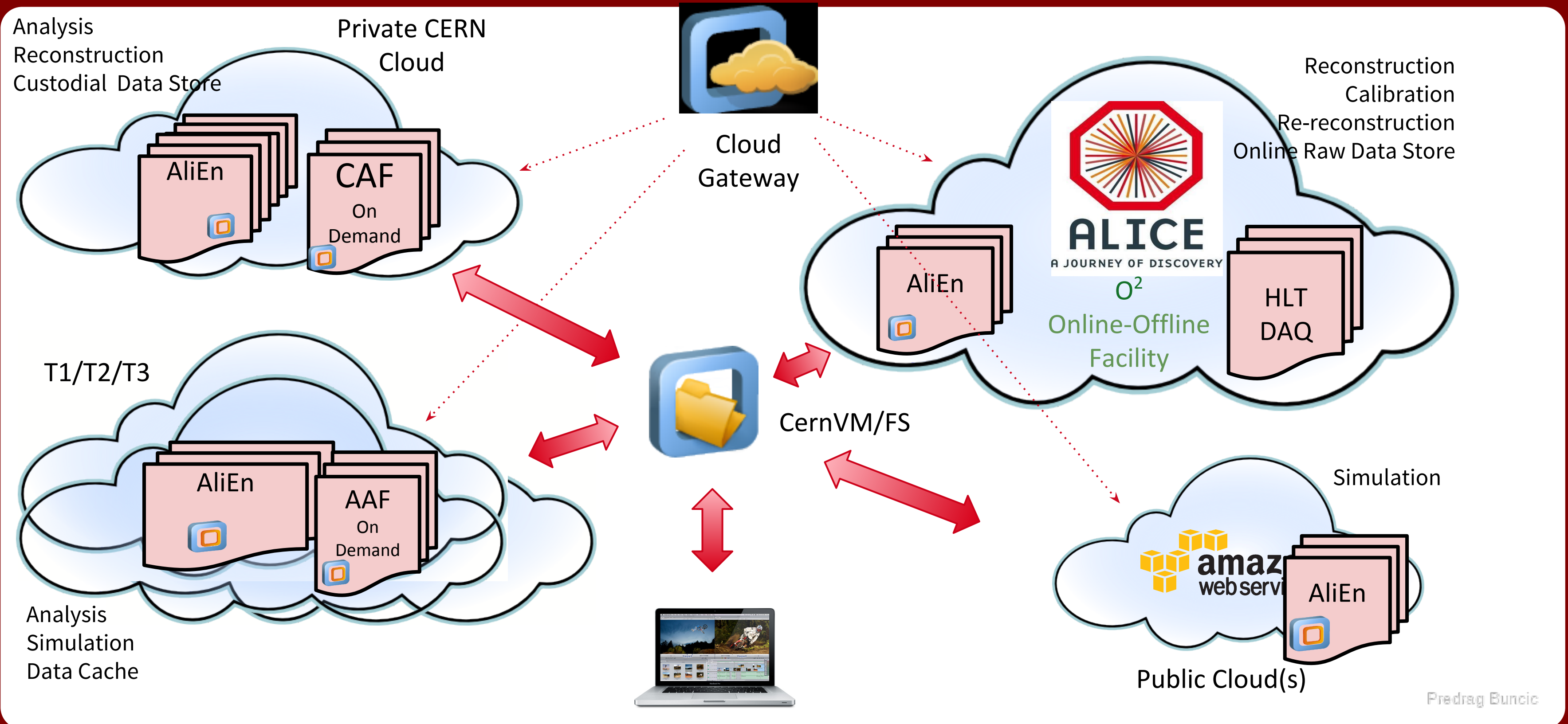
Scalable cloud without dedicated storage.

D.Batkovich, M.Kompaniets, A.Zarochentsev
St.Petersburg State University

Outline

- Future of ALICE experiment IT infrastructure/ Motivation
- Scalable cloud for Tier-3
    - Distributed storages
- Current status / Problems

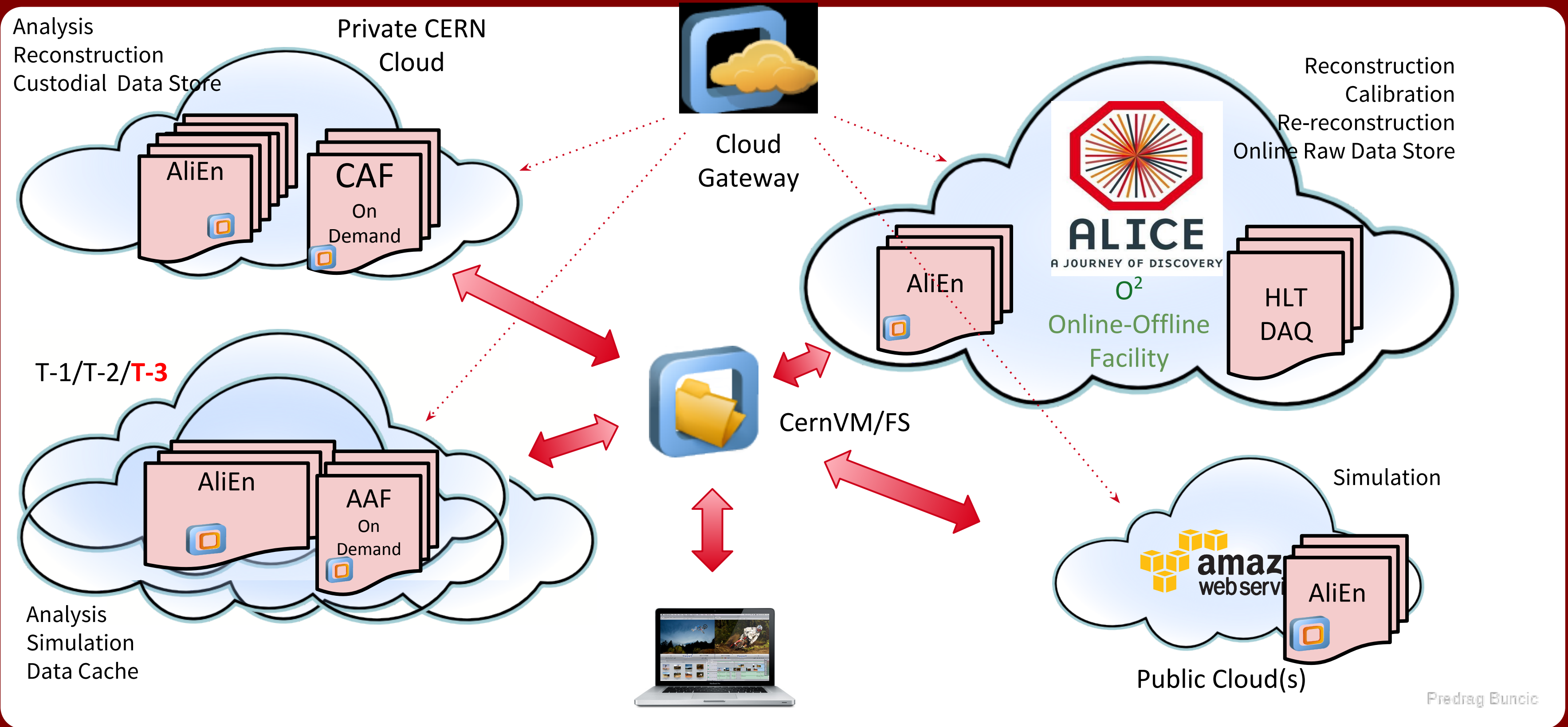
ALICE IT infrastructure plans for Run3



Why clouds?

- More efficient resources utilization
    - e.g. HLT farm is heavy loaded during runs but almost free during shutdowns
- Unified software deployment method (mCernVM/CVMFS)
    - easy upgrades
    - maintaining different software versions on the same facility (LTDP)
- Elasticity
    - share jobs with public/private clouds using unified API

ALICE IT infrastructure plans for Run3



Tier-3 resources

Lots of small institutions / groups

- limited computational resources
- limited finances
- lack of qualified IT personnel

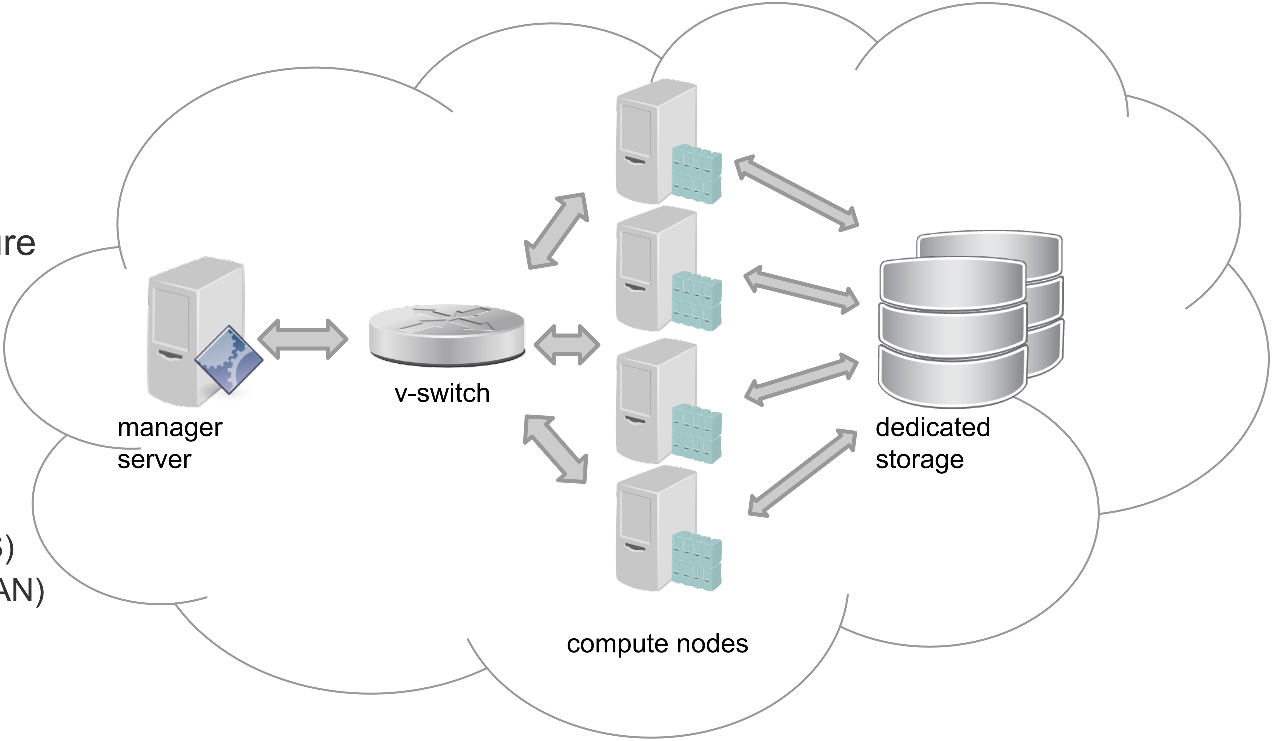
Requirements for cloud:

- runs on commodity hardware
- easy deployment and maintenance
- scalable
- fault tolerant
- Amazon EC2 API support

Cloud with dedicated storage

Cloud (in general):

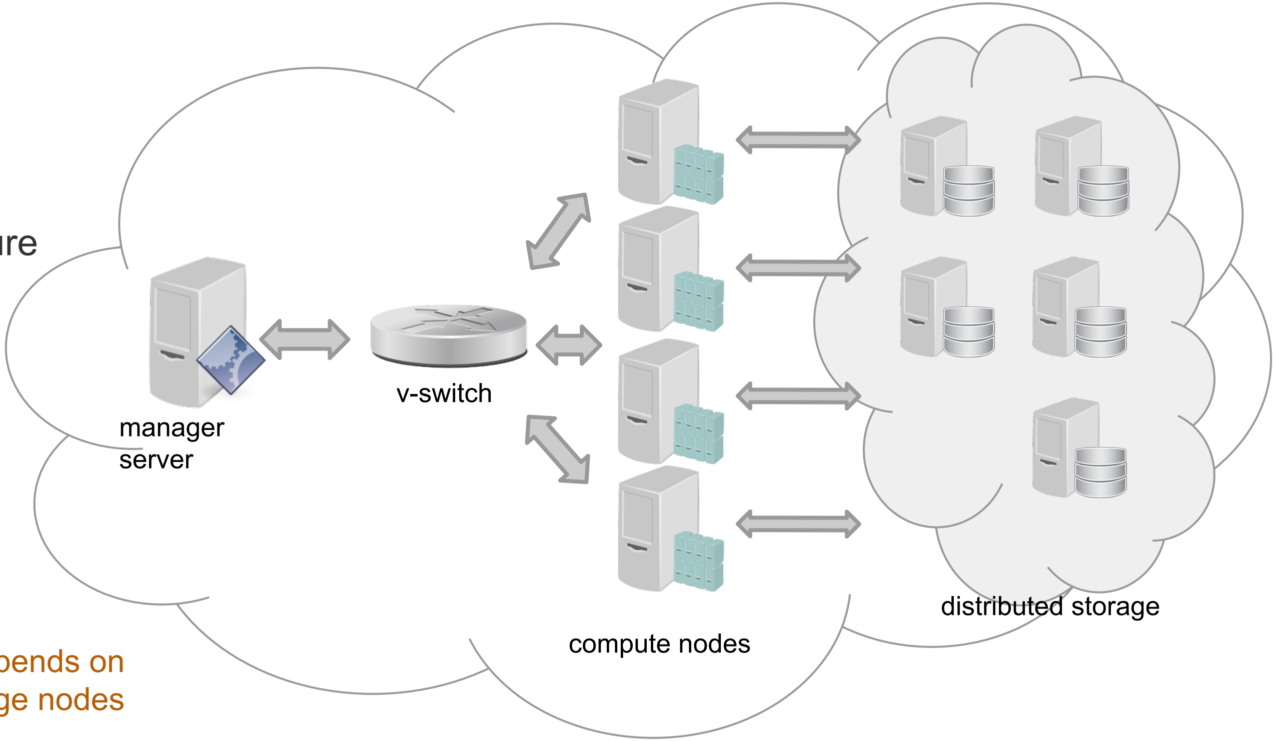
- Manager server
- Network infrastructure
- Computing nodes
- Storage for block devices/images
    - either slow (NFS)
    - or expensive (SAN)



Cloud with distributed storage

Cloud (in general):

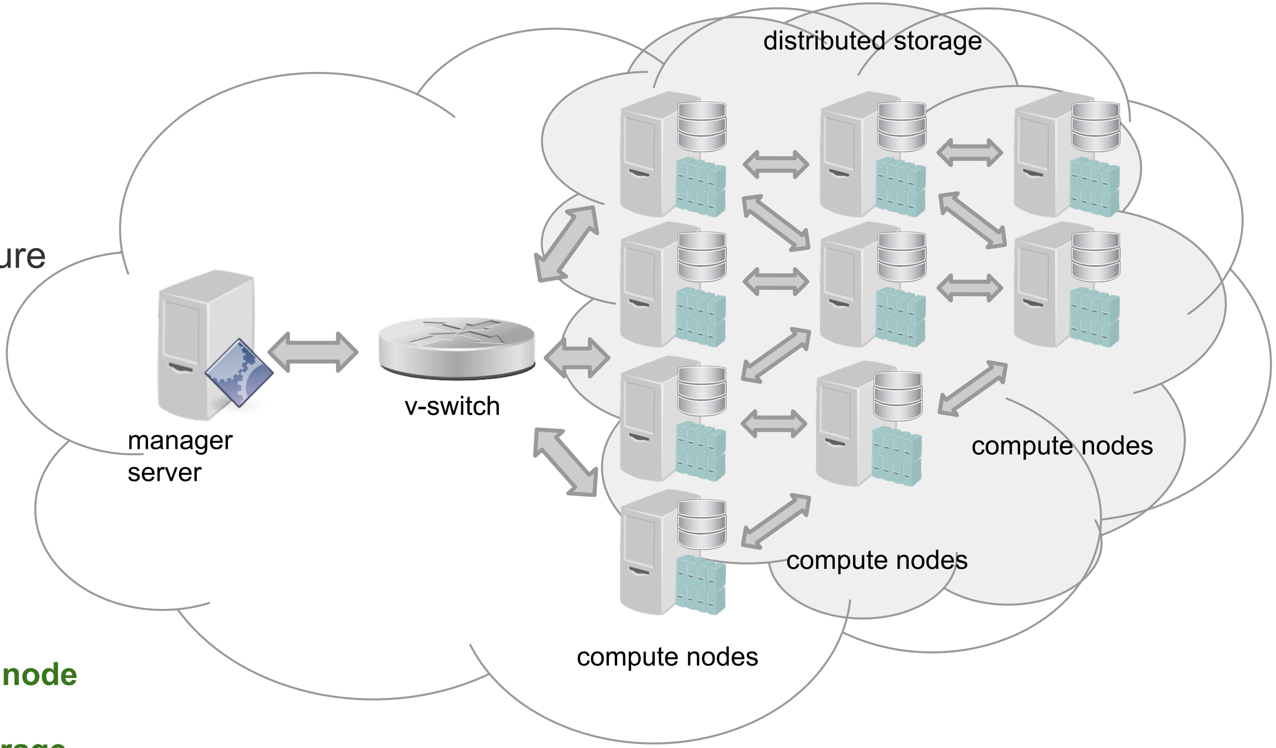
- Manager server
- Network infrastructure
- Computing nodes
- Storage for block devices/images
    - inexpensive
    - scalable
    - fault tolerant
    - performance depends on number of storage nodes



Cloud without dedicated storage

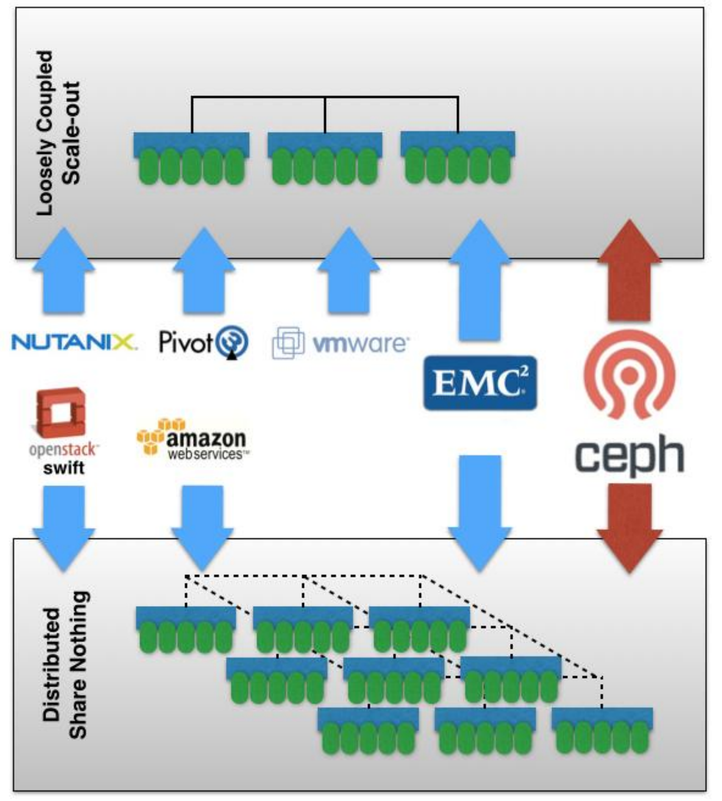
Cloud (in general):

- Manager server
- Network infrastructure
- Computing nodes
- Storage for block devices/images
    - inexpensive
    - scalable
    - fault tolerant
    - **each compute node contributes to distributed storage**



Distributed storages

- OpenStack Swift
- CEPH
- GlusterFS
- EMC ScaleIO
- VMWare Virtual SAN
- Nuantix
- Pivot3
- ….



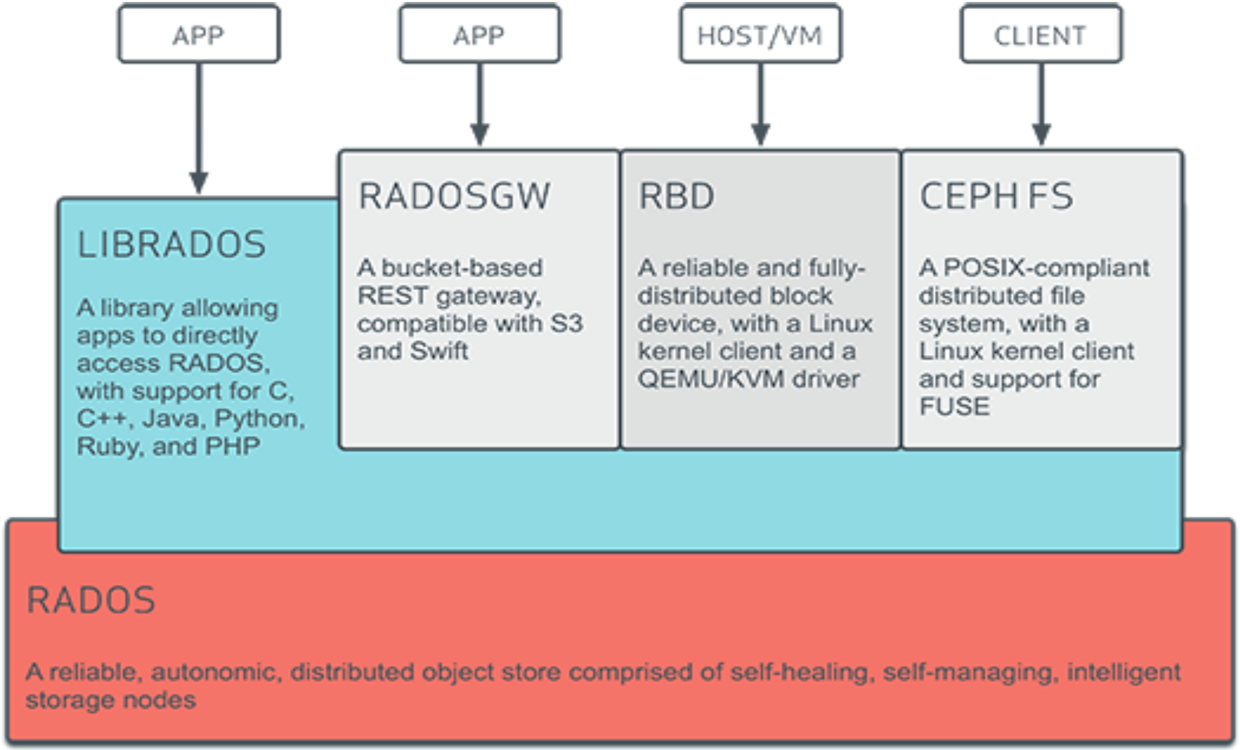
http://pinrojas.com/2014/05/27/ceph-versus-distributed-share-nothing-storage-architectures/

Object stores: Swift vs CEPH

	Swift	CEPH
Replication	Yes	Yes
Max. obj. size	5Gb(bigger objects segmented)	Unlimited
Multi Data Center installation	Yes	No
Replicas management	No	Yes
Writing algorithm	Synchronous	Synchronous
Amazon S3 compatible API	Yes	Yes
Data placement method	Ring(static mapping structure)	CRUSH(algorithm)

http://www.mirantis.com/blog/object-storage-openstack-cloud-swift-ceph/

CEPH is more than an object store



http://ceph.com/community/more-than-an-object-store/

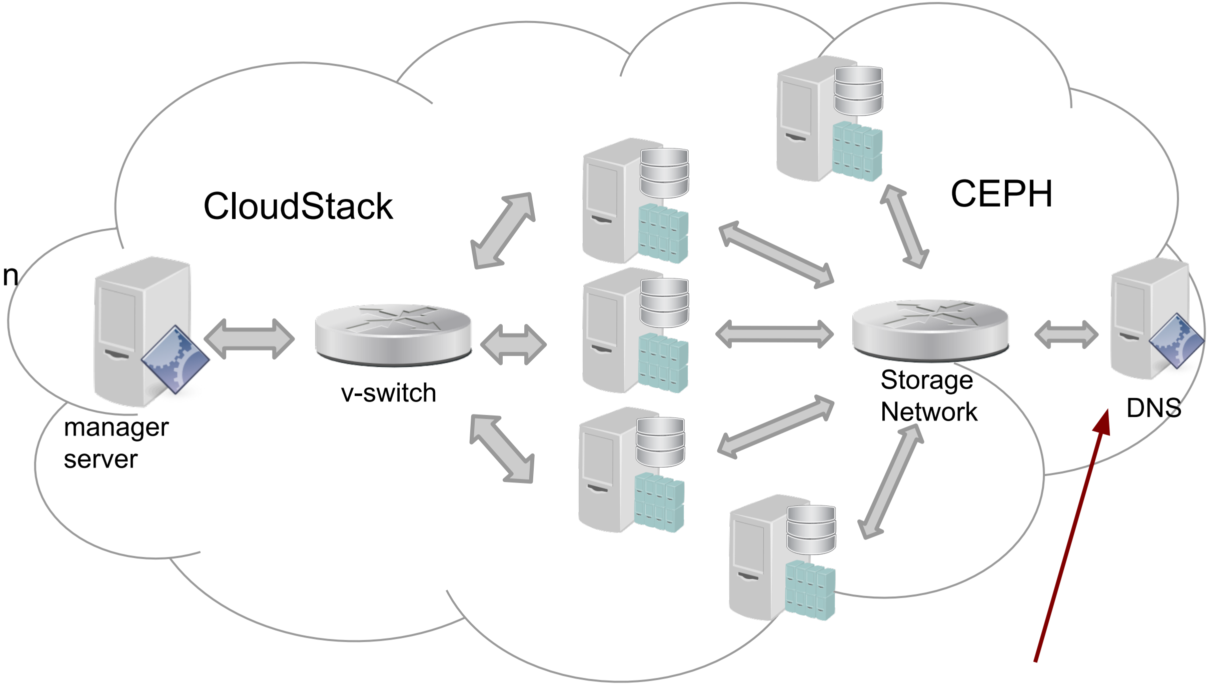
CloudStack and CEPH (PoC)

Our current setup:

- manager server
- 5 mixed compute|storage nodes
- Cloud network
- Storage network with DNS in round robin mode

Points of failure:

- manager server
- storage DNS



CloudStack has **limited** CEPH support, and for **fault tolerance** requires additional DNS in round-robin mode to locate CEPH monitors

Summary

Requirements for cloud:

runs on commodity hardware	yes
easy deployment and maintenance	automated compute|storage node deployment
scalability	yes
fault tolerance	to node fault
Amazon EC2 API support	limited

Problems to be solved:

fault tolerance	multiple manager node, storage DNS
easy deployment and maintenance	CloudStack automated deployment broken about 1 year ago Requires deep code revisioning

Summary2

We are not satisfied with CloudStack code quality

- lots of regression in code
- limited CEPH support
- limited support of Amazon EC2 API (required for CernVM)
- Cli interface is not fully covers functionality

OpenStack:

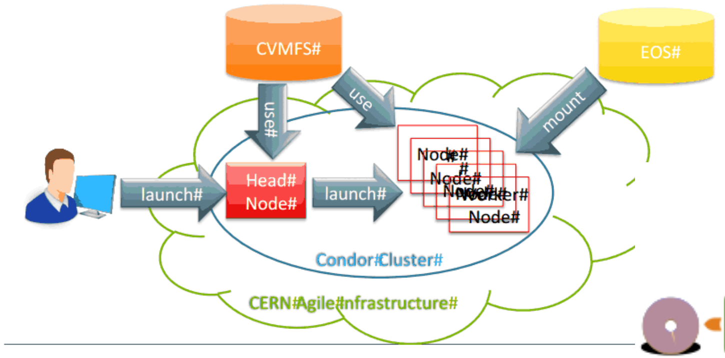
- CEPH backend is fully supported in OpenStack
- Inktank (CEPH) is acquired by Red Hat
- Red Hat will use CEPH as block device storage for OpenStack

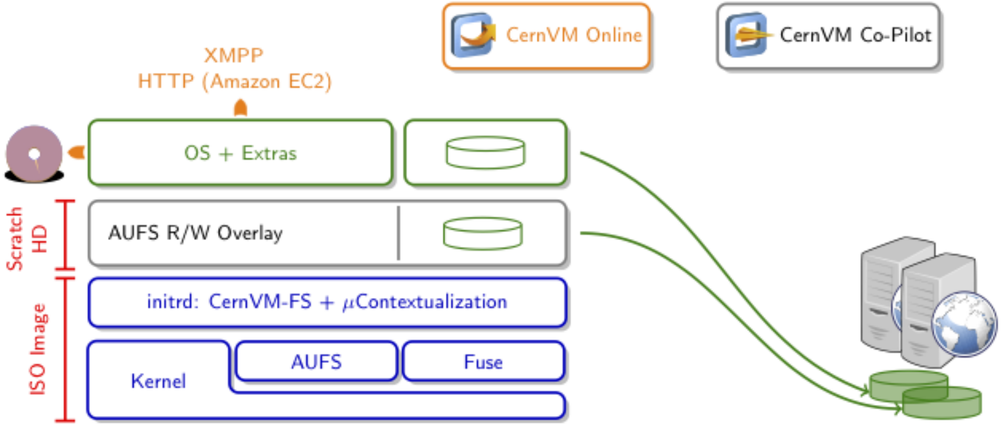
Thank you for attention!

Backup

mCernVM

Virtualized cluster for QA





Storage architectures

- **Distributed Share Nothing**: This type of architectures works on independent controllers no sharing memory resources between nodes. This sort of solution has been made for ***Non-transactional data*** and brings distributed data protection features. Object Storage is a solution that fits with this description and you have several options like OpenStack Swift or Ceph Object Storage or Amazon S3.

- **Loosely Coupled Scale-Out**: Similar description to the other, but it’s aimed to store ***transactional data***. The data is distributed through all nodes in blocks or pieces and you get consistency writes and reads among the nodes. Some part of the software maps the location of the pieces of the data and help you to put it together to have a coherent read operation. The performance and the capacity scale out adding nodes and usually you can control the importance of every node into the whole cluster depending on its hardware features and its contribution to the overall performance. Some examples are: EMC ScaleIO, Ceph Block Storage, VMWare Virtual SAN, Nutanix and Pivot3.

OpenStack and CEPH