XRootD plug-ins at ALICE AF Prototype

T1-T2 workshop 2018 / Derby

Jan Knedlik, Paul Kramp

GSI

19. April 2018

Introduction

- Multipurpose HPC centre @ GSI (~32K logical cores, ~14 PiB Lustre storage)
- Contributing storage & computing resources as an ALICE-Tier 2 (T2) centre, as a NAF and possibly as an Analysis Facility (AF) in the future, to ALICE
- HPC requirements/challenges: OS/software dependencies, performant file access using Grid methods, limited connectivity to the outside, etc.

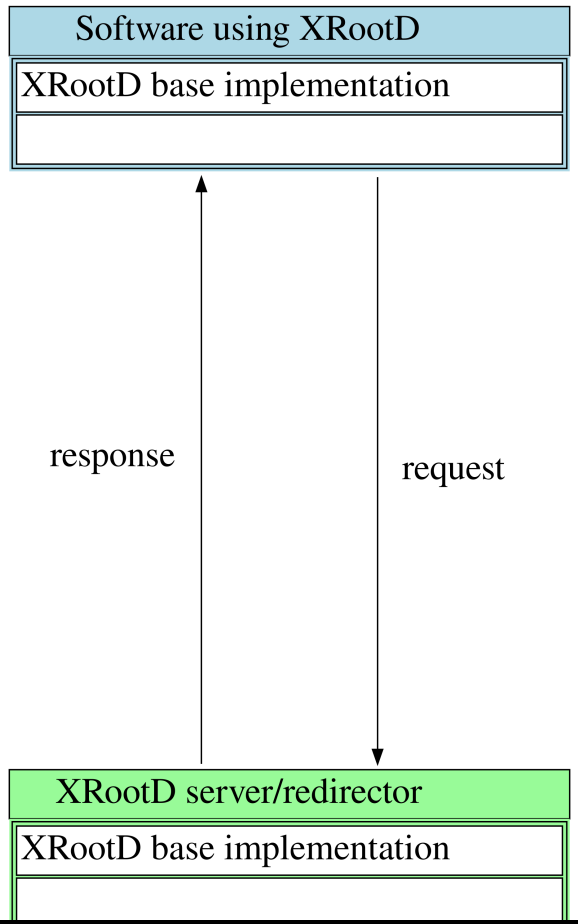
Introduction

- Multipurpose HPC centre @ GSI (~32K logical cores, ~14 PiB Lustre storage)
- Contributing storage & computing resources as an ALICE-Tier 2 (T2) centre, as a NAF and possibly as an Analysis Facility (AF) in the future, to ALICE
- HPC requirements/challenges: OS/software dependencies, performant file access using Grid methods, limited connectivity to the outside, etc. => solutions to some of those

XRootD

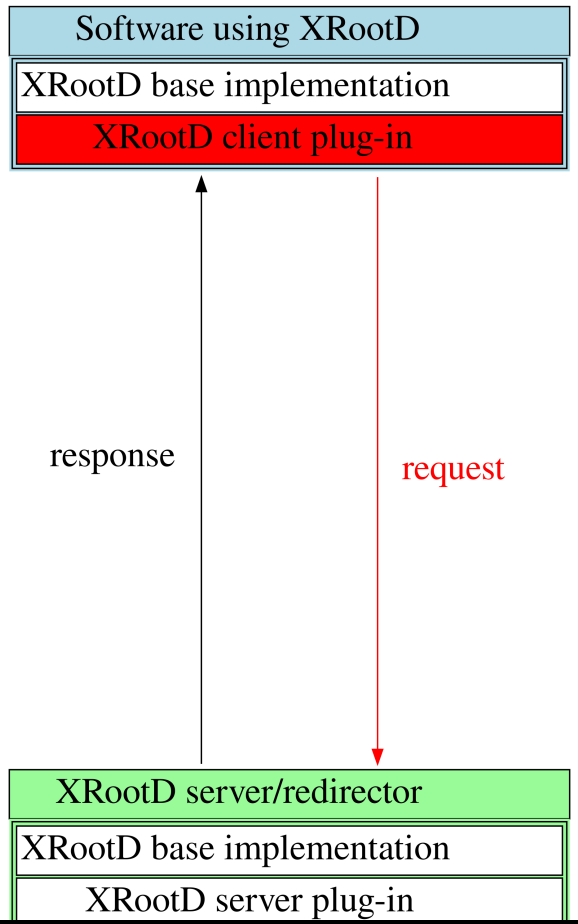
- XRootD established itself as a software standard for WAN data access in HEP and HENP
- We run two similar setups for ALICE T2 and AF prototype, that consists of 2 proxy servers, 2 redirectors and 3 data servers

XRootD



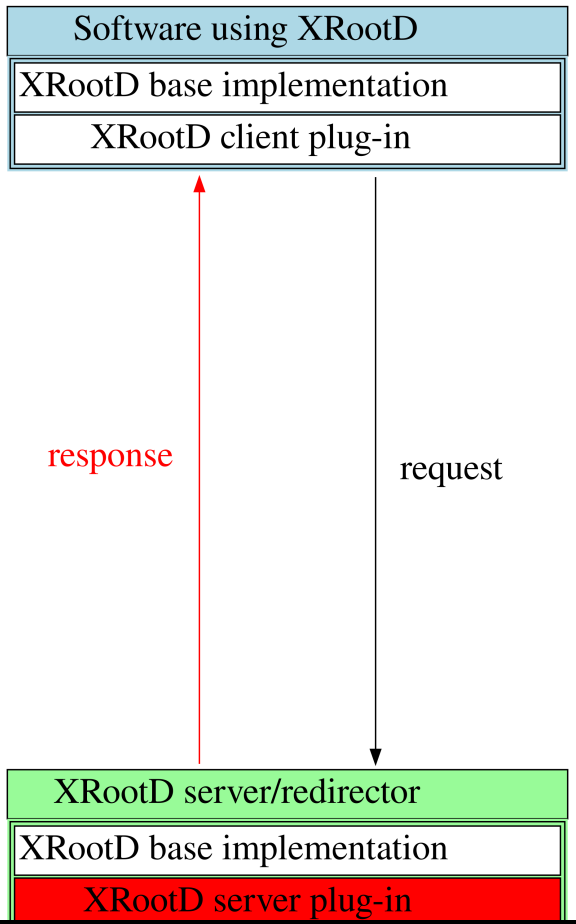
- XRootD communication uses a typical request-response model

XRootD Client Plug-ins



- Change the underlying client implementation
- Are transparent to higher level applications
- Adapt client request behaviour

XRootD Server/Redirector Plug-ins



- Change the underlying server implementation
- Adapt server request handling behaviour

Lustre Quota as storage statistics

One Lustre-related problem came up: Apmon sent wrong usage statistics

- XRootD redirectors/ data servers were agnostic towards Lustre and quota space reserved for ALICE
- => Gave misleading information (the whole lustre space) / might be a problem when the actual max quota is reached

LustreOssWrapper - Plug-in

Solution: An XRootD data server plug-in

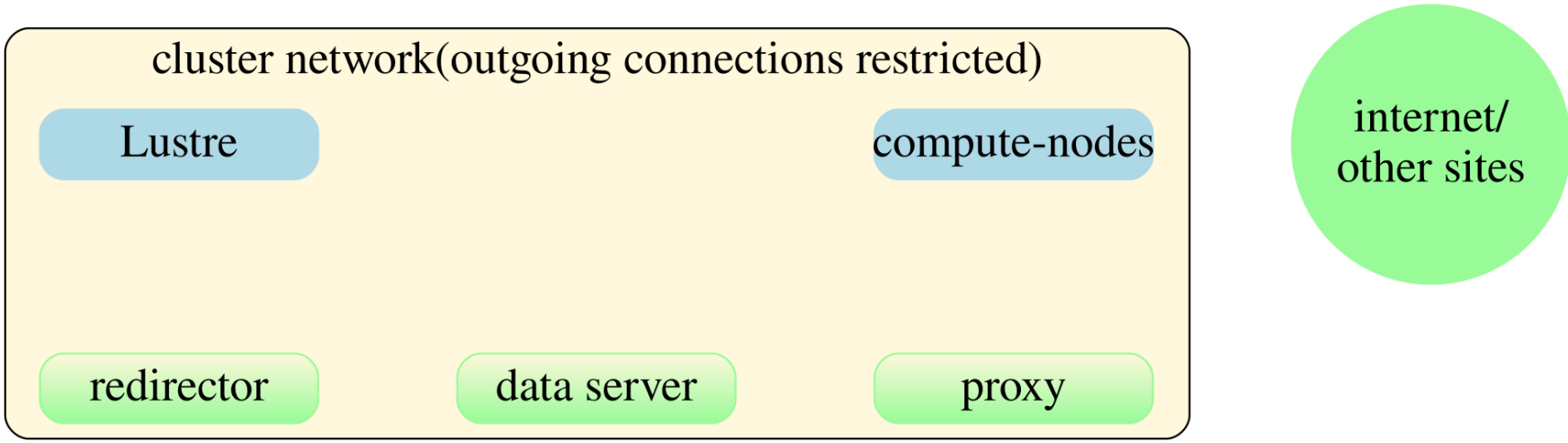
- Data server (ofs.osslib) plug-in that changes the usage statistics implementation
- Calls the Lustre-API for usage statistics to the current user's group's quota instead
- Configure server with ofs.osslib /path/to/libLustreOss.so & LustreOss.lustremount /path/to/lustre/mount

```
int LustreOss::StatVS(XrdOssVSInfo* sP, const char* sname, int
    updt) {
    char* buf = strdup(lustremount.c_str());
    struct qsStruct qs = getQuotaSpace(buf);
    sP->Total = qs.Total * 1024;
    sP->Usage = qs.Curr * 1024;
    sP->LFree = sP->Free = sP->Total - sP->Usage;
    return XrdOssOK;
}
```

XrdAliceTokenAcc is library, part of the standard alice SE XRootD installation

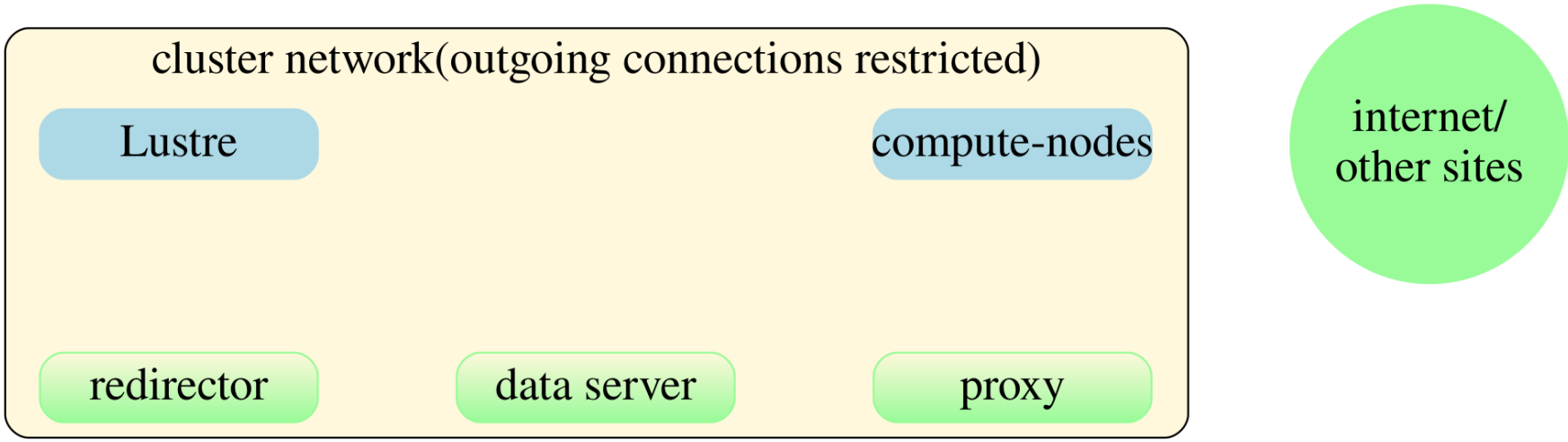
- Also allows the creation of symlinks during file access authorization
- Checks the envelope for read/write access
- Problems: symlinks became orphan when files are removed
- Solution: We implemented a corresponding symlink removal functionality (checks the envelope for delete access)

Accessing data via XRootD



- All jobs run inside a very restricted network
- Analysis jobs are I/O-bound => heavily depend on I/O performance

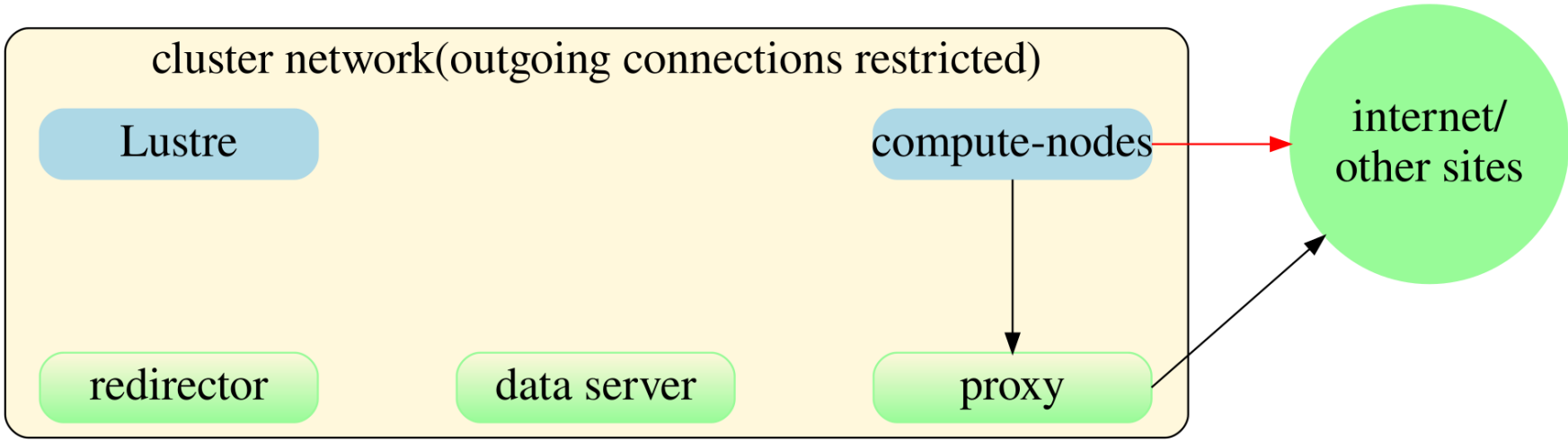
Accessing data via XRootD



Two kinds of data accesses

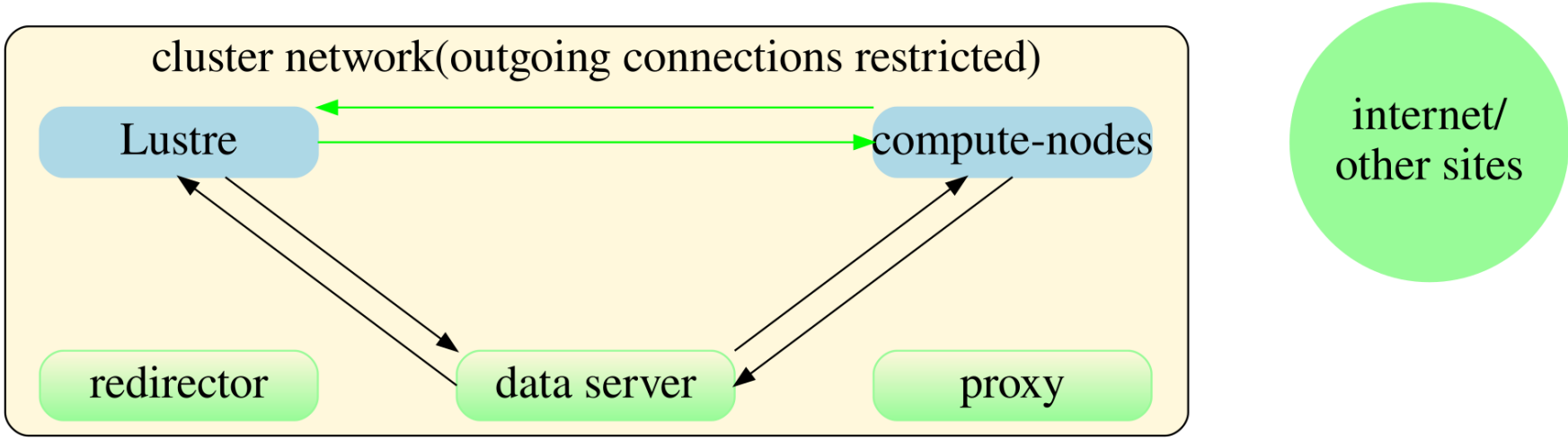
- Accessing data that is not stored at GSI
- Accessing data that is stored at GSI

Accessing data not stored at GSI



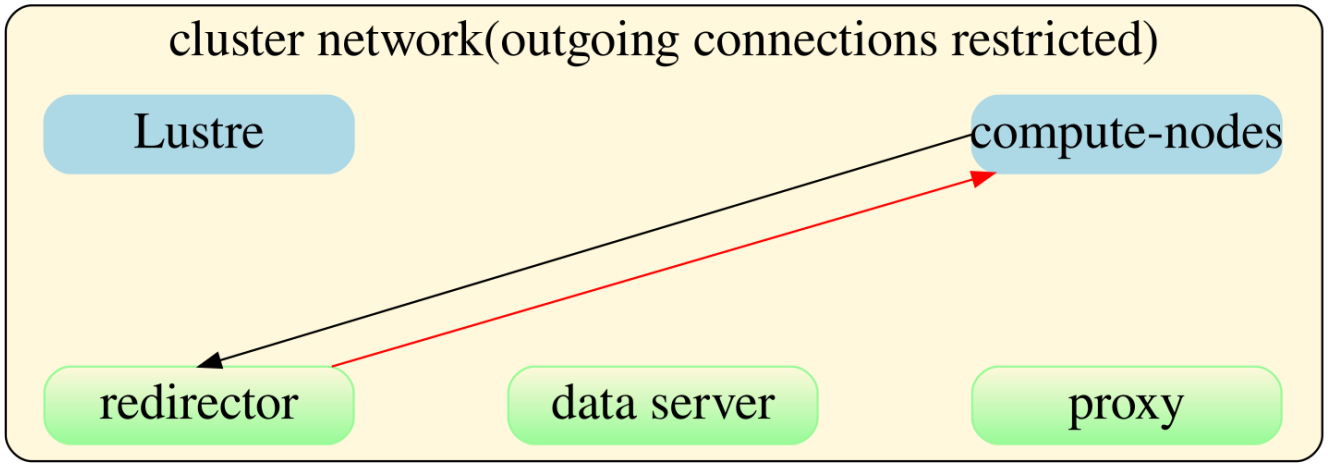
- GSI Computing nodes have very restricted connectivity
- => XRootD proxy tunnels traffic from the HPC environment to SE's outside GSI

Accessing data stored at GSI



- Doubles network traffic inside the infiniband network
- This procedure is a bottelneck in CPU & bandwidth to our setup (need to scale data servers with # of clients)

A redirector plug-in to redirect requests locally



RedirPlugin & Changes in XRootD

- Redirector(cms.ofslib) plug-in that redirects the client to the shared filesystem
- Plug-in only redirects newer clients to the shared filesystem, pre v4.8.0 clients are redirected to the data servers
- You need to use the v4 Client API (XrdCl)
- When the fs hangs, fs dependent behaviour (block indefinitely?...)
  -> XRootD devs are going to implement a timeout inside the v4 client, so the client can return an error and try another source

RedirPlugin & Changes in XRootD

RedirPlugin depends on XRootD v4.8.0

- A new part of the XrootD Client(XrdCl), the Localfilehandler, implements POSIX operations which allows circumvention of the XRootD dataserver by local redirections
- It was developed and integrated into the XRootD base in collaboration with the XRootD core developers (see commit 76108af & ef28e28 on xrootd/xrootd Github)

RedirPlugin::Locate 1/2

```
int RedirPlugin::Locate(XrdOucErrInfo &Resp, const char *path,
    int flags,XrdOucEnv *EnvInfo) {

    int rcode = nativeCmsFinder->Locate(Resp, path, flags,
        EnvInfo);
    // Get regular target host

    int pversion = Resp.getUCap() & 0x0000ffff;
    // Mask out client protocol version

    const char *ppath = theSS->Lfn2Pfn(path, buff,
        maxPathLength, rc);
    //get pfn
    ...
```

RedirPlugin::Locate 2/2

```
...
XrdNetAddr target(-1);
target.Set(Resp.getErrText());
 if (ClientPrivate && TargetPrivate &&
     pversion>=784 && !(flags & SFS_O_STAT) &&
         !(readOnlyredirect && !(flags & SFS_O_RDONLY)))
             {
      Resp.setErrInfo(
          -1,
        // set to -1 to redirect to a local path
          (localroot + std::string(ppath)).c_str()
        //set to the complete local path
                        );
   return SFS_REDIRECT;
 }
  return rcode;
```

```
#incomplete redirector config file
#set path to where you put the plugin

cms.ofslib="/usr/lib/xrootd/libRedirLocal.so"

#set the plug-in Localroot to your global,
#so it can redirect correcly

RedirLocal.Localroot="/sharedfs/xrootddata"

# Set to true/false, if you want redirect reads only (default is
     true)
RedirLocal.readOnlyredirect="true"
```

Local redirections & ROOT (TAlienFile)

- AF jobs will have to use a ROOT v6 with XRootD >= v4.8.0 and the new client API (XrdCl)
- Good news: ROOT6/Aliroot compiles just fine with new XRootD
- Bad news: TAlienFile is derived from TXNetFile (old XrdClient)

Local redirections & ROOT (TAlienFile)

Solution:

- Let TAlienFile be derived from TNetXNGFile(new XrdCl) instead
- Add a LFN parameter to TNetXNGFile ctor to pass LFN to TFile/TArchiveFile (like in TXNetFile)

Status: Working (including archive files), but:
hangs sometimes for 30s to 10m somewhere in TFile::Open() without error => We will investigate it
Then: fix it, create a PR to ROOT and run testjobs using a prototype environment in the meantime.

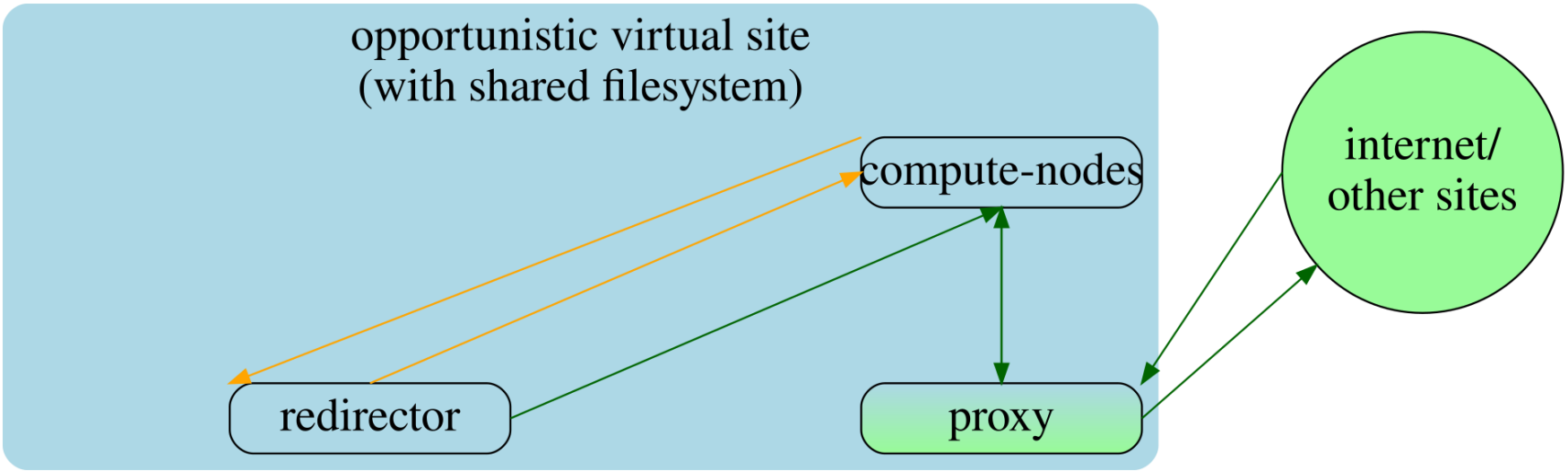
RedirPlugin

https://github.com/pkramp/RedirPlugin

Disk caching proxy for opportunistic resources

- Developed in collaboration with KIT
- Utilize opportunistic resources such as clouds for CMS
- =>Virtual site
- Challenges: Limited connectivity, data locality -> disk caching proxy

Disk caching proxy for opportunistic resources



Conclusion

- Lustre quota statistics plugin https://github.com/jknedlik/XrdLustreOssWrapper Status: deployed @AF Prototype
- AliceTokenAcc/tktokenauthz plug-in(symlink removal) Status: deployed @AF Prototype
- Plug-in to redirect clients to a shared filesystem directly https://github.com/pkramp/RedirPlugin Status: deployed @AF Prototype
- Local redirects through TAlienFile/ROOT Status:fix the hanging client
- Plug-ins and a test setup to utilize opportunistic resources https://git.gsi.de/dc/XRootD-Reps/xrootd-disk-cache-local-access Status: prototype running @bwHPC cluster NEMO(Freiburg)

Outlook

- Solve problems with TAlienFile
- Run performance tests with local redirections/TAlienFile & create a PR to ROOT
- Create a PR for XrdAliceTokenAcc

Thank you for your attention