FROM HPC TO CLOUD … AND BACK AGAIN?

SAGE WEIL – PDSW 2014.11.16

AGENDA

- A bit of history and architecture
  - Technology
  - Community
- Challenges
- Looking forward

CEPH

APP

HOST/VM

CLIENT

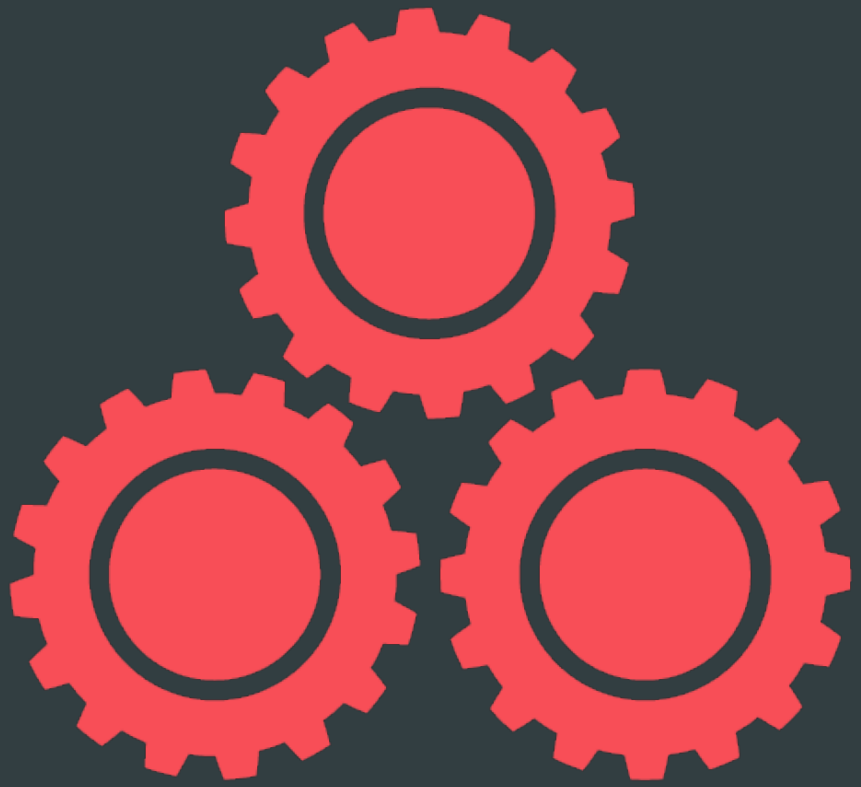
RGW
A web services gateway for object storage, compatible with S3 and Swift

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



SOME HISTORY

...AND ARCHITECTURE

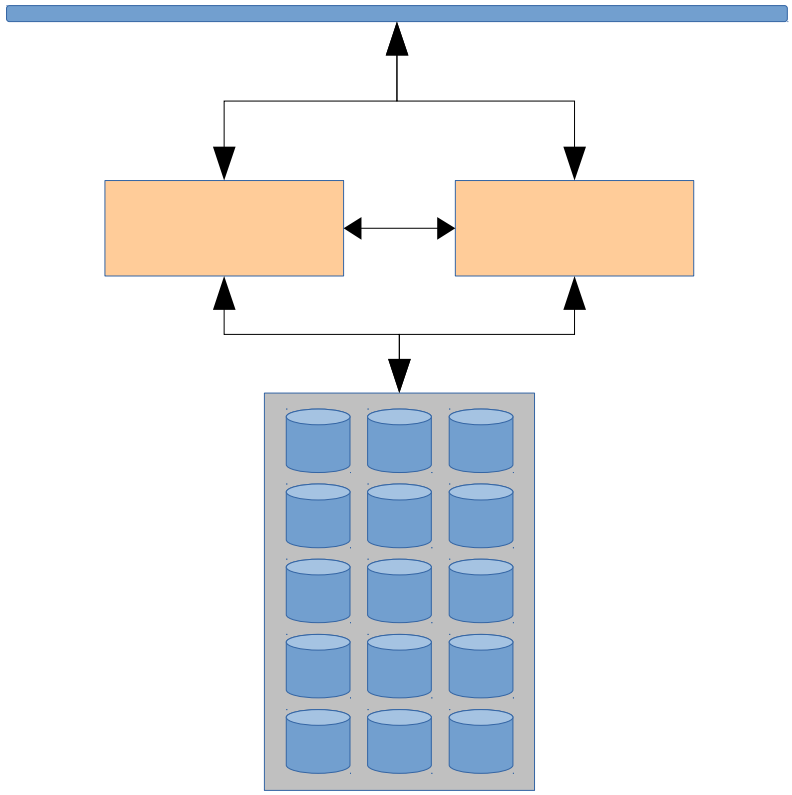
ORIGINS

- Petascale object storage
  - DOE: LANL, LLNL, Sandia
  - Scalability, reliability, performance
- Scalable metadata management
- First line of Ceph code
  - Summer internship at LLNL

MOTIVATING PRINCIPLES

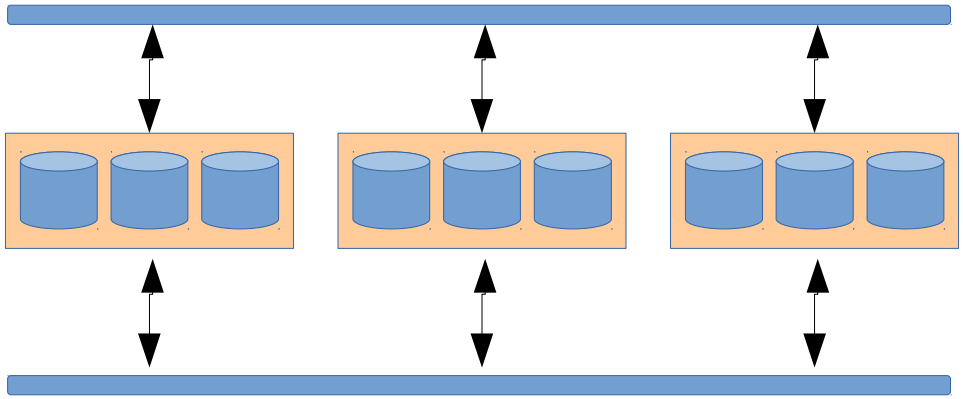
- “Intelligent” everything
  - Smart disks
  - Smart MDS
  - Dynamic load balancing
- Design tenets
  - All components must scale horizontally
  - There can be no single point of failure
  - Self-manage whenever possible
- Open source
  - The solution must be hardware agnostic

CLIENT / SERVER



“clients stripe data across reliable things”

CLIENT / CLUSTER



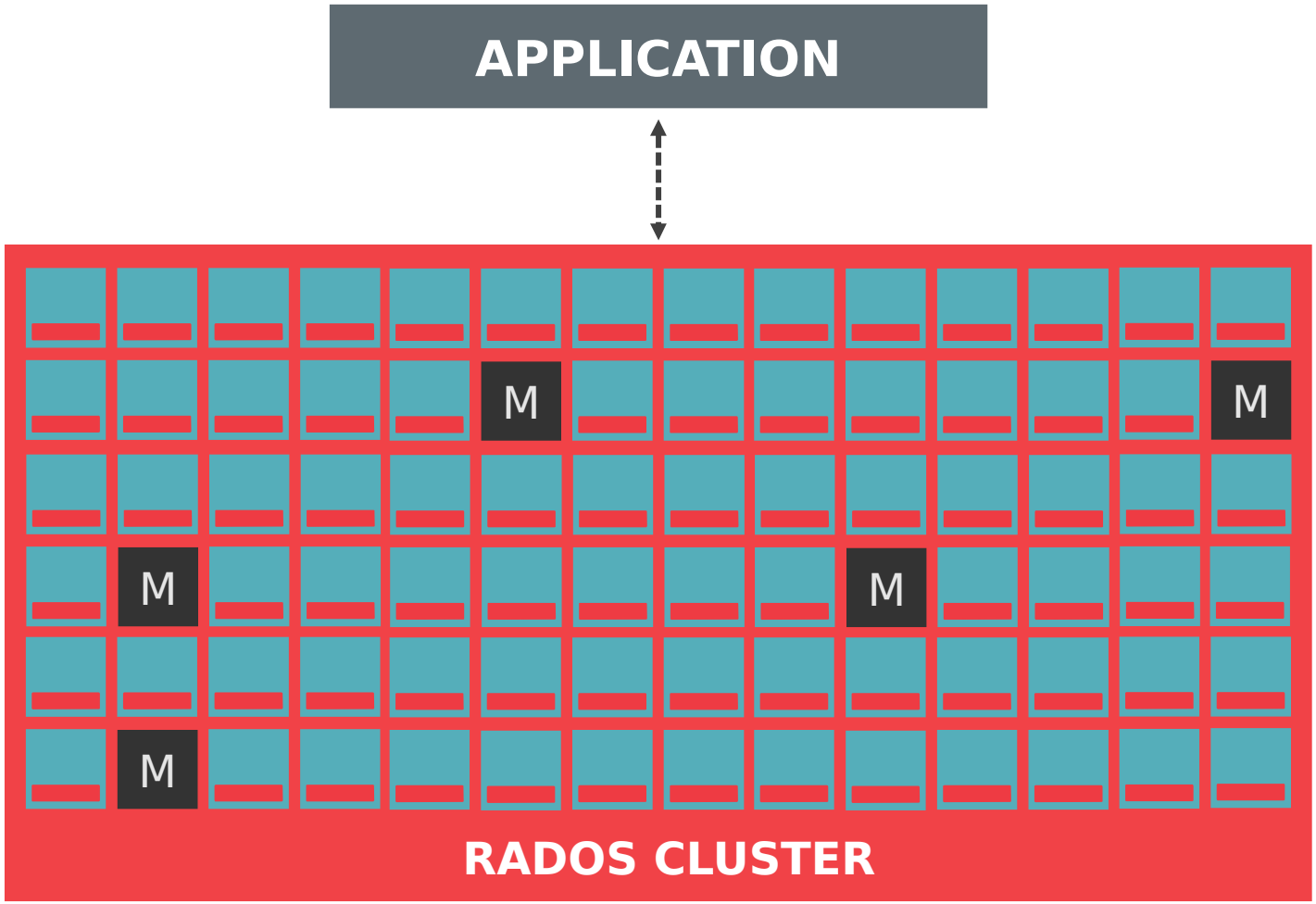
access network

mortal targets

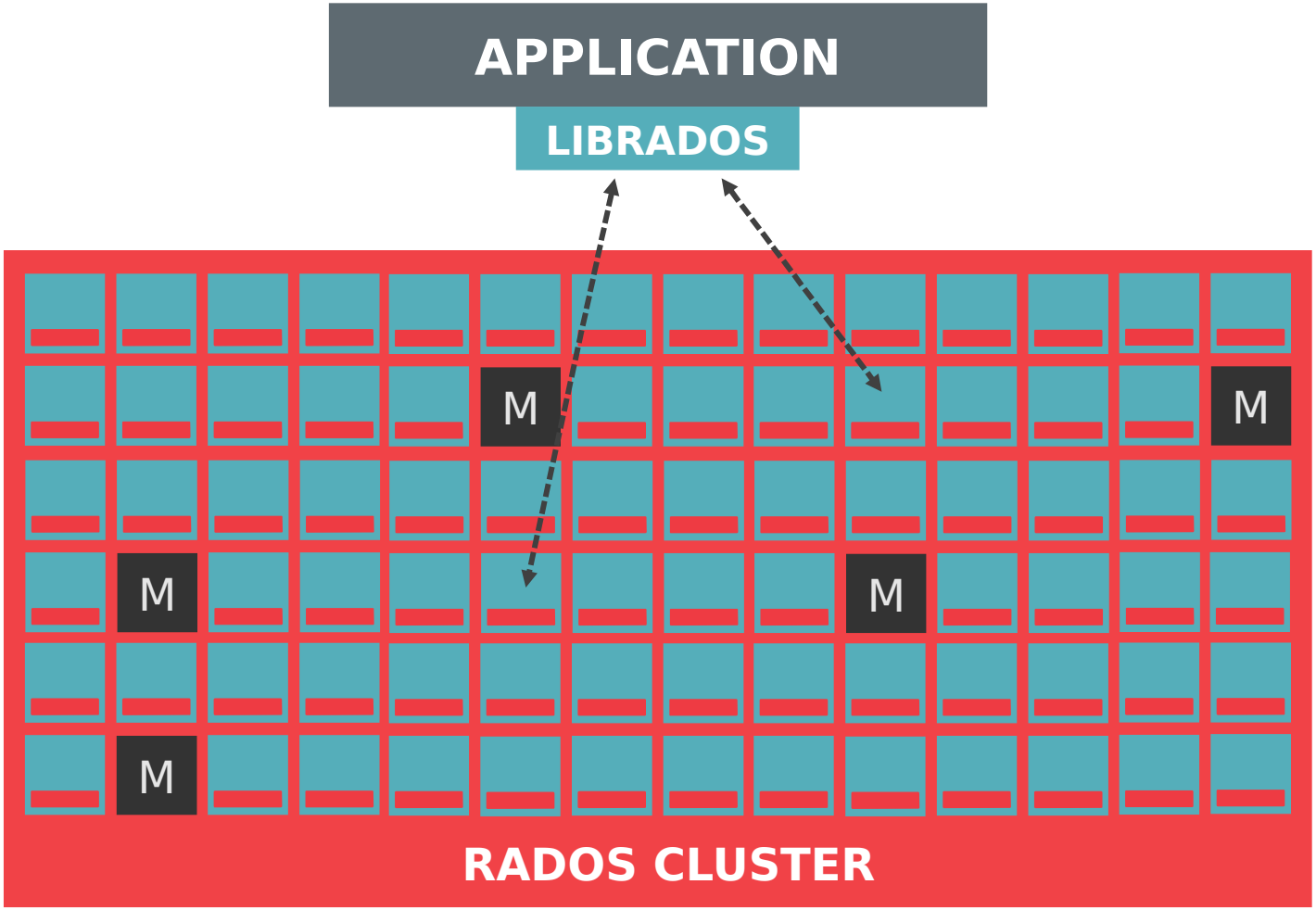
backside network
(optional)

“client stripe across unreliable things”
“servers coordinate replication, recovery”

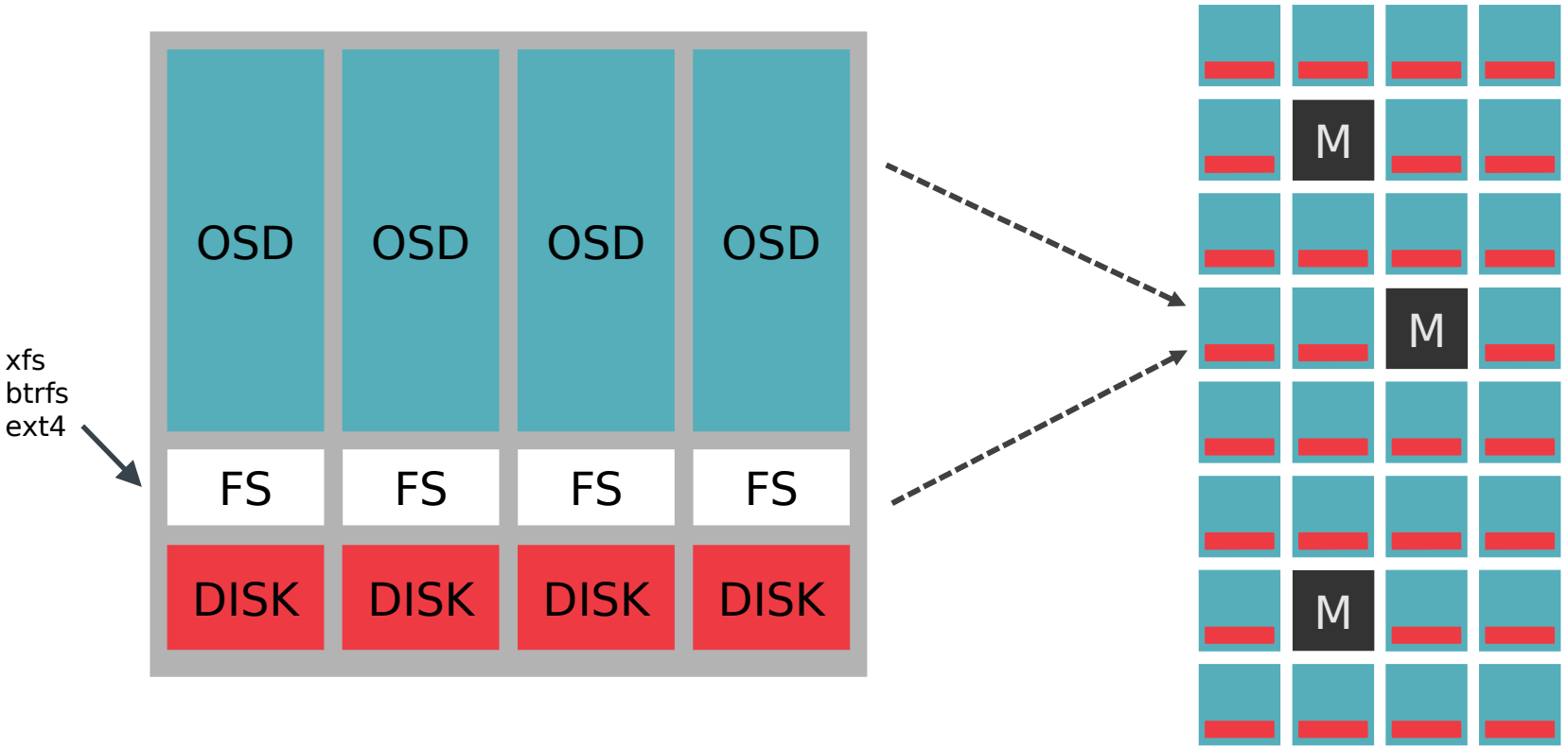
RADOS CLUSTER



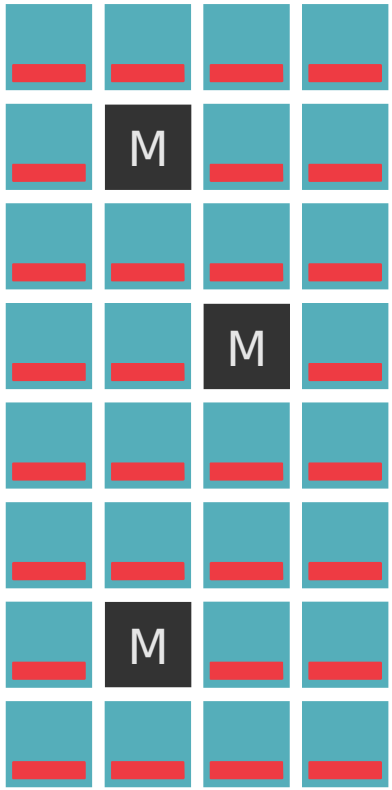
RADOS CLUSTER



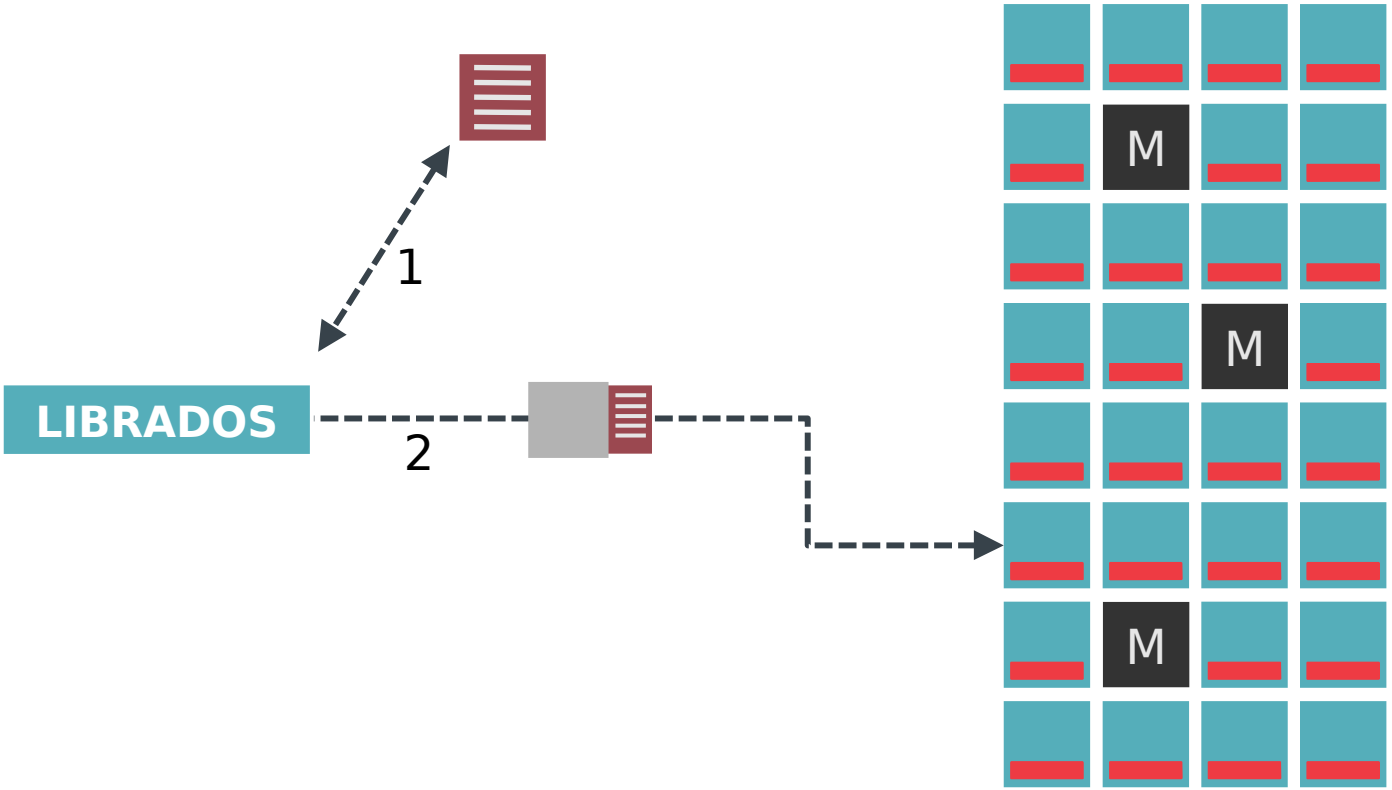
MANY OSDS PER HOST



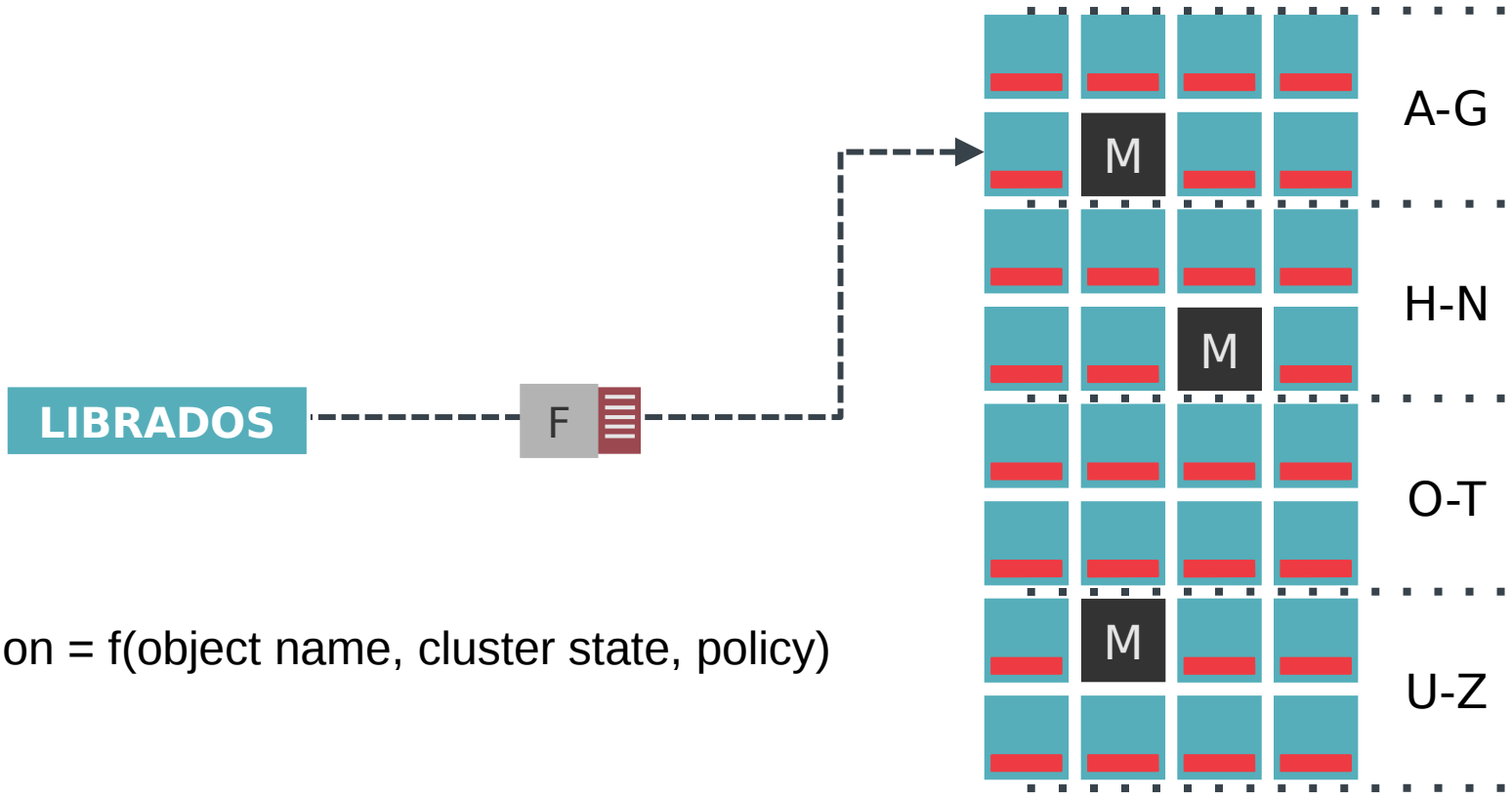
WHERE DO OBJECTS LIVE?



A METADATA SERVER?

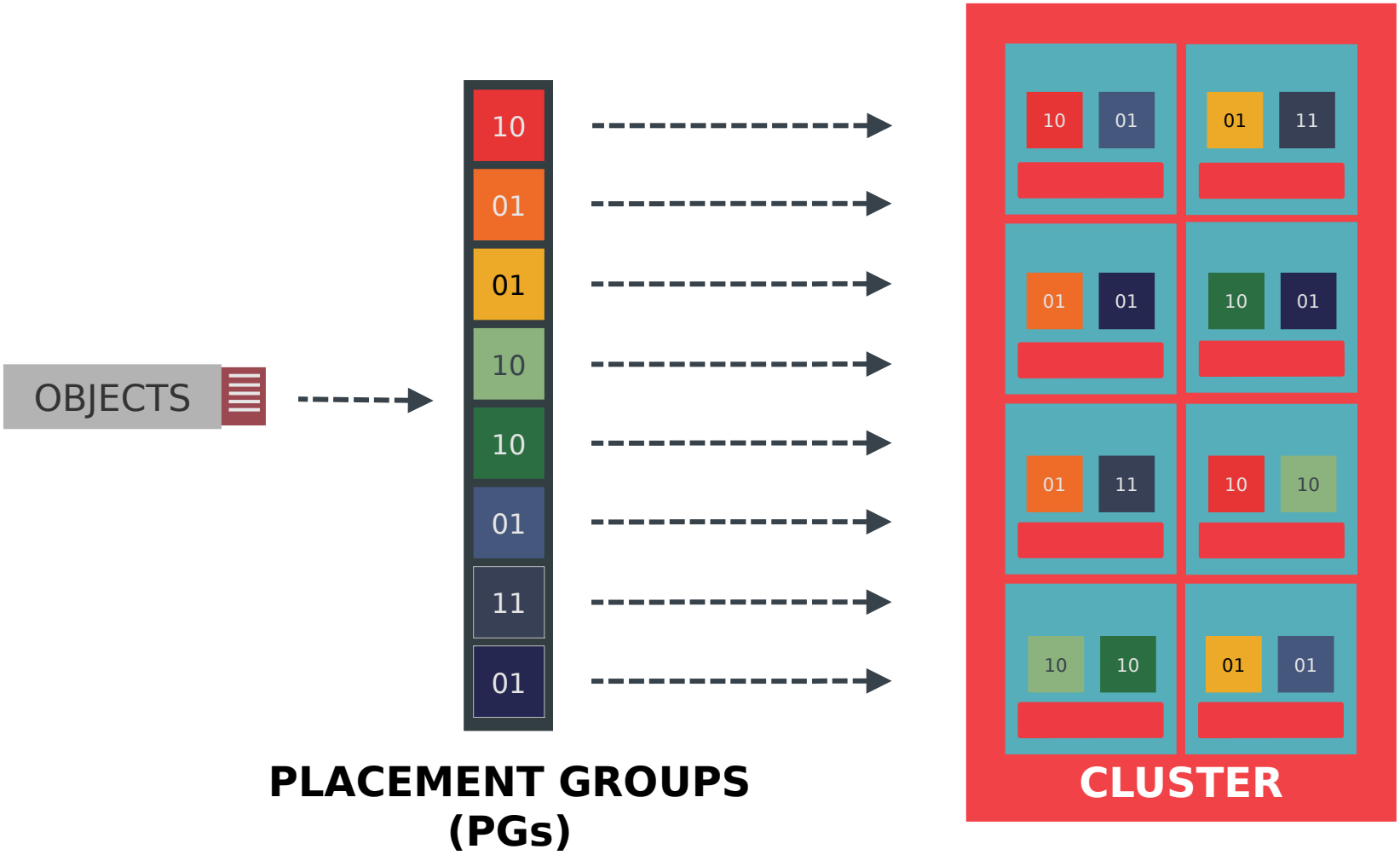


CALCULATED PLACEMENT

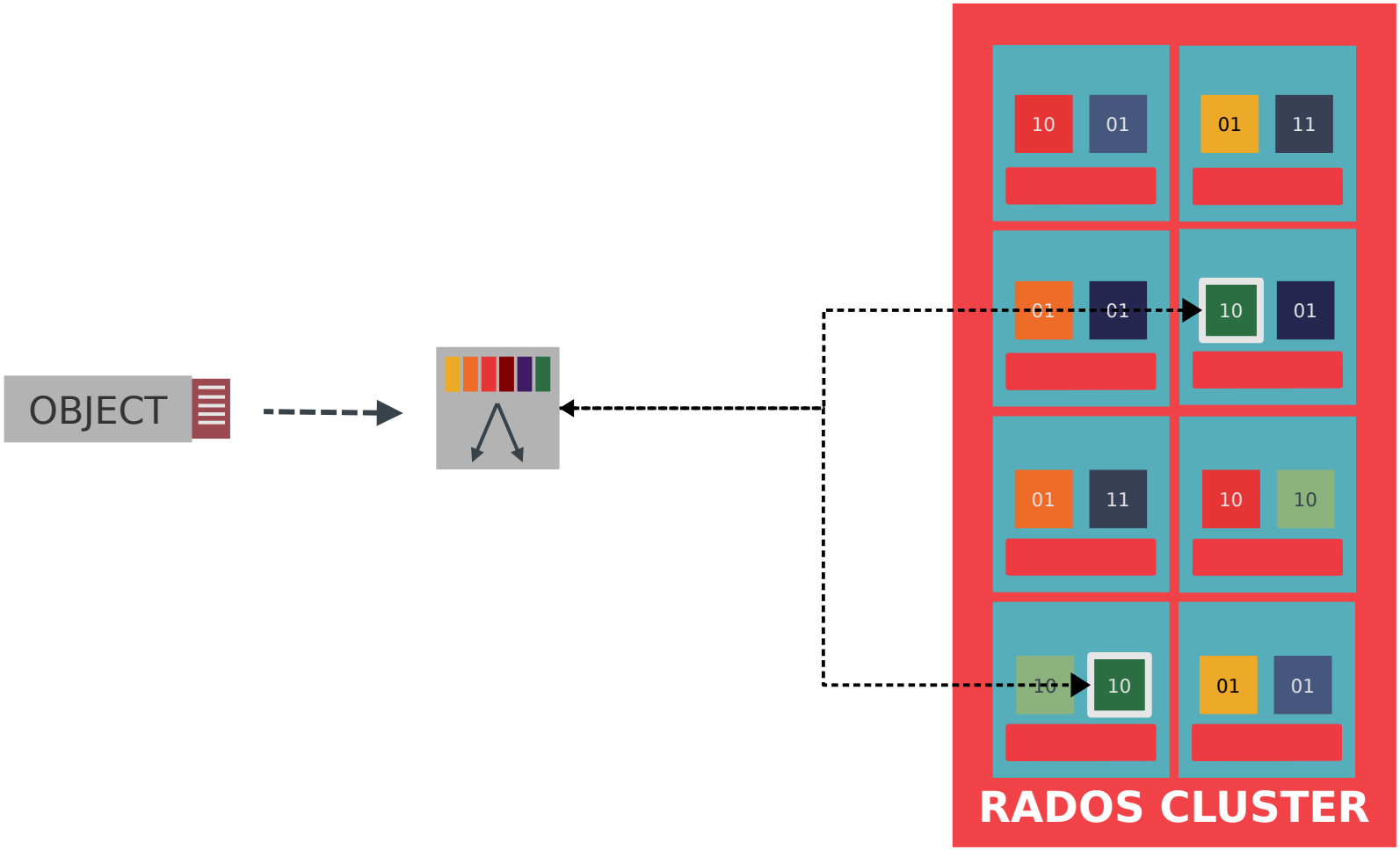


location = f(object name, cluster state, policy)

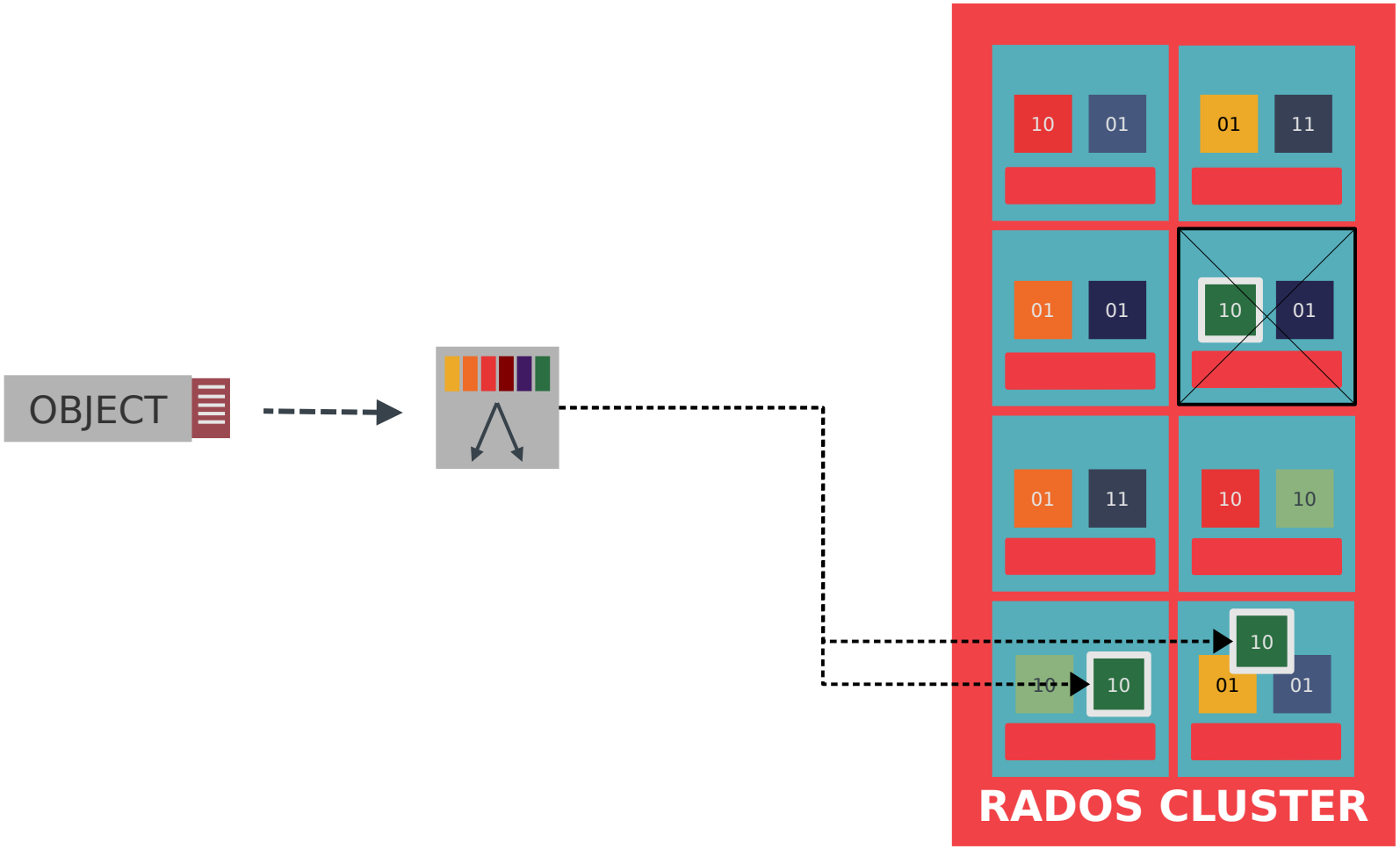
CRUSH


OBJECTS
10
01
01
10
10
01
11
01
PLACEMENT GROUPS
(PGs)
10 01
01 11
01 01
10 01
01 11
10 10
10 10
01 01
CLUSTER

CRUSH IS A QUICK CALCULATION

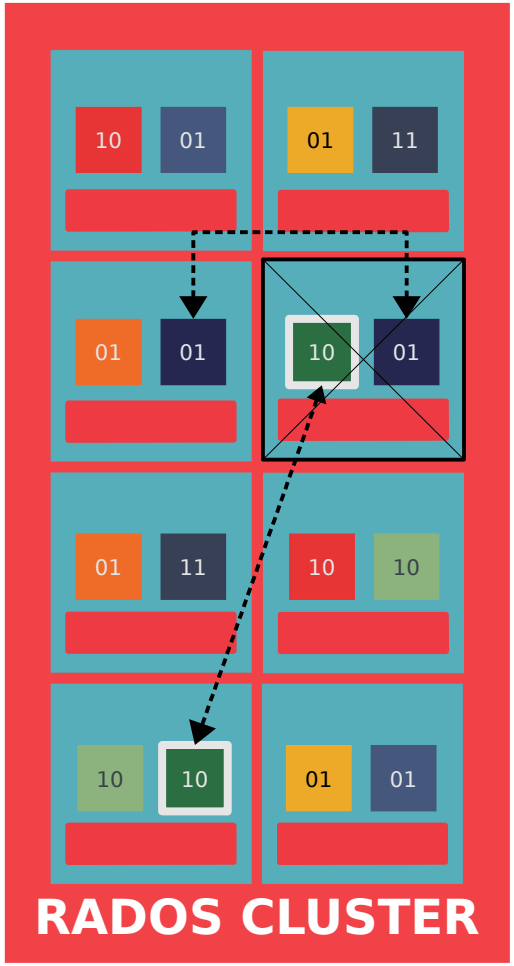


CRUSH AVOIDS FAILED DEVICES

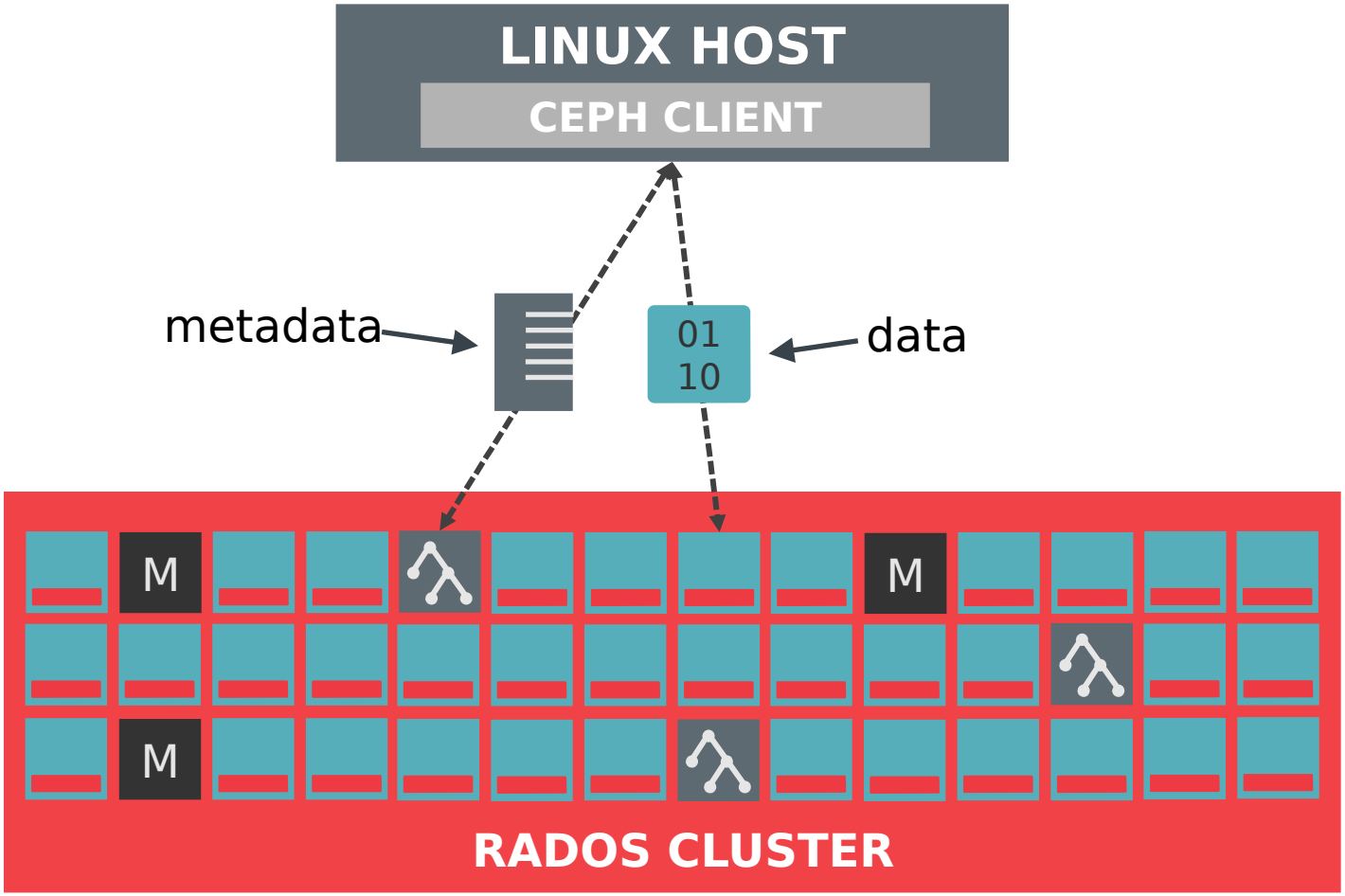


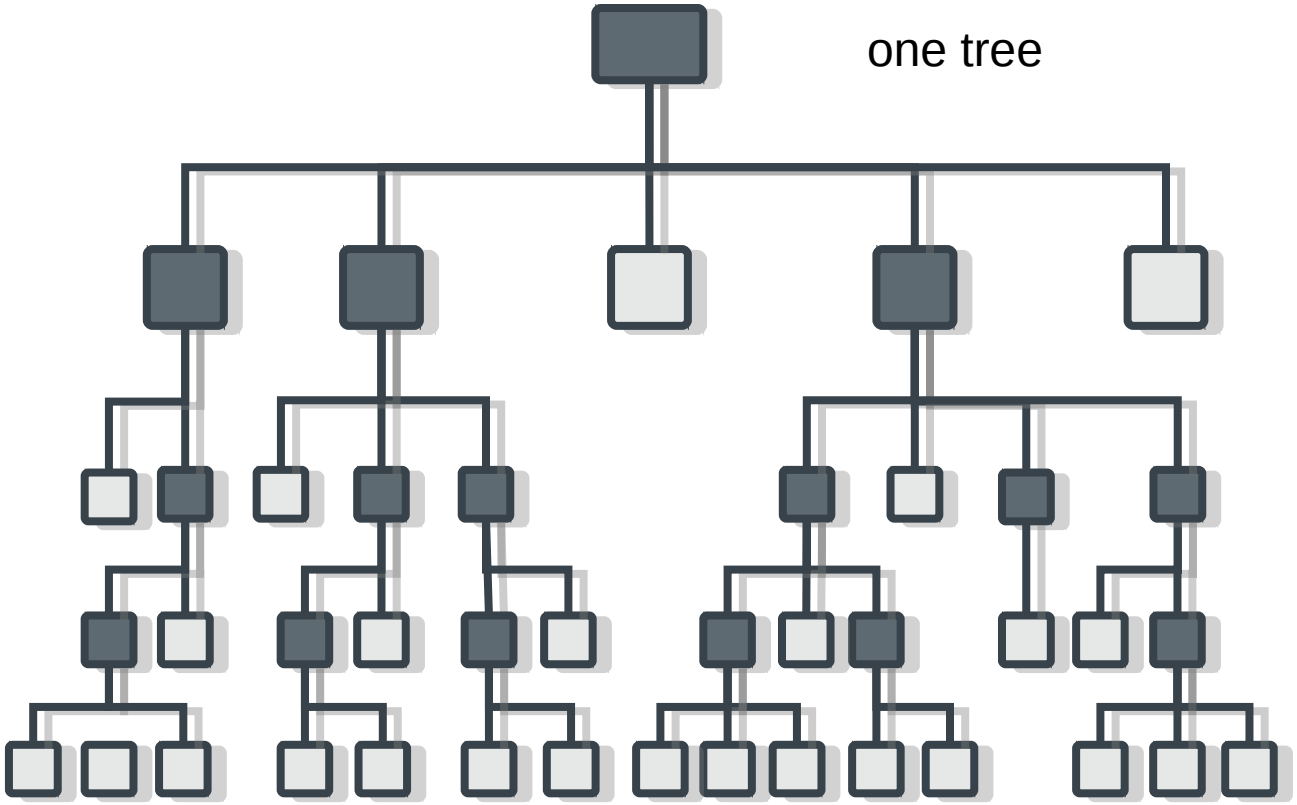
DECLUSTERED PLACEMENT

- OSDs store many PGs
- PGs that map to the same OSD generally have replicas that do not
  - No spares
  - Highly parallel recovery
- Recovery is loosely coordinated
  - Monitors publish new CRUSH map
    - “OSD.123 is now down”
  - OSDs migrate data cooperatively
  - With strong client consistency

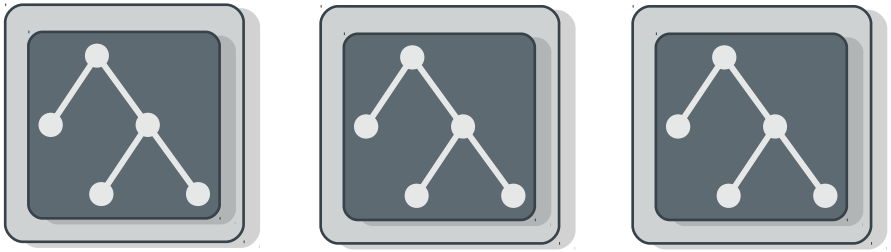


FILE SYSTEM

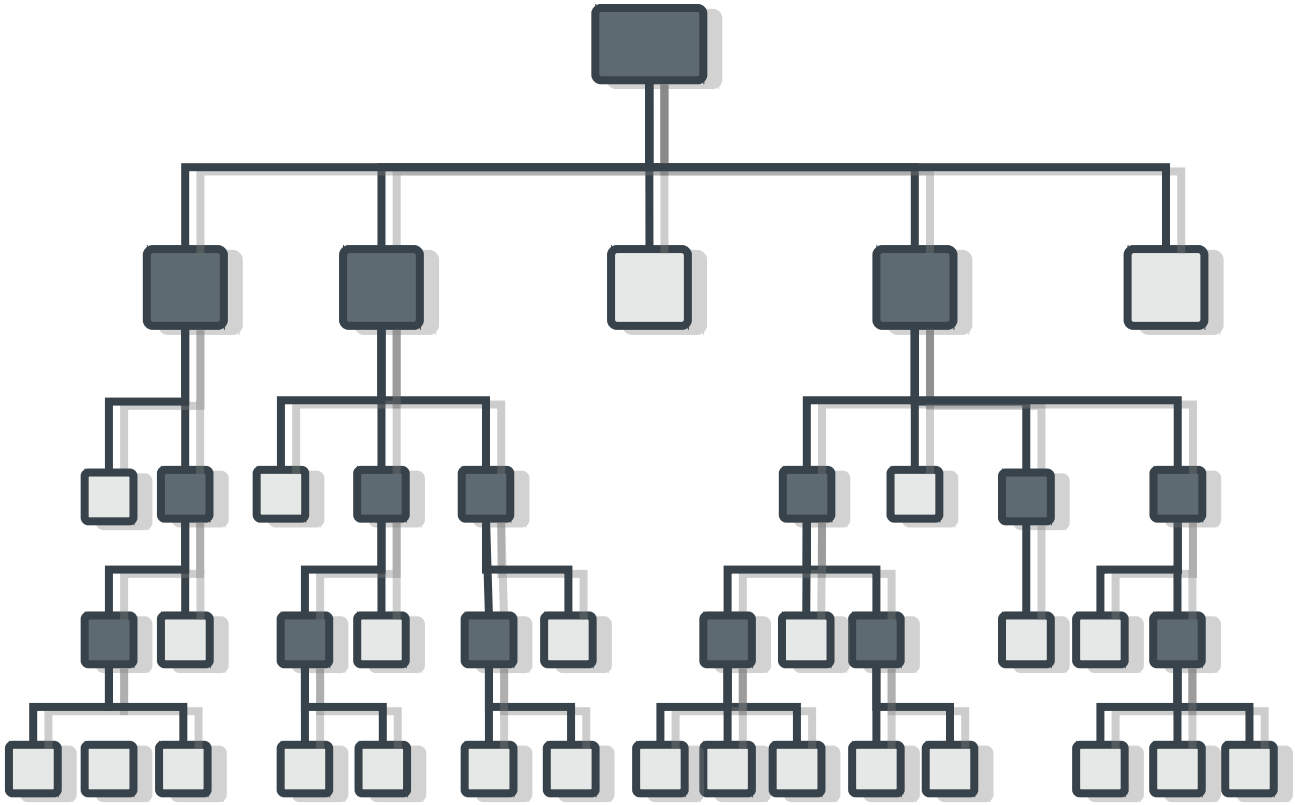

LINUX HOST
CEPH CLIENT
metadata
01
10
data
M
M
M
RADOS CLUSTER

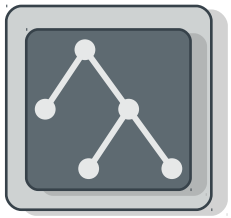


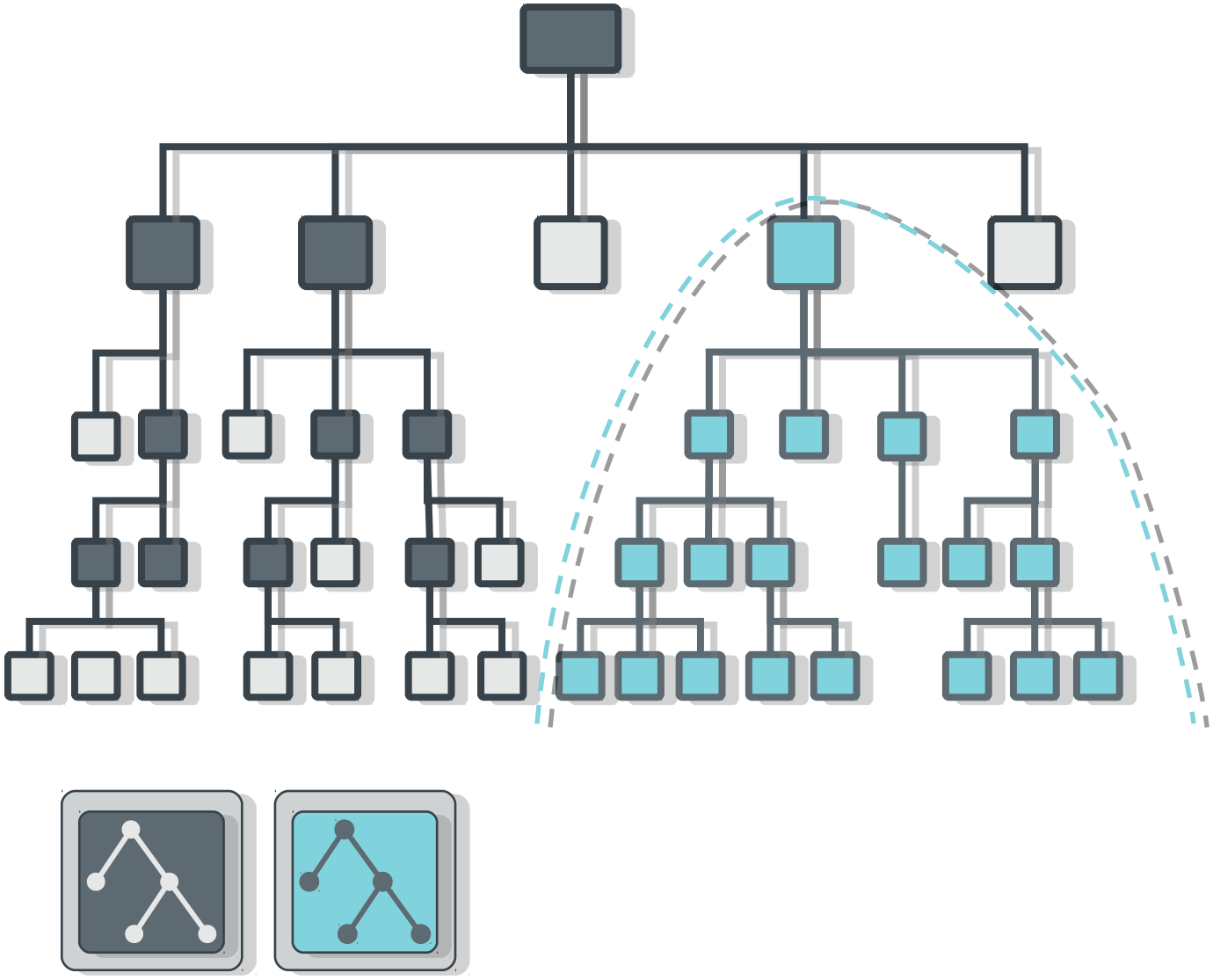
three metadata servers

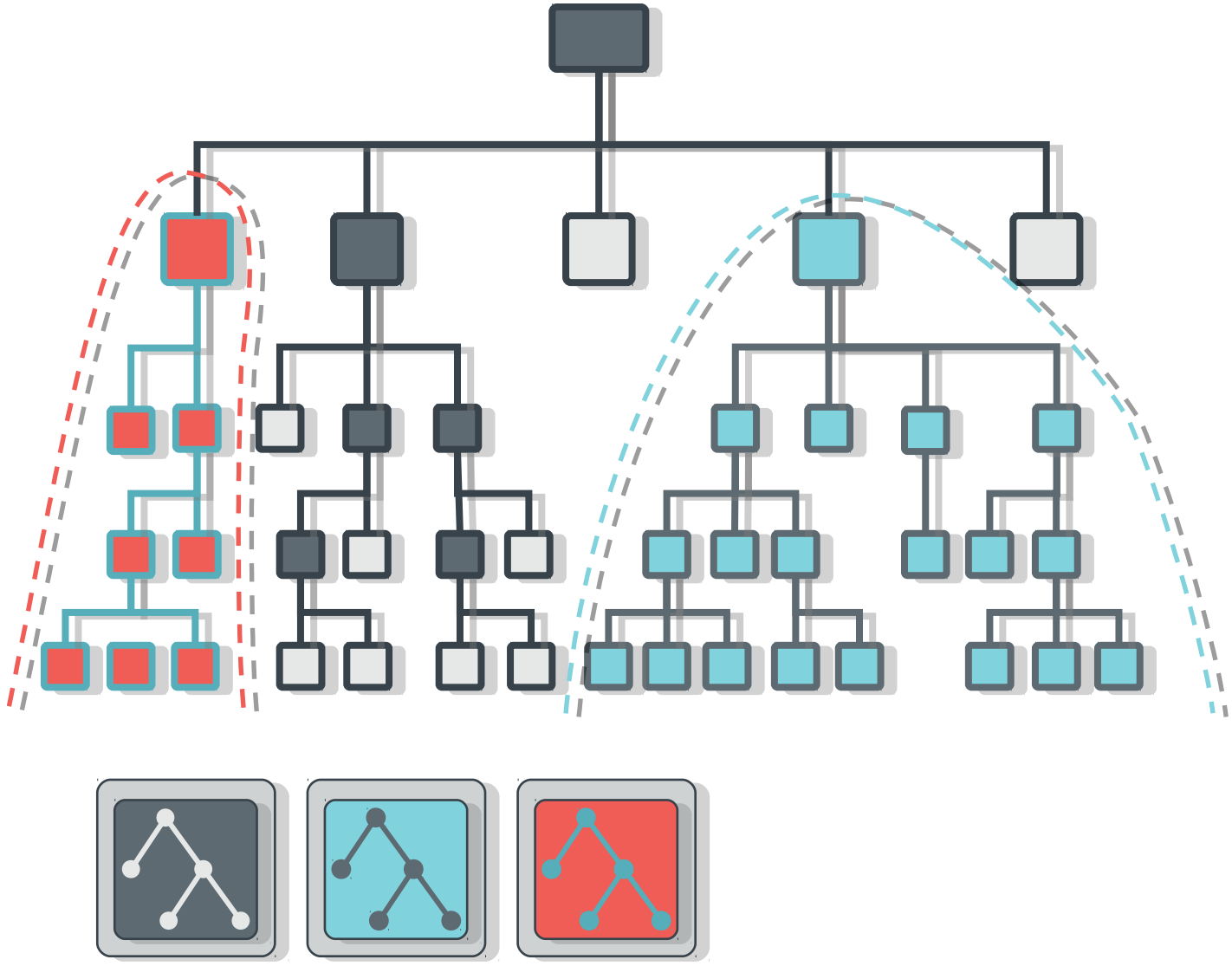


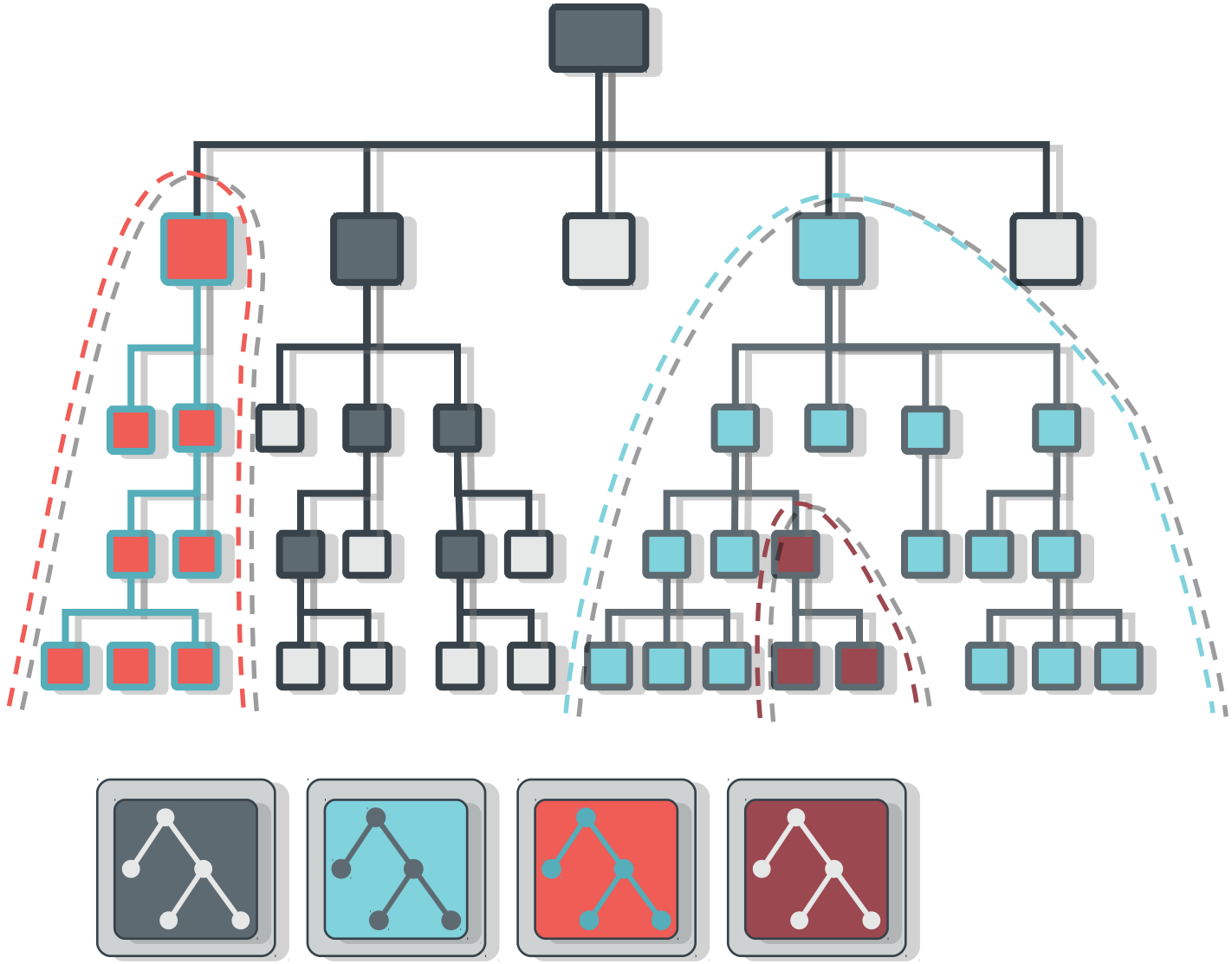
??

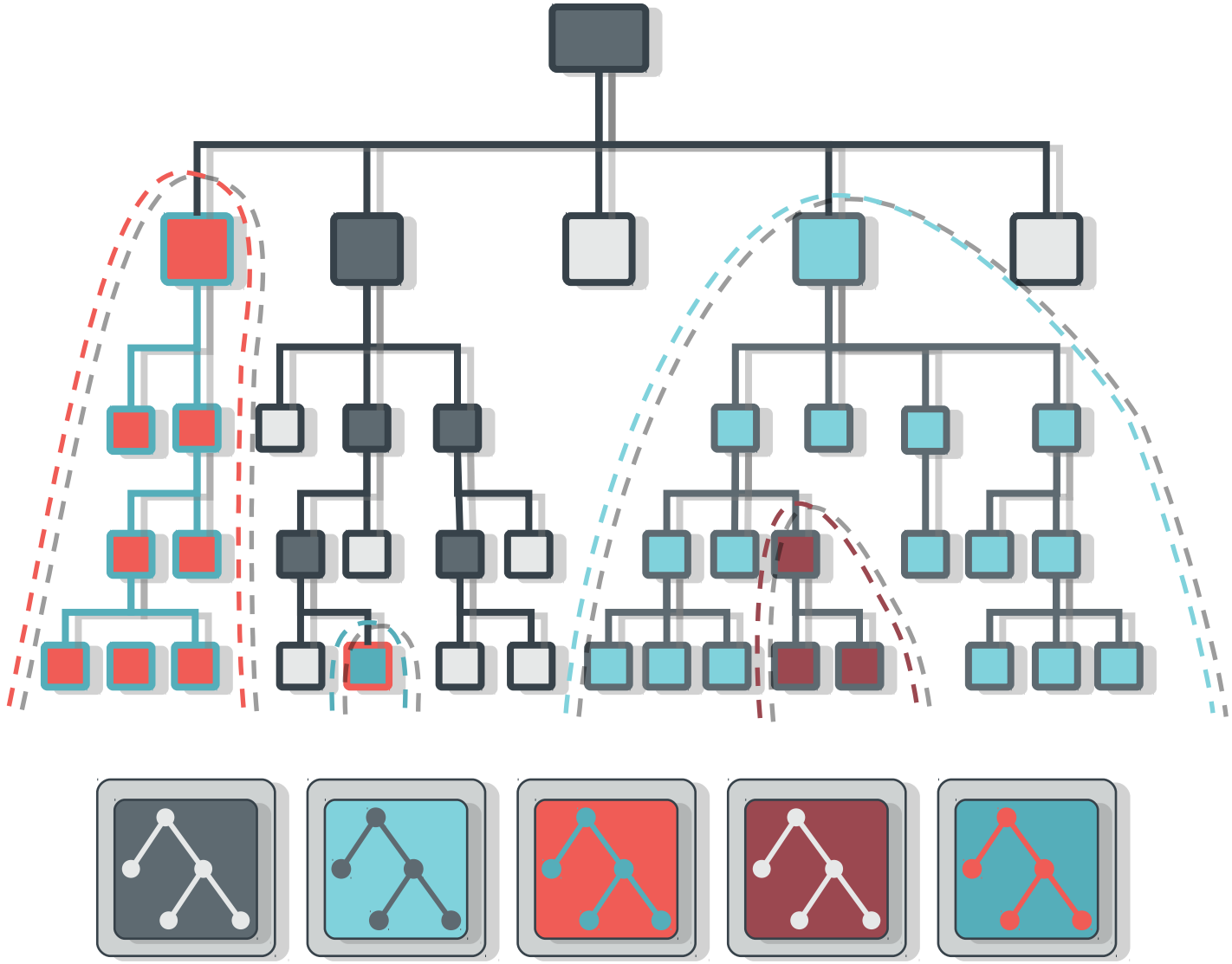












DYNAMIC SUBTREE PARTITIONING

WHAT NEXT?

WHAT CLIENT PROTOCOL?

- Prototype client was FUSE-based
  - Slow, some cache consistency limitations
- Considered [p]NFS
  - Abandon ad hoc client/MDS protocol and use a standard?
  - Avoid writing kernel client code?
- pNFS would abandon most of the MDS value
  - Dynamic/adaptive balancing, hot spot mitigation, strong fine-grained coherent caching
- Built native Linux kernel client
  - Upstream in ~2.6.36

FOSS >> OPEN STANDARDS

- Open source client *and* server
- Unencumbered integration
  - Linux, Qemu/KVM
- No need to adopt standard legacy protocols
  - iSCSI, NFS, CIFS are client/server
- Lesson:
  - standards critical for proprietary products
  - offer no value to end-to-end open solutions

- Intelligent OSDs can do more than read/write blocks
  - What else should they do?

INCUBATION (2007-2011)

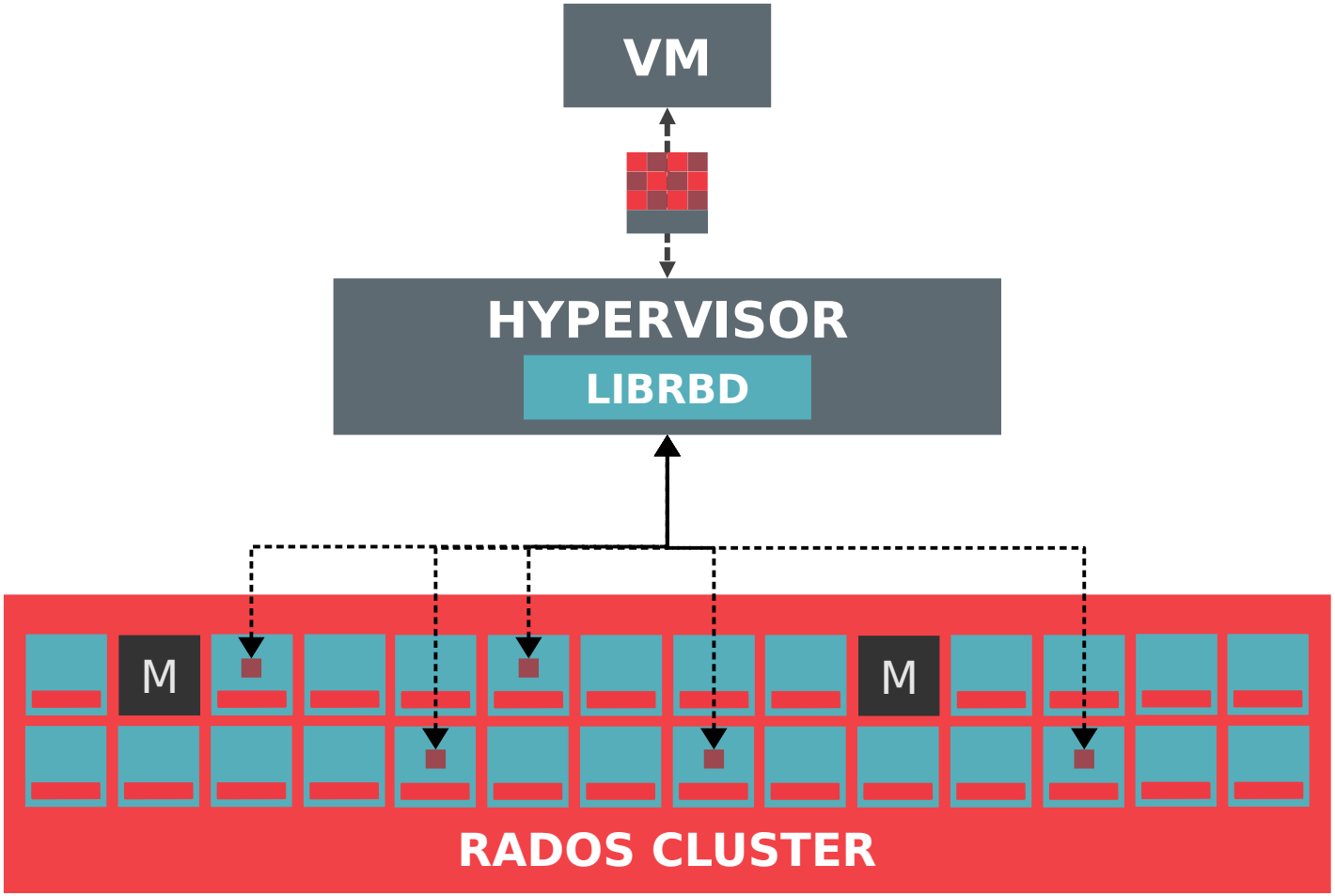
- Skunkworks project at DreamHost
  - Native Linux kernel client (2007-)
  - Per-directory snapshots (2008)
  - Recursive accounting (2008)
  - librados (2009)
  - radosgw (2009)
  - Object classes (2009)
  - strong authentication (2009)
  - RBD: rados block device (2010)

DreamHost®

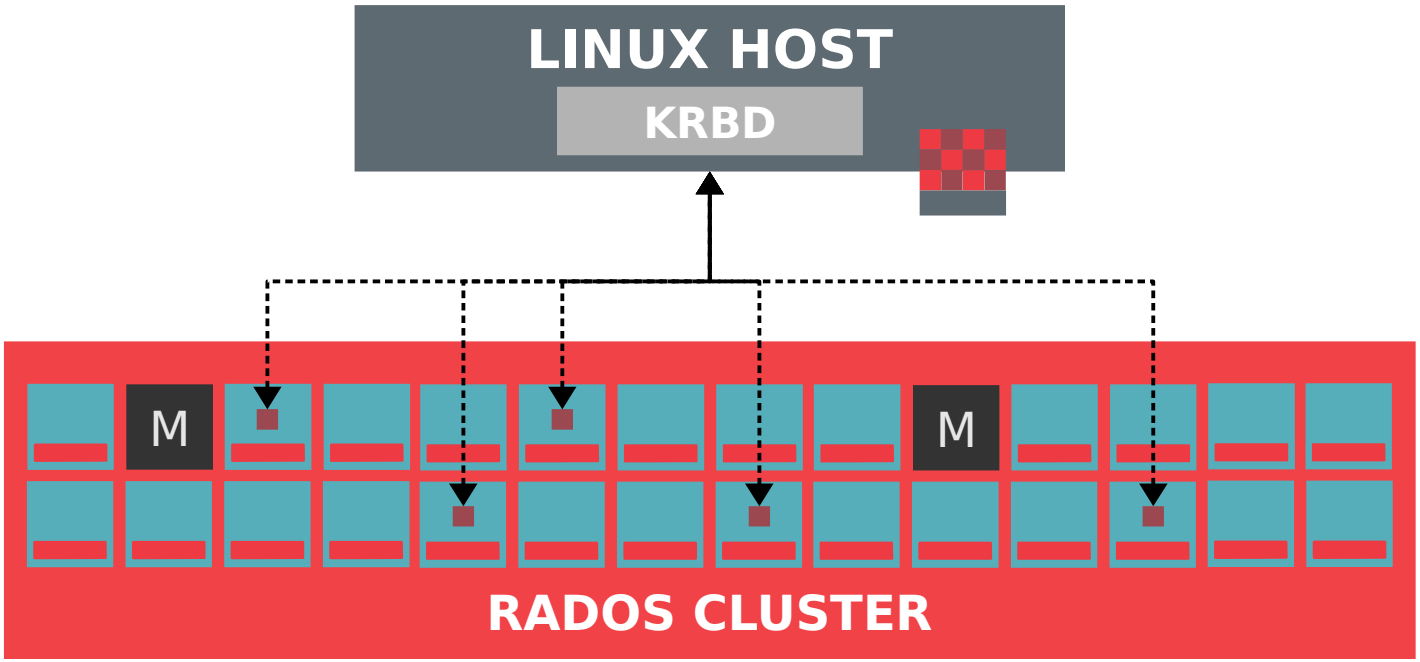
LINUX KERNEL SUPPORT

- Began attending LSF (Linux Storage and File systems) workshops
- Hear stories about early attempts to upstream Lustre
- Engage community with their processes
- Eventually merged into mainline in 2.6.34

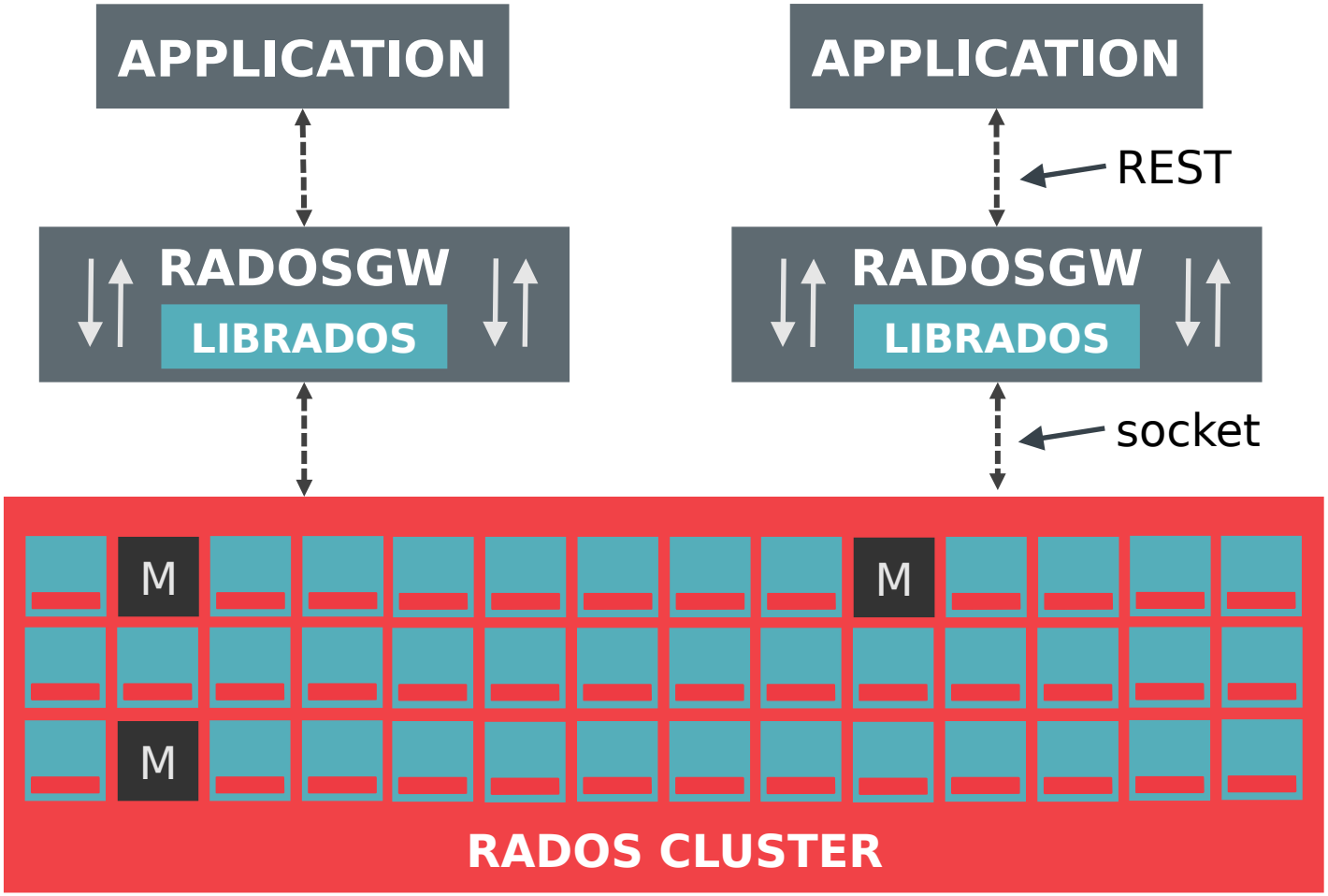
RBD – VIRTUAL BLOCK DEVICES



RBD KERNEL MODULE



THE RADOS GATEWAY



RADOS OBJECTS

- Flat object namespace within logical pools
- Rich data model for each “object”
  - Byte array
  - Attributes (small inline key/value data)
  - Bulk key/value data
- Mutable objects
  - Partial overwrite of existing data
- Single-object “transactions” (compound operations)
  - Atomic reads or updates to data and metadata
  - Atomic test-and-set, conditional updates

RADOS CLASSES

- “Objects” in the OOP sense of the word (data + code)
- RADOS provides basic “methods”
  - Read, write, setattr, delete, ...
- Plugin interface to implement new “methods”
  - Via a dynamically loaded .so
- Methods executed inside normal IO pipeline
  - Read methods can accept or return arbitrary data
  - Write methods generate an update transaction
- Moving computation is cheap; moving data is not

RADOS LUA CLASS

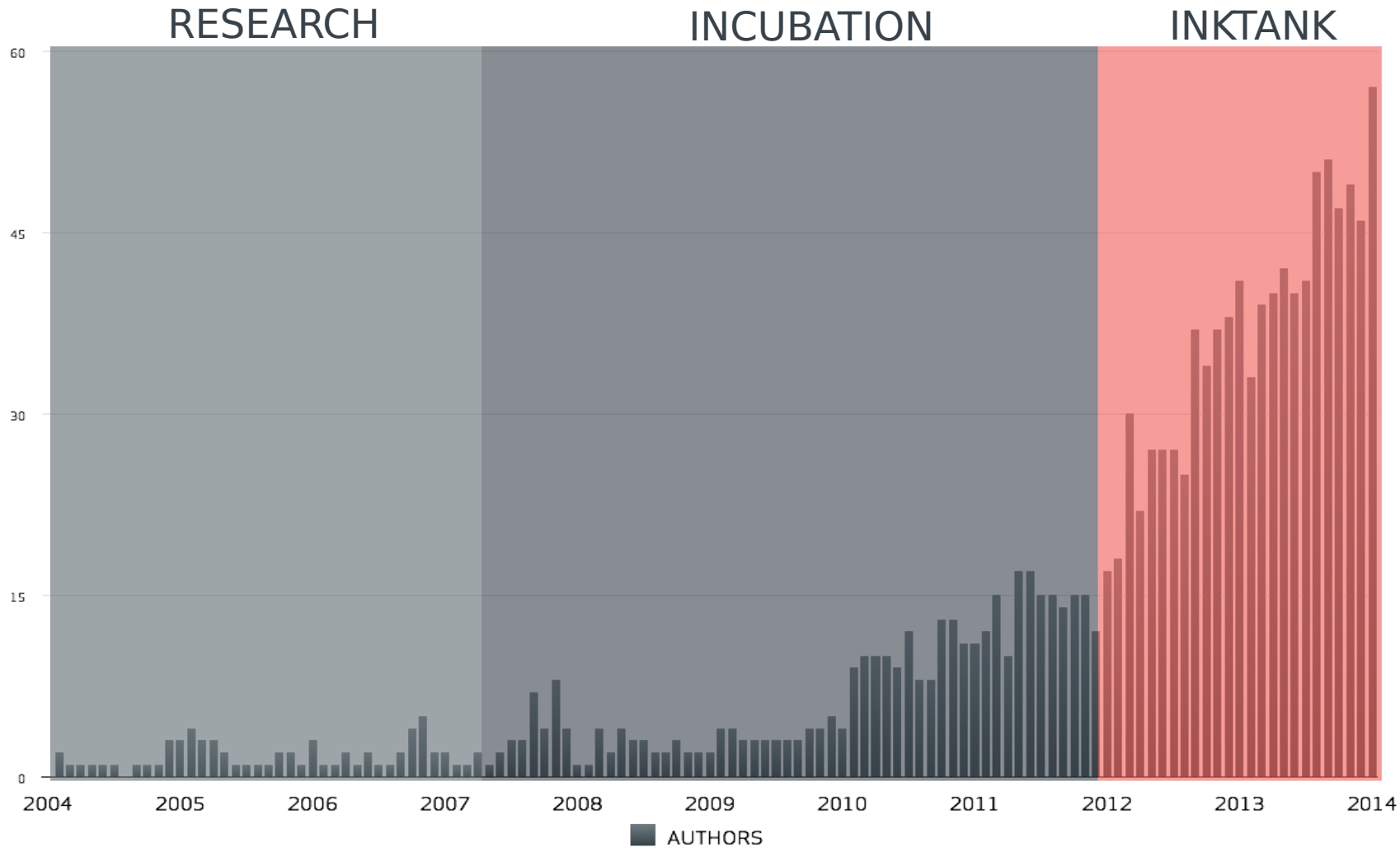
- Noah Watkins (UCSC)
- RADOS class links embedded LUA interpreter
- Clients can submit arbitrary script code
- Simple execution environment
  - Can call existing methods (like read, write)
- Caches compiled code

INKTANK (2012-2014)

- Spinout in 2012
  - DreamHost a poor fit to support open source software
  - Funding from DreamHost, Mark Shuttleworth
- Productize Ceph for the enterprise
  - Focus on stability, testing automation, technical debt
  - Object and block “Cloud” use-cases
- Real users, real customers

inktank

CONTRIBUTORS / MONTH



HPC?

- Lustre works
- Lustre hardware model a poor match for Ceph
  - Redundancy within expensive arrays unnecessary
  - Ceph replicates or erasure codes across devices
    - More disks, cheaper hardware
  - Ceph uses NVRAM/flash directly (not buried in array)
- ORNL experiment
  - Tune Ceph on OSTs backed by DDN array
  - Started terrible; reached 90% of theoretical peak
  - Still double-writing, IPoIB, ...
  - Inefficient HW investment

LINUX?

- Did kernel client investment engage Linux community?
  - Not really
  - Developers have small environments
- Red Hat bought Gluster Inc.
  - CephFS not stable enough for production
- Canonical / Ubuntu
  - Pulled Ceph into supported distro for librbd
  - Mark Shuttleworth invested in Inktank

THE CLOUD

- OpenStack mania
- Inktank focus on object and block interfaces
  - Start at bottom of stack and work up
  - Same interfaces needed for IaaS
- Helped motivate Cinder (block provisioning service)
  - Enable support of RBD image cloning from Cinder
  - No data copying, fast VM startup
- Ceph now #1 block storage backend for OpenStack
  - More popular than LVM (local disk)
- Most Inktank customers ran OpenStack
- Lesson: find some bandwagon to draft behind

RED HAT

- Red Hat buys Inktank in April 2014
  - 45 people
  - $190MM
- OpenStack

CHALLENGES

SUPPORTABILITY

- Distros
  - Ubuntu 12.04 LTS at Inktank launch
- Dependencies
  - Leveldb suckage – reasonably fast moving project, distros don't keep up
- Kernels
  - Occasionally trigger old bugs
- Rolling upgrades
  - Large testing matrix
  - Automation critical
- Lesson: not shipping hardware makes QA & support harder

USING DISK EFFICIENTLY

- OBFS: simpler data model → faster
- Ebofs: userspace extent and btree-based object storage
  - Transaction-based interface
- Btrfs: how do expose transactions to userspace?
  - Start and end transaction ioctls
  - Pass full transaction description to kernel
  - Snapshot on every checkpoint; rollback on restart
  - Still need ceph-osd's full data journal for low latency
- XFS: stable enough for production
  - Need journal for basic atomicity and consistency
- Lesson: interfaces can tend to clean and respectable
  - ...but implementations generally do not

MAKING IT WORK AT SCALE

- Goal: manage to a steady state
  - Declare desired state of system; components move there
  - System may never be completely “clean”
- Dynamic / emergent behaviors
  - Various feedback loops in autonomic systems
  - Equilibrium may be unstable
- Lesson: importance of observability
  - Convenient state querying, summaries
- Lesson: operator intervention
  - Need ability to suspend autonomic processes

ENTERPRISE

- Ecosystem
  - Ubuntu dominated early OpenStack
  - RHEL/CentOS dominate enterprise
- Vendor needs a compelling product
  - Simple support on open code is a difficult model
  - Conundrum: better software → reduces product value
  - Engineering expertise is necessary but not sufficient
- Inktank Ceph Enterprise
  - Management layer, GUI (proprietary add-ons)
  - Enterprise integrations (SNMP, VMWare, Hyper-V)
- Legacy
  - Back to talking about iSCSI, NFS, CIFS (as gateway drug)

COMMUNITY BUILDING

- User community
  - Huge investment in making things easy to deploy
  - Documentation
  - Hand-holding over email, IRC
- Developer community
  - Forcing tight developer team to use open processes
  - Email, IRC, public design and code review
- Ceph Developer Summits
  - 100% online Google hangout, IRC, wiki
  - Every few months
- Lessson: developers need employers; partners matter

LOOKING FORWARD

PERFORMANCE

- Have demonstrated Ceph works; now users would like it to be faster
- Polish internal APIs; replace original implementations
  - OSD backend (XFS + leveldb)
  - Message passing interface
- Modularity helpers new developers engage
- Critical mass of developer community stakeholders
  - Intel, Mellanox, Fujitsu, UnitedStack
  - Challenge is in shepherding efforts

NEW HARDWARE COMING

- Flash and NVRAM for high IOPS
  - Locking and threading → improve parallelism
- Low-power processors for cold storage
  - Limit data copies, CRC → reduce memory bandwidth

- Challenge: remain hardware agnostic
  - Keep interface general

- Lesson: LGPL is great for infrastructure software

ETHERNET DISKS

- Ethernet-attached HDDs
  - On-board, general purpose ARM processors
  - Standard form factor, ethernet instead of SATA
  - Eliminate usual Intel-based host tier
- Seagate Kinetic
  - New key/value interfaces to move beyond block
  - Well-suited to new shingled drives
  - Strategy: define a new “standard” interface
- HGST open ethernet drives
  - General purpose Linux host on HDD
  - Standard block interface from host
  - Strategy: build ecosystem of solutions around an open disk architecture
- Prediction
  - Hiding drive capabilities behind new APIs will limit innovation, adoption
  - Opportunity to leverage existing “software defined” platforms

MULTIPLE VENDORS

- Avoiding vendor lock-in resonates with users
- Hardware vendor independence
  - Architect system for commodity hardware
  - Customers can buy piecemeal or full solutions
- Open source → software vendor independence
  - Code is free (as in speech and beer)
- Need credible competitors
  - Linux: Red Hat, SUSE, Canonical
  - Ceph: Red Hat, ?
- Lesson: being too successful undermines your value prop

ACADEMIA → FOSS PIPELINE

- Incredible innovation in graduate programs
- Most academic work based on open platforms
- Very little work survives post-thesis to become free or open source software
- I see three key problems
  - Lack of engagement and education about FOSS communities
  - Pool of employers are dominated by non-free software vendors
  - Gap between prototype code that is typical at thesis stage and the production quality needed for paying users or venture investors

PARTING THOUGHTS

- Ceph is awesome.
- Building a successful community around open source technology is just as challenging as the technology.
- Successful business model (and business environment) is a huge catalyst to driving community.
- Sacrificing software freedoms to enable the business opportunity is frequently tempting, but unnecessary.

THANK YOU!

Sage Weil
CEPH PRINCIPAL ARCHITECT


sage@redhat.com


@liewegas