Cloud object storage in Ceph

Orit Wasserman
owasserm@redhat.com
Fosdem 2017

AGENDA

- What is cloud object storage?
- Ceph overview
- Rados Gateway architecture
- Questions

Cloud object storage

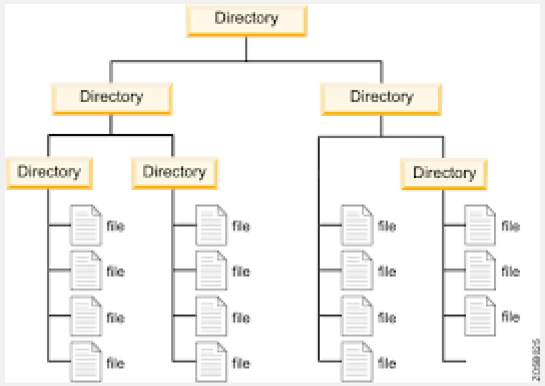
Block storage

- Data stored in fixed blocks
- No metadata
- Fast
- Protocols:
  - SCSI
  - FC
  - SATA
  - ISCSI
  - FCoE

File system

- Users
- Authentication
- Metadata:
  - ownership
  - Permissions/ACL
  - Creation/Modification time
- Hierarchy: Directories and files
- Files are mutable
- Sharing semantics
- Slower
- Complicate

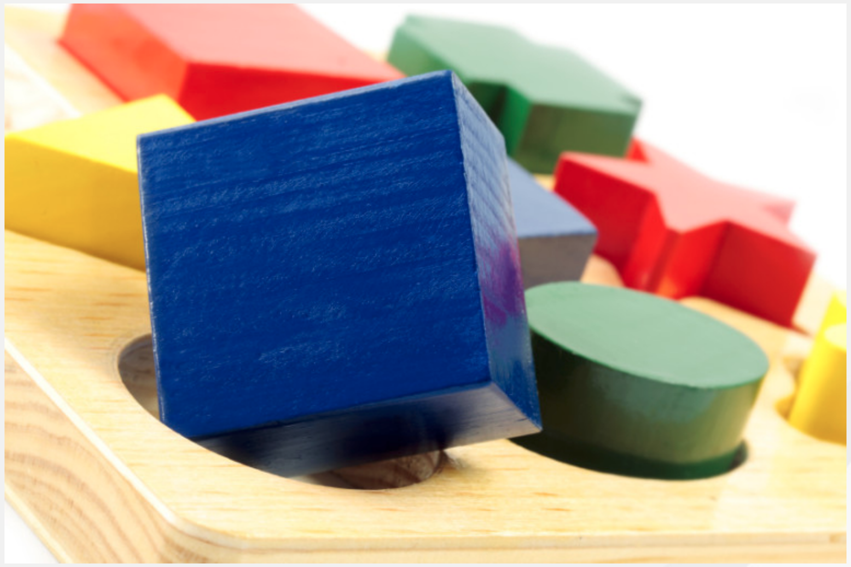
- Protocols:
  - Local: ext4,xfs, btrfs, zfs, NTFS, …
  - Network: NFS, SMB, AFP



Object storage

- Restful API (cloud)
- Flat namespace:
  - Bucket/container
  - Objects
- Users and tenants
- Authentication
- Metadata:
  - Ownership
  - ACL
  - User metadata
- Large objects
- Objects are immutable

- Cloud Protocols:
  - S3
  - Swift (openstack)
  - Google Cloud storage



S3 examples

Create bucket

```
PUT /{bucket} HTTP/1.1
Host: cname.domain.com
x-amz-acl: public-read-write
Authorization: AWS {access-key}:{hash-of-header-and-secret}
```

Get bucket

```
GET /{bucket}?max-keys=25 HTTP/1.1
Host: cname.domain.com
```

S3 examples

Delete bucket

```
DELETE /{bucket} HTTP/1.1
Host: cname.domain.com

Authorization: AWS {access-key}:{hash-of-header-and-secret}
```

S3 examples

Create object

```
PUT /{bucket}/{object} HTTP/1.1
```

Copy object

```
PUT /{dest-bucket}/{dest-object} HTTP/1.1
x-amz-copy-source: {source-bucket}/{source-object}
```

S3 examples

Read object

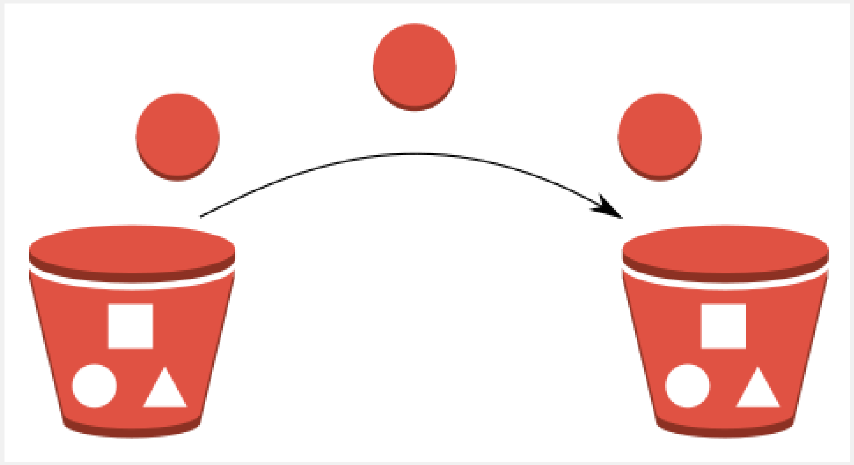
```
GET /{bucket}/{object} HTTP/1.1
```

Delete object

```
DELETE /{bucket}/{object} HTTP/1.1
```

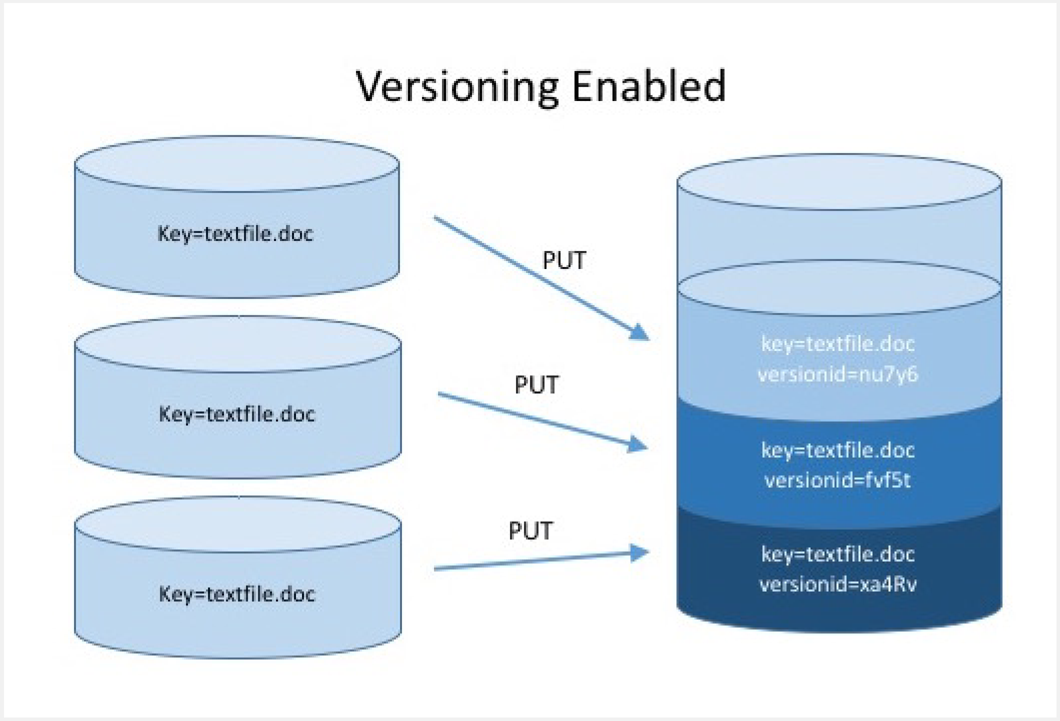
Multipart upload

- upload a single object as a set of parts
- Improved throughput
- Quick recovery from any network issues
- Pause and resume object uploads
- Begin an upload before you know the final object size



Object versioning

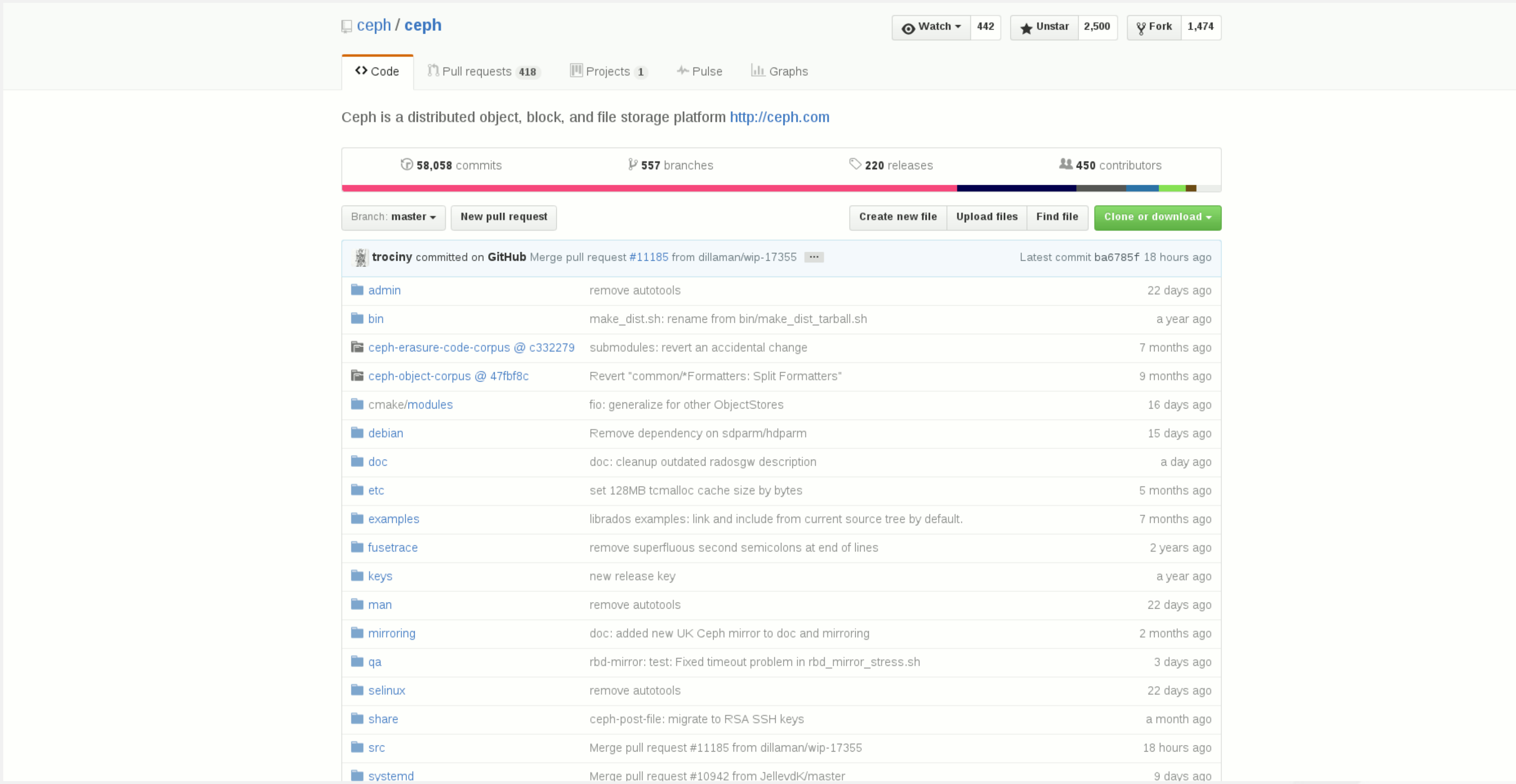
- Keeps the previous copy of the object in case of overwrite or deletion



Ceph

Cephalopod

Ceph



ceph / ceph

Watch 442 | Unstar 2,500 | Fork 1,474

Code | Pull requests 418 | Projects 1 | Pulse | Graphs

Ceph is a distributed object, block, and file storage platform http://ceph.com

58,058 commits | 557 branches | 220 releases | 450 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

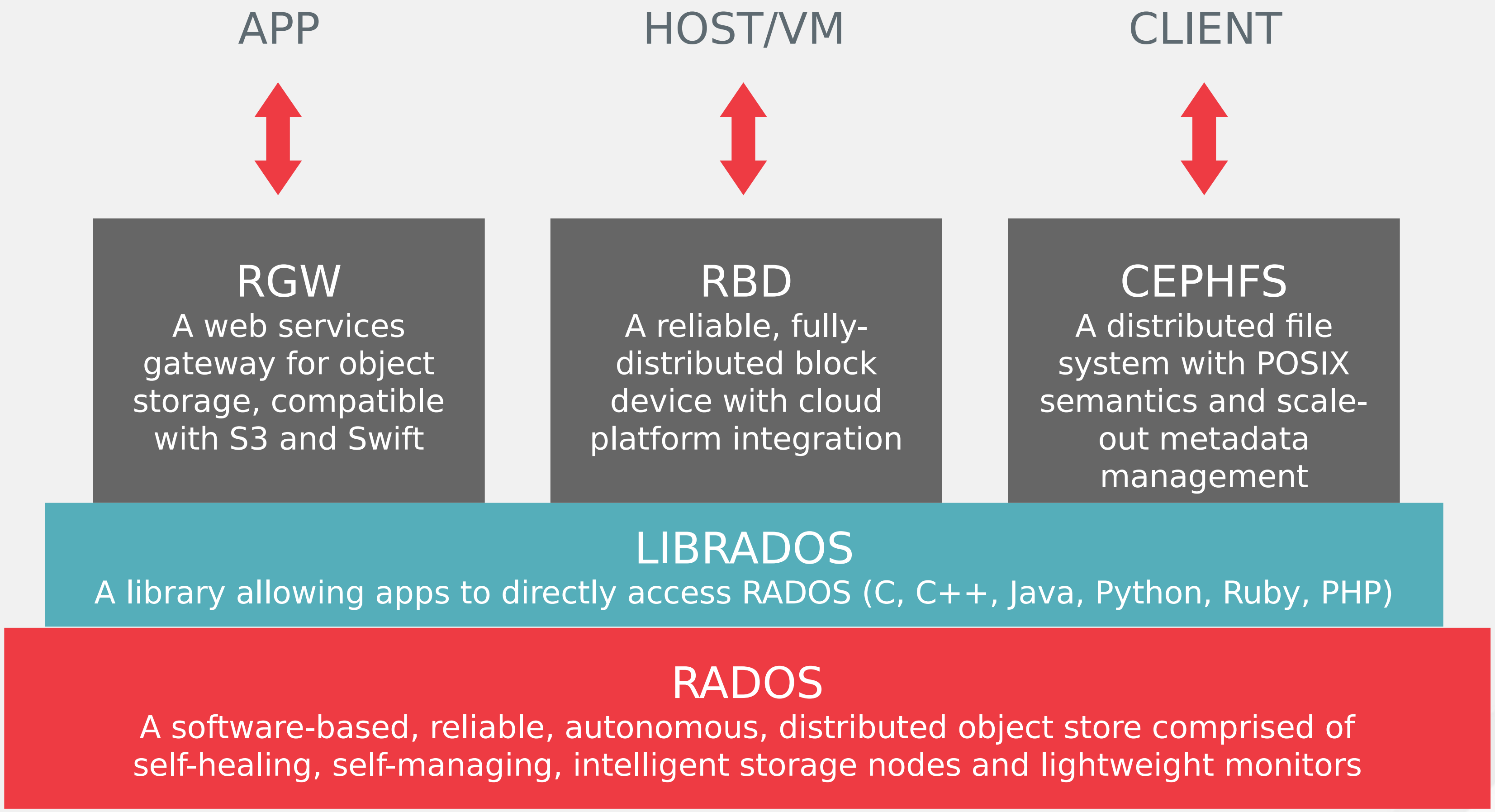
trociny committed on GitHub Merge pull request #11185 from dillaman/wip-17355 … Latest commit ba6785f 18 hours ago

admin	remove autotools	22 days ago
bin	make_dist.sh: rename from bin/make_dist_tarball.sh	a year ago
ceph-erasure-code-corpus @ c332279	submodules: revert an accidental change	7 months ago
ceph-object-corpus @ 47fbf8c	Revert "common/*Formatters: Split Formatters"	9 months ago
cmake/modules	fio: generalize for other ObjectStores	16 days ago
debian	Remove dependency on sdparm/hdparm	15 days ago
doc	doc: cleanup outdated radosgw description	a day ago
etc	set 128MB tcmalloc cache size by bytes	5 months ago
examples	librados examples: link and include from current source tree by default.	7 months ago
fusetrace	remove superfluous second semicolons at end of lines	2 years ago
keys	new release key	a year ago
man	remove autotools	22 days ago
mirroring	doc: added new UK Ceph mirror to doc and mirroring	2 months ago
qa	rbd-mirror: test: Fixed timeout problem in rbd_mirror_stress.sh	3 days ago
selinux	remove autotools	22 days ago
share	ceph-post-file: migrate to RSA SSH keys	a month ago
src	Merge pull request #11185 from dillaman/wip-17355	18 hours ago
systemd	Merge pull request #10942 from JelleydK/master	9 days ago

Ceph

- Open source
- Software defined storage
- Distributed
- No single point of failure
- Massively scalable
- Replication/Erasure Coding
- Self healing
- Unified storage: object, block and file

Ceph architecture

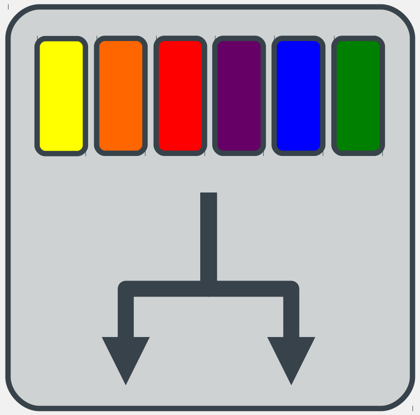

APP
HOST/VM
CLIENT
RGW
A web services gateway for object storage, compatible with S3 and Swift
RBD
A reliable, fully-distributed block device with cloud platform integration
CEPHFS
A distributed file system with POSIX semantics and scale-out metadata management
LIBRADOS
A library allowing apps to directly access RADOS (C, C++, Java, Python, Ruby, PHP)
RADOS
A software-based, reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes and lightweight monitors

Rados

- Reliable Autonomous Distributed Object Storage
- Replication/Erasure coding
- Flat object namespace within each pool
  - Different placement rules
- Strong consistency (CP system)
- Infrastructure aware, dynamic topology
- Hash-based placement (CRUSH)
- Direct client to server data path

Crush

- Controlled Replication Under Scalable Hashing
- Pseudo-random placement algorithm
- Fast calculation, no lookup
- Ensures even distribution
- Repeatable, deterministic
- Rule-based configuration
    - specifiable replication
    - infrastructure topology aware
    - allows weighting



OSD node

- Object Storage Device
- 10s to 1000s in a cluster
- One per disk (or one per SSD, RAID group…)
- Serve stored objects to clients
- Intelligently peer for replication & recovery

Monitor node

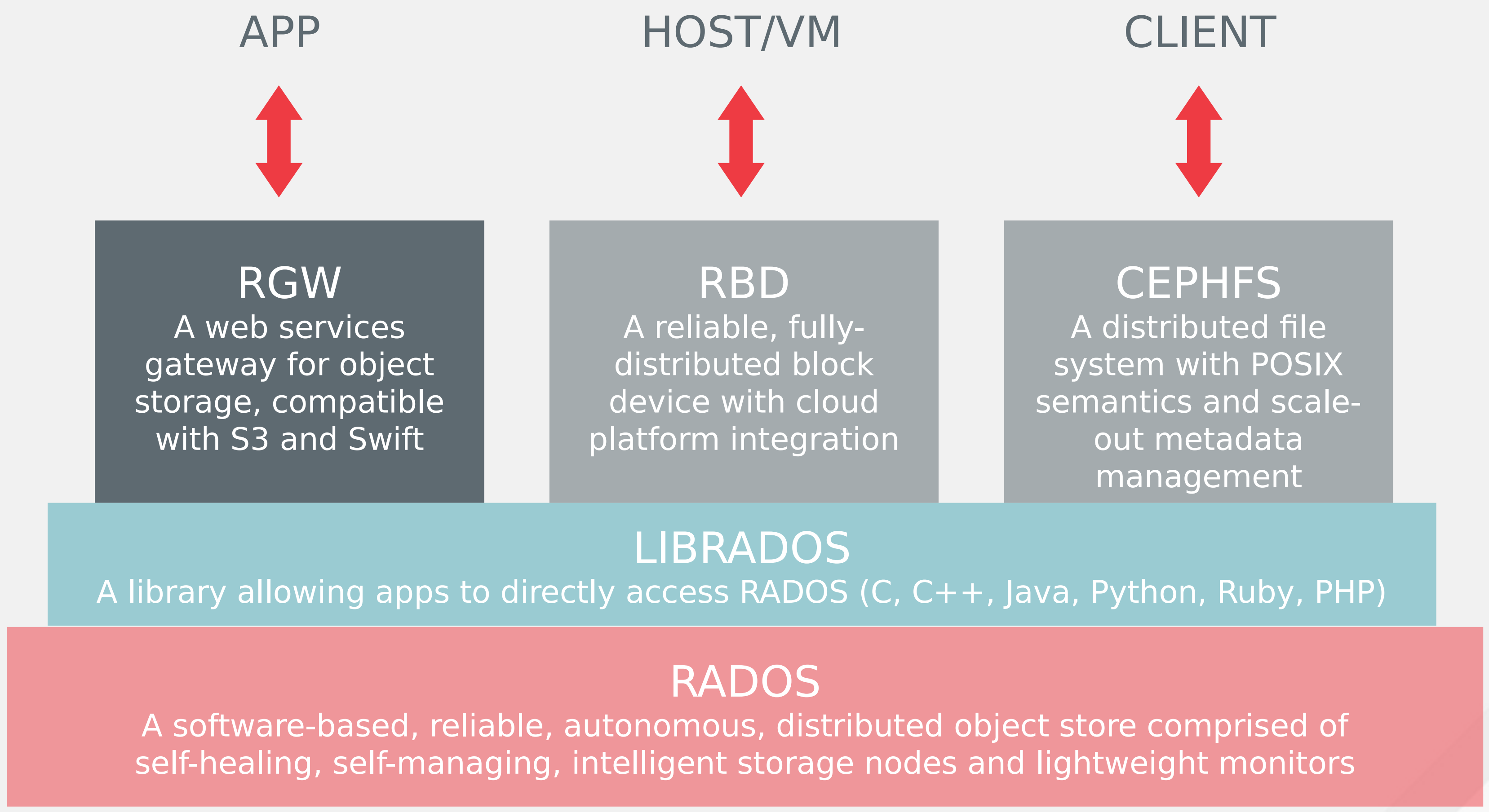
- Maintain cluster membership and state
- Provide consensus for distributed decision-making
- Small, odd number
- These do not serve stored objects to clients

Librados API

- Efficient key/value storage inside an object
- Atomic single-object transactions
  - update data, attr, keys together
  - atomic compare-and-swap
- Object-granularity snapshot infrastructure
- Partial overwrite of existing data
- RADOS classes (stored procedures)
- Watch/Notify on an object

Rados Gateway

Rados Gateway



APP

HOST/VM

CLIENT

RGW
A web services gateway for object storage, compatible with S3 and Swift

RBD
A reliable, fully-distributed block device with cloud platform integration

CEPHFS
A distributed file system with POSIX semantics and scale-out metadata management

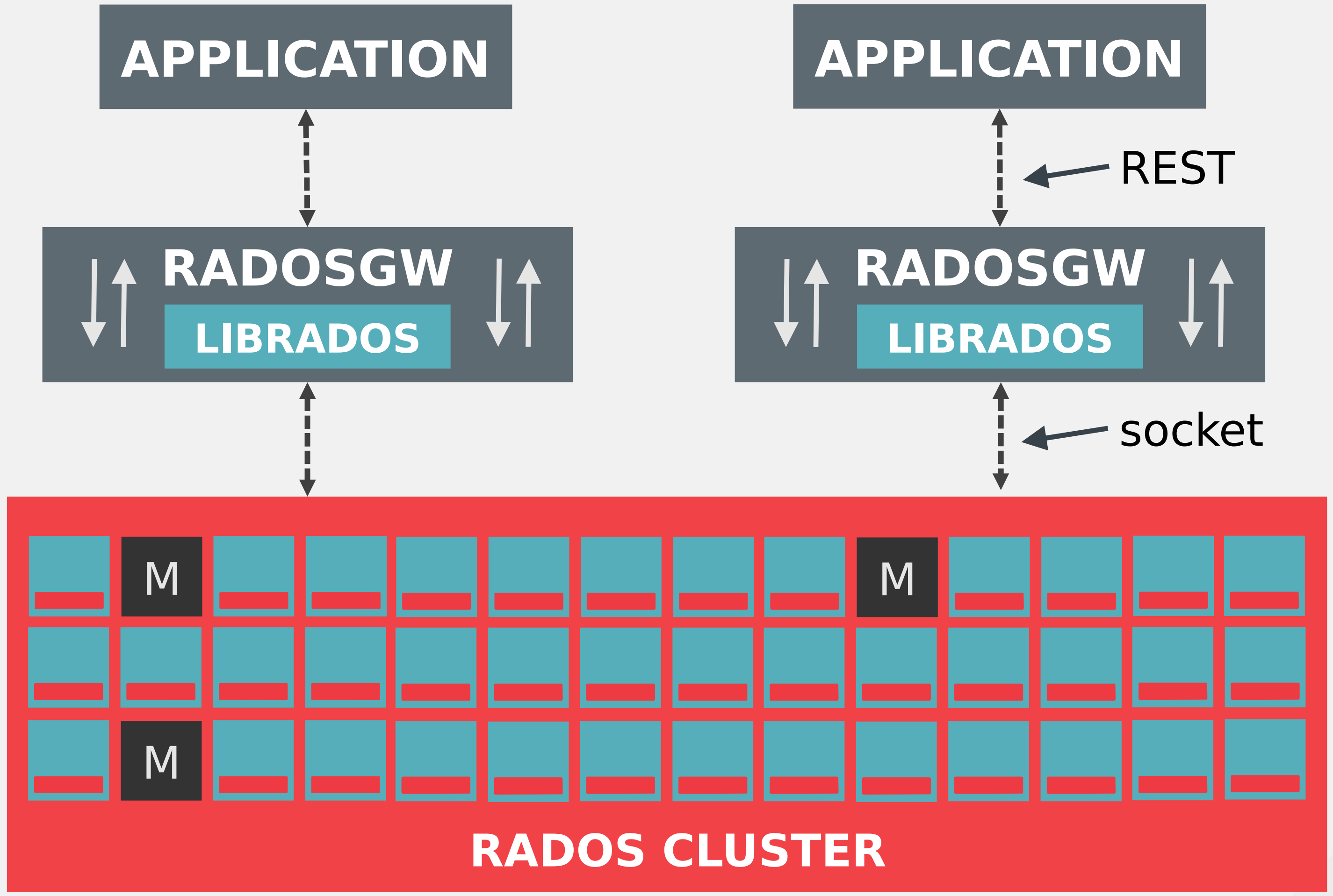
LIBRADOS
A library allowing apps to directly access RADOS (C, C++, Java, Python, Ruby, PHP)

RADOS
A software-based, reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes and lightweight monitors

RGW vs RADOS objects

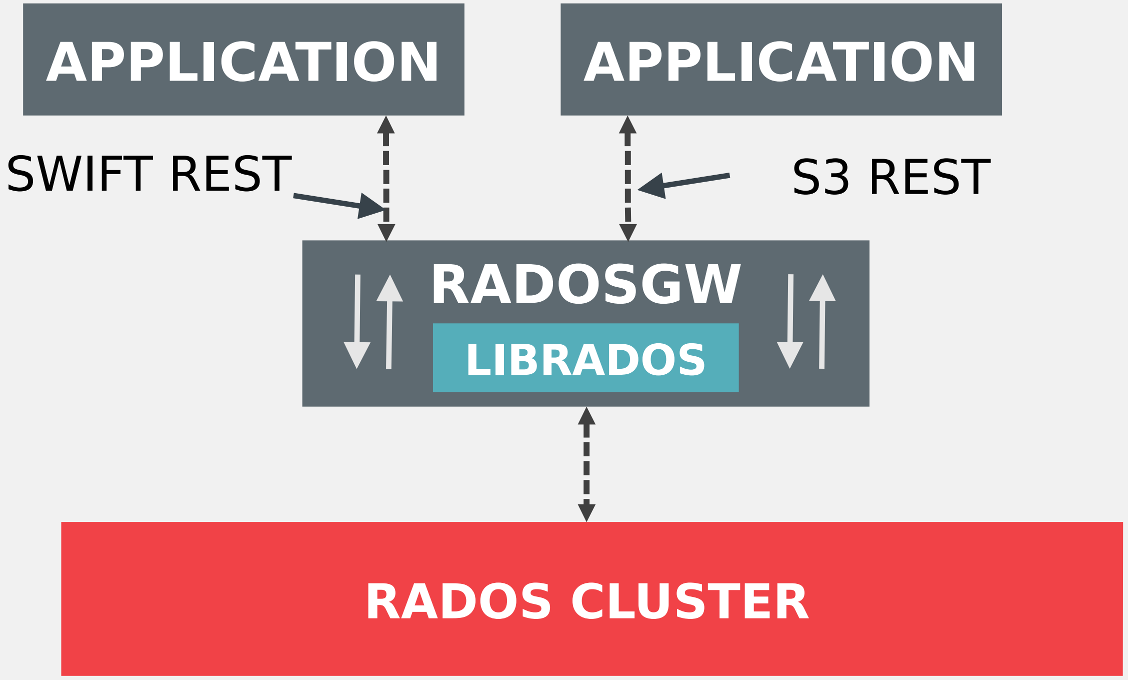
- RADOS
  - Limited object sizes (4M)
  - Mutable objects
  - Not indexed
  - per-pool ACLs
- RGW
  - Large objects (TB)
  - Immutable objects
  - Sorted bucket listing
  - per object ACLs

Rados Gateway

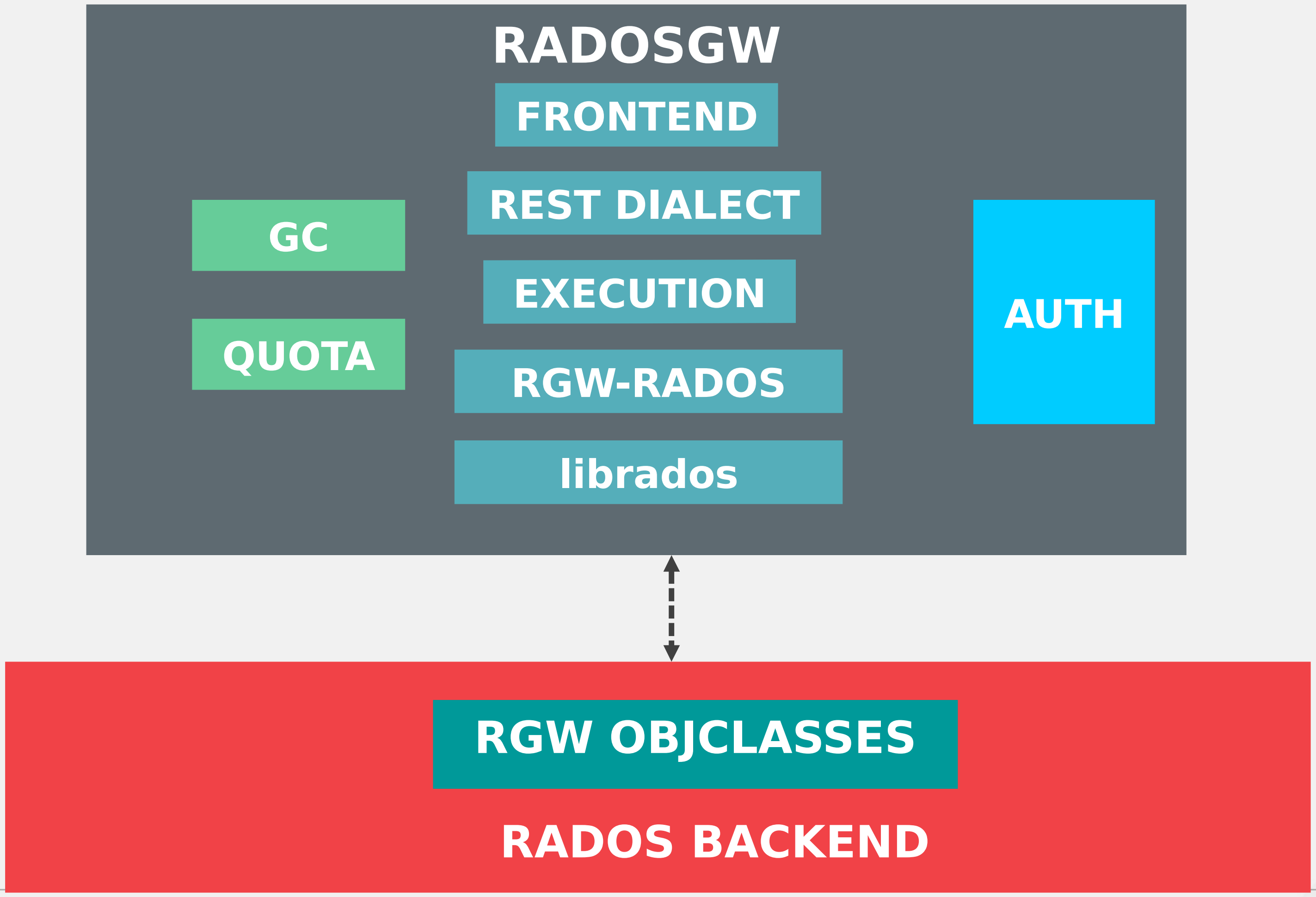


RESTful OBJECT STORAGE

- Users/Tenants
- Data
  - Buckets
  - Objects
  - Metadata
  - ACLs
- Authentication
- APIs
  - S3
  - Swift
  - NFS



RGW


RADOSGW
FRONTEND
REST DIALECT
EXECUTION
RGW-RADOS
librados
GC
QUOTA
AUTH
RGW OBJCLASSES
RADOS BACKEND

RGW Components

- Frontend
  - FastCGI - external web servers
  - Civetweb – embedded web server
- Rest Dialect
  - S3
  - Swift
  - Other API (NFS)
- Execution layer – common layer for all dialects

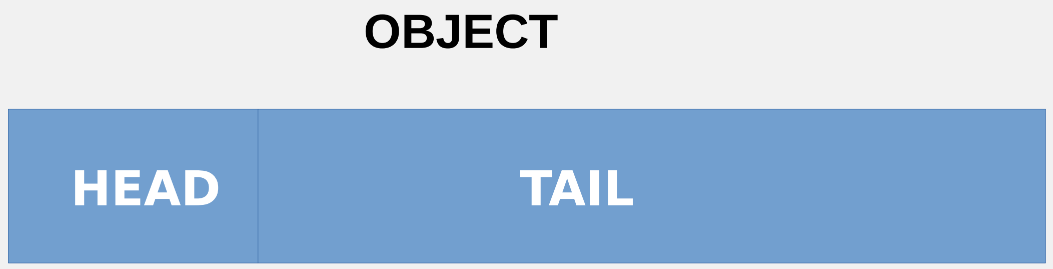
RGW Components

- RGW Rados – manages RGW data by using rados
  - object striping
  - atomic overwrites
  - bucket index handling
  - Object classes that run on the OSDs
- Quota - handles user or bucket quotas.
- Authentication - handle users authentication
- GC - Garbage collection mechanism that runs in the background.

RGW objects

- Large objects
- Fast small object access
- Fast access to object attributes
- Buckets can consist of a very large number of objects

RGW objects



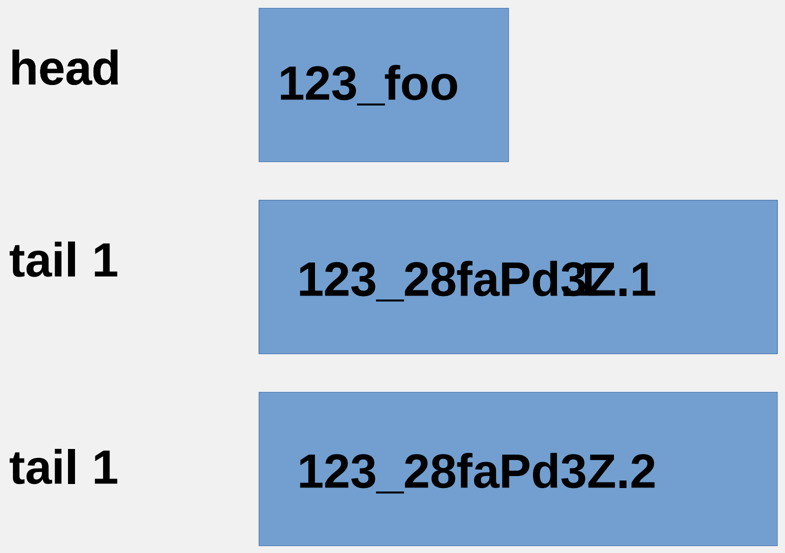
- Head
  - Single rados object
  - Object metadata (acls, user attributes, manifest)
  - Optional start of data
- Tail
  - Striped data
  - 0 or more rados objects

RGW Objects

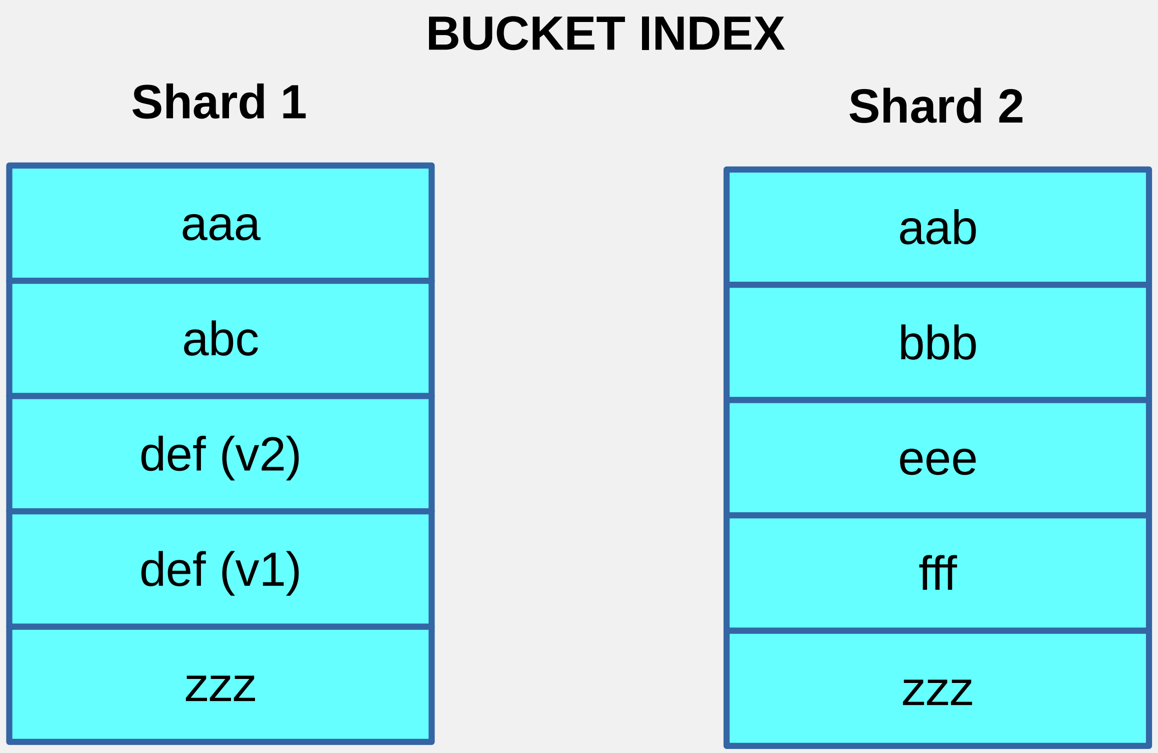
OBJECT: foo

BUCKET: boo

BUCKET ID: 123



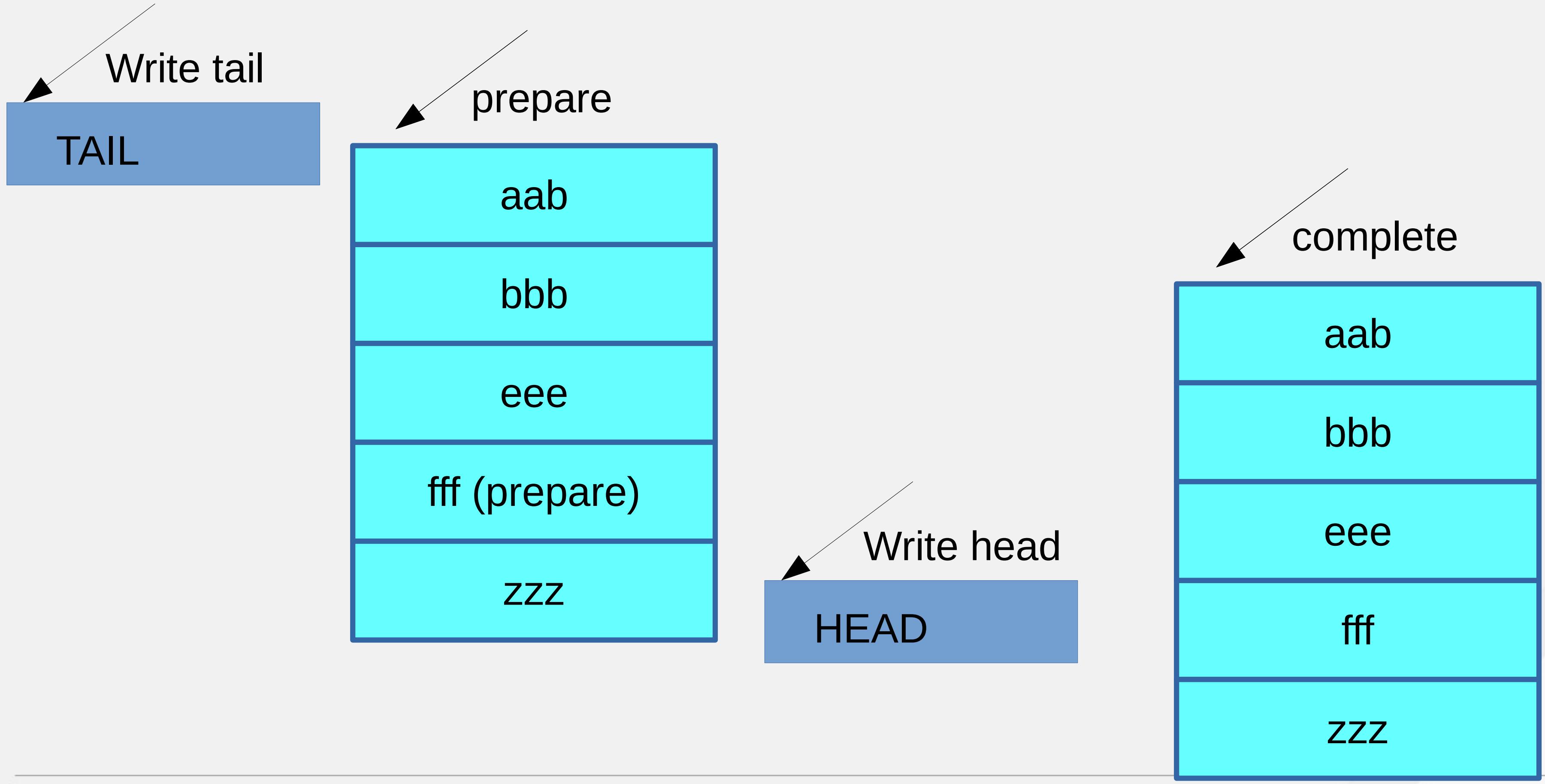
RGW bucket index



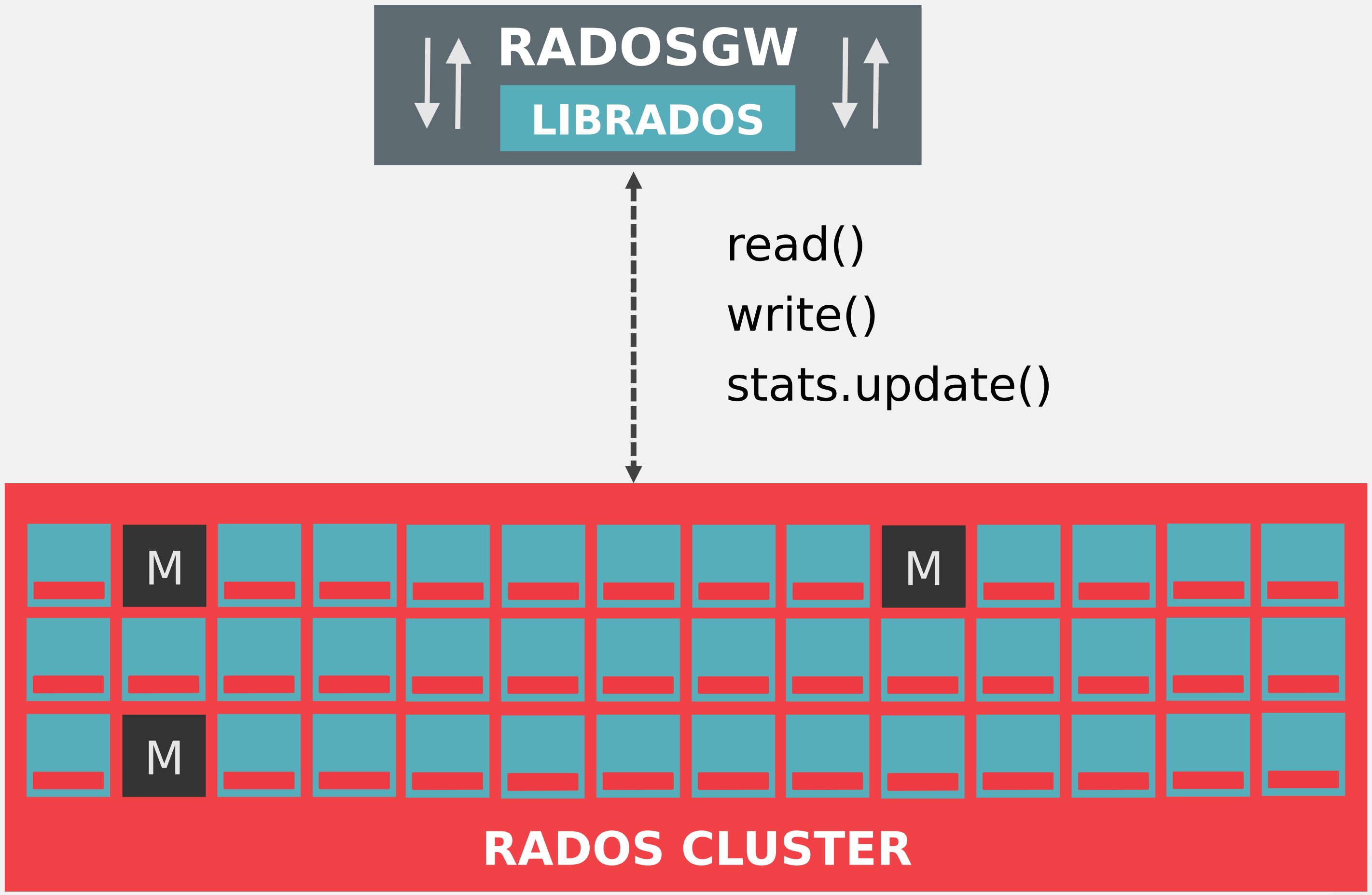
RGW object creation

- Update bucket index
- Create head object
- Create tail objects
- All those operations need to be consist

RGW object creation



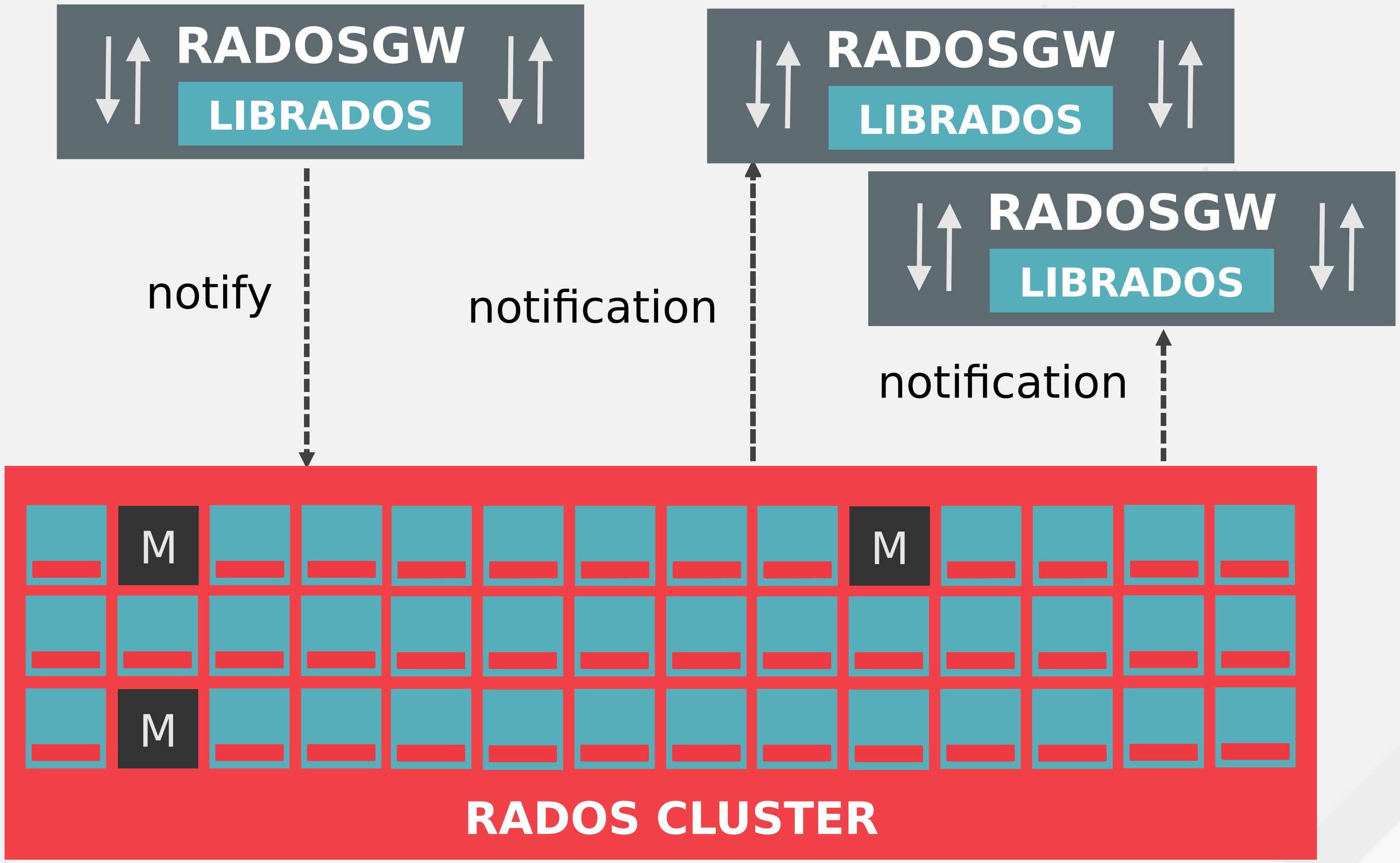
RGW quota


RADOSGW
LIBRADOS
read()
write()
stats.update()
M
M
M
RADOS CLUSTER

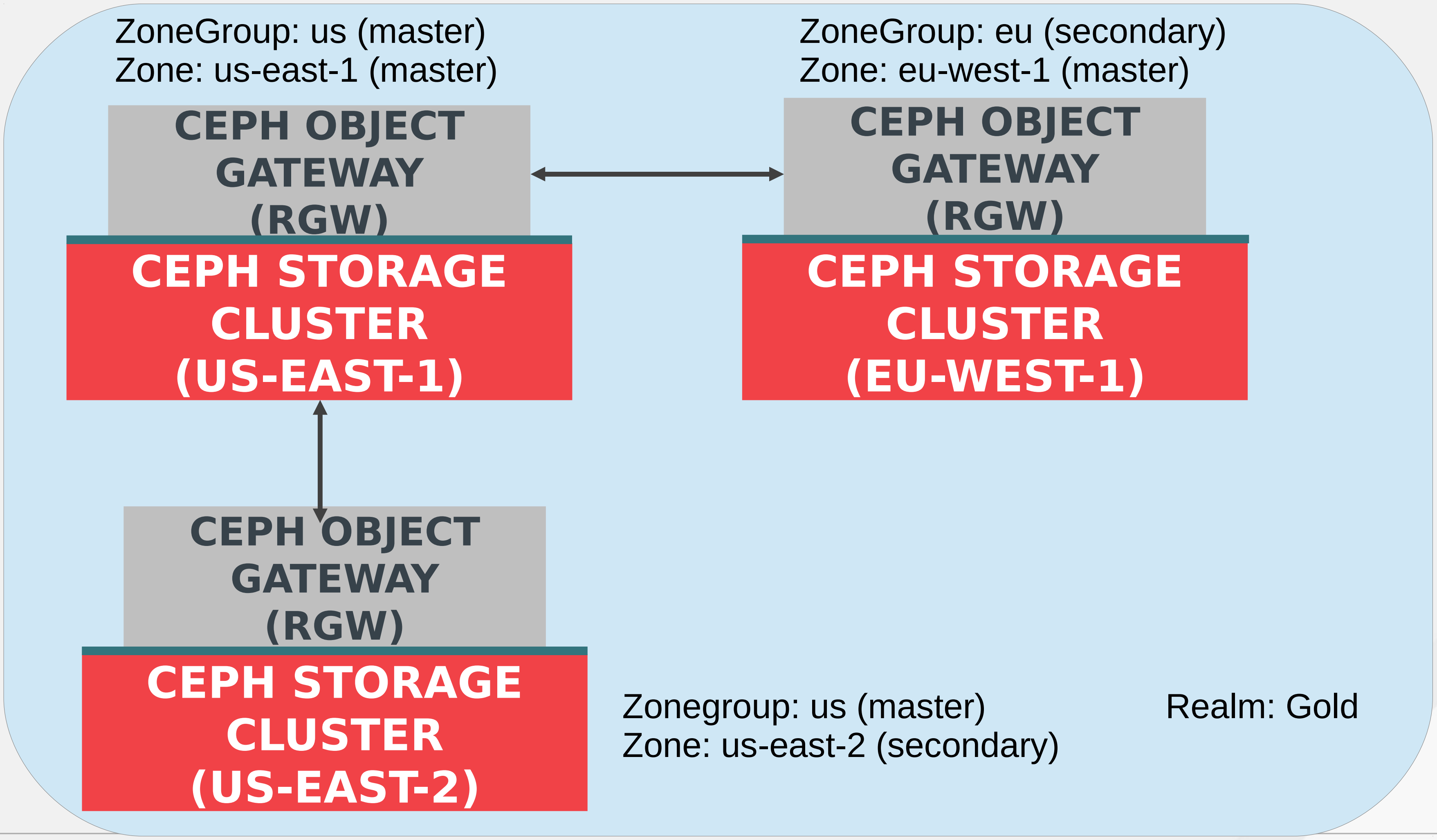
RGW metadata cache

- Metadata needed for each request:
  - User Info
  - Bucket Entry Point
  - Bucket Instance Info

RGW metadata cache



Multisite environment


ZoneGroup: us (master)
Zone: us-east-1 (master)
CEPH OBJECT GATEWAY (RGW)
CEPH STORAGE CLUSTER (US-EAST-1)
ZoneGroup: eu (secondary)
Zone: eu-west-1 (master)
CEPH OBJECT GATEWAY (RGW)
CEPH STORAGE CLUSTER (EU-WEST-1)
CEPH OBJECT GATEWAY (RGW)
CEPH STORAGE CLUSTER (US-EAST-2)
Zonegroup: us (master)
Zone: us-east-2 (secondary)
Realm: Gold

multisite

- Implementation as part of the radosgw (in c++)
- Asynchronous (co-routines)
- Active/active support
- Namespaces
- Failover/failback
- Backward compatibility with the sync agent
- Meta data sync is synchronous
- Data sync is asynchronous

More cool features

- Object life cycle
- Object copy
- Bulk operations
- Encryption
- Compression
- Torrents
- Static website
- Metadata search
- Bucket resharding

THANK YOU

Ceph mailing lists:

Ceph-users@ceph.com

ceph-devel@ceph.com

IRC:

Irc.oftc.net #ceph #ceph-devel