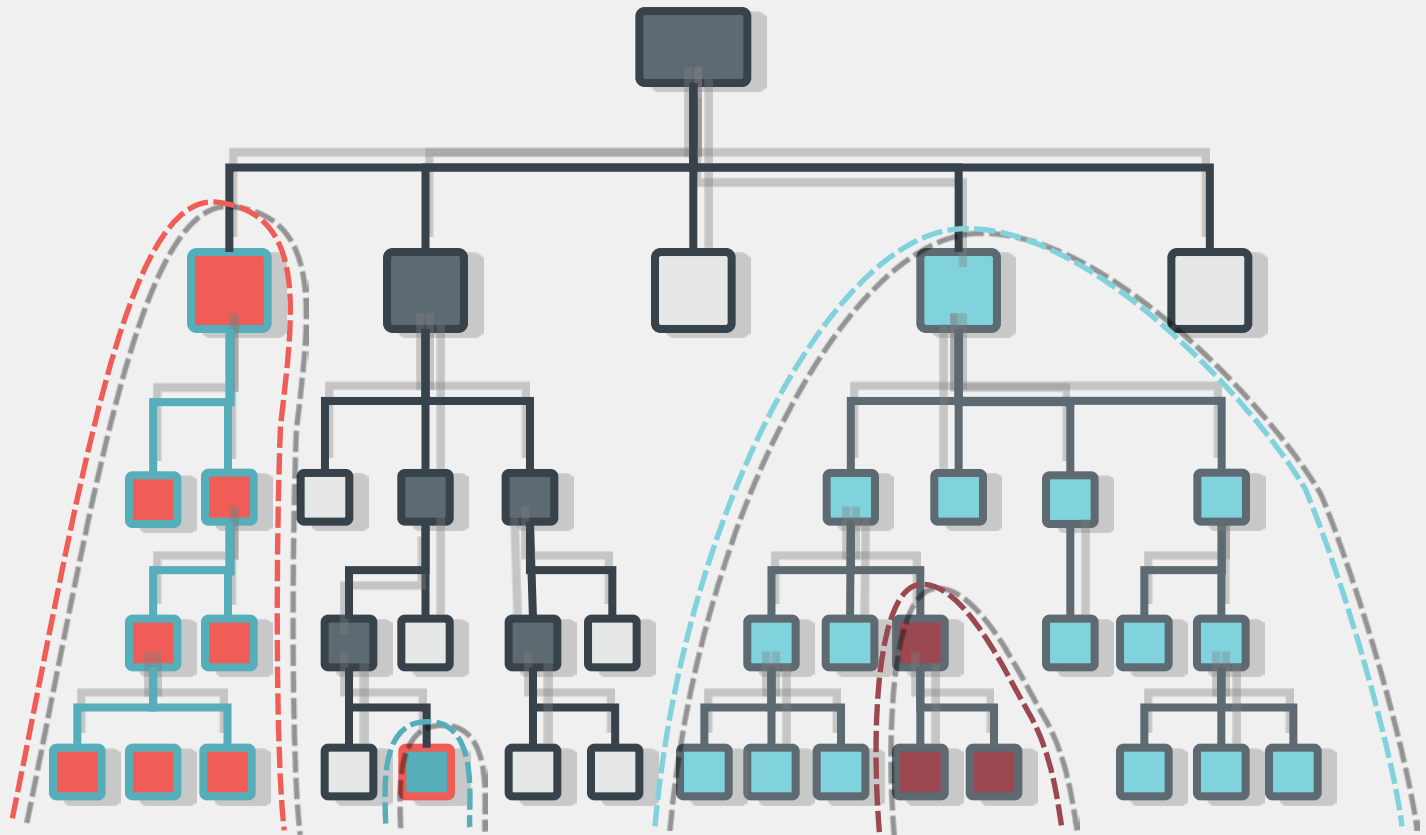
??











DYNAMIC SUBTREE PARTITIONING

Ceph: 21st Century Technology

- Performance
 - direct, striped, parallel I/O
 - well distributed over a large cluster
- Reliability and Availability
 - configurable replication and persistence policies
 - automatic failure-domain aware placement
 - no single points of failure
 - prompt, fully automatic recovery from common failures
- Scalable
 - no architectural bottle-necks
 - maximum independence and parallelism
 - efficient use of all available storage/processing
 - self-healing, self-balancing, self-managing



Hands-on tutorial



Hands-on Tutorial Prep

Download VM image

<http://ceph.com/tutorial>

tutorial.img.tar.gz (KVM/Qemu)

tutorial.vdi.gz (Virtualbox, ...)

2GB RAM

Attach 4 additional disks (~8GB each)



Q&A



Thanks!

sage weil

sage@inktank.com

@liewegas

<http://github.com/ceph>

<http://ceph.com/>

inktank