redhat.

INTRO TO DISTRIBUTED STORAGE WITH CEPH AND GLUSTER

RHUG Montreal

Dustin L. Black, RHCA
Principal Cloud Success Architect
2015-10-29

DUSTIN L. BLACK

dustin@redhat.com
@dustinlblack
linkedin.com/in/dustinblack
people.redhat.com/dblack

RED HAT CERTIFIED ARCHITECT

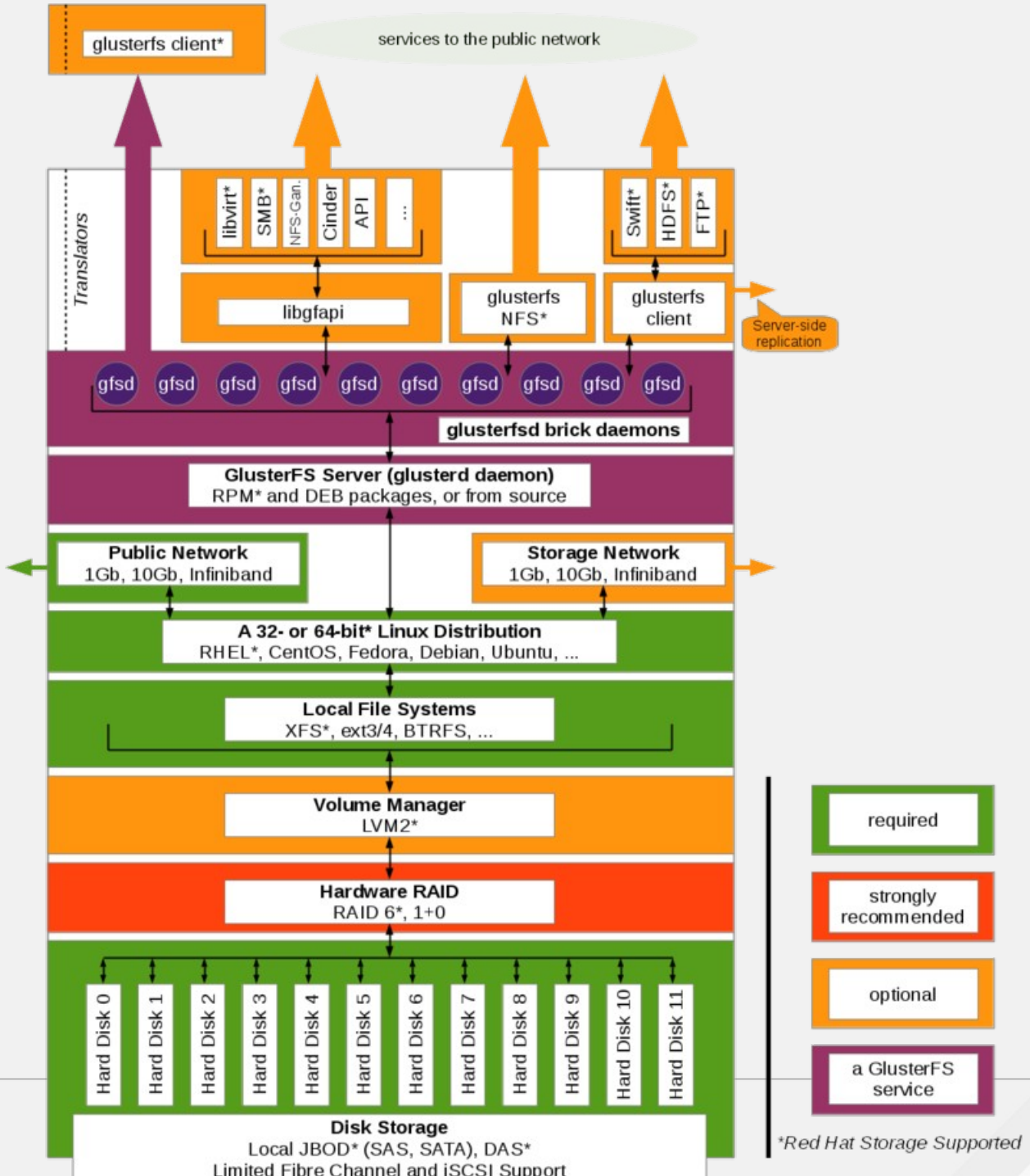
redhat

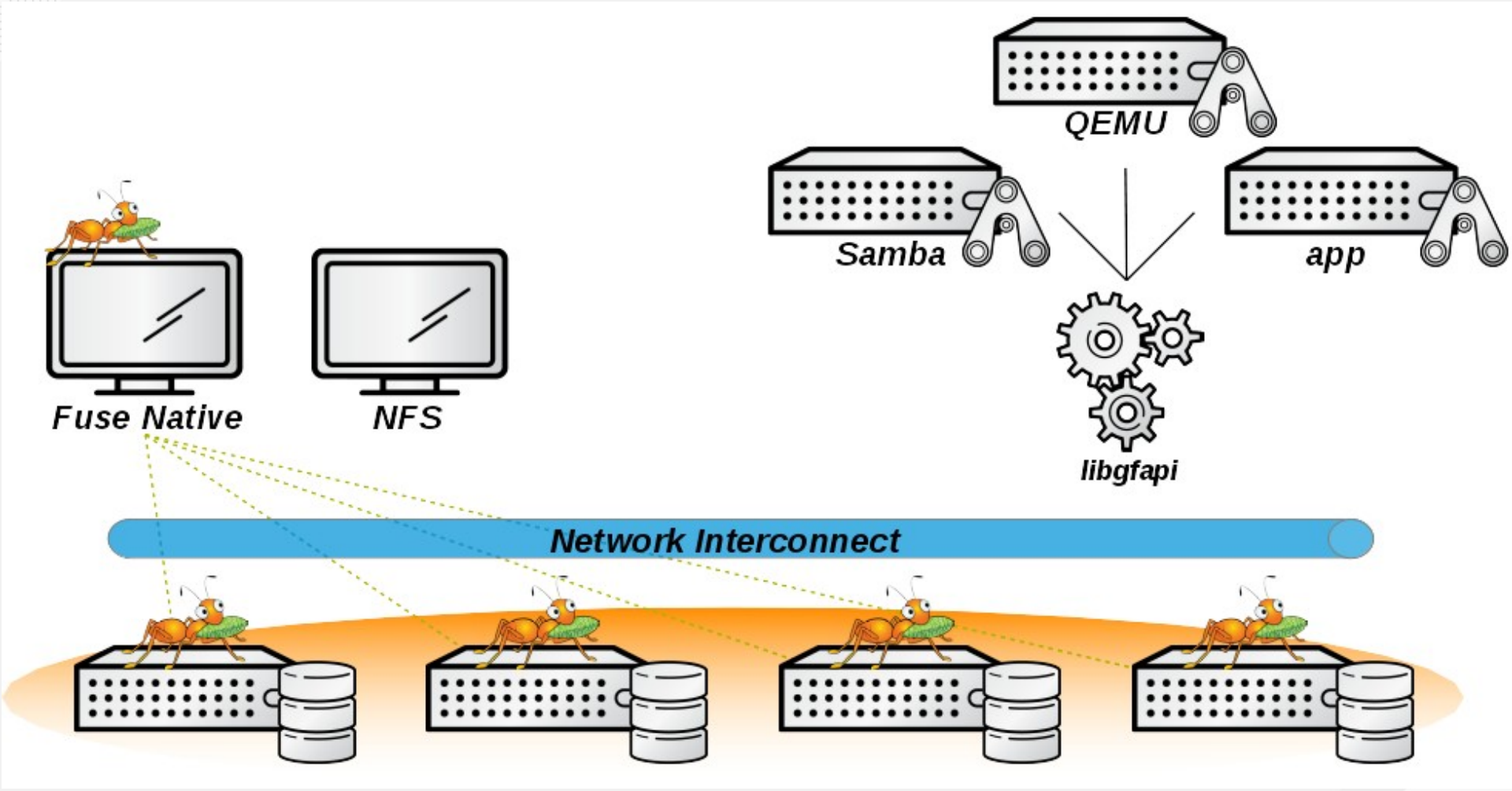
LET'S TALK DISTRIBUTED STORAGE

- Decentralize and Limit Failure Points
- Scale with Commodity Hardware and Familiar Operating Environments
- Reduce Dependence on Specialized Technologies and Skills

GLUSTER

- Clustered Scale-out General Purpose Storage Platform
- Fundamentally File-Based & POSIX End-to-End
- Familiar Filesystems Underneath (EXT4, XFS, BTRFS)
- Familiar Client Access (NFS, Samba, Fuse)
- No Metadata Server
- Standards-Based – Clients, Applications, Networks
- Modular Architecture for Scale and Functionality

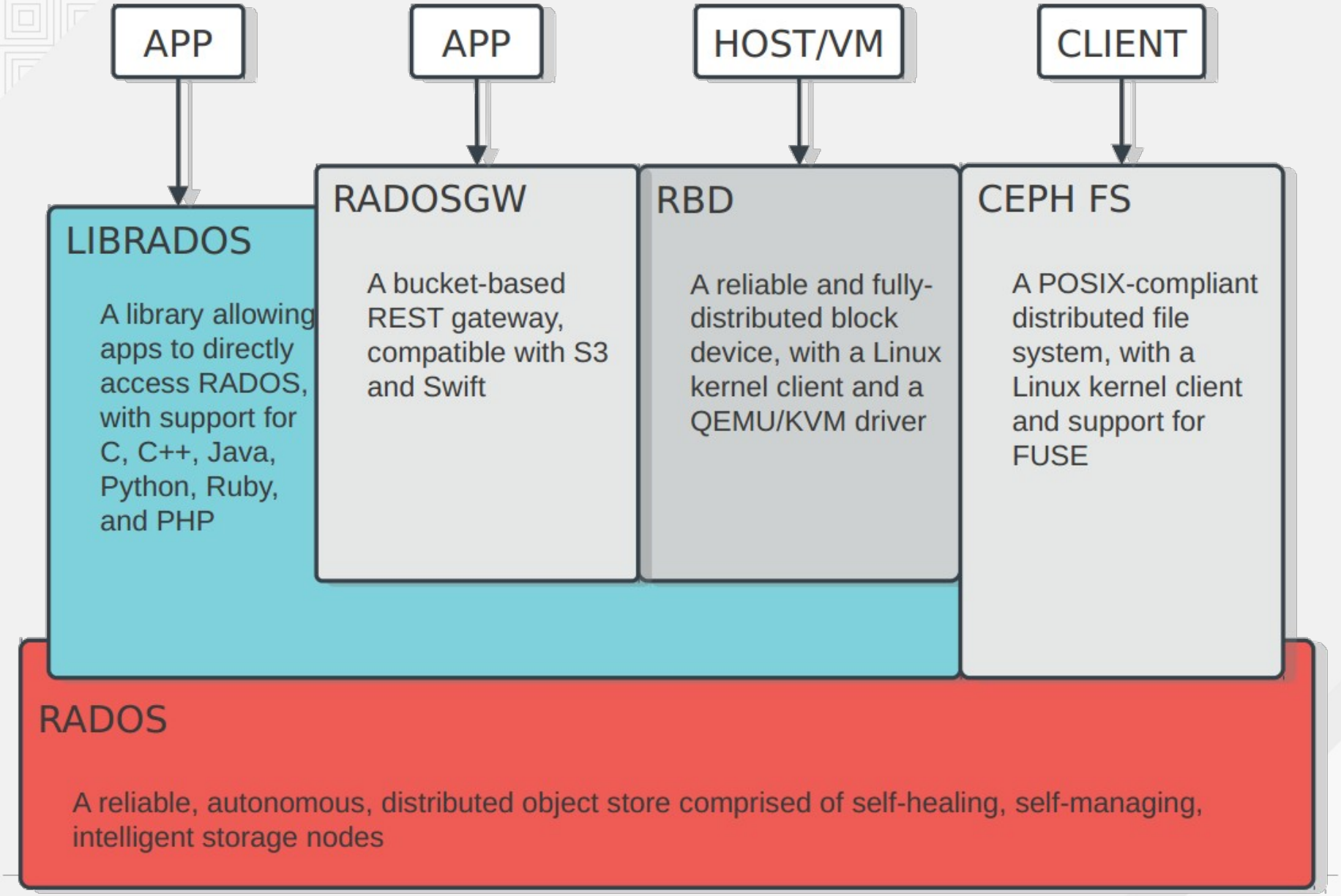

glusterfs client*
services to the public network
Translators
libvirt*
SMB*
NFS-Gan.
Cinder
API
...
Swift*
HDFS*
FTP*
libgfapi
glusterfs NFS*
glusterfs client
Server-side replication
gfsd
glusterfsd brick daemons
GlusterFS Server (glusterd daemon)
RPM* and DEB packages, or from source
Public Network
1Gb, 10Gb, Infiniband
Storage Network
1Gb, 10Gb, Infiniband
A 32- or 64-bit* Linux Distribution
RHEL*, CentOS, Fedora, Debian, Ubuntu, ...
Local File Systems
XFS*, ext3/4, BTRFS, ...
Volume Manager
LVM2*
Hardware RAID
RAID 6*, 1+0
Hard Disk 0
Hard Disk 1
Hard Disk 2
Hard Disk 3
Hard Disk 4
Hard Disk 5
Hard Disk 6
Hard Disk 7
Hard Disk 8
Hard Disk 9
Hard Disk 10
Hard Disk 11
Disk Storage
Local JBOD* (SAS, SATA), DAS*
Limited Fibre Channel and iSCSI Support
required
strongly recommended
optional
a GlusterFS service
*Red Hat Storage Supported

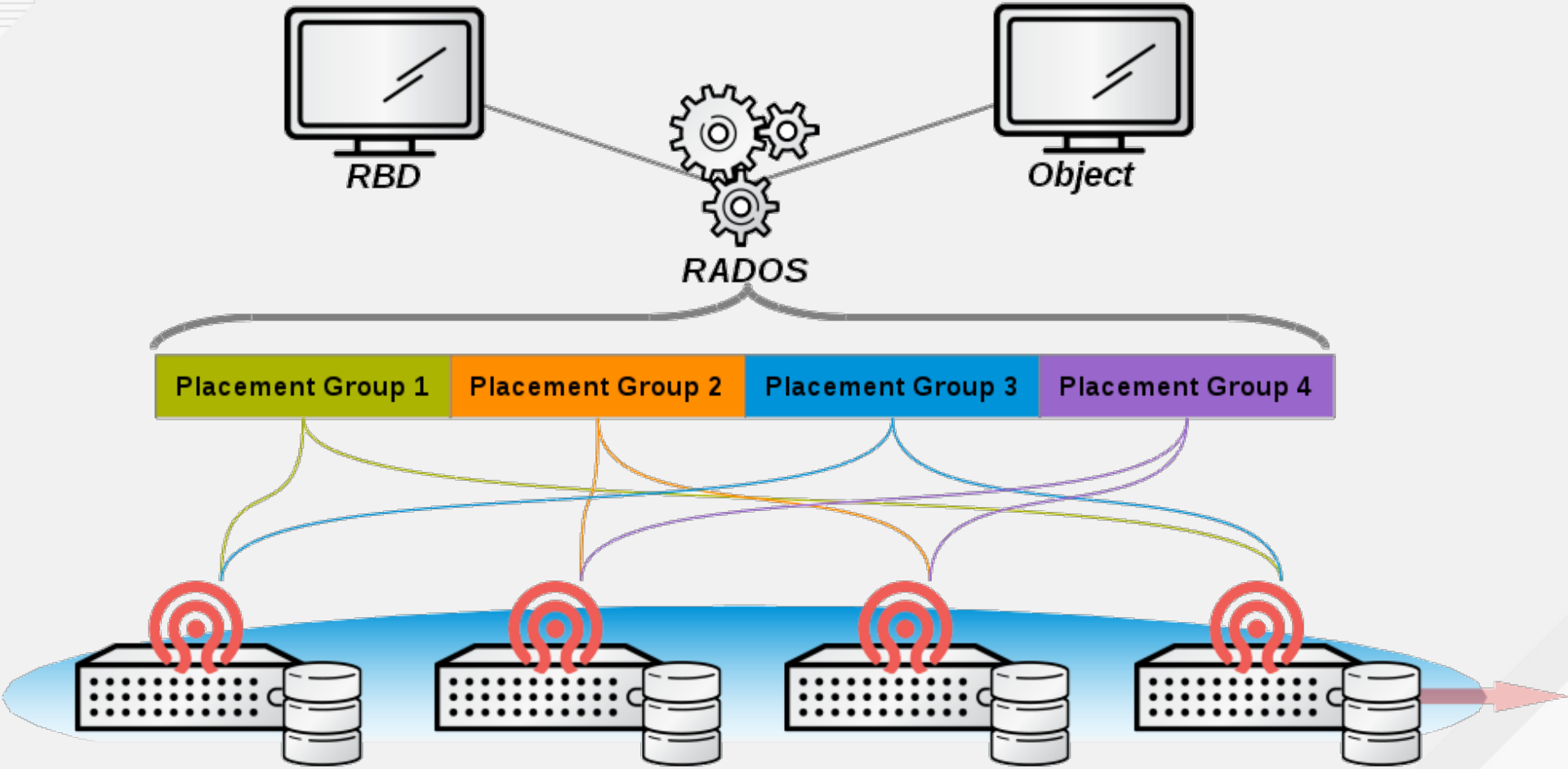

QEMU
Samba
app
Fuse Native
NFS
libgfapi
Network Interconnect

MORE ON GLUSTER LATER...

CEPH

- Massively scalable, software-defined storage system
- Commodity hardware with no single point of failure
- Self-healing and Self-managing
- Rack and data center aware
- Automatic distribution of replicas
- Block, Object, File
- Data stored on common backend filesystems (EXT4, XFS, etc.)
- Fundamentally distributed as objects via RADOS
- Client access via RBD, RADOS Gateway, and Ceph Filesystem


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


RBD
Object
RADOS
Placement Group 1
Placement Group 2
Placement Group 3
Placement Group 4


CEPH IN MORE DETAIL

CEPH BASICS

Commodity Server Hardware

Standard X86-64 servers are typically used

Two major “node” types

- Monitor nodes (maintain and provide cluster map)
- OSD nodes (nodes that provide storage OSD's)

Graphical management

- Calamari host

CEPH BASICS

Monitor nodes

Monitors should have of an odd number of servers for quorum

Either 3 or 5 monitors are typically used

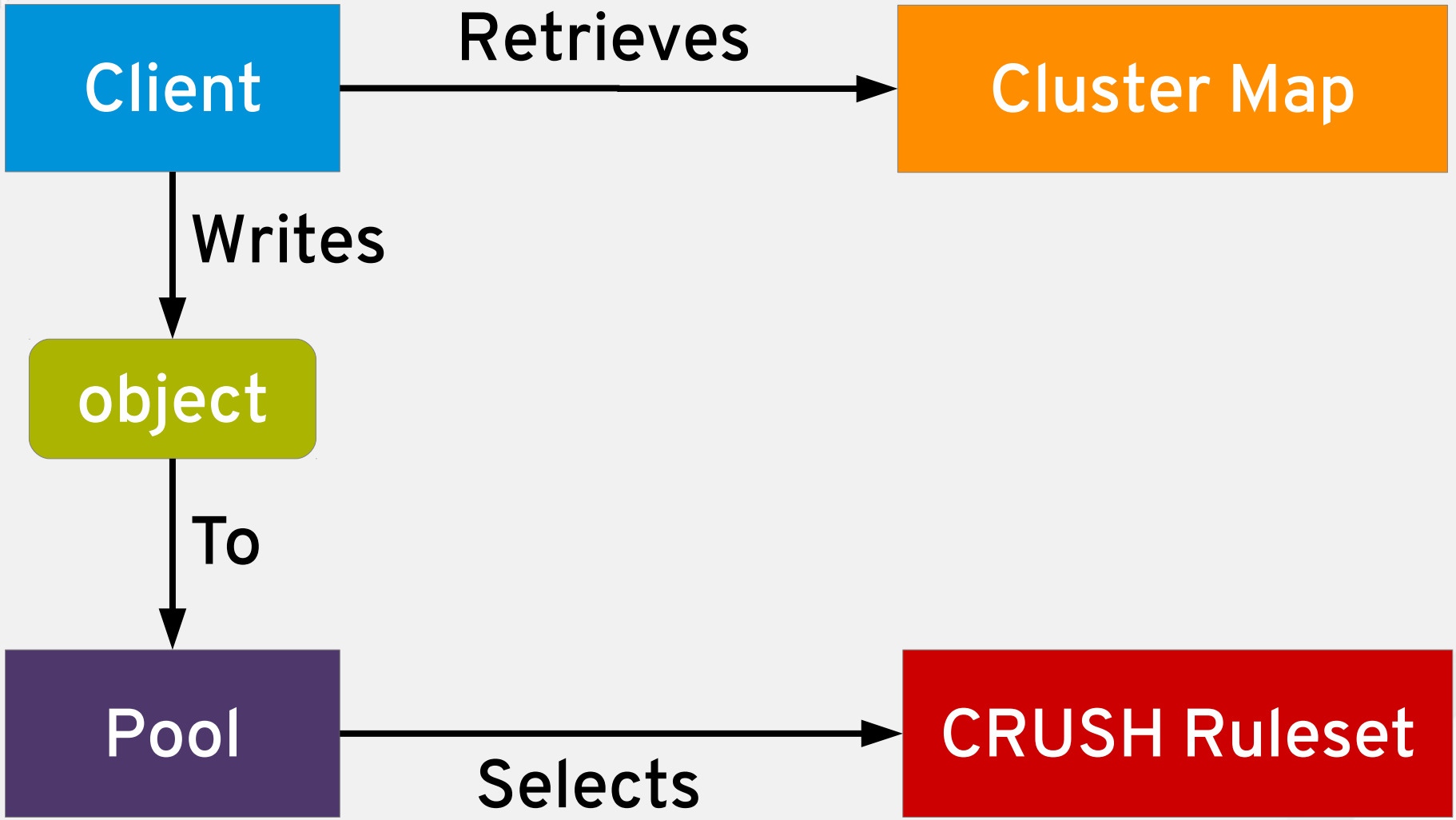
Paxos protocol

Monitor nodes always work by consensus.

They will need to agree with each other, using Paxos.

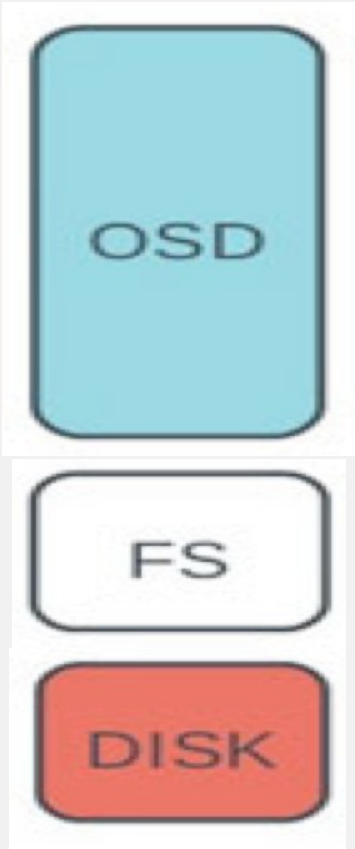
Paxos is a family of protocols for solving consensus in a network of unreliable processors. Consensus is the process of agreeing on one result among a group of participants (source: Wikipedia)

CEPH BASICS


Client
Retrieves
Cluster Map
Writes
object
To
Pool
Selects
CRUSH Ruleset

CEPH OSD

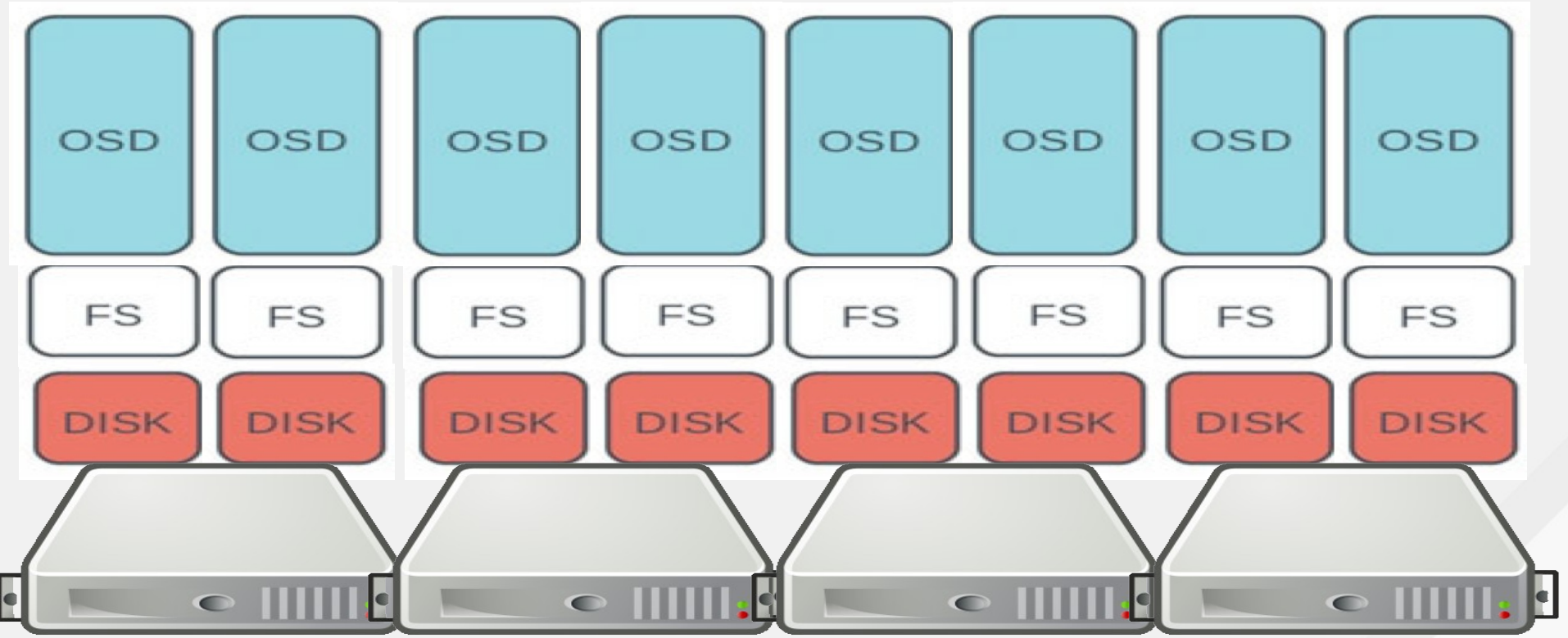
Object Storage Daemon



CEPH OSD

RADOS

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



CEPH RADOS

Reliable Autonomic Distributed Object Store

RADOS

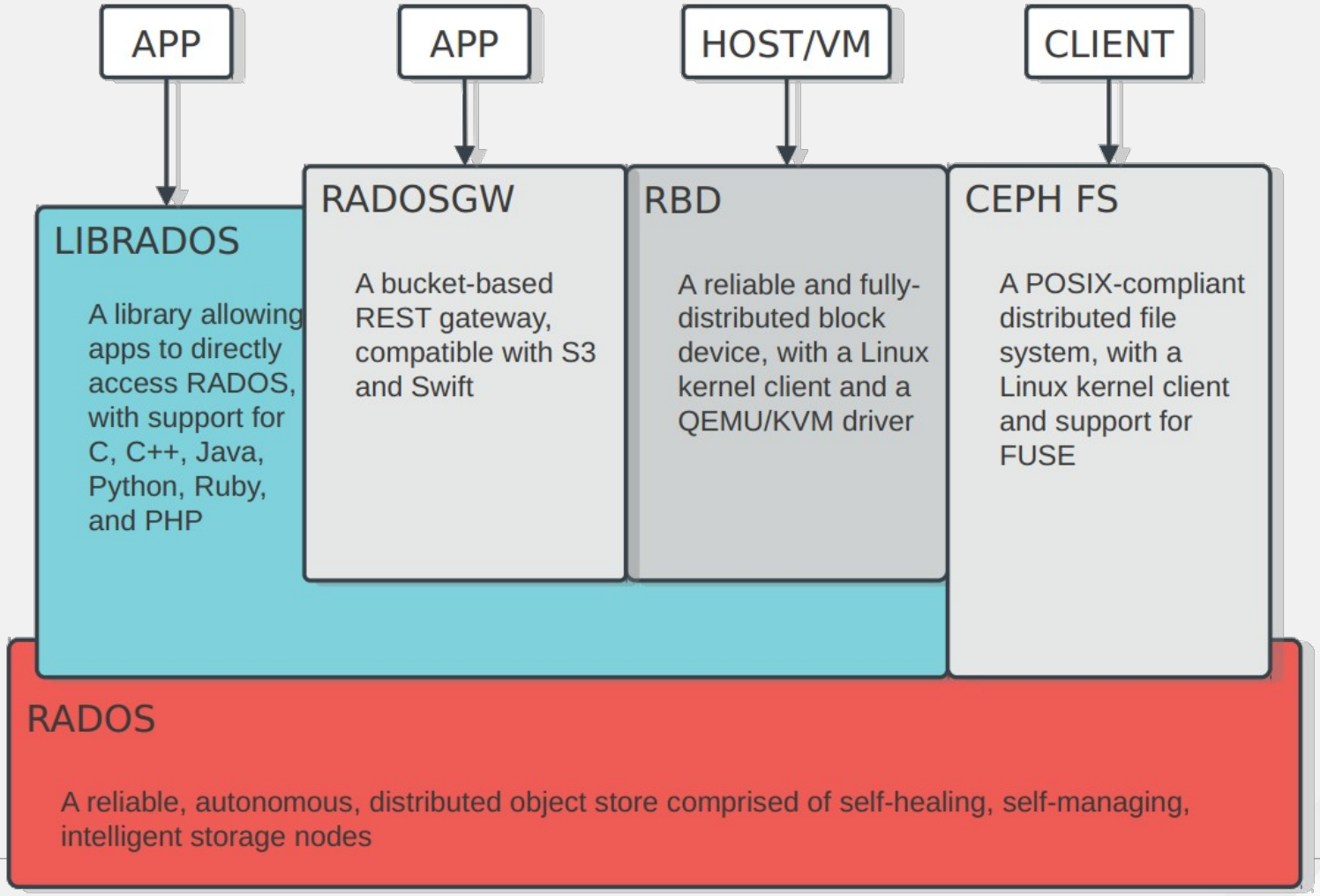
A reliable, autonomic, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

RADOS is an object storage service, able to scale to thousands of hardware devices by running Ceph software on each individual node.

RADOS is an integral part of the Ceph distributed storage system.

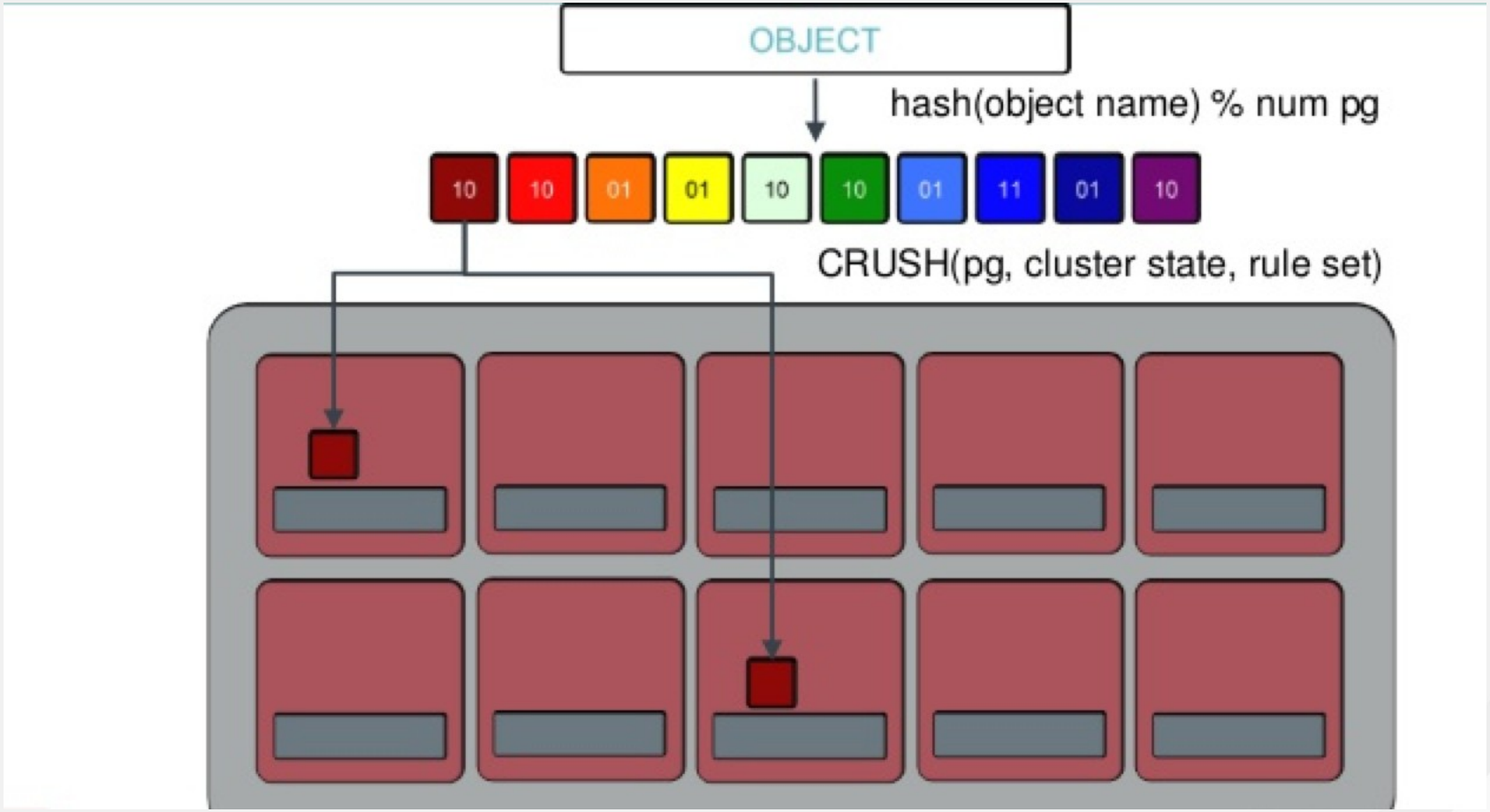
CEPH LIBRADOS

RADOS Library

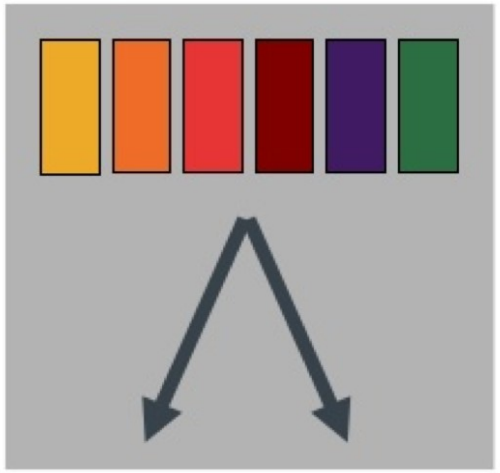


CEPH CRUSH

Controlled Replication Under Scalable Hashing



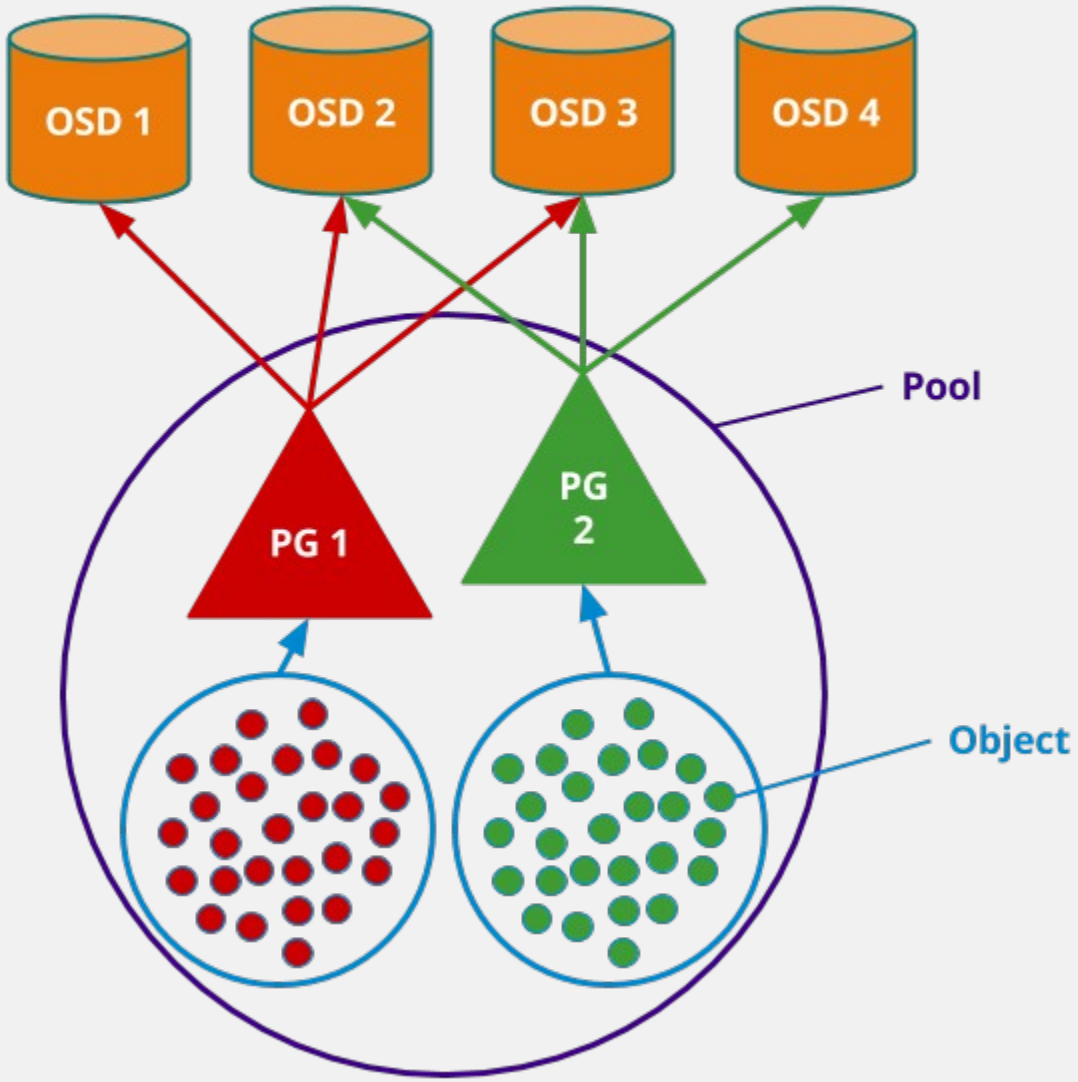
CEPH CRUSH



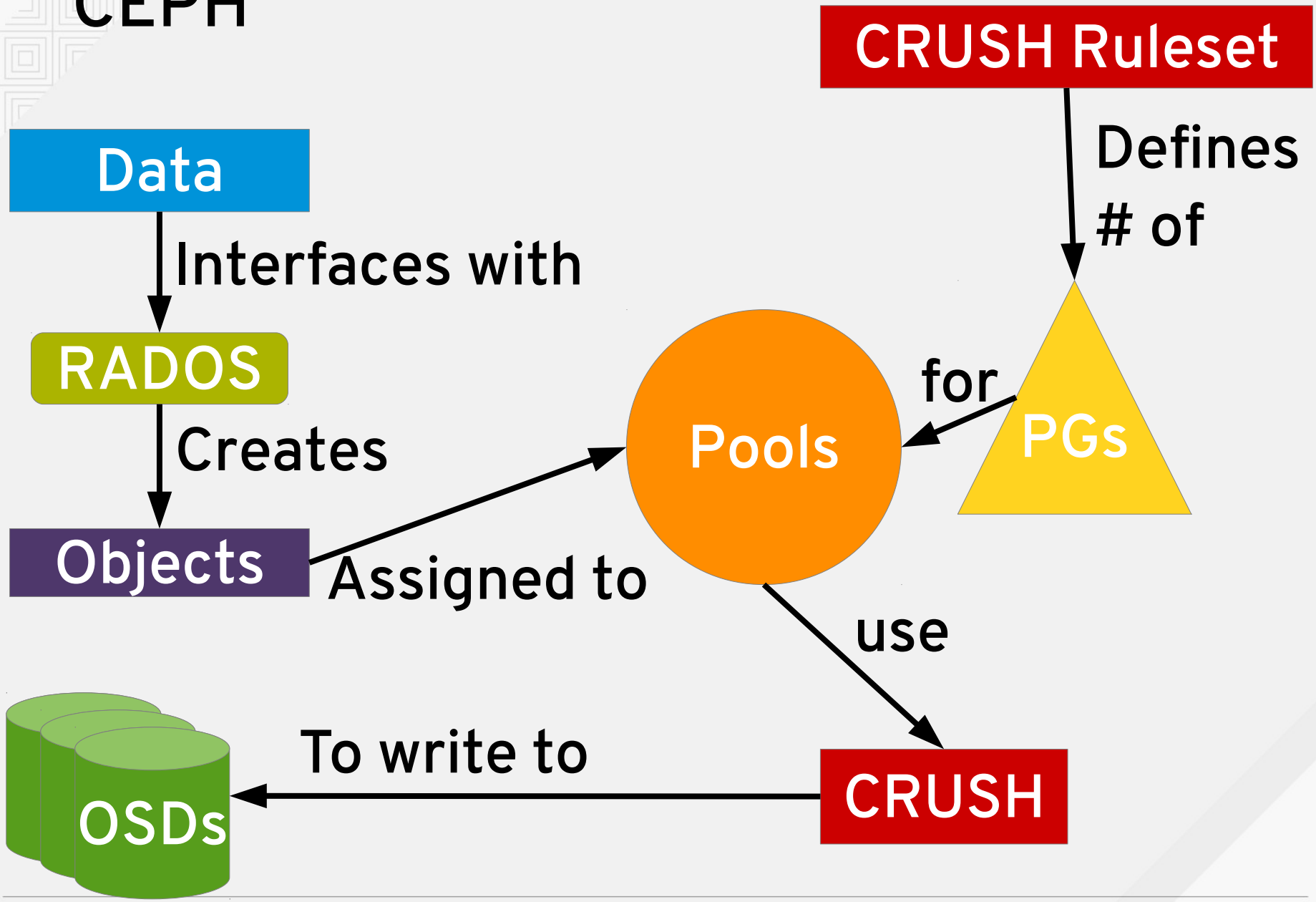
CRUSH:

- Pseudo-random placement algorithm
  - Fast calculation, **no lookup**
  - Repeatable, deterministic
- Statistically uniform distribution
- **Stable** mapping
  - Limited data migration on change
- Rule-based configuration
  - Infrastructure **topology aware**
  - Adjustable replication
  - Weighted devices (different sizes)

CEPH POOLS & PLACEMENT GROUPS



CEPH



**Disclaimer: Beta slide – Accuracy not guaranteed*

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos