DATA TRANSFERS, STORAGE SYSTEMS AND CONFIGURATIONS

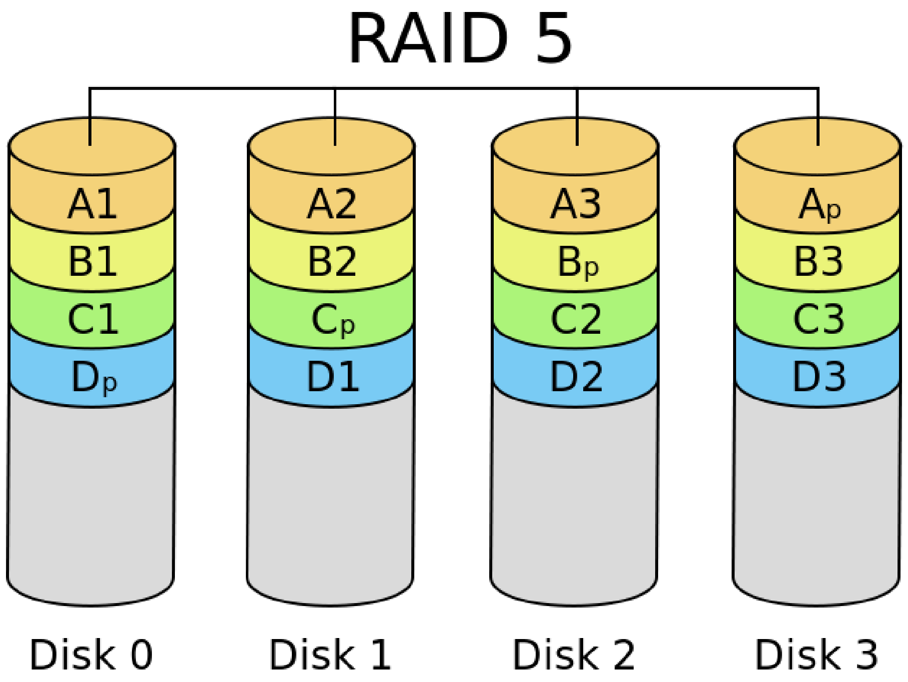
RON TROMPERT

Introduction

- A number of years ago:
  - Servers with a hardware raid controller and disks in the same box or in a separate storage array
  - Everyone used RAID5 to cope with disk failure

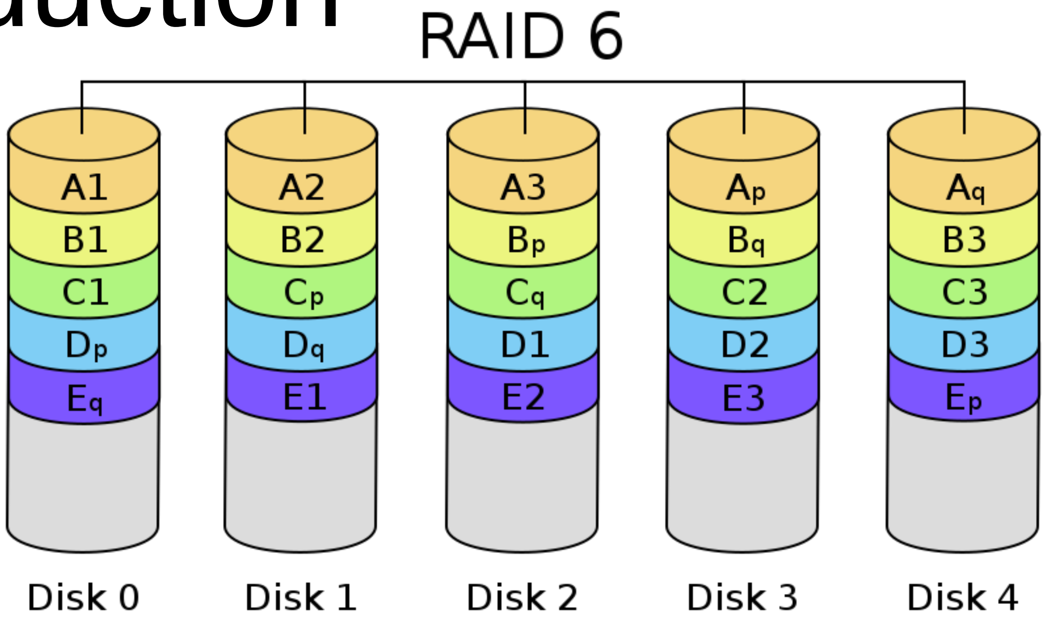
Introduction

- Store extra parity bits
- In case of a single disk failure, use parity bits on other disk to reconstruct the contents of the failed disks → rebuilding
- H/W RAID controller waits for the broken disk to be replaced
- After disk is replaced, the RAID controller starts rebuilding
- In the period between disk failure and the end of the rebuild proces no disk failure can be tolerated



Introduction

- But then disks got bigger and bigger and rebuilding took longer and longer
- Chances increased of a second disk failure before the end of a rebuild
- No problem->RAID6
- Store two extra parity bits
- Now two disks can fail
- In case of disk failure, use parity bits on other disks to reconstruct the contents of the failed disk → rebuilding



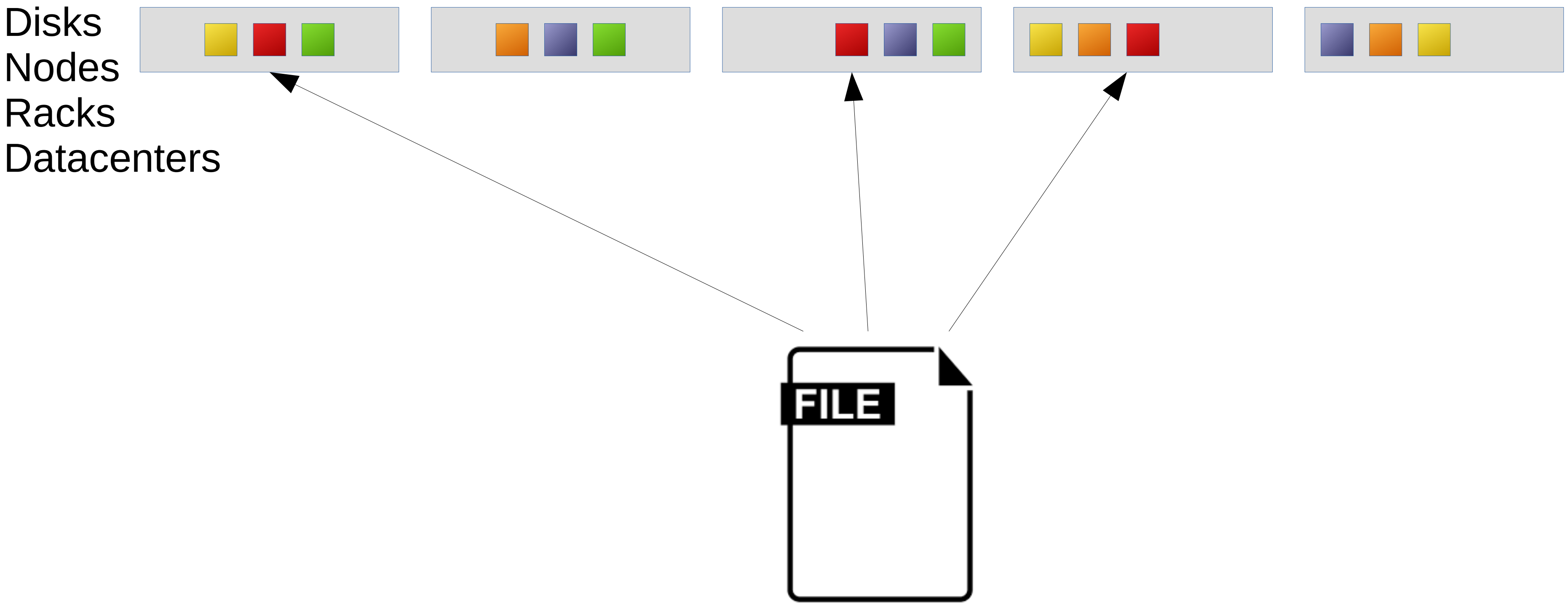
Introduction

- But the disks are still getting bigger and bigger and rebuilding still takes longer and longer
- So even rebuilding for RAID6 is going to take too long
- What do we do now?

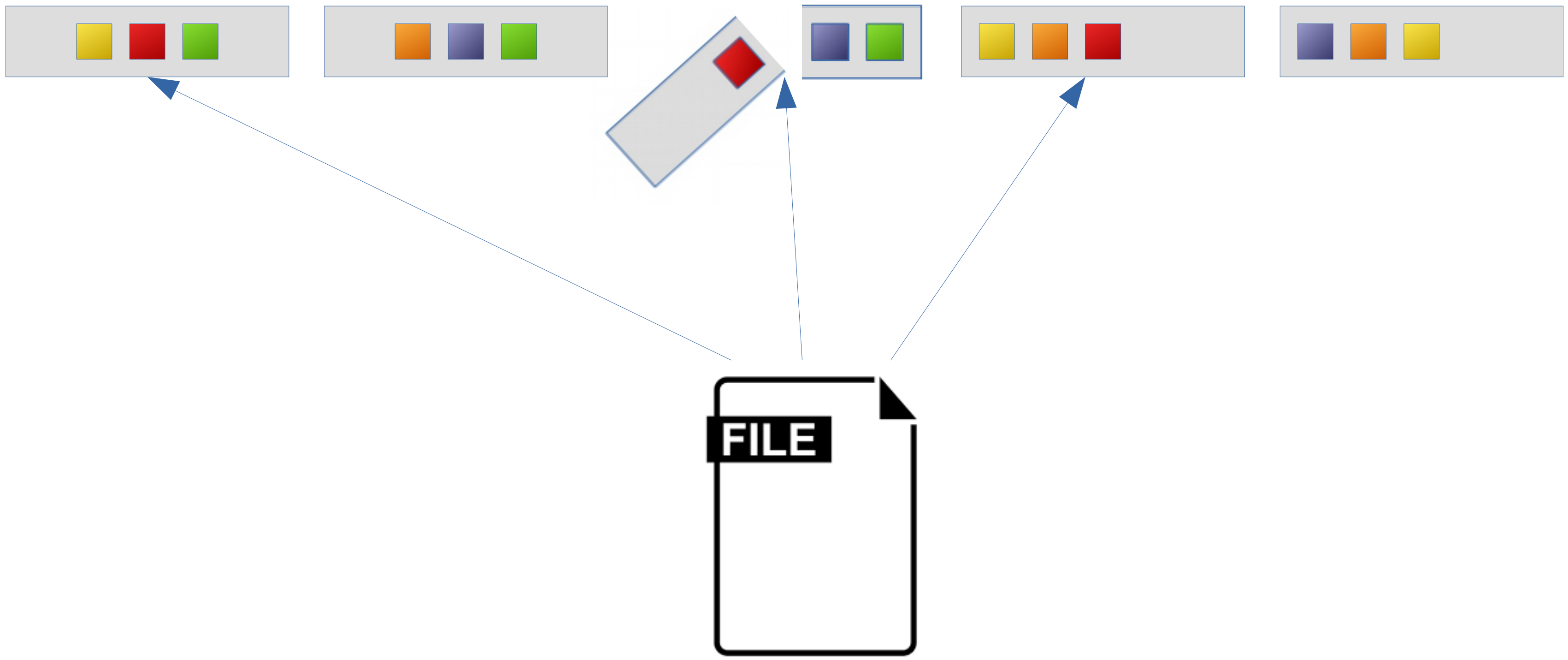
Data durability

- Redundancy not in single RAID set but distributed over nodes in a cluster

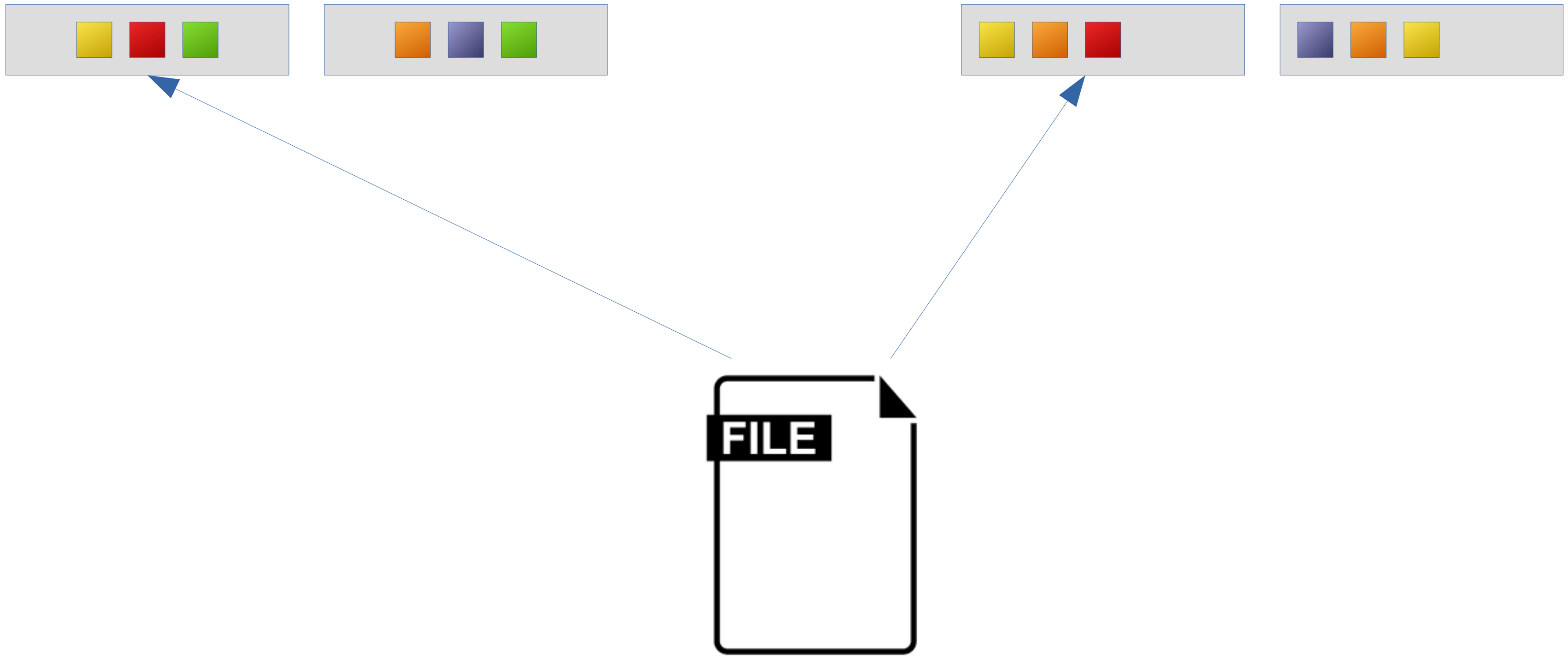
Data durability



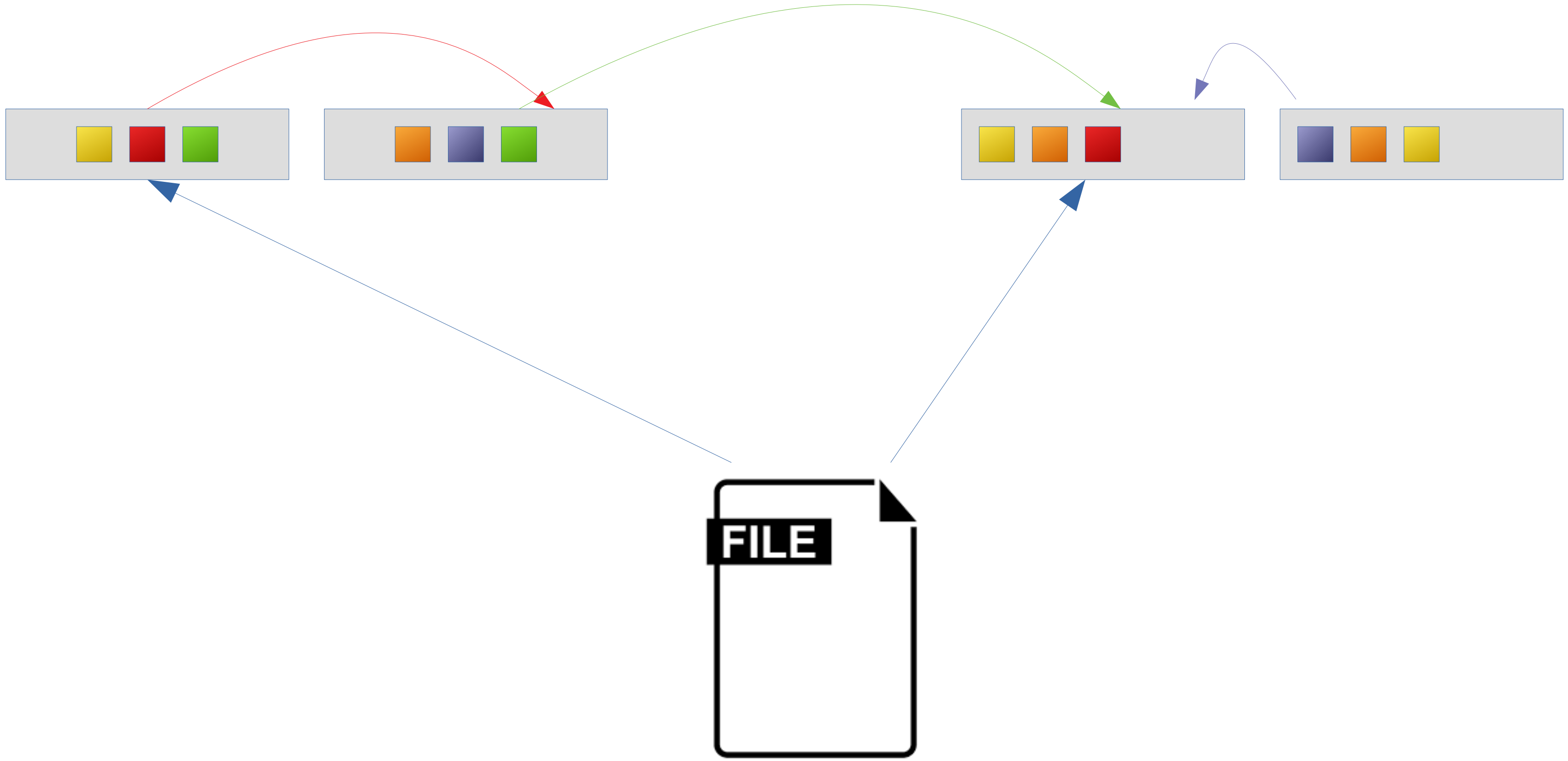
Data durability



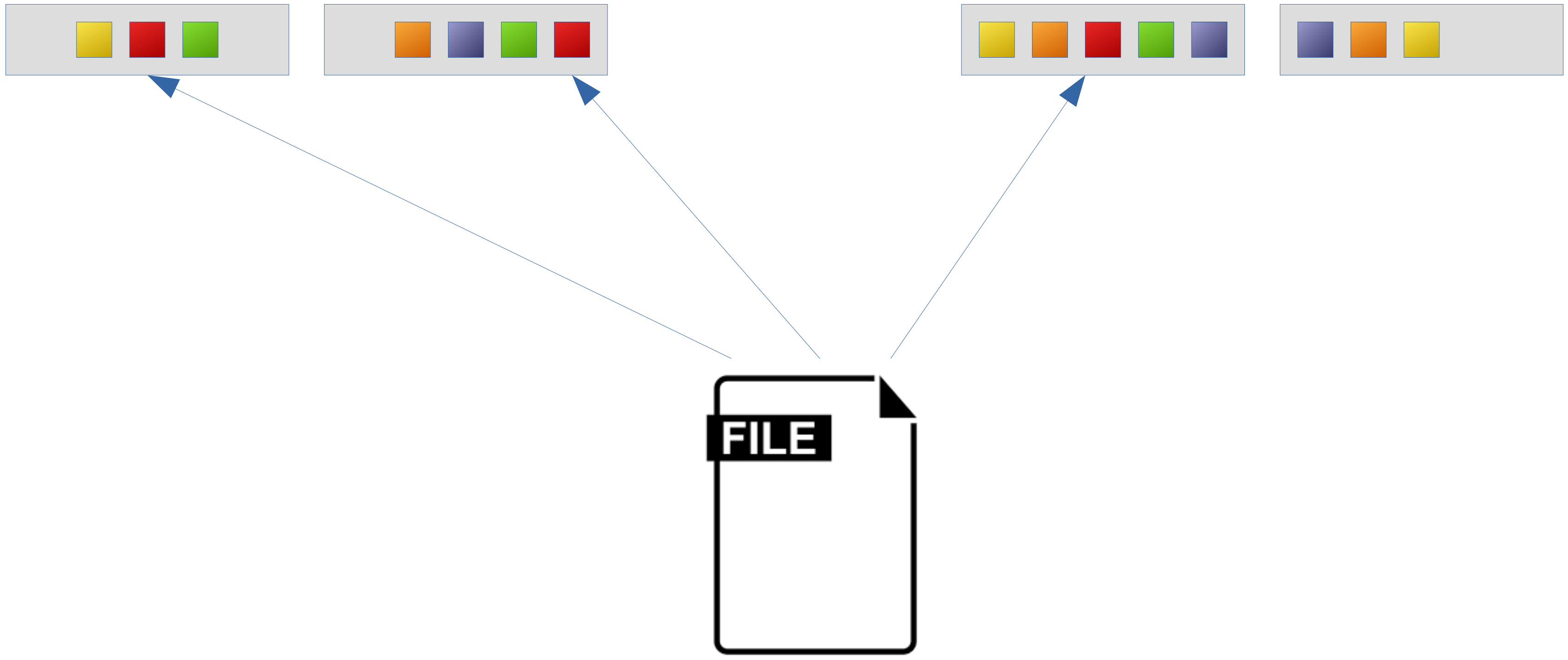
Data durability



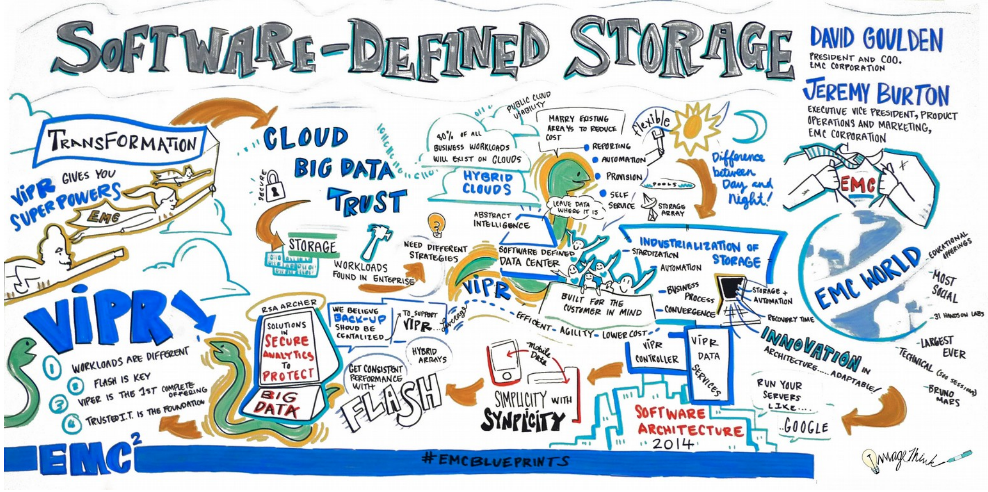
Data durability



Data durability



Data durability



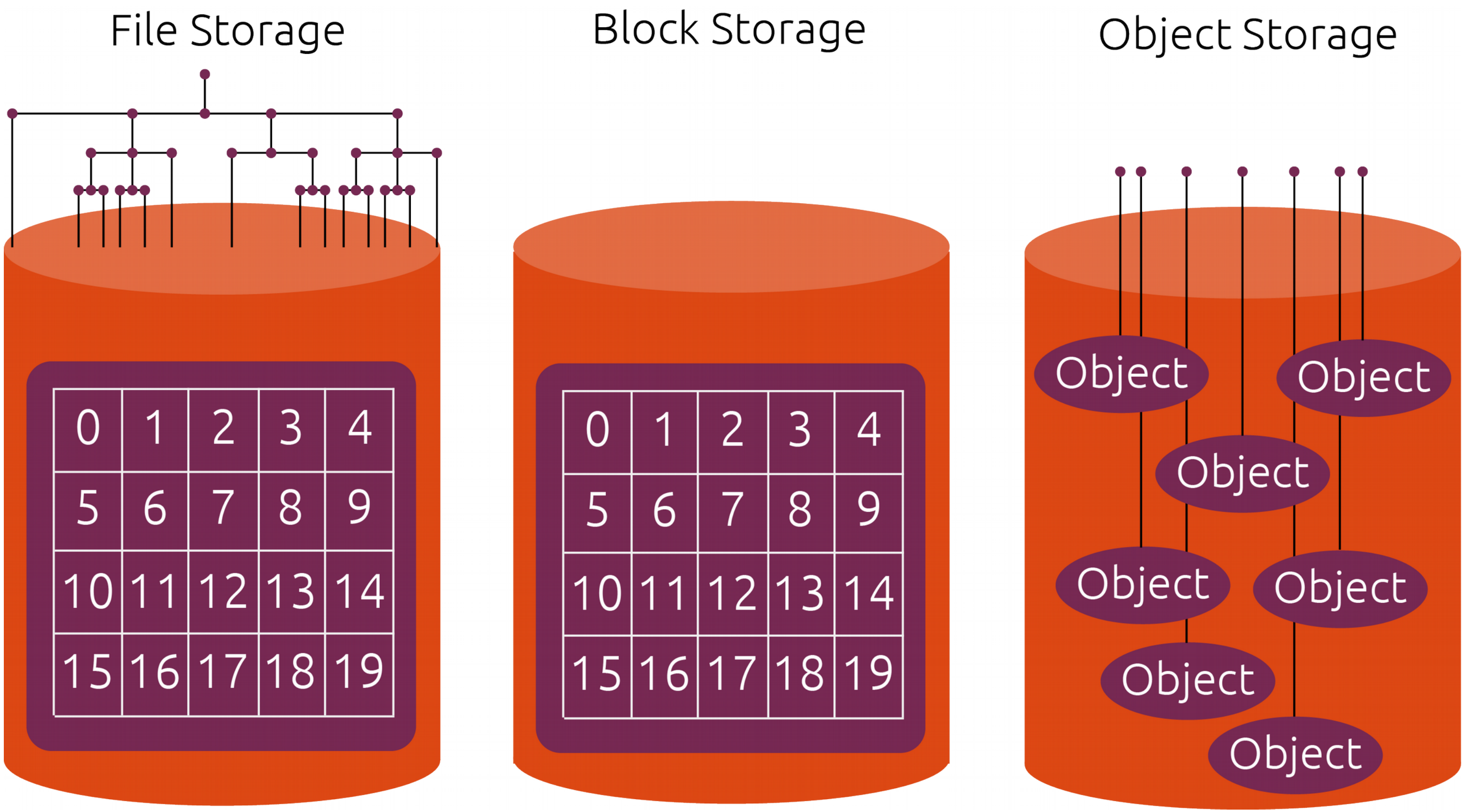
- Software Defined Storage
  - Auditting processes
  - Failures are handled by the storage system itself. No manual intervention required. Saves a lot of effort :)
  - No individual disk or node is responsible for the durability of the data
- CEPH, SWIFT, IBM Spectra Scale (declustered array), Netapp StorageGrid, Huawei OceanStor 9000, FUJITSU Storage ETERNUS CD10000 S2,…..
- Currently testing dCache with CEPH storage backend


GLORY
GLORY COLLISION
GLORYKICKBOXING.COM
WATCH PPV LIVE
DEC 10TH
SWIFT VS CEPH
$29.95 4PM ET / 1PM PT
UFC.TV/GLORYCOLLISION
UFC TV

CAP theorem

- CAP theorem (Eric Brewer)
  - Consistency
    (all nodes see the same data at the same time)
  - Availability
    (every request receives a response about whether it succeeded or failed)
  - Partition Tolerance
    (the system continues to operate despite arbitrary partitioning due to network failures)
- You can get only 2 out of 3
- SWIFT drops consistency to get availability, partition tolerance
- CEPH drops availability to get consistency and partition tolerance

Types of storage



SWIFT & CEPH

- SWIFT is only an object store and nothing else
- CEPH can be an object store (RGW), file-based storage (CEPHFS) and block storage (RBD)
- Both run on commodity hardware
- Both are Software Defined Storages
- Both have no SPOFs
- Both are self-healing

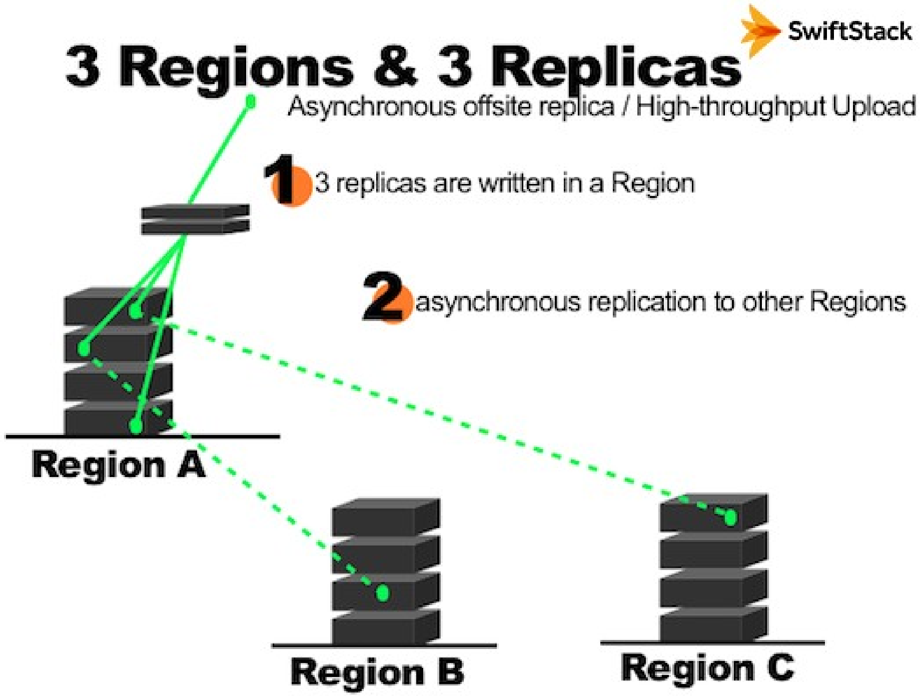
SWIFT

- CRUD – Create Read Update Delete
- Objects are accessible through an API (via URLs and https)
- Object locations as URLs for scalability of storage system
  - https://proxy.swift.surfsara.nl/v1/KEY_05b2aafab5a745eab2726d88649d95fe/mycontainer/myobject

SWIFT

- Unstructured data
  - Text, video, scientific data backups,websites,.....
- Highly available
- Eventual consistent
  - No transactional data
- Speaks its own SWIFT protocol and S3
- Massive scalability,
  SWIFT scales to the

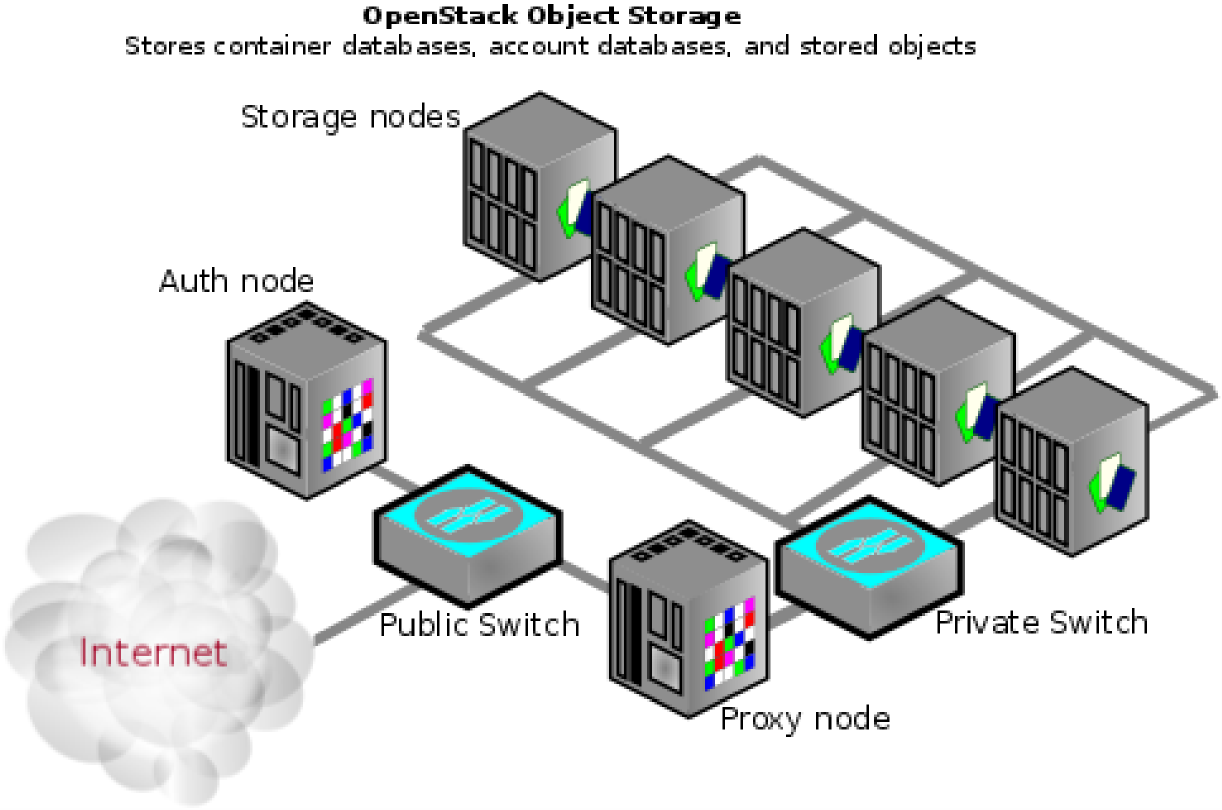
SWIFT


SwiftStack
3 Regions & 3 Replicas
Asynchronous offsite replica / High-throughput Upload
1 3 replicas are written in a Region
2 asynchronous replication to other Regions
Region A
Region B
Region C

SWIFT

- Single name space
- Geographically distributed

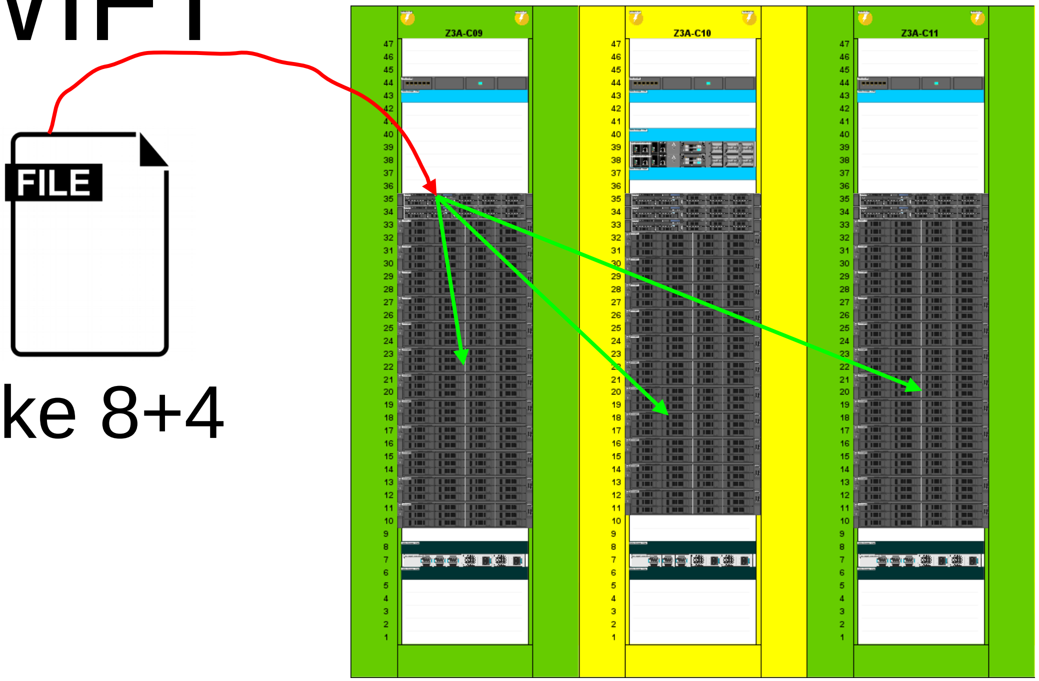
SWIFT



OpenStack Object Storage
Stores container databases, account databases, and stored objects
Storage nodes
Auth node
Internet
Public Switch
Proxy node
Private Switch

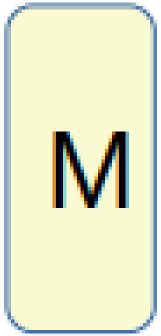
SWIFT

- Storage policies
  - 3 replica's →
  - Also Erasure coding like 8+4
  - SSDs/HDDs
  - Geographic location

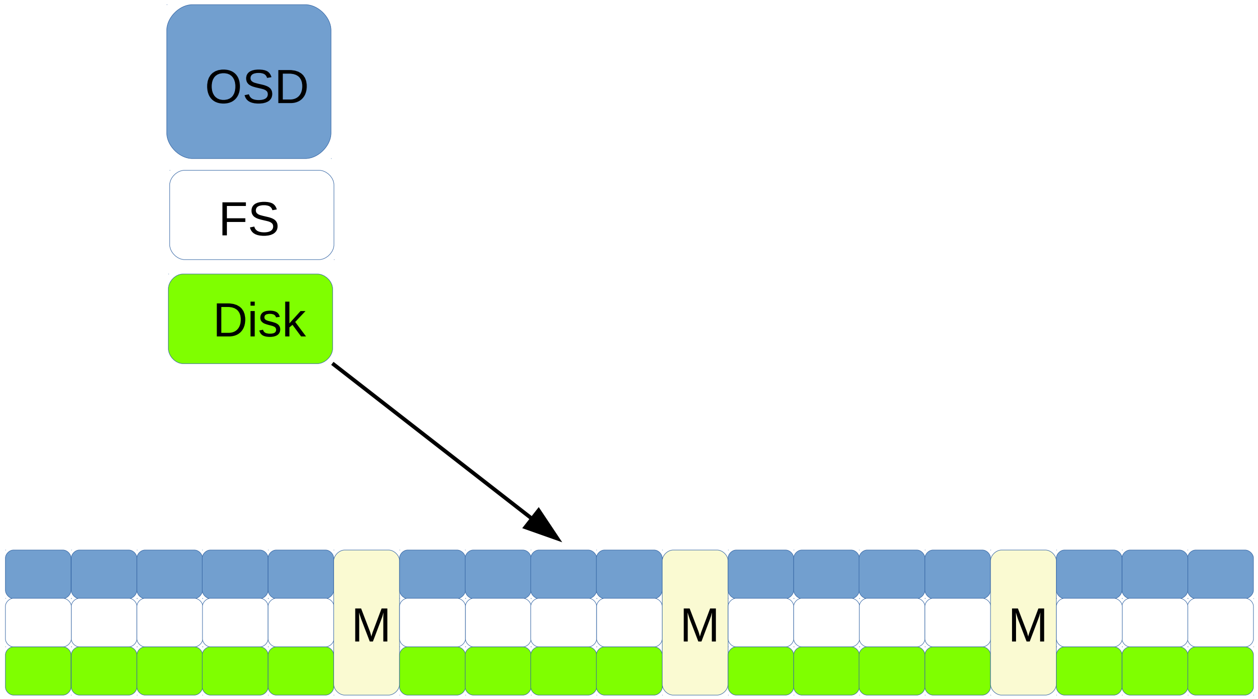


CEPH

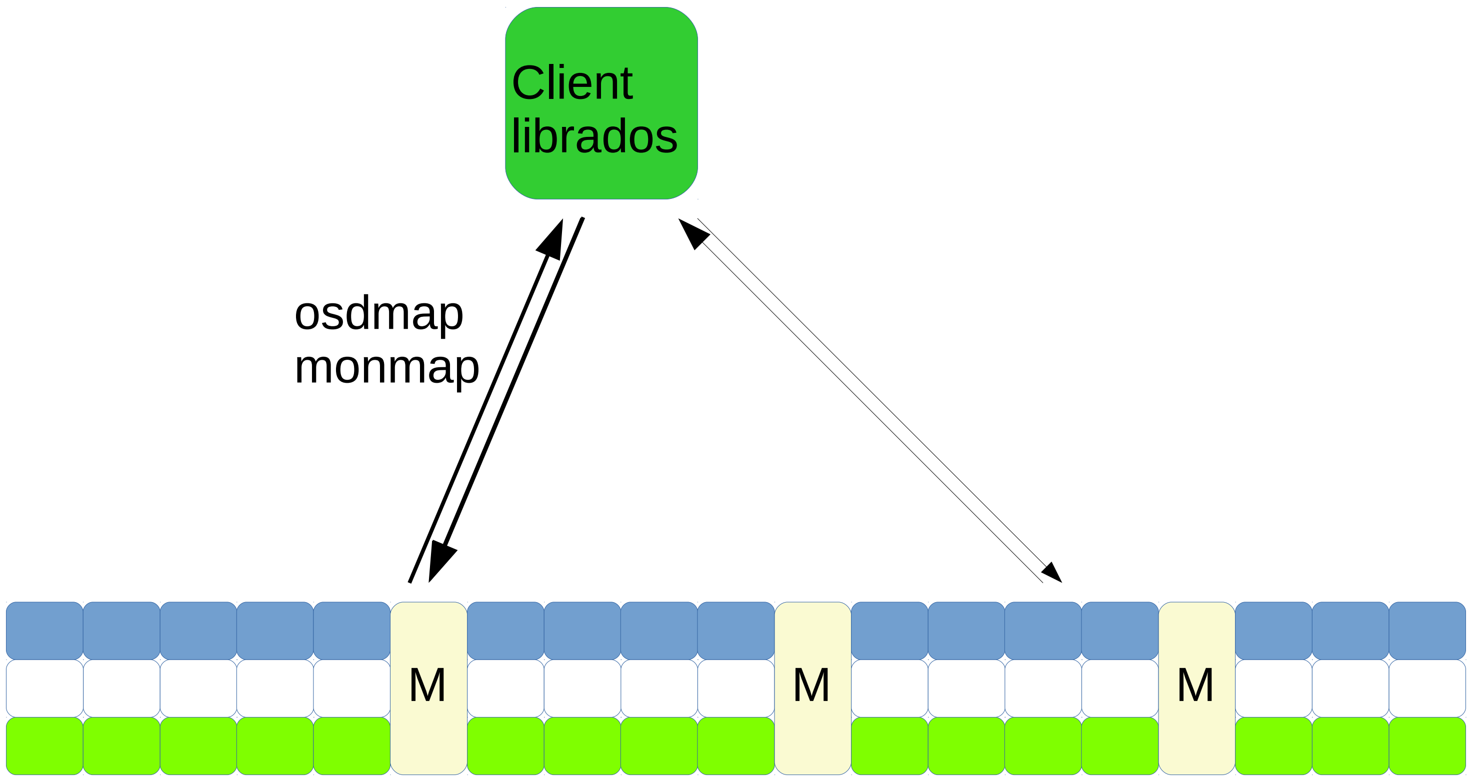
- CEPH components
  - Monitor/Manager
    - Management
    - Statistics
    - Consensus distributed decicion making
    - Cluster membership and state
    - Odd number
  - OSD
    - 1 per disk
    - Serves objects to clients
    - Replication and recovery



CEPH

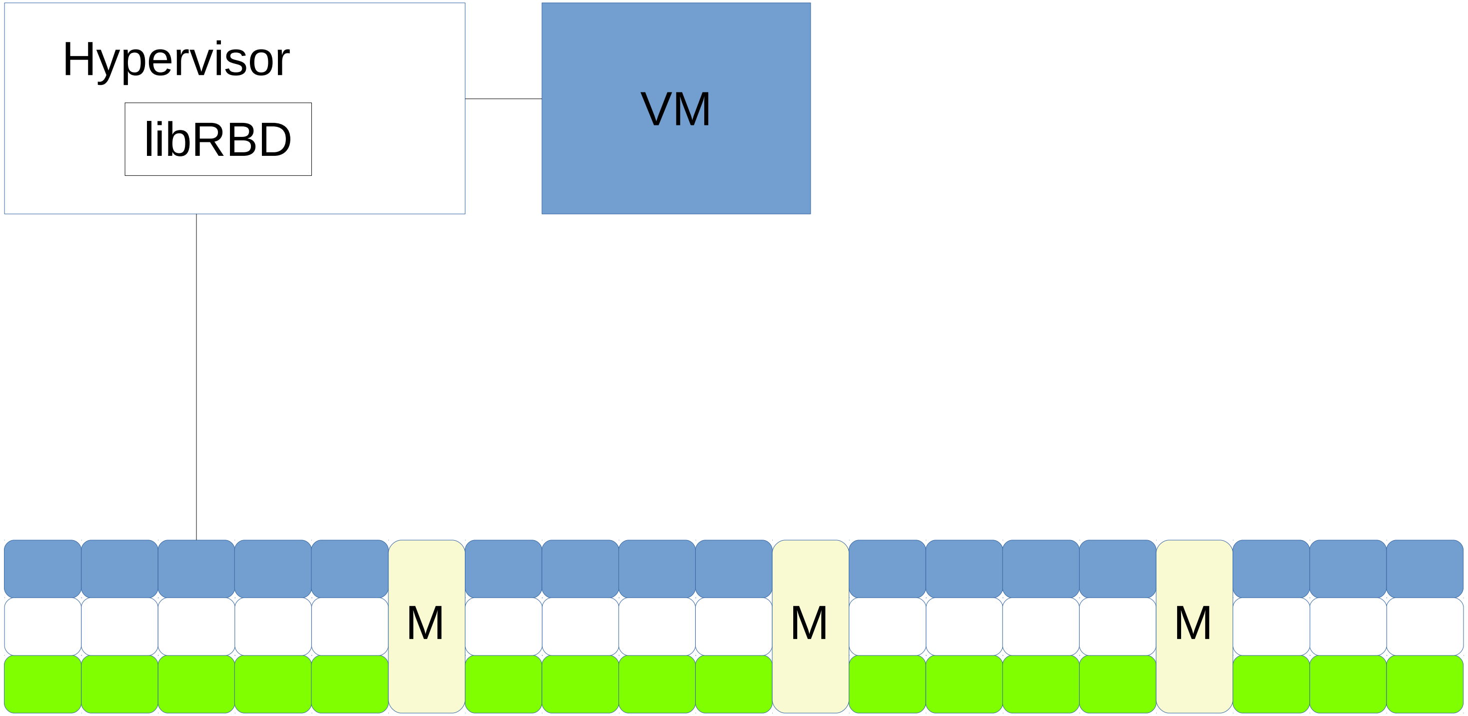


CEPH



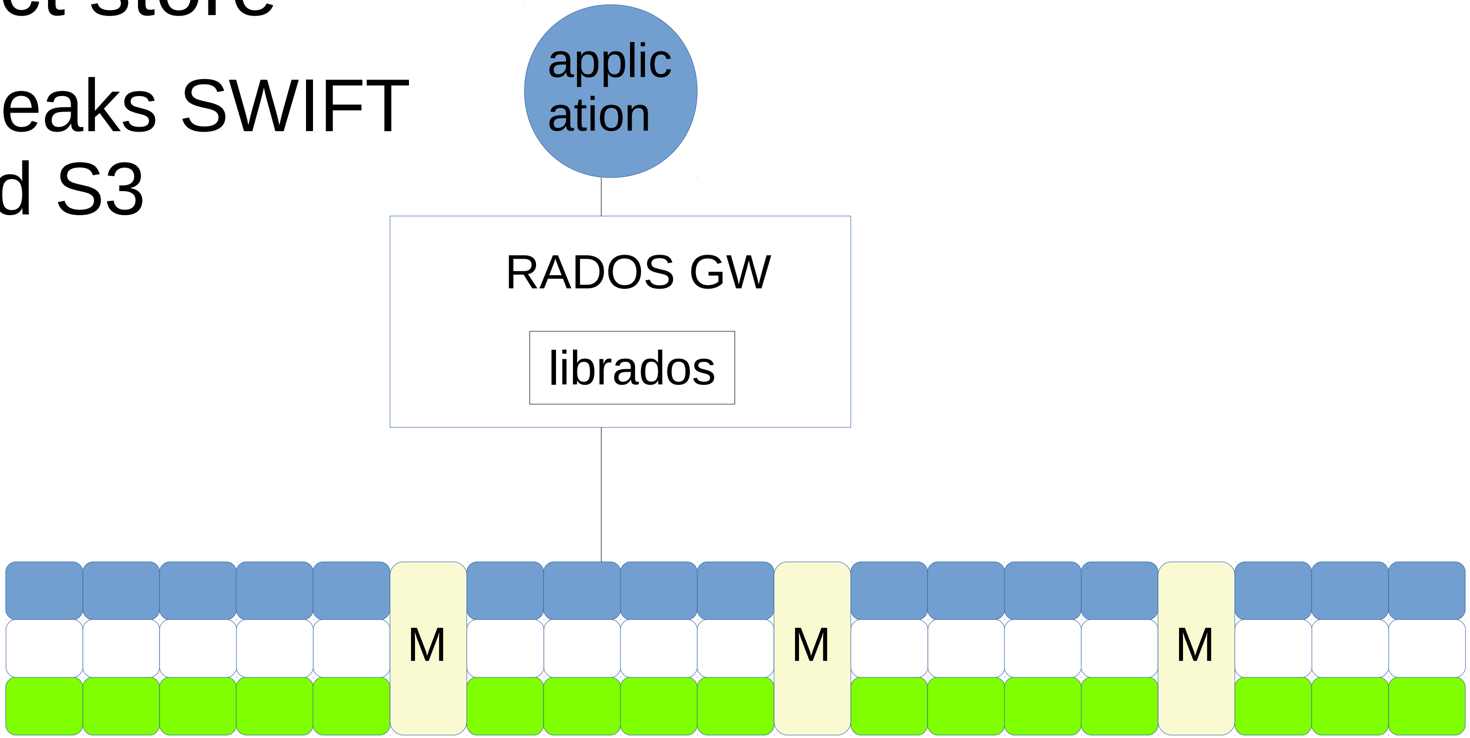
CEPH

- Block device



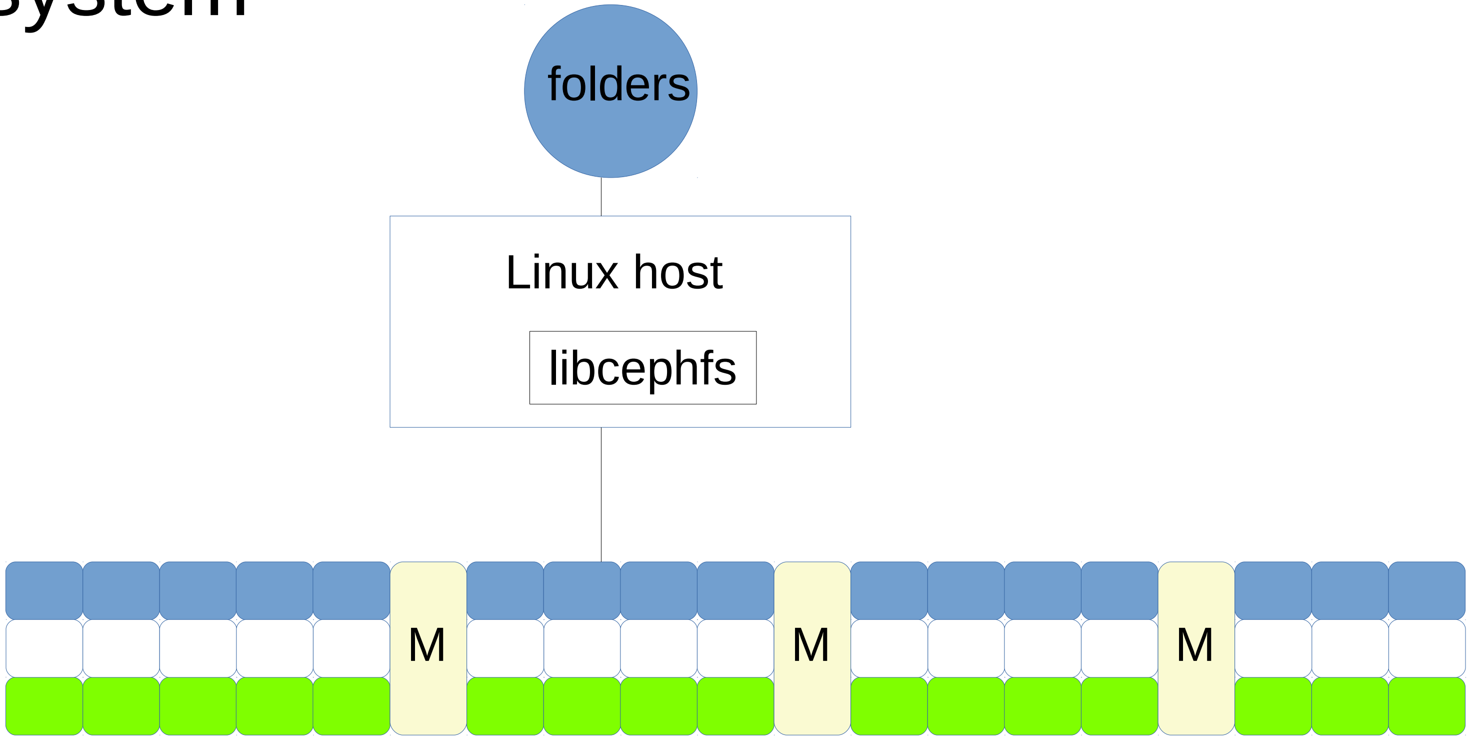
CEPH

- Object store
  - Speaks SWIFT and S3



CEPH

- File system



FTS

FTS

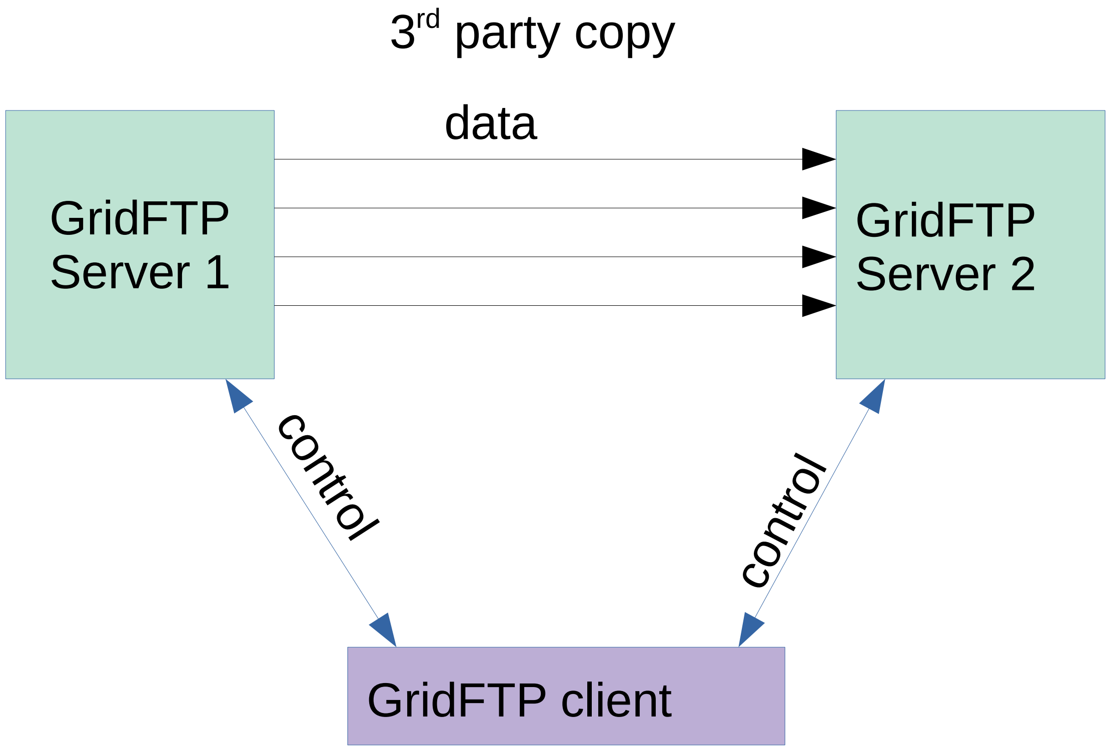
- Developed by CERN
- Responsible for distributing the majority of the data from the Large Hadron Collider over the world. Handles more than 35PB per month.
- Reliable bulk transfers of files between sites
- Zero configuration required :)

FTS

- Authentication by x509 proxy delegation
  - Credentials never leave your machine
  - SciTokens and  are underway.
- Multi protocols
  - Gridftp, WebDAV (HTTP 3rd party copy for dCache and DPM), XrootD, S3

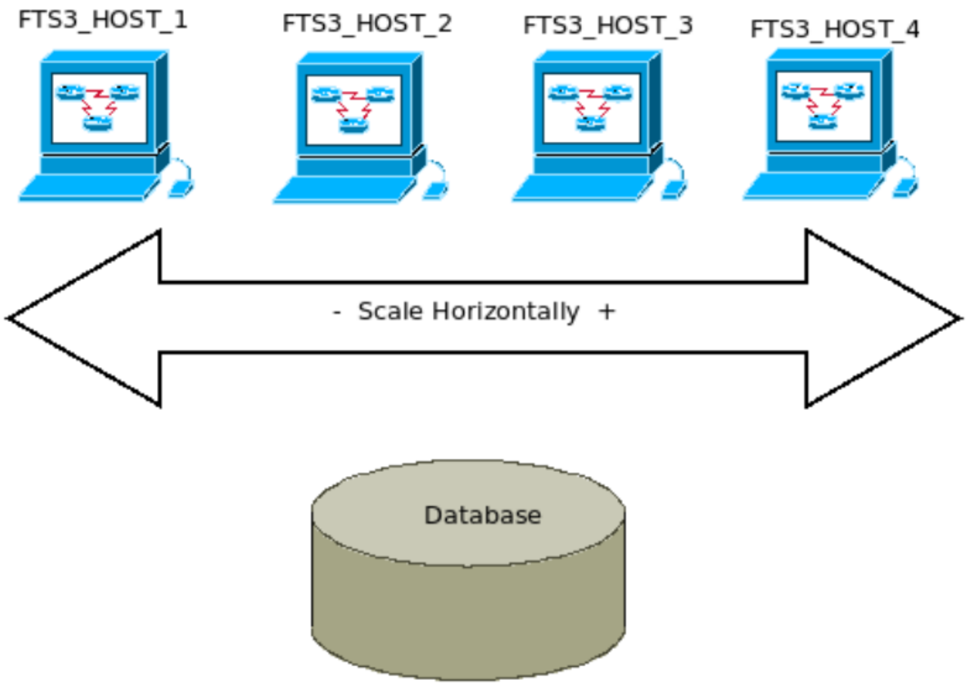
FTS

- Multi protocols
  - Gridftp
  - WebDAV (HTTP 3rd party copy for dCache and DPM)
  - XrootD
  - S3



FTS

- Horizontally scalable



FTS

- Session reuse
- Multihop transfers
- Interacts with tape archives
- Retries
  - Retry times, retry delay

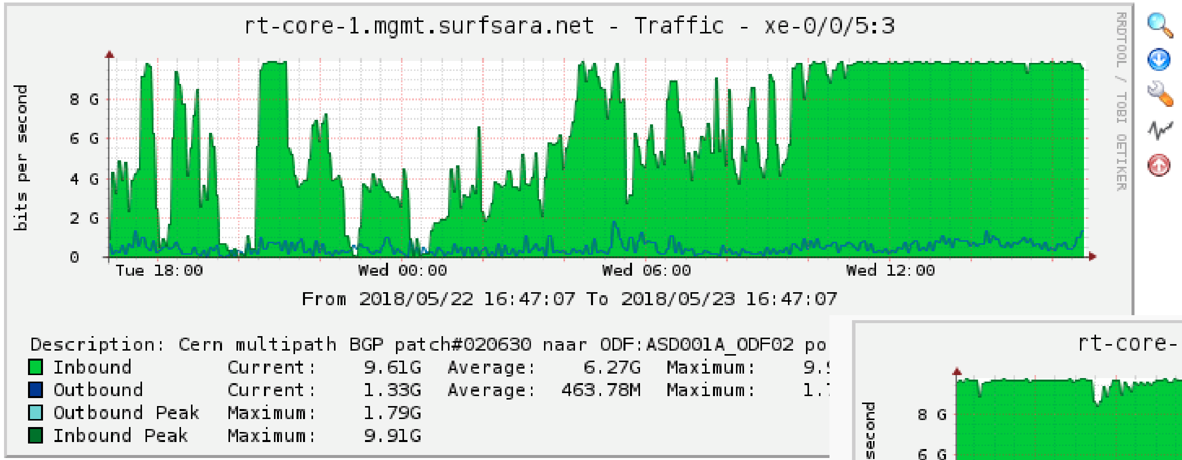
- REST API
- Python bindings
- CLI
- WebFTS

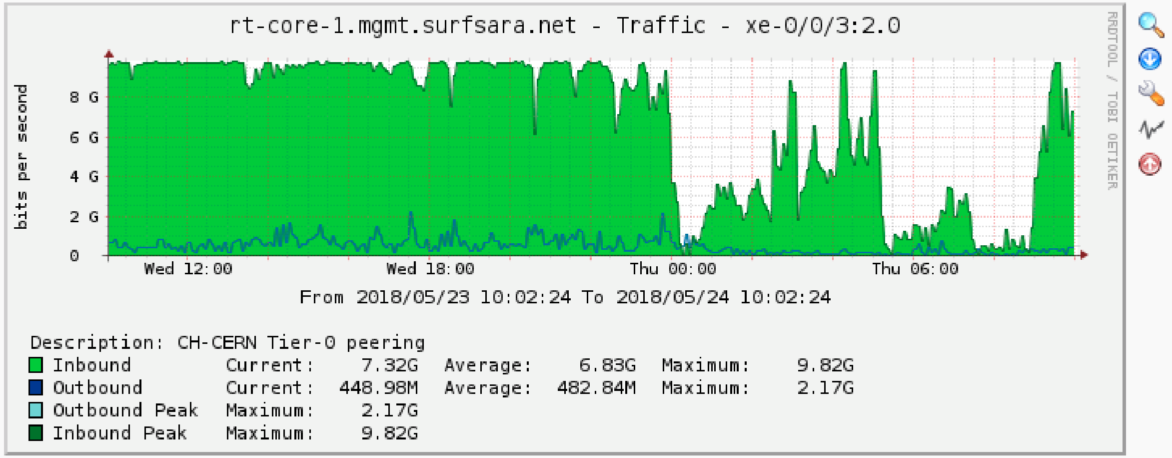
FTS

- Optimizer
  - Configuration might be possible if you would have a detailed knowledge about network topology and the available resources
  - Circumstances may change due to other users’ activities
  - Empirical approach to maximise throughput
    - Increases the number of parallel file transfers to maximise throughput
    - Decreases the number of parallel file transfers when the number of recoverable errors increase

FTS

- The optimizer in action





Send 100 files from Amsterdam to Barcelona and from there to Vancouver

FTS commandline

https://youtu.be/FcvE4G-jAX4

FTS monitoring

https://youtu.be/T_RzhXxSVZE

WebFTS

https://youtu.be/PcvVTiw8h8w