

INTERNAL Submission plugin

ARC 6 camp

Umeå 7-9.11 2018

INTERNAL Submission plugin

- Especially aimed for restrictive HPC sites
- “Self-contained” ARC site: intended to run with ARC Control Tower installed on frontend
 - The local aCT pulls jobs from PanDA, and submits the jobs to ARC (running on the same machine, or at least sharing filesystem)

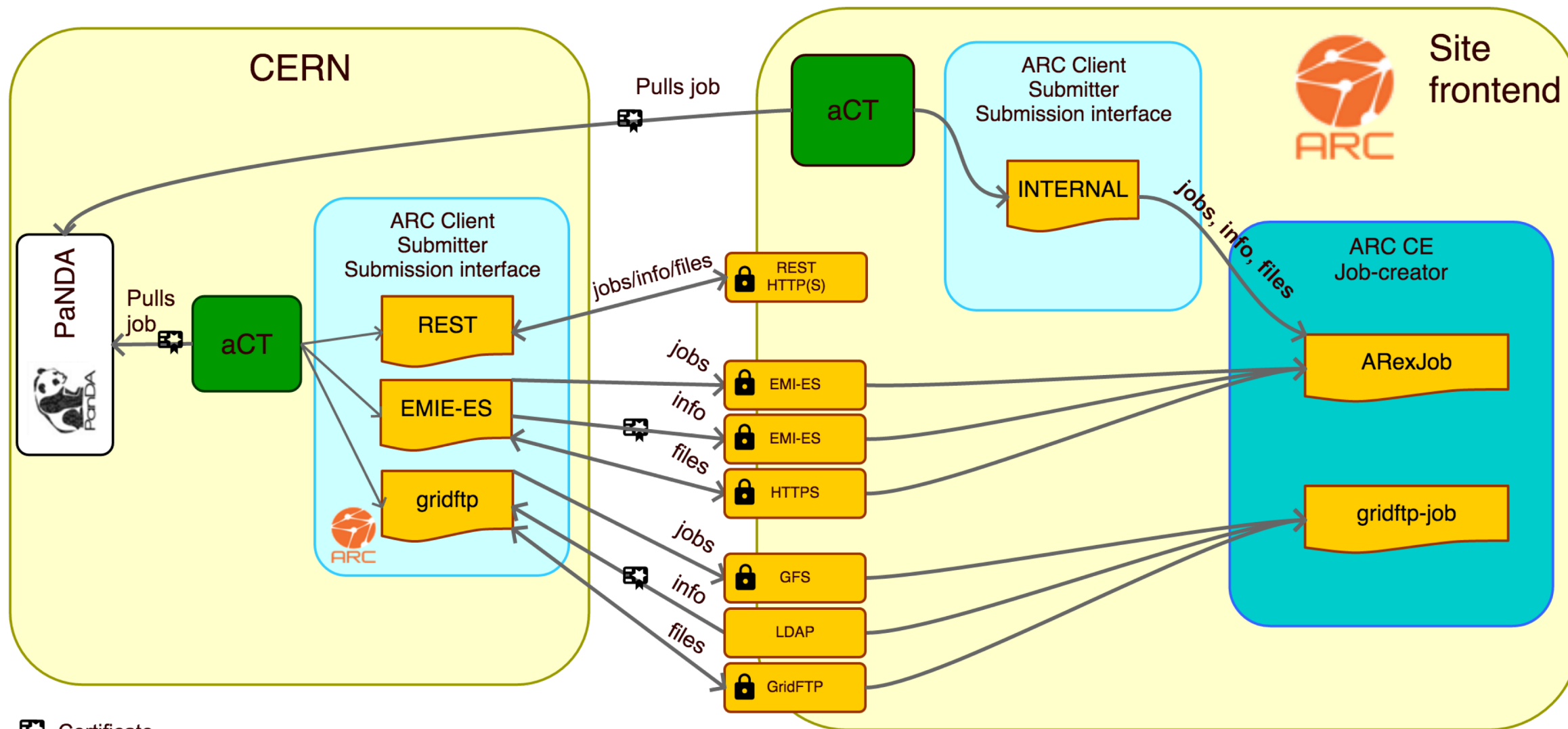
Lightweight:

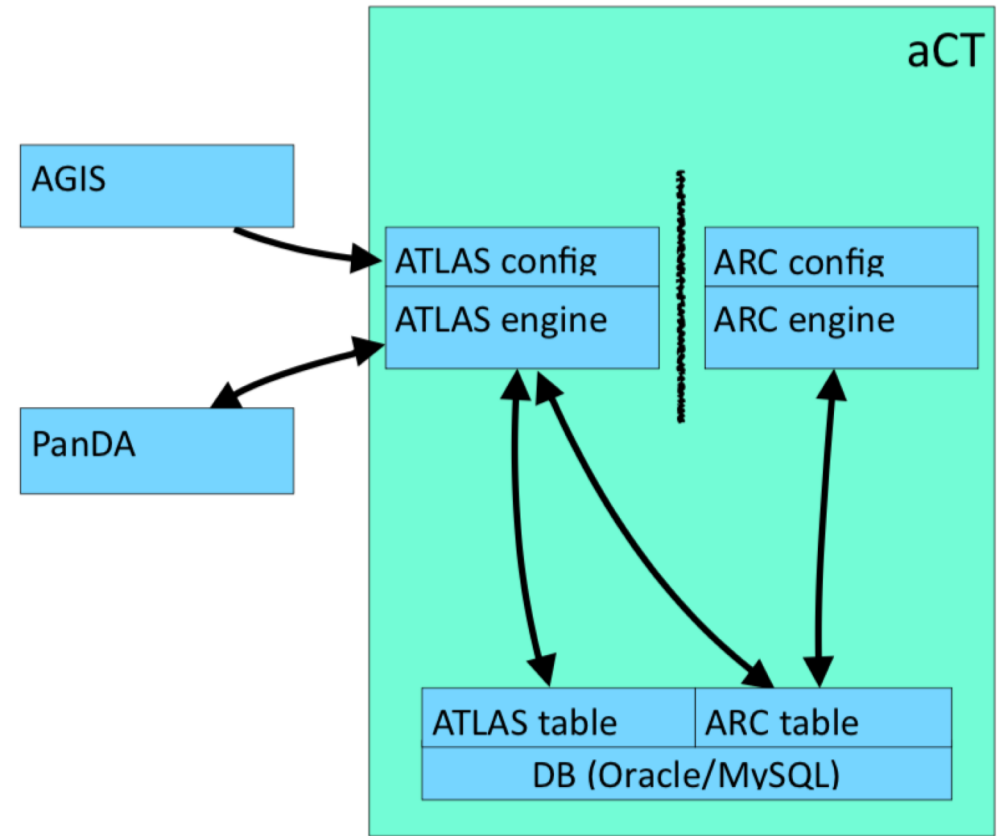
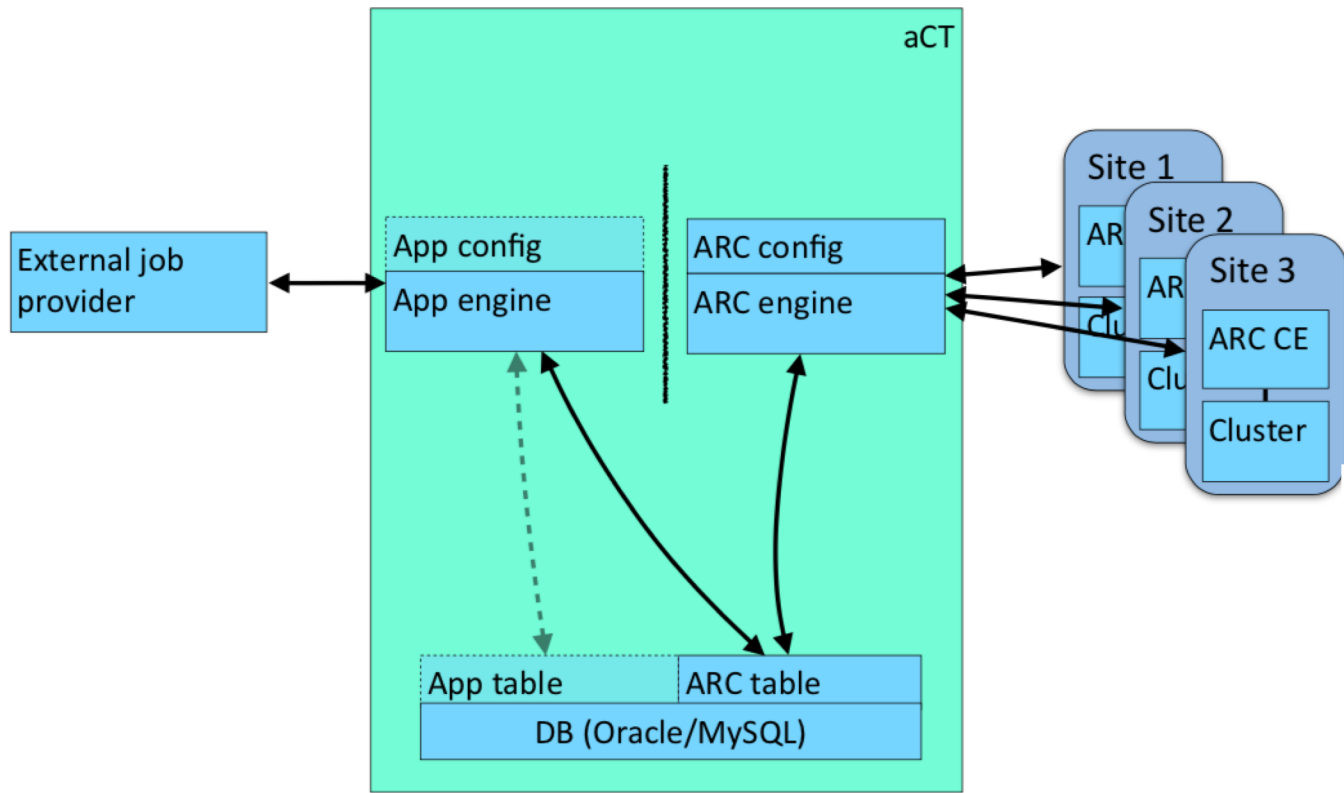
- No host certificate needed
- No webservice or gridftp service needed
- No ldap information system needed since the site has its own aCT and hence needs not advertise its existence
- ARC and aCT can be installed by normal user

Minimal set of services, no gridftp server, no emi-es, no ldap, no host certificate

→ Lightweight ARC-CE beneficial for installation, configuration and maintenance

Overview of the ARC-CE submission interfaces





Implementation overview

- The INTERNAL submission plugin (part of the ARC client) interacts with the parent plugin classes using the same API as the other plugins (e.g. gridftp plugin or emi-es).
- The INTERNAL plugin interacts directly with the A-REX memory and methods, and therefore is integrated as part of the A-REX service which belongs to the ARC-CE.
 - Both the ARC client and the ARC CE must be installed and on the same machine for the INTERNAL submission plugin interface to function.
 - The internal plugin is separated in its own package and must be installed in addition to nordugrid-arc-arex and nordugrid-arc-client nordugrid-arc-plugins-internal
- All interaction between the client and A-REX happens directly via files in the controldir or via A-REX memory.

INTERNAL plugin constructor

- SetAndLoadConfig()
 - Get config-file (arc.conf)
 - Push through parser to set up default values
- SetEndPoint()
 - config->Controldir()
- MapLocalUser()
 - Setup mapping of local user (user submitting job) to mapped user from arc.conf
 - Details: Aleksandr
- PrepareArexConfig()
 - From the config prepared in SetAndLoadConfig(), and arexconfig object is created.
 - Needed for the creation of the ArexJob()

Example job submission (manual)

- When a job is submitted via the INTERNAL submission interface the plugin creates an A-REX job object, which then takes care of creating all necessary files (like for instance the ARC job description in the ARC CE's controldir) and folders (sessiondir) for the job, in addition to creating a job ID.
- Once these files are present in the controldir A-REX adds the job to its joblist, and takes over the handling of the job from there.
- The INTERNAL plugin places any input files local to the client in the newly created sessiondir (with Arc::FileCopy)
 - Remaining remote input files are downloaded by the DTR

```
$ arctsub -c localhost -S org.nordugrid.internal hello.xrls
```

```
Job submitted with jobid:
```

```
file:///wlcg/session/po2LDmoAHWtnrp02tmaBI5UnABFKDmABFKDmQeMKDmABFKDmtdHX5n
```

Accessing Information About Jobs

- Job information evoked by calling `arcstat` is extracted from a combination of information stored in A-REX memory (job state) and the `job.ID.local` file in the `controldir` (session, stagein and stageout directories). An example call and its results is

```
$ arcstat
file:///wlcg/session/po2LDmoAHWtnrp02tmaBI5UnABFKDmABFKDmQeMKDmABFKDmtdHX5n
Job: file:///wlcg/session/po2LDmoAHWtnrp02tmaBI5UnABFKDmABFKDmQeMKDmABFKDmtdHX5n
Name: hello_internal
State: Accepted
Status of 1 jobs was queried, 1 jobs returned information
```


Retrieving Service Information

- As a site running in the INTERNAL mode is not accessible from the outside, any service retrieval information can only be done from within the site.

```
arcinfo -c localhost
```

Computing service:

Information endpoint: file://localhost

Submission endpoint: file://localhost (status: ok, interface:
org.nordugrid.internal)

- When arcinfo is called, the INTERNAL submission interface extracts the site information by directly accessing the info.xml file in the controldir.
 - TargetInformationRetrieverPluginINTERNAL class
 - The INTERNAL plugin reads the info.xml and outputs information in xml-format to the ARC client, which in turn displays it to the user.
 - TO-DO – Should the INTERNAL service at all be extracted to info.xml?

Controlling Job Execution

- Killing, cleaning and resubmitting jobs is initiated by direct call the existing ARexJob methods: Kill(), Clean(), Resume().
- These methods all place files in the controldir that the grid-manager acts upon, such as job.jobid.clean mark or job.jobid.cancel mark.

TO-DO

- aCT
 - Create ini-style configuration
 - Use sqlite instead of mysql
 - How to handle authentication with PanDA – at moment our test site is using central aCT long-lived proxy
- ARC INTERNAL plugin
 - Write UNIT test
 - Make INTERNAL submission plugin more fault tolerant + have some code review
 - Make packages installable as normal user to non-default path
 - rpm2cpio?
- Ansible scripts for installing and setting up INTERNAL ARC + aCT available, but only used by me – and only prepared for Red-hat based OS-es (partially for Debian)

Configuration file examples

```

<config>
<db>
  <type>mysql</type>
  <name>act</name>
  <user>centos</user>
  <password>xxxxxxx</password>
  <host>localhost</host>
  <port>3306</port>
</db>

<loop>
  <periodicrestart>
    <actsubmitter>120</actsubmitter>
    <actstatus>600</actstatus>
    <actfetcher>600</actfetcher>
    <actcleaner>600</actcleaner>
  </periodicrestart>
</loop>

<tmp>
  <dir>/home/centos/software/aCT/act-test/tmp</dir>
</tmp>

<actlocation>
  <dir>/home/centos/software/aCT/src/</dir>
  <pidfile>/home/centos/software/aCT/act-test/act.pid</pidfile>
</actlocation>

<logger>
  <level>debug</level>
  <arcllevel>debug</arcllevel>
  <logdir>/home/centos/software/aCT/act-test/log</logdir>
  <rotate>25</rotate>
</logger>

<atlasgiis>
  <timeout>20</timeout>
</atlasgiis>

```

```

<voms>
  <vo>atlas</vo>
  <roles>
    <item>production</item>
  </roles>
  <bindir>/grid/software/bin</bindir>
  <proxylifetime>345600</proxylifetime>
  <minlifetime>259200</minlifetime>
  <proxypath>/grid/atlact1.rfc.long.proxy</proxypath>
  <cacertdir>/etc/grid-security/certificates</cacertdir>
  <proxystoredir>/grid/software/aCT/proxies</proxystoredir>
</voms>

```

aCTConfigARC

```

<config>

<executable>
  <ptarurl>http://pandaserver.cern.ch:25080;cache=check/cache/pilot/pilotcode.tar.gz</ptarurl>
  <ptarurlrc>"http://project-atlas-gmsb.web.cern.ch;cache=no/project-atlas-gmsb/pilotcode-rc.tar.gz"</ptarurlrc>
</executable>

<joblog>
  <urlprefix>/var/www/html/act/jobs</urlprefix>
  <dir>/data/user/act/www/jobs</dir>
</joblog>

<agis>
  <server>http://atlas-agis-api.cern.ch/request/pandaqueue/query/list/&#63;json&#38;preset&#61;schedconf.all</server>
  <objectstores>http://atlas-agis-api.cern.ch/request/ddmendpoint/query/list/&#63;json&#38;type&#91;&#93;&#61;OS_LOGS&#38;type&#91;&#93;&#61;OS_ES</objectstores>
  <jsonfilename>/grid/software/aCT/tmp/agis.json</jsonfilename>
  <osfilename>/grid/software/aCT/tmp/oses.json</osfilename>
  <pilotmanager>aCT</pilotmanager>
  <maxjobs>0</maxjobs>
</agis>

<panda>
  <server>https://pandaserver.cern.ch:25443/server/panda/</server>
  <heartbeattime>1800</heartbeattime>
  <threads>1</threads>
  <getjobs>1</getjobs>
  <schedulerid>aCT-atlact1-1</schedulerid>
  <timeout>60</timeout>
  <minjobs>10</minjobs>

  <sites>
    <site>
      <name>UIO_CLOUD</name>
      <endpoints>
        <item>local://localhost:/main</item>
      </endpoints>
      <maxjobs>20</maxjobs>
      <corecount>4</corecount>
    </site>
  </sites>
</panda>

</config>

```

aCTConfigATLAS

arc.conf example

```
[common]
hostname=localhost

[authgroup:any]
all=yes

[mapping]
map_to_user = any centos:centos

[lrms]
lrms=slurm

[arex]
controldir=/grid/control
sessiondir=/wlcg/session
runtimedir=/wlcg/runtime
shared_scratch=/wlcg
cachedir=/wlcg/cache
cachelifetime=1d

logfile=/grid/arclog/arex_arc6.log
joblog=/grid/arclog/gm-jobs_arc6.log

[infosys]
logfile=/grid/arclog/infoprovider_arc6.log

[infosys/cluster]
cluster_alias=UIO_CLOUD
nodeaccess=outbound

[queue:main]
```

Configuration of ARC (NB! ARC 5 example!)

```
# This file is managed by ansible
# UiO UH-sky arc.conf

[common]
hostname="localhost"
lrms="SLURM"
slurm_wakeupperiod="60"
x509_cert_dir="/etc/grid-security/certificates"
shared_filesystem="yes"
tmpdir='${LOCALTMP}'

[grid-manager]
debug="5"
pidfile="/grid/grid-manager.pid"

controldir="/grid/control"
sessiondir="/wlcg/session"
runtimedir="/wlcg/runtime"
shared_scratch="/wlcg"

logfile="/grid/grid-manager.log"
wslogfile="/grid/ws-interface.log"
joblog="/grid/gm-jobs.log"
logsize="1000000 52"

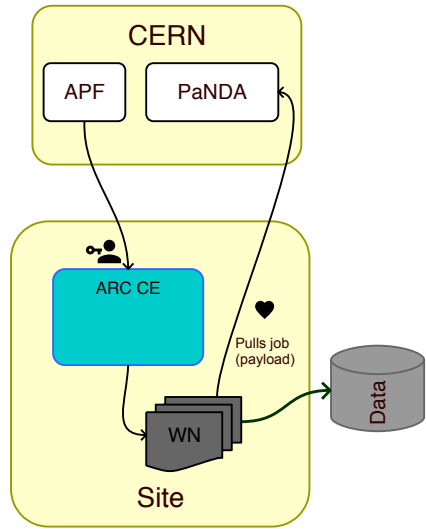
cachedir="/wlcg/cache"
cachesize="80 70"
cachelifetime="5d"
shared_filesystem="yes"

maxjobs="10 10"
wakeupperiod="30"
enable_arc_interface="no"
enable_emies_interface="no"
maxtransfertries="10"
```

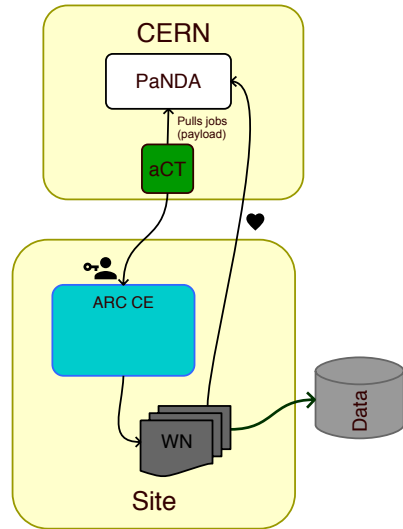
```
[infosys]
lrms="SLURM"
providerlog="/grid/infoprovider.log"

[cluster]
hostname="localhost"
cluster_alias="UIO_CLOUD"
cluster_location="NO-0316"
cluster_owner="University of Oslo (USIT)"
nodeaccess="outbound"
architecture="x86_64"
opsys="CentOS"

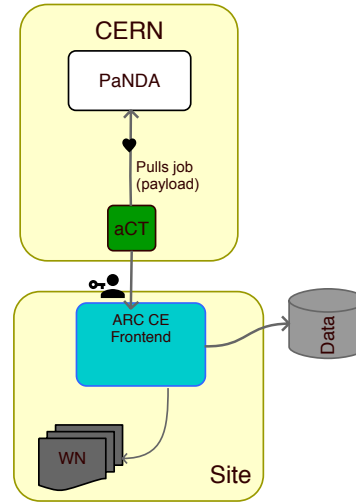
[queue/main]
queue_name="main"
comment="Main queue"
scheduling_policy="FIFO"
comment="UH-cloud LOCAL queue for grid jobs"
architecture="x86_64"
cachetime="90"
timelimit="60"
sizelimit="5000"
totalcpus="16"
```

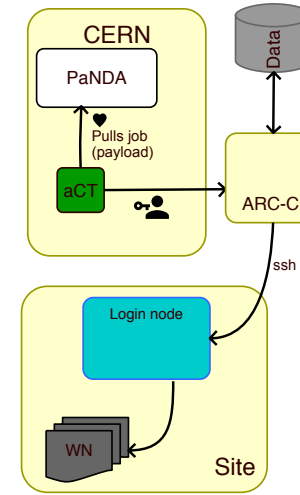
Pilot factory



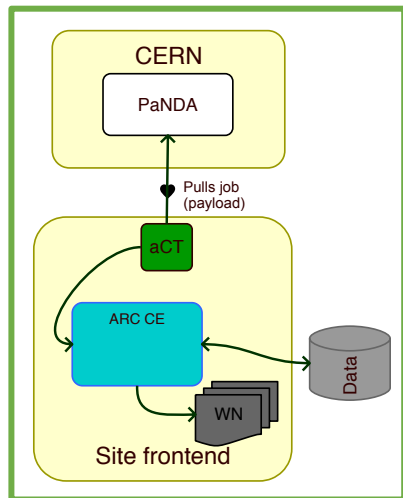
True pilot



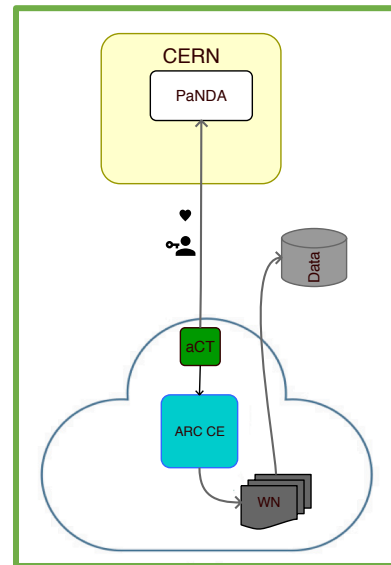
NDGF mode



ssh-mode



INTERNAL mode HPC



INTERNAL mode cloud

Nordugrid ARC CE modes

