JAliEn status

ALICE T1/T2 Workshop
Seoul, 2024-04-16

costin.grigoras@cern.ch

Deployment status

~3 years since we have started the transition of Grid sites to JAliEn

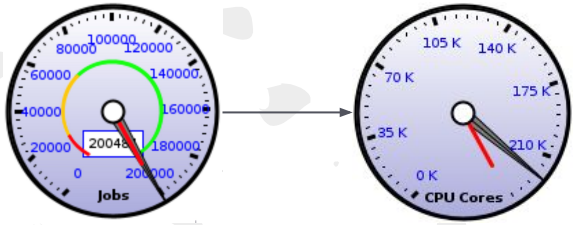
- 28 tags since the last workshop, 1.5.8 to 1.8.6

Fully deployed on all sites

- 8 whole node queues
- All others are 8+ CPU cores queues

In the process of migrating to EL 9

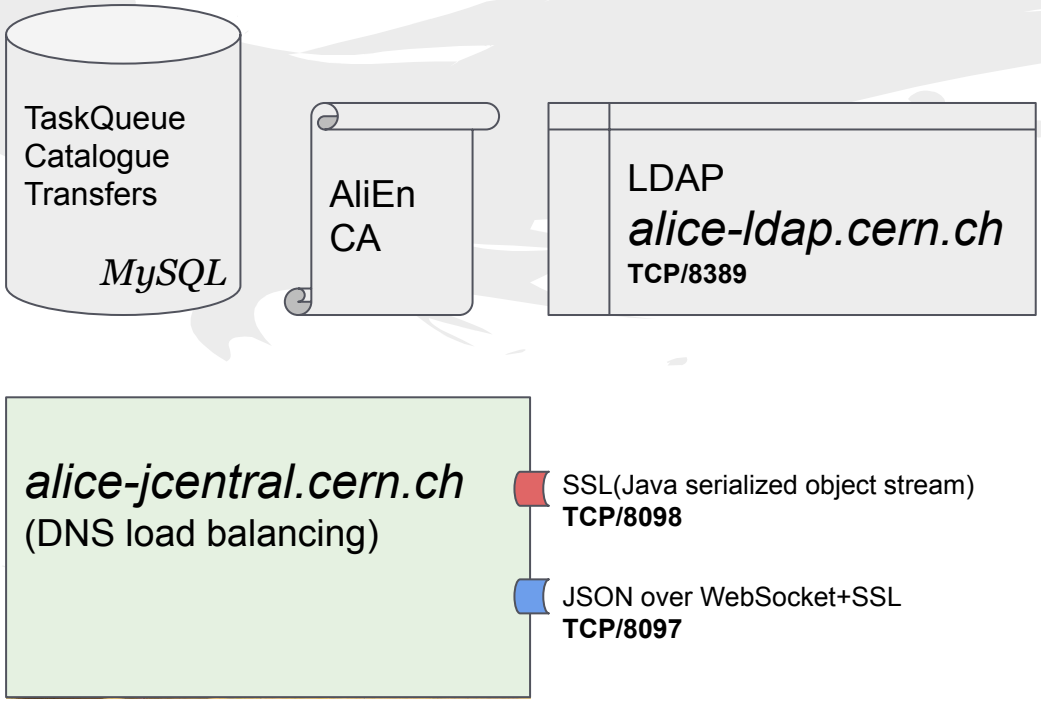
- 38% RHEL and Alma 9
- 58% CentOS and SL 7
- 4% Alma 8 (EPN nodes) and other variants



OS not a problem any more with containers used for everything

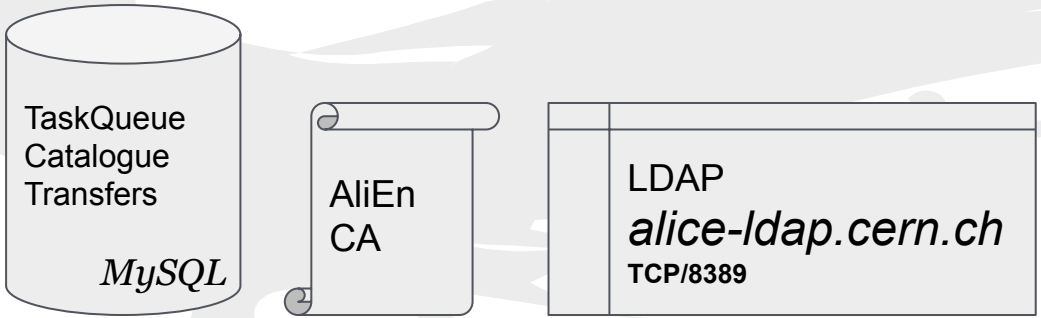
- But only **EL 9** fully supports *cgroupsv2*

Central services layout



- Identical client-facing services
- Background “optimizers”
- Transfer agents (3rd party or relay)
- Easily scalable
- Same *.jar* from CVMFS can be used as server, agent or embedded as a library
- Two connection options
  - Native Java ObjectStream
  - WebSockets + JSON
- Embedding an Apache Tomcat engine
- Exclusive use of X.509 authentication
  - No proxies!
- Same old SE envelopes for per-file and per-operation
  - Soon **JWT**

Central services layout



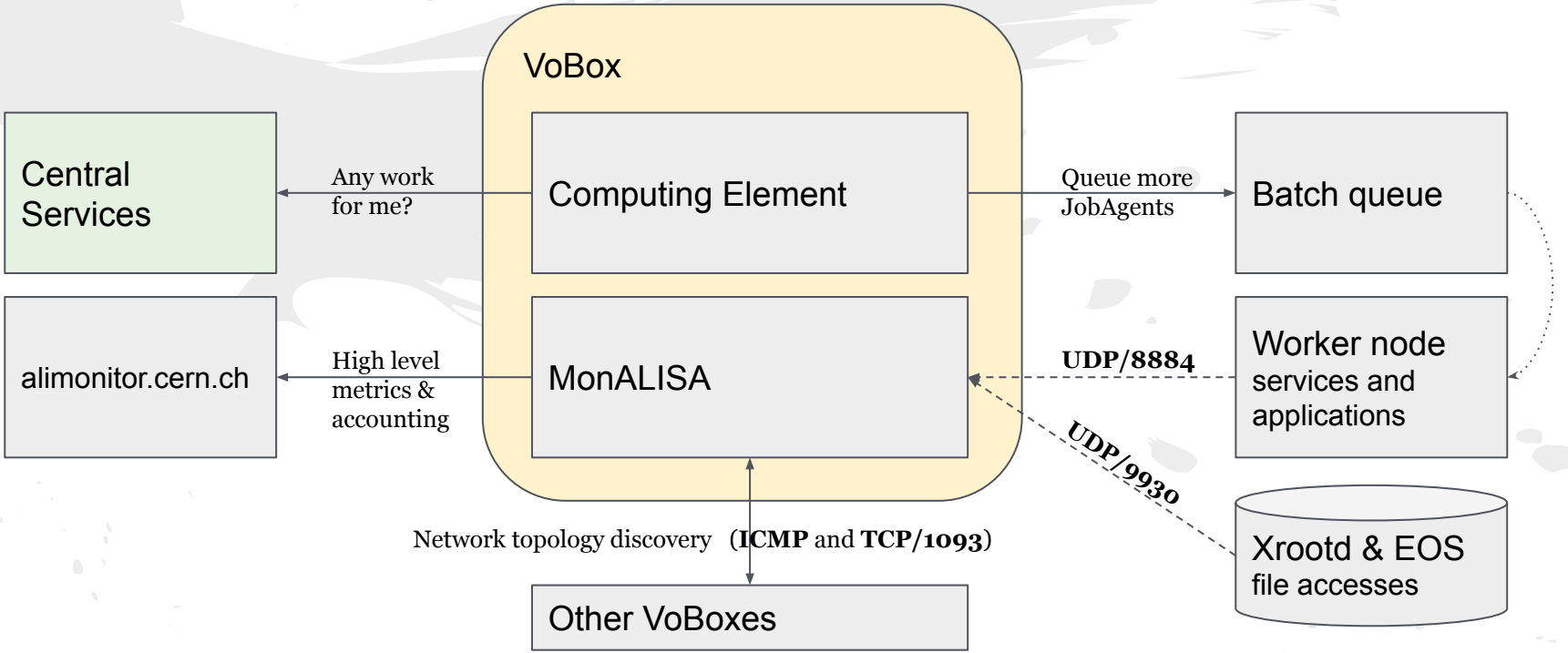
SSL(Java serialized object stream)
TCP/8098

JSON over WebSocket+SSL
TCP/8097

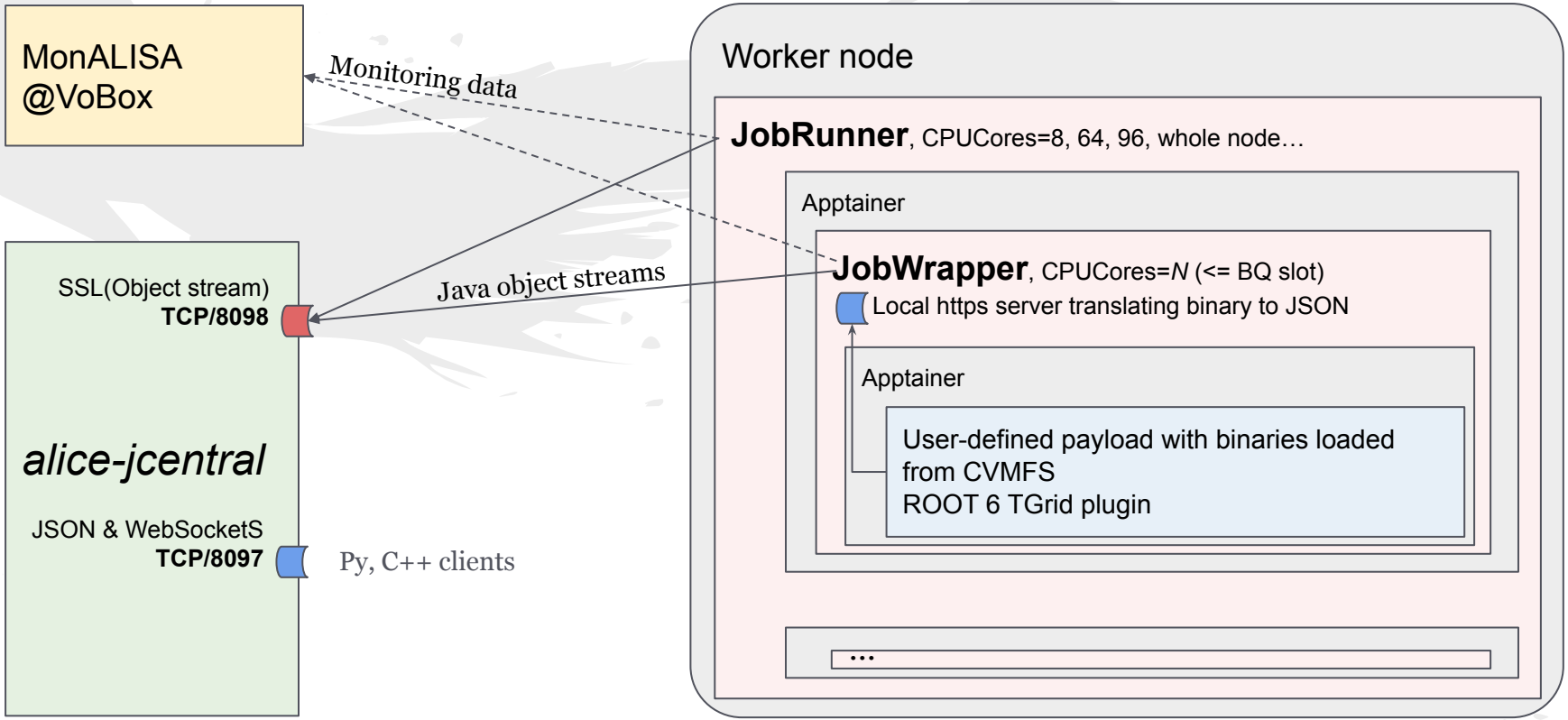
`128.142.249.0/24`
`2001:1458:301:54::/64`

- Identical client-facing services
- Background “optimizers”
- Transfer agents (3rd party or relay)
- Easily scalable
- Same *.jar* from CVMFS can be used as server, agent or embedded as a library
- Two connection options
  - Native Java ObjectStream
  - WebSockets + JSON
- Embedding an Apache Tomcat engine
- Exclusive use of X.509 authentication
  - No proxies!
- Same old SE envelopes for per-file and per-operation
  - Soon **JWT**

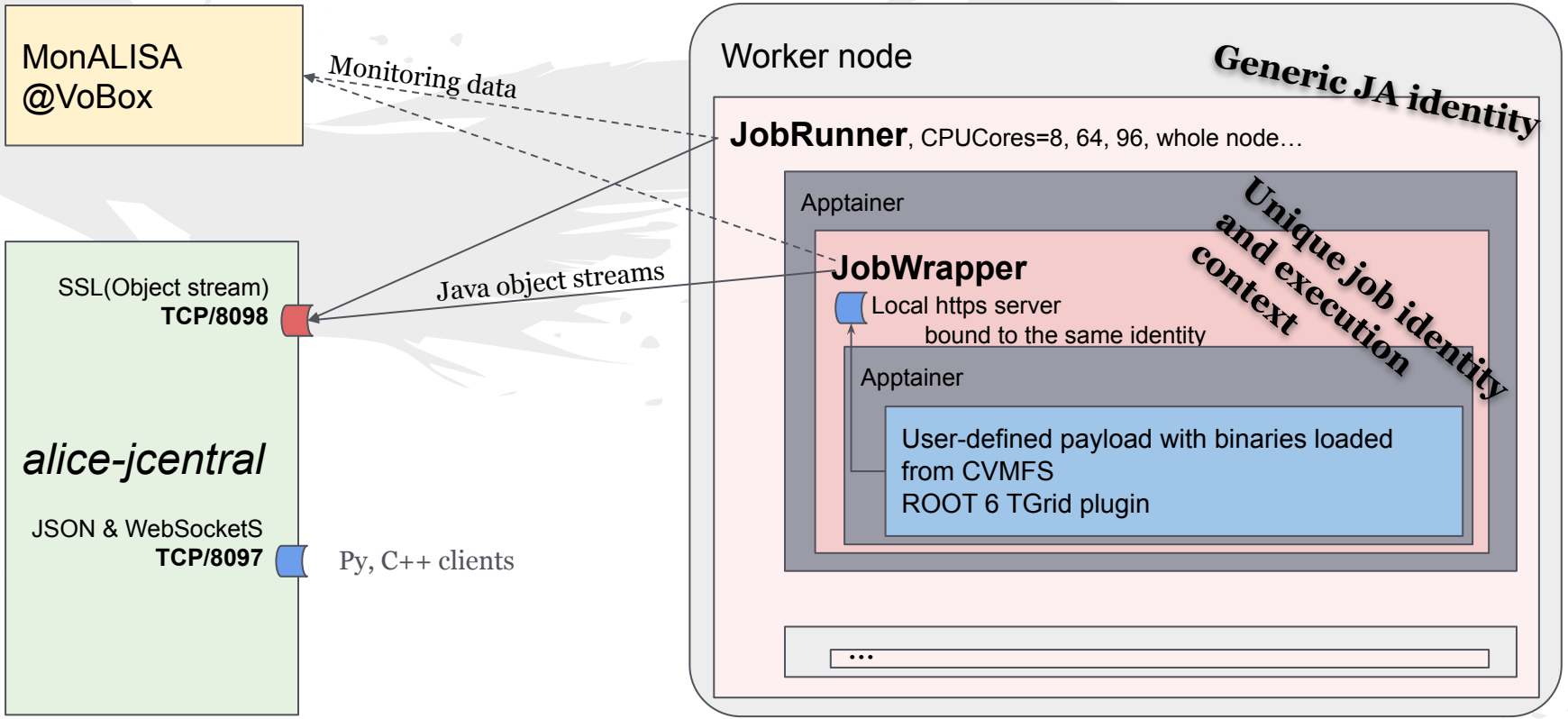
Site services / VoBox



Worker node components



Worker node components



Worker node execution

- Configurable slot size (no. of CPU cores)
  - Automatically adapt to **whole node** resources where available
  - Our preferred strategy to accommodate varying computing and memory needs from jobs
- Free resources advertised to the JobBroker
- A job can match any number of cores <= slot size
- Isolated instances
  - Container execution (container image picked up from CVMFS *f*(software deps))
  - Per job credentials (X.509 pair issued by our CA), with limited capabilities
  - Constrained resources (Core pinning with *taskset* -> *cgroupsv2* when possible)
- Continuous monitoring of payload resource usage
  - Accounting and preempting them if running over boundaries

Changelog highlights

1.5.9 Look for *core* files in the sandbox -> `ERROR_E`

1.6.1 Expose NVidia and AMD cards to the containers

1.6.2 *@inheritlocation* JDL output specs -> same as input

1.6.3 Start using *cgroupsv2* for memory limiting

1.6.4 CPU pinning detection & set using *taskset*

1.6.6 Support for arbitrary resource brokering, based on SiteSonar

1.6.9 Assign unused CPU cores in a slot to the remaining running payloads

1.7.1 Package publishing integrated with the build system

1.7.2 ARC 7 support

1.7.5 Nested containers to isolate execution environments

1.7.7 Optimize tape recalls with vector operations

1.7.9 Experimental *aarch64* support

1.8.0 SuperFacilityAPI as a new BQ implementation

1.8.1 Build-platform aware packages

1.8.2 Support *cgroupsv2* delegation in HTCondor 23.1+

1.8.3 Default container is EL9, fallback to C7 for old sw

aarch64 container and Apptainer binaries

1.8.5 Job optimizers (accounting, priority, expired, splitting)

cgroupsv2 delegation in HTCondor & Slurm for CPU and memory

1.8.6 SciTag decoration of all file accesses

Grid clients

Java native: `alienv enter JAliEn`

- And use `$CLASSPATH` as your only dependency

Any other language: the WebSocketS + JSON endpoint

- One command line request -> JSON formatted reply
- Default for users: *alice-jcentral.cern.ch:8097*
- For jobs: localhost access on `$JALIEN_HOST:$JALIEN_WSPORT`

Use the full Grid certificate, user token or the job token indicated by

- `$JALIEN_TOKEN_KEY / $JALIEN_TOKEN_CERT`

Fully implemented for ROOT and in use since we moved to ROOT6

And as the Python+Xrootd bindings Grid shell (or library)

- `alienv enter xjalienfs`

alienpy client shell

By Adrian

Current version : 1.6.0, diff relative to 10 Nov 2023 (1.5.0)

27 files changed, 855 insertions(+), 296 deletions(-)

Have own tests battery and have them run in *alidist* CI

XRootD::Cp : catalogue MD5 can be missing, so ignore it

XRootD::Cp : act on value of `ALIEN_SITE` env var

ccdb :: command improvements and fixes

Use RCT information when run number is requested

Add utilities for converting run number->timestamp and content mirroring

`jobInfo` : fixes for timestamp and extraction of node hostname

`ALIENPY_NO_HISTORY` : disable loading and usage of *readline* module

remove requirement of *gnureadline*

On Linux, the system *readline* is already _always_ present on both EL and Debian strains (the devel should be installed for python compilation to pick up readline usage)

On Darwin (macos) is the only usable *readline*, but not compatible with python 3.12 at this moment : if not present, then it will not be used

Site Sonar

A flexible and extensible Grid infrastructure monitoring tool

Reports data from ~10,000 Grid nodes daily

Invoked at the beginning of the execution of the *JobRunner* to collect the information of the current node and report to Central Services

Collected information is fed to an ELK stack daily

Data is monitored, analyzed and visualized using Elasticsearch and Kibana

38 probes currently defined (`/cvmfs/alice.cern.ch/sitesonar/`)

Now also used for job brokering on arbitrary resource specifications

- AVX support, GPU or CPU model etc.

IPv6 support

SiteSonar WN probe

- 61% of the WNs are ok (3% can't resolve IPv6 and 36% cannot connect)

All components are IPv6 ready

- Java, Python, Xrootd 5+ (client and server)

Central services see ~70% connections on IPv6

- These are job slots

IPv4 still required for legacy binaries (ROOT5 and Xrootd 3.x)

Exactly half of the VoBoxes are dual stacked

92% of the storage volume is dual stacked & working (== **1.5 y ago**)

- 6 sites still don't have it, others show various IPv6 errors

Data management

- File crawler for a sampling of current problems
  - Health (exists, can be read, checksum matches)
  - Performance (throughput and stat time)
- Remote operations to repair content
  - Dark data can be inferred with recursive `ls`
  - Lost data recovery from lists provided by sites
  - 3rd party transfers to re-establish consistency
  - Catalogue cleanup if that's not possible

Data and ops over http

EOS features

- `fsck` for reporting and repairing
- HTTP endpoint for data access
- Access to `fsck` reports over the same http port
  - Unprivileged account with access just to this list
- Available from 5.2+
- See Andreea's talk at the EOS workshop

fsck report and repair

Set *scaninterval* for space and all filesystems

```
space config <space-name>(ex:default) space.scaninterval=<sec>
fs config <fs-id> scaninterval=<sec>
```

Activate collection or repair threads

```
fsck config toggle-collect [<threads_number>]
fsck config toggle-repair [<threads_number>]
```

HTTP(s) data and REST API

```
xrd.protocol XrdHttp:1094 /usr/lib64/libXrdHttp.so
http.exthandler EosMgmHttp /usr/lib64/libEosMgmHttp.so
eos::mgm::http::redirect-to-https=1
xrd.tls  /etc/grid-security/daemon/hostcert.pem /etc/grid-security/daemon/hostkey.pem
xrd.tlsca  certdir /etc/grid-security/certificates/
http.gridmap /etc/grid-security/grid-mapfile
EOS_MGM_ENABLE_REST_API=1
```

(in /etc/sysconfig/eos_env)

Xrootd and HTTP can run on the same port, no need to set up a different firewall for it

SciTags

Experiment and activity accounting of network usage

Two methods of reporting

- IPv4: flow marking, UDP firefly sent to R&E collectors
- IPv6: flow label, part of the IPv6 header, no need for extra packets

EOS only supports flow marking, for both protocols

All ALICE operations are tagged with values from here

- `xrdcp root://….?authz=<token>&scitag.flow=330&eos.app=JobWrapper`
- 5<<6 + 10 (5==ALICE, 10==Data access)

Can be enabled on recent EOS and Xrootd versions (5.6.7+)

Data to be sent to the nearest collector

`{eu,us,global}.scitags.org` - more to come

SciTags config in EOS/Xrootd

```
xrootd.pmark use firefly scitag
xrootd.pmark domain any
xrootd.pmark debug
xrootd.pmark trace
xrootd.pmark ffdest 198.128.151.27:10514
```

For now use the IPv4 address of `global.scitags.org`

- Be ready to change the config at a later time
- `eosalice.cern.ch` is running it in production

JWT for data access

Industry standard, cross-experiments implementation

No need for an extra auth plugin for ALICE

Similar to our current tokens, 1-to-1 mapping of operations

Under development in collaboration with the EOS team

```
<authz>
  <file>
    <access>read</access>
    <turl>root://eosalice.cern.ch:1094//…-2c44fd849358</turl>
    <lfn>/alice/cern.ch/user/g/grigoras/wn.xml</lfn>
    <size>755</size>
    <guid>CF54C1A0-5E90-11E8-BEE3-2C44FD849358</guid>
    <md5>6f0d829a0f3fc8295f48c204a8053a75</md5>
    <pfn>/00/44960/cf54c1a0-5e90-11e8-bee3-2c44fd849358</pfn>
    <se>ALICE::CERN::EOS</se>
  </file>
</authz>
```

```
{
  "aud": "https://wlcg.cern.ch/jwt/v1/any",
  "sub": "aliprod",
  "nbf": 1711492372,
  "scope": "storage.write:/eos/dev/alice/test1",
  "iss": "https://alice-jcentral.cern.ch:8098/",
  "exp": 1711495972,
  "iat": 1711492372,
  "jti": "NDdjNmYy…hmMmE1"
}
```

Credits

Performance analytics	Elena
Resource hunter	**Kalana**
Master of Grid operations	**Maarten**
Farseer expert	Volodymyr
Task maestro	**Haakon**
Dept of corrections chief	**Marta**
Job optimizer	**Jørn-Are**
Production coordinator	**Irakli**
Monitoring guru	Cristi
Supercomputer hero	**Sergiu**
Data wizard	Alice
Visual effects coordinator	Mateea
Shipping expert	**Maksim**
Customer success manager	Adrian
Data doctor	Andreea