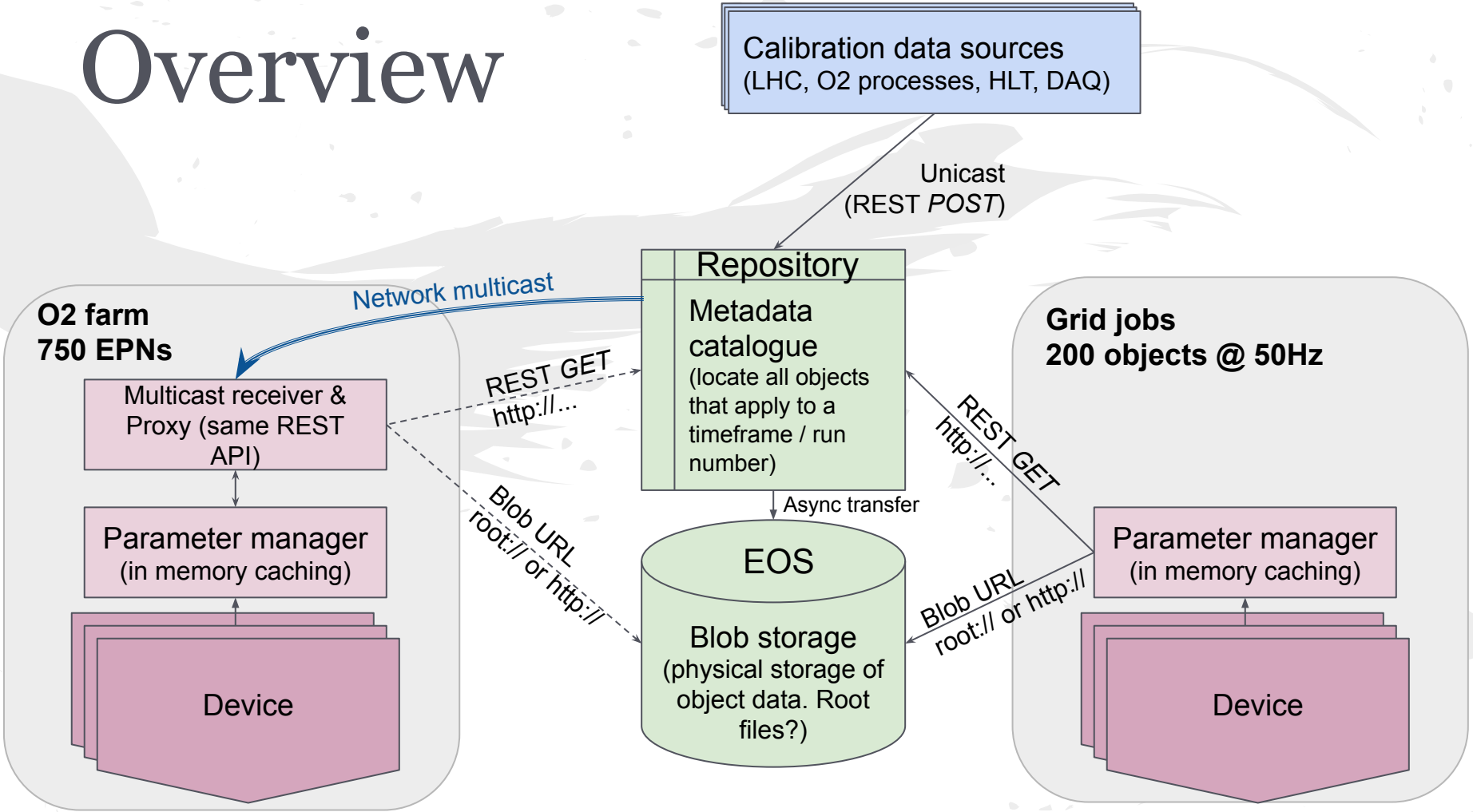
WP10

CCDB
Conditions DB for Run 3

costin.grigoras@cern.ch

Overview


Calibration data sources
(LHC, O2 processes, HLT, DAQ)
Unicast
(REST *POST*)
Repository
Metadata catalogue
(locate all objects that apply to a timeframe / run number)
Network multicast
O2 farm
750 EPNs
Multicast receiver & Proxy (same REST API)
Parameter manager
(in memory caching)
Device
REST GET
http://...
Blob URL
root:// or http://
Async transfer
EOS
Blob storage
(physical storage of object data. Root files?)
Grid jobs
200 objects @ 50Hz
REST GET
http://...
Blob URL
root:// or http://
Parameter manager
(in memory caching)
Device

Status

- Central repository test instance is online:
  - http://ccdb-test.cern.ch:8080/
  - More details in these slides
- First client implementations from Giulio and Laurent
- GSoC project on the multicast object distribution

DPL client

- An initial implementation of client code is available in DPL
- Key (URL path) is automatically generated from on the class name
- Client code has to know the object type to cast to (templated call)

DPL client example

Full example here

```
19  // This is how you can define your processing in a declarative way
20  WorkflowSpec defineDataProcessing(ConfigContext const&)
21  {
22    return WorkflowSpec{
23      {
24        "A",
25        { InputSpec{ "somecondition", "TST", "FOO", 0, Lifetime::Condition },
26          InputSpec{ "sometimer", "TST", "BAR", 0, Lifetime::Timer } },
27        { OutputSpec{ "TST", "A1", 0, Lifetime::Timeframe } },
28        AlgorithmSpec{
```

DPL client (todo list)

- Extract the code as a standalone library
- Merging with Barth's code
- Implement ETag-based internal caching mechanism to optimise access to the repository
- Use object metadata to describe serialization method and object type

Muon local object repo

- Using the local filesystem-based CCDB repository for development purposes
- Importing the OCDB objects as starting point
- Using the run number-based validity intervals
- 1307 objects in 1290 runs (~220MB)
- Packaged in a container together with the respective web server to access them

Muon local object repo (2)

- Limitations found and fixed in how the filesystem timestamps were used in matching validity intervals
  - Not all filesystems support high precision timestamps, some truncate the values
  - Server code automatically switches to using the *.properties* (object metadata) content in these cases => slower but good enough for development

Multicast distribution

- Highest rate so far: TPC IDC
  - 1MB @ 50Hz (Assuming 2x compression and aggregation at timeframe granularity)
  - 0.5Gbps for distribution of the CCDB objects
  - 50TB / data taking period
  - x number of external replicas, should be taken into account when planning for the storage

Multicast distribution (2)

- Network topology will impact the design
  - Single broadcast domain? If not we might need another ‘repeater’ service in each.
- Submitted a GSoC project for implementing the sender and the EPN receiver+local cache
- Application deadline in one week, selection in progress, more news in a few weeks
  - 6 complete applications so far