Xrootd in the distributed cloud storage

D.Batkovich[1],M.Kompaniets[1],**O.Shadura**[2],S.Svistunov[2],
V.Yurchenko[2], A.Zarochentsev[1]

SPBSU & BITP

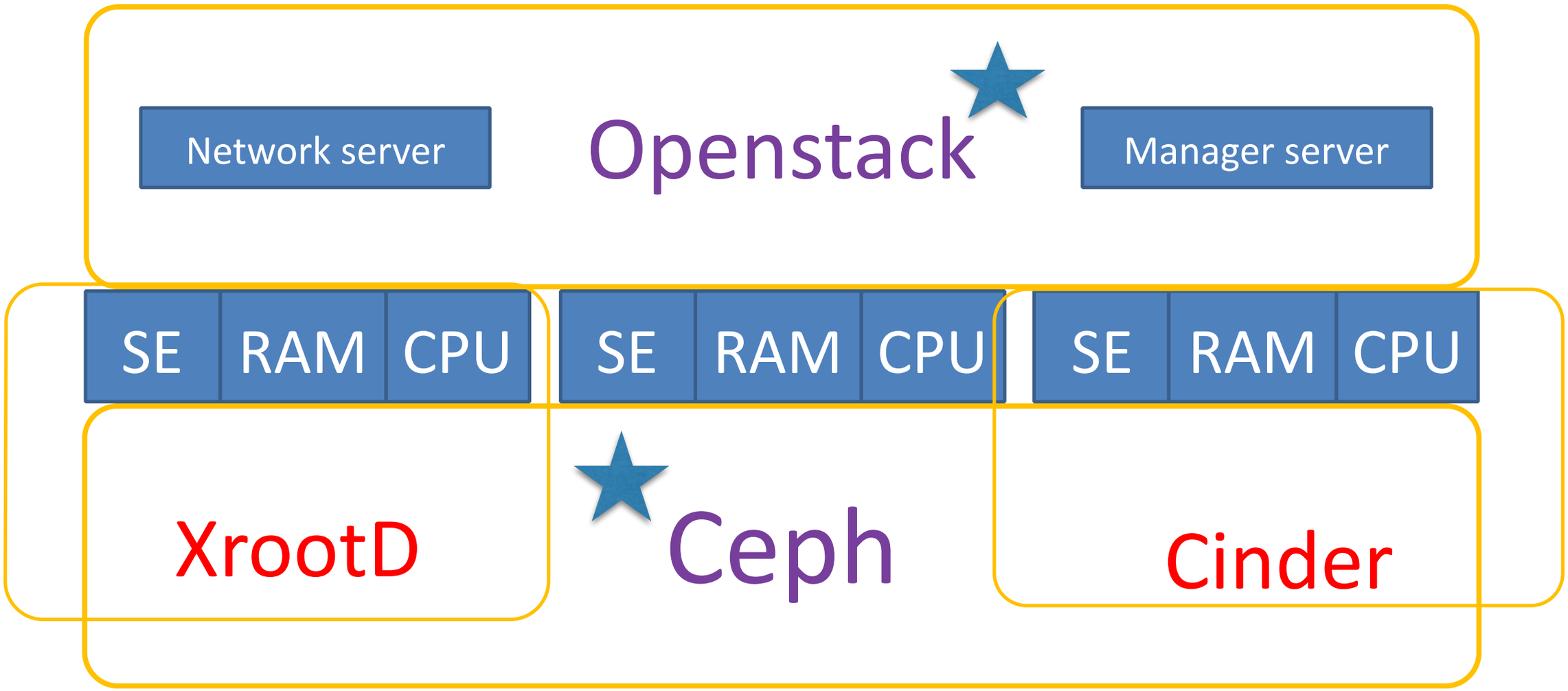
Goals & benefits

- Easy deployed and flexible Tier 3 for ALICE data analyses needs:
  - Full automatisation & fast deployment of management tools
  - Cheap cloud computing power for small research groups
  - Integration XrootD with Ceph backend as a data storage

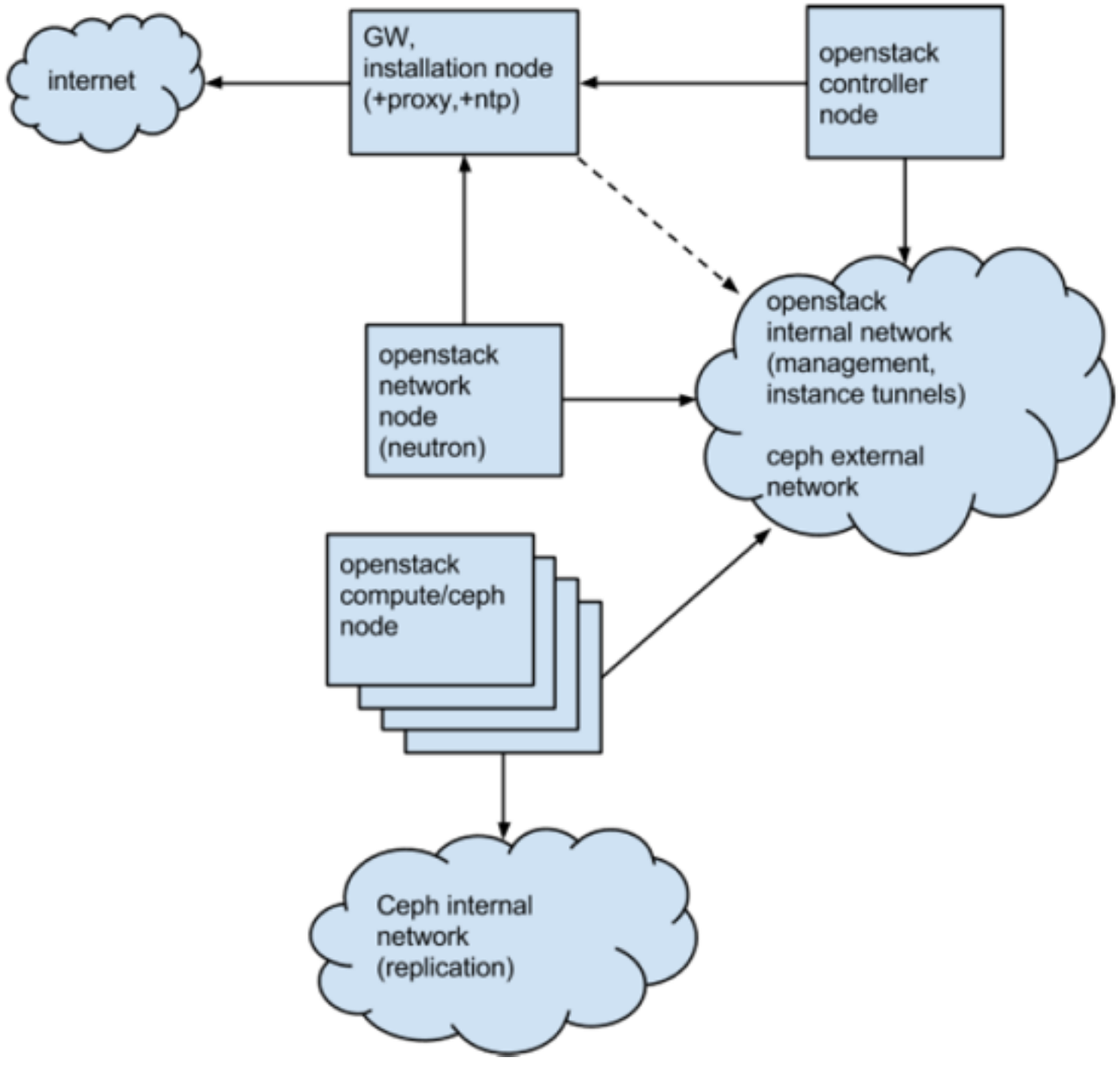
Why Ceph?

- Ceph is based on Rados (Reliable Autonomic Distributed Object Store):
  - Benefits: replicates objects for fault tolerance
  - Benefits: Fast & scalable storage
  - Benefits: Ceph can be used as a storage for Openstack Cinder and as a data analysis storage based on XrootD

Symbiosis of cloud CE & SE



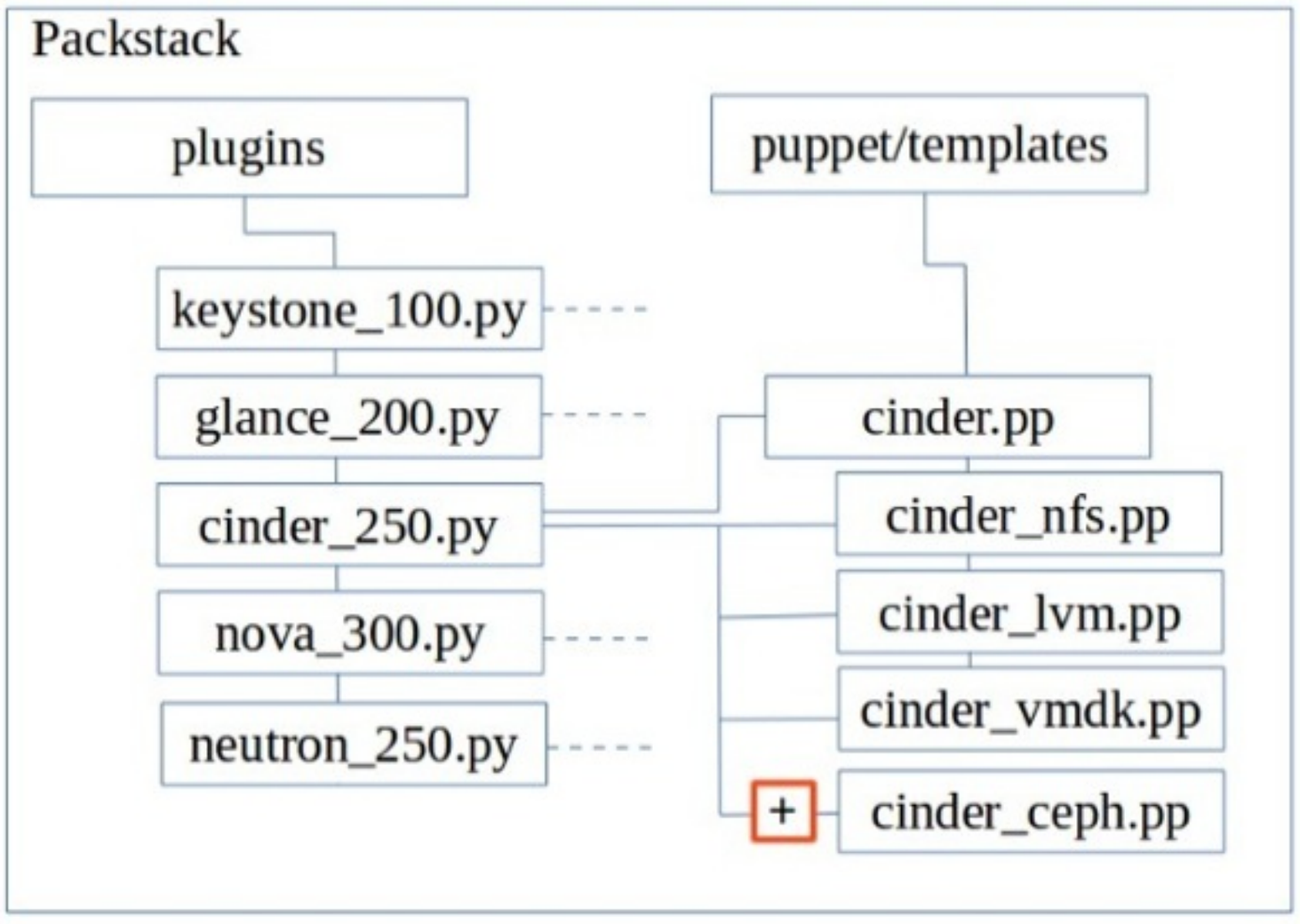
Deployment scheme



Packstack

*Fast Puppet based Openstack solution provided by RDO (RHEL)

* Modifications:



Packstack modifications

1. ML2 network updates
1. Update of list of repositories
2. Installation of needed packages and dependencies
3. Installation of Ceph on main node and storage nodes
4. Installation and start Ceph monitor on separated storage nodes
5. Installation and start Ceph OSD on all storage nodes
6. Final configuration of Ceph cluster
7. Configuration Libvirt virtualization software
8. Modification in all configuration files and reboot of Openstack services

Xrootd on object storage

- For organisation interaction of Xrootd and object storage we are using intermediate layer: RadosFS

- RadosFS - A filesystem library based in librados that offers a simple interface for file operations on top of a Ceph Cluster.

- Written by Joaquim Rocha (IT, CERN): https://github.com/joaquimrocha/radosfs

radosfs-python library

radosfs-python[1] is the Python wrapper for Rados Filesystem[2].

- written on Cython. Doesn't require any compilation and C++ => easy to use for scripts and hand working
- supports Python ≥ 2.6 (including Python 3)
- provides basic operations (IO, CRUD).

TODO: implement full set of operations available in radosfs, build RPM package

1. https://github.com/batya239/radosfs-python
2. https://github.com/joaquimrocha/radosfs

radosfs-python examples

- FS tree access example snippet:

```
{
        import radosfs

        fs = radosfs.RadosFs(username, ceph_conf)
        fs.add_data_pool(data_pool, "/", size=size)
        fs.add_metadata_pool(metadata_pool, "/")

        my_dir = fs.dir("/my-dir").create(-1, True, owner_uid=1000, owner_gid=1000)
        print my_dir.is_writable()
        print fs.dir("/").entries()
}
```

- Files access example snippet:

```
{
        import radosfs

        fs = radosfs.RadosFs(username, ceph_conf)
        fs.add_data_pool(data_pool, "/", size=size)
        fs.add_metadata_pool(metadata_pool, "/")

        my_file = fs.file("/my.txt", radosfs.OpenMode.READ_WRITE).create(384, pool=data_pool)
        my_file.write("A long time ago in a galaxy far, far away...", offset=0)

        print "file content:", my_file.read()
}
```

Xrootd server & Ceph for mCernVM

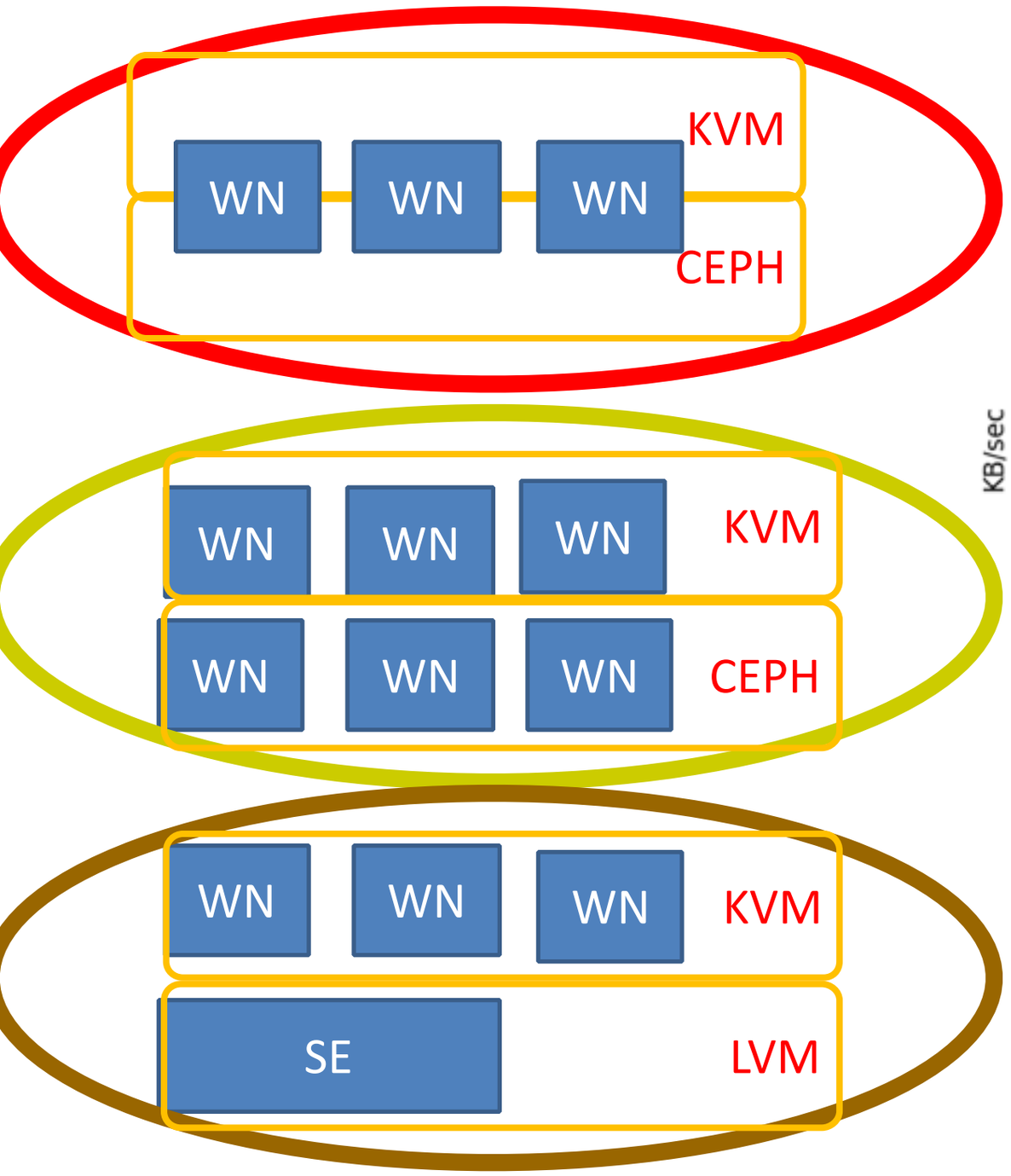
- Due to performance reasons, XrootD will be located on each VM (XRootD will provide option of “proxy” to Ceph)

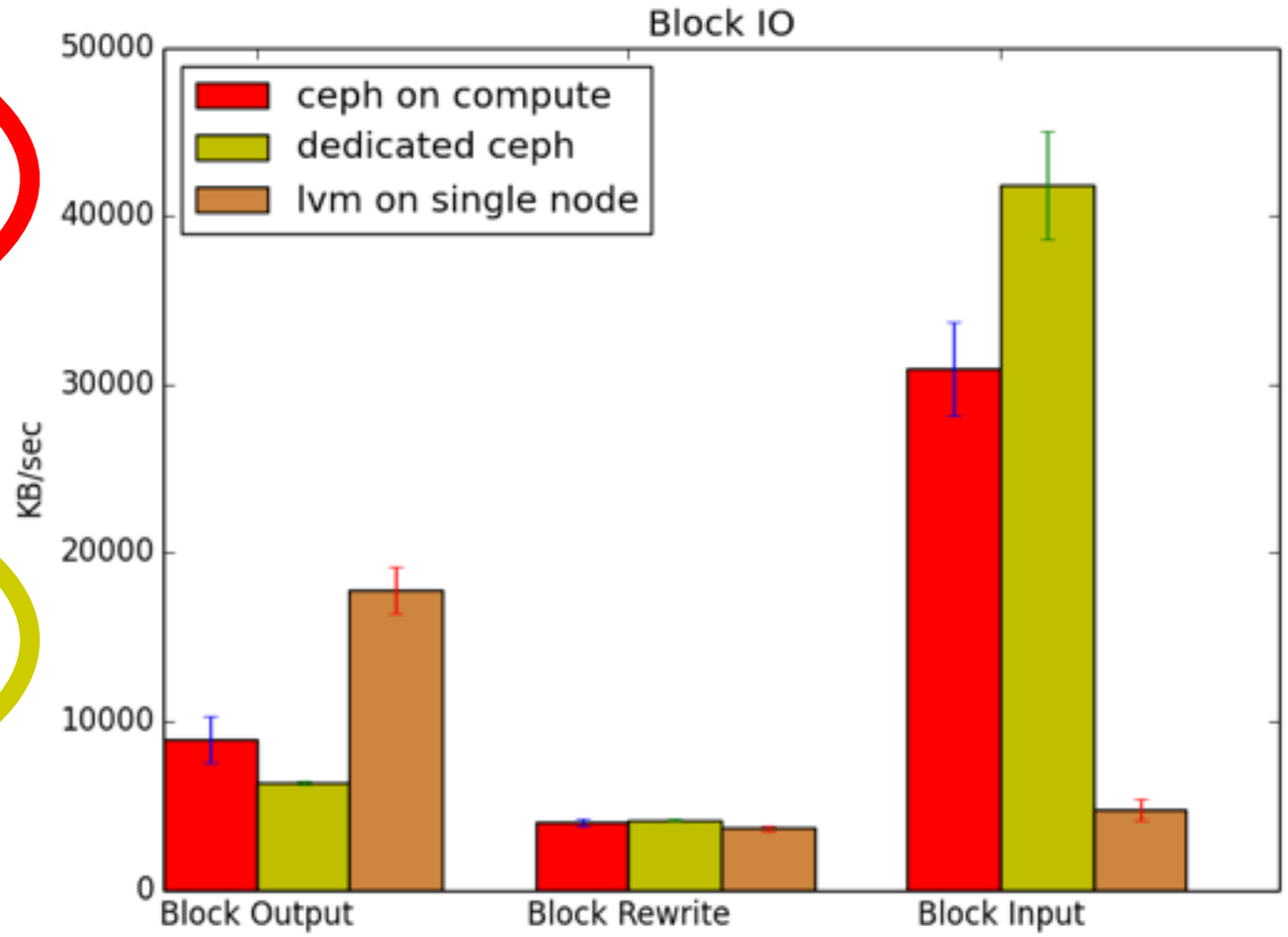
- BITP has own CVMFS server for local needs

- Branch: xrootdceph.bitp.kiev.ua with installed xrootd & librados rpms

- Next step: to test mCernVm with local CVMFS (local CernVm Online)

Comparison of SE's by Bonnie++ test





- 3 mixed compute/ceph nodes
- 3 compute, 3 ceph nodes
- 3 compute, 1 LVM node

Known Issues

Ceph is massively scalable, open source, distributed storage system but:

- Think about cache management?
  - Cache pool tiering
  - Flashcache (block cache for Linux)
- Think about high availability for Openstack?
  - HAProxy | Keepalived
  - Mysql Galera..

TBD: January 2015

- Commit changes to Packstack repo
- Create separate CVMFS server with radosfs,ceph-libs and xrootd packages
- Check work with mCernVM
- Check ALICE authorisation for XrootD/Ceph
- Test implementation of the FUSE plug-in to use RadosFS
- Check again all together :)