OSBCONF 2014
Cloud backup with Bareos

Sep 23, 2014

OSBCONF 23/09/2014

- Content:
  - Who am I
  - Quick overview of Cloud solutions
  - Bareos and Backup/Restore using Cloud Storage
  - Bareos and Backup/Restore of Cloud Storage

Who am I

- Marco van Wieringen
- Studied datacommunications from 1989-1995
- Author of Linux diskquotas and BSD accounting.
- Employed as Senior Consultant for 15+ years
- Mostly worked in the banking sector on Solaris systems (new environments and migrations)

Who am I (2)

- Active with Bacula since 2008
- Forked the codebase in 2012
- #3 independent contributor to Bacula
- One of 5 founders of Bareos GmbH & Co KG

Cloud storage landscape

- Started looking into the cloud storage landscape in February 2014 after FOSDEM
- Two aspects:
  - Using as backup storage
  - Backup content of Cloud storage
- No particular favorite
- Performance important
- Only looked at API and usability

Cloud storage landscape (2)

- GlusterFS
- CEPH
- Amazon S3
- Swift
- Commercial OpenSource like RHS
  - Now GlusterFS
  - Future CEPH ?

GlusterFS

- POSIX-Like Distributed File System
- No Metadata Server
- Network Attached Storage (NAS)
- Heterogeneous Commodity Hardware
- Aggregated Storage and Memory
- Standards-Based – Clients, Applications, Networks

GlusterFS (2)

- Flexible and Agile Scaling
  - Capacity – Petabytes and beyond
  - Performance – Thousands of Clients
- Single Global Namespace
- Run fully in userspace
- Next to network connectivity also support for InfiniBand.

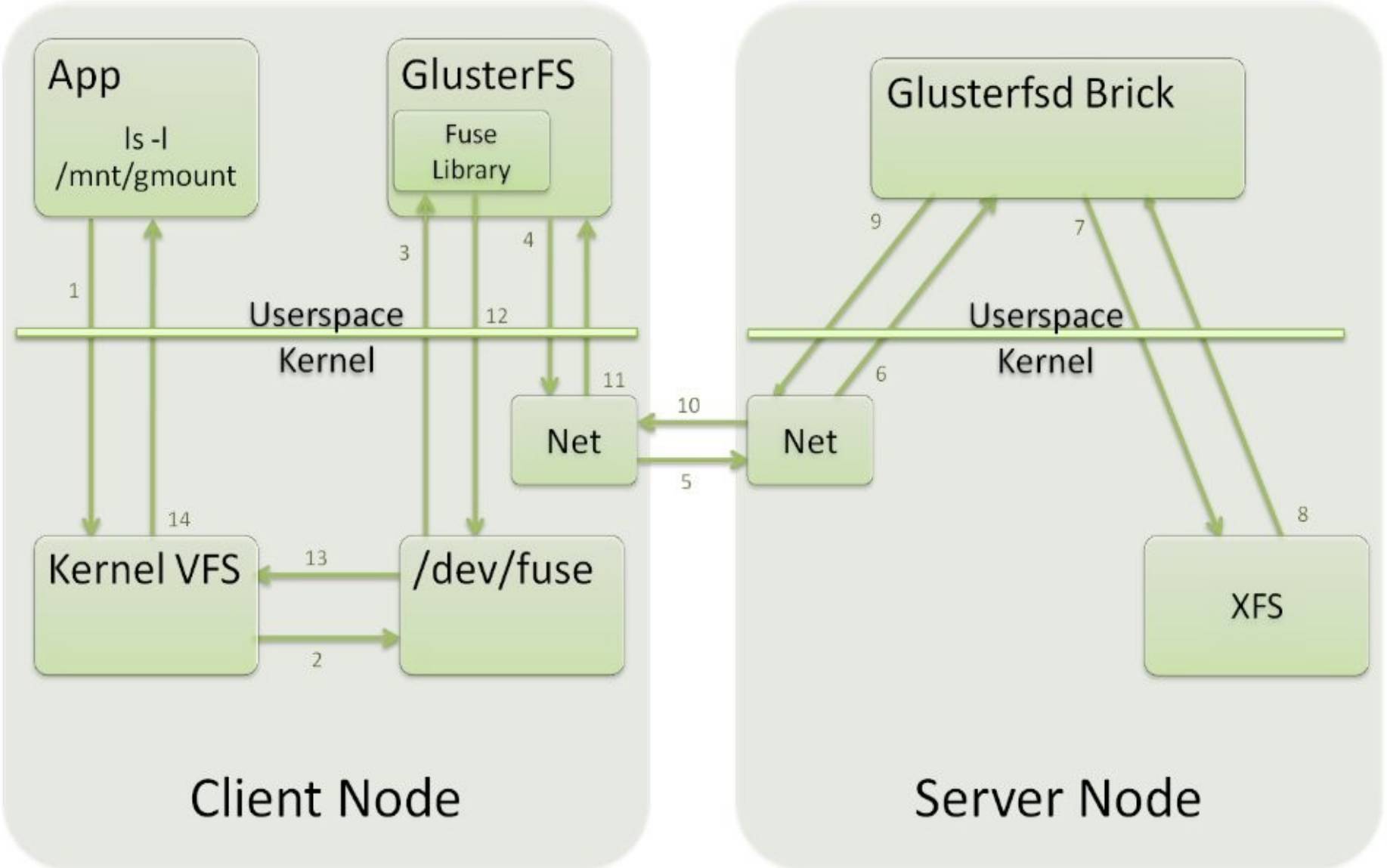
Data Access Overview

- GlusterFS Native Client
  - Filesystem in Userspace (FUSE)
- NFS
  - Built-in Service
- SMB/CIFS
  - Samba server required
- Unified File and Object (UFO)
  - Simultaneous object-based access

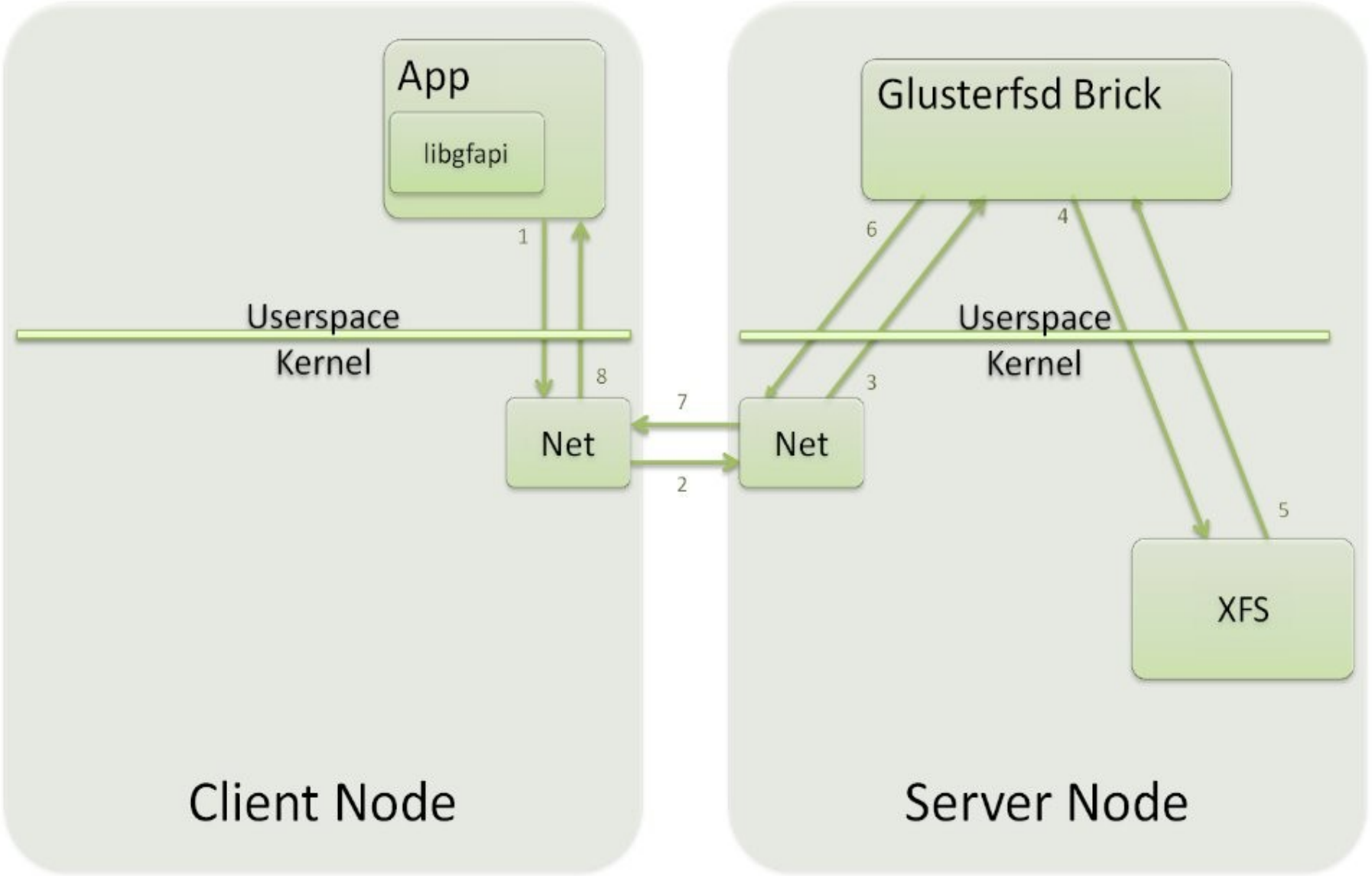
Use as backup storage

- Using FUSE Gluster volume can be mounted as POSIX filesystem and used as file storage today.
- A faster datapath is wanted to integrate into the storage server so we bypass the FUSE layer.
- Linus stake on FUSE is that its used for toy filesystems and not production.

Data Access via FUSE



Data Access via GFAPI



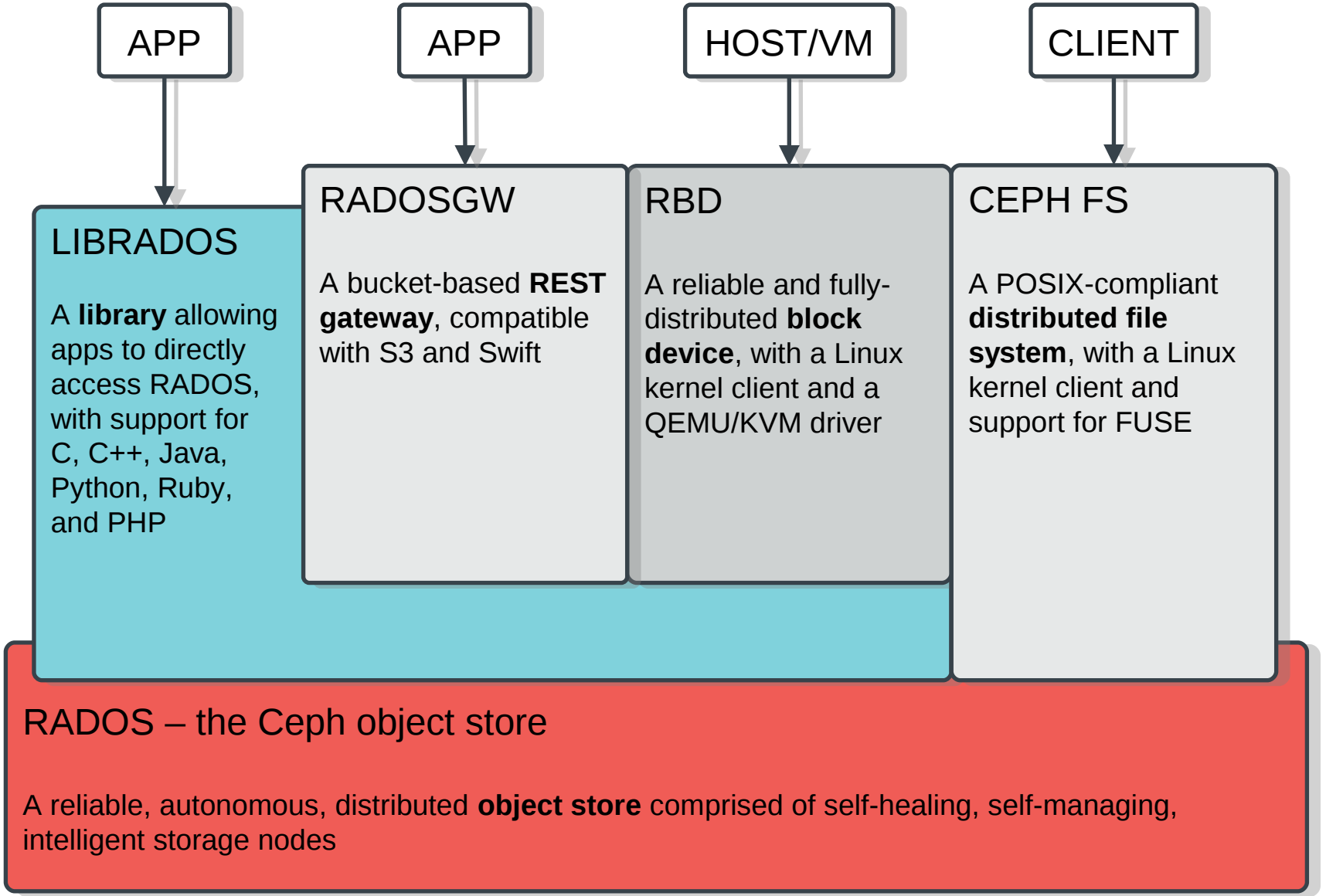
CEPH

- Most interesting things already said by previous speaker.
- Distributed Storage System
  - Large scale
  - Fault tolerant
    - No single point of failure
    - Commodity hardware
  - Self-managing, self-healing

CEPH (2)

- Unified storage platform
  - Objects
    - Rich native API
    - Compatible RESTful API
  - Block
    - thin provisioning, snapshots, cloning
  - File
    - strong consistency, snapshots

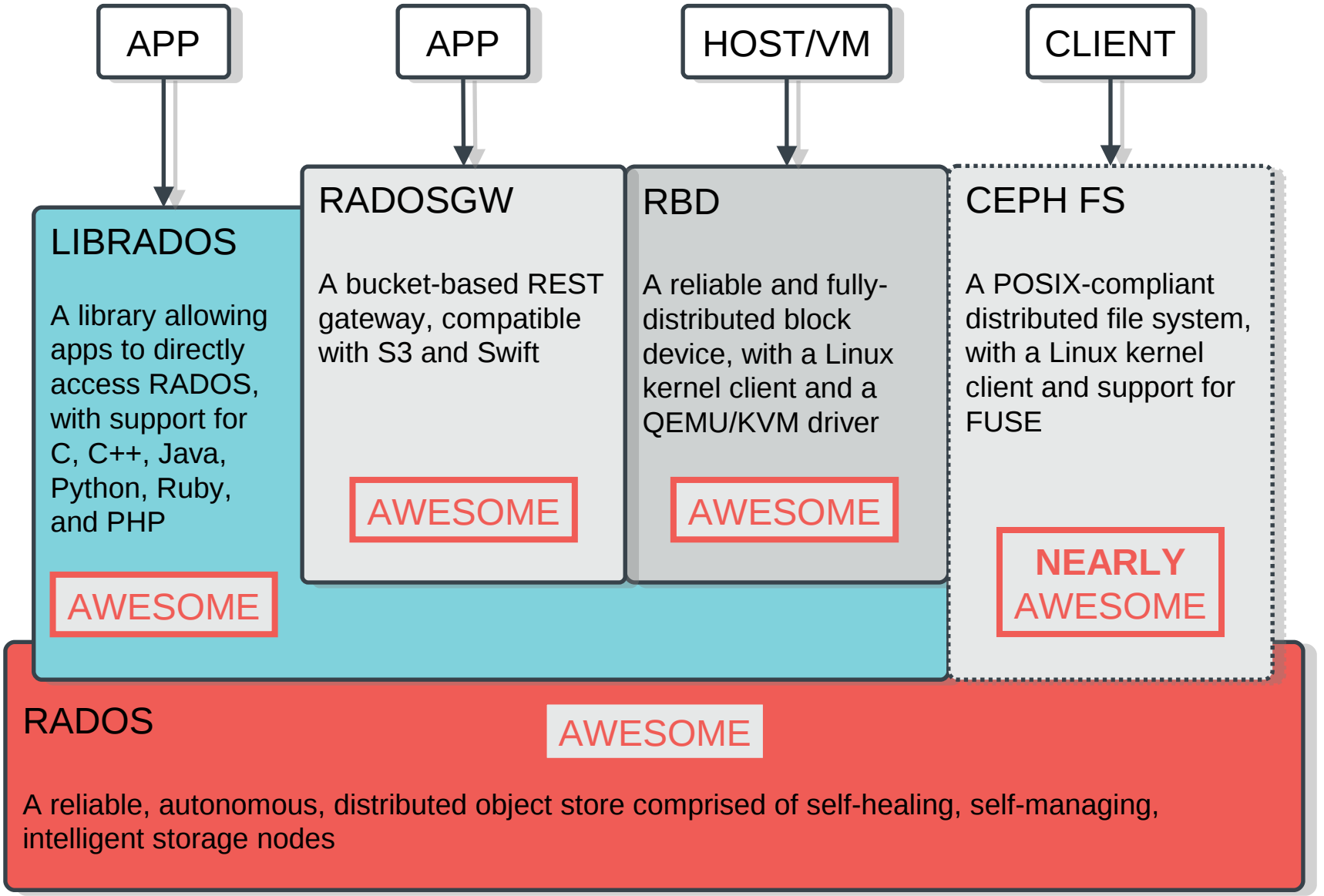
Data Access Overview



Data Access Overview (2)

- Filesystem on top of RBD (Rados Block Device)
- CEPH FS via either FUSE or native kernel module.
- RADOSGW (S3/Swift)
- Librados (direct low level access)

Data Access Overview (3)



Use as backup storage

- Create filesystem on a RDB device and use as filestorage.
- CEPHFS via FUSE or native kernel module as filestorage.
- Native storage using librados
- Native storage using libcephfs
- Last two preferable.

Object Storage

- Multiple vendors
- Most API's for new style languages (PHP/Python/Java)
- HTTP/HTTPS based protcol
- Only able to PUT full new object not appendable
- Most implementations use CURL as lowlevel library

Object Storage (2)

- Most promising library for native support seems to be libdroplet of Scality
  - Abstraction for different object store providers
  - Native C/C++ API
  - POSIX file storage implemented but unusable due to the low level protocol not supporting POSIX operations.
  - Missing a local caching layer for file operations.

Use as backup storage

- Use S3FS(C++) or S3QL (Python) and mount via FUSE.
- Use S3backer which has some interesting caching but is only one file.
- Caution with costs because of storage and upload/download costs involved with Amazon S3.

Cloud Storage and Bareos

- Use cloud storage as backing store
  - Storage Daemon can save data to cloud
  - Already possible today with filesystem abstraction.
- Backup content of the Cloud
  - Backing up full cloud probably not feasible
  - Subset of data can already be backup without support in Bareos today.
  - Doesn't use specific features available.

Cloud Storage and Bareos

- Cloud storage as backing store for data
  - Close integration with Storage Daemon
  - Shortest low level path for storage
  - Minimum number of data copying
  - For now POSIX interface only
  - No data format changes just save data as normal disk based volume format.
  - Future may bring other formats and other clever tricks.

Storage Daemon Changes

- Support for multiple storage backends in Storage Daemon.
- Abstraction of different type of devices
- Seperate packaging of different backends
- Analog to the already existing database backends dynamically loaded on reference in the config.

Storage Daemon Changes

- Available as technology preview in 14.2.1
  - File backend (always compiled in)
  - Tape backend (in separate package)
  - Fifo backend
  - GFAPI backend (Gluster)
  - RADOS backend (CEPH)
  - CEPHFS backend (CEPH)
  - OBJECT backend (S3/Swift using droplet)
    - Not working needs more work on droplet library.

Storage Daemon Changes

- Media stored on CEPH and Gluster have same format as normal file based volumes
- By using normal tools of CEPH (rados get) and gluster (mount glusterfs) the file volumes can be copied to normal disk storage and used as file storage.
- Replication of backup data can be done using native tools of Gluster (Geo Replication) etc.
- Possibly in future allow read-only devices to access data in remote datacenter

Future of Cloud backends

- Work out Object Storage plugin
  - Probably some chunking method
  - Local caching
- Windows Azure
  - Seems to support blobs
  - Elasto library may fit (https://code.google.com/p/elasto/)

Cloud Storage and Bareos

- Backup content of the Cloud
  - Due to dependecies implement as Plugins
  - Plugin interface as exists today to limited to write proper backup and restore filesystem like data
  - Implement cloud backup and restore functionality in File Daemon
  - Reuses configure logic added for Storage Daemon Backends

File Daemon Changes

- Generic Hardlink handling also for plugins
- ACL Backup and Restore handling for plugins
- XATTR Backup and Restore handling for plugins

File Daemon Changes

- Hardlink code refactored and available in master branch.
  - Dropped duplicate hash table code
  - Reused generic htable for all hash tables
- ACL and XATTR extensions for plugin mostly coded and on its way for review and merge into master tree. (Including Python plugin code)

RADOS FD Plugin

- Initial coding done.
  - Uses librados
  - Uses enumerator to list objects in certain RADOS pool
  - Snapshots rados pool before starting backup
  - Early stage of testing

GFAPI FD Plugin

- For GlusterFS
- Initial coding done
  - Uses Glusters GFAPI
  - POSIX API
  - First phase only regular files
- Currently scans filesystem for changed files
  - Analog to what the normal filedaemon does
  - Including support for accurate etc.
- Future maybe support for changelog tracking

CEPHFS FD Plugin

- For CEPHFS
- Initial coding done
  - Uses CEPH libcephfs
  - POSIX API (almost the same as GFAPI)
  - First phase only regular files
- Currently scans filesystem for changed files
  - Analog to what the normal filedaemon does
  - Including support for accurate etc.

Future of Cloud FD plugins

- Merge new code after further testing.
- Beta program
- Get some sponsoring
- When we work out object storage as backend maybe create a object store FD plugin.

Questions ?

Links

- https://code.google.com/p/elasto/
- https://code.google.com/p/s3fs/
- https://fedorahosted.org/s3fs/
- https://code.google.com/p/s3backer/
- https://github.com/scality/Droplet
- https://forge.gluster.org/
- http://ceph.com/