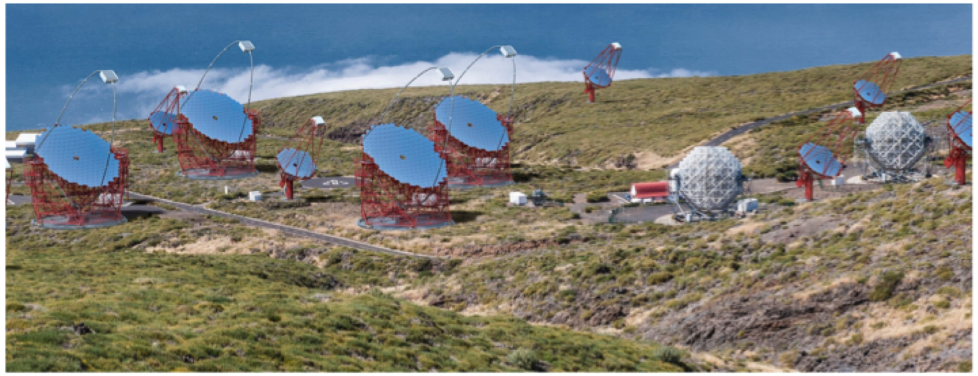
Provenance
LST1 Large Size Telescope for CTA

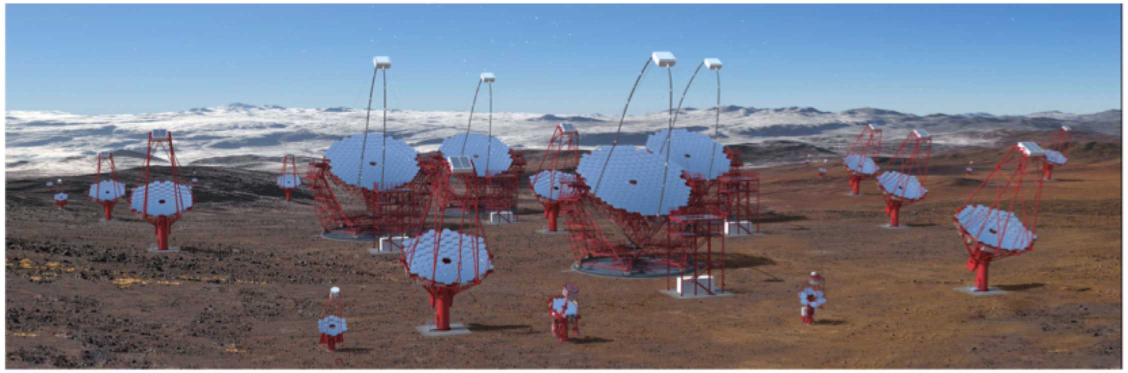
José Enrique Ruiz (IAA – CSIC)

ESCAPE WP4 Provenance Workshop

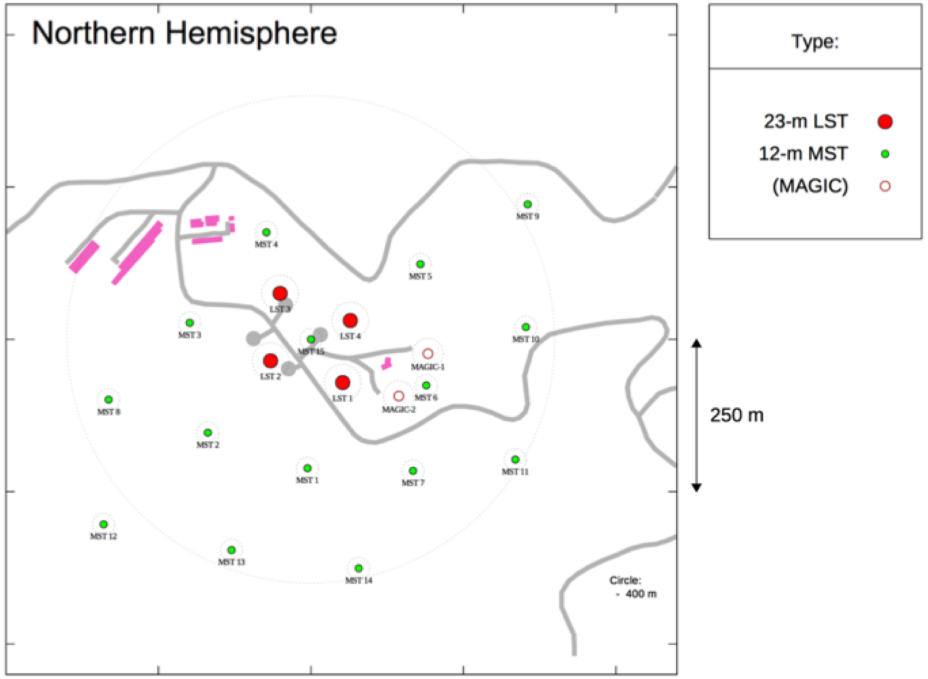
08/09/2020

Observatory sites and arrays



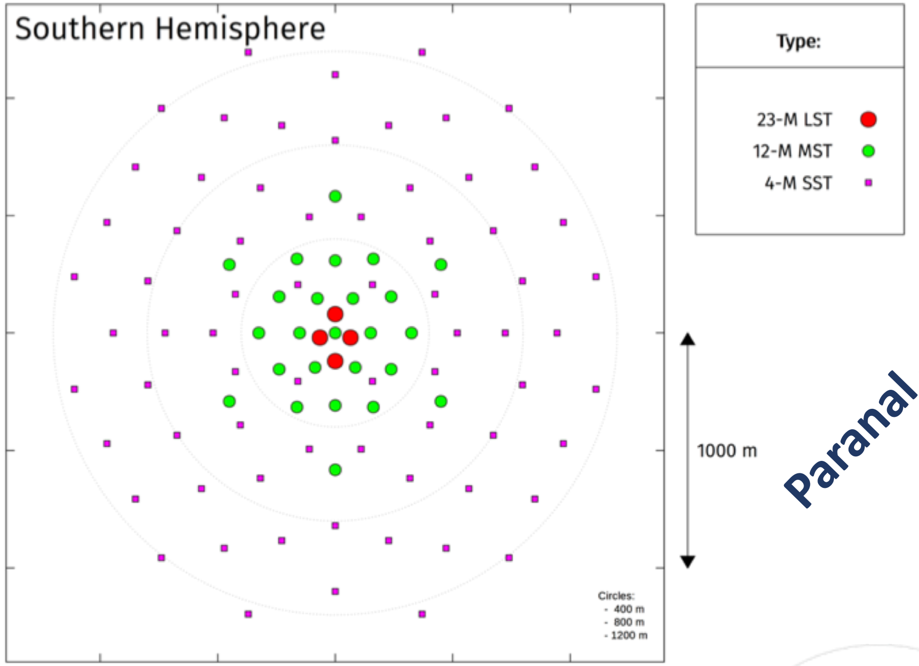


La Palma - Spain



4 LS4 LSTs, 15 MSTs

Paranal - Chile



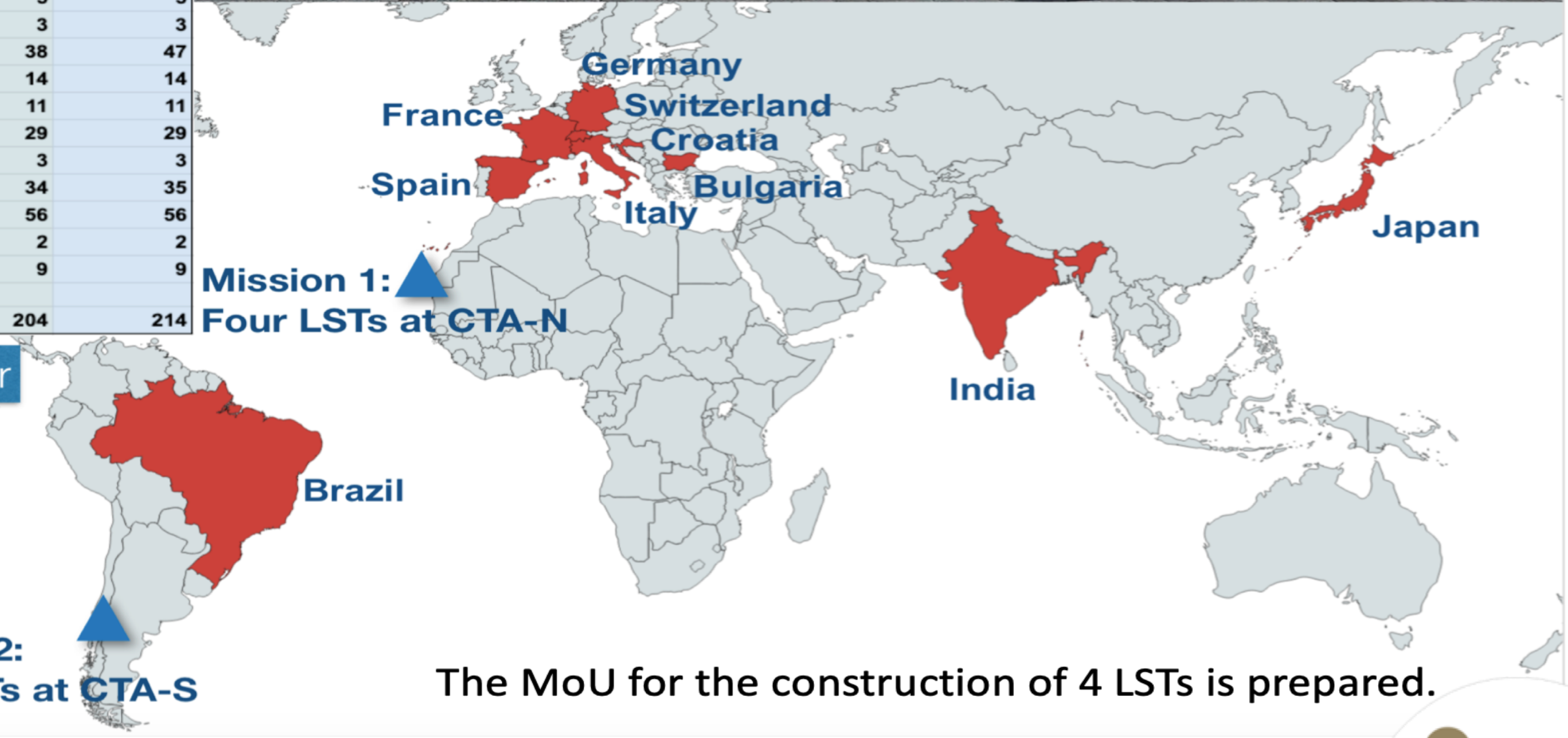
4 LSTs, 25 MSTs, 70 SSTs

LST sub-consortium

LST statistics

	Members	Scientist + Students	Authors
Bulgaria	5	5	5
Brazil	4	3	3
Spain	75	38	47
France	34	14	14
Croatia	11	11	11
Germany	36	29	29
India	3	3	3
Italy	39	34	35
Japan	60	56	56
Poland	2	2	2
Switzerland	9	9	9
Total	278	204	214

~80 FTE per year



Mission 1:
Four LSTs at CTA-N

Mission 2:
Four LSTs at CTA-S

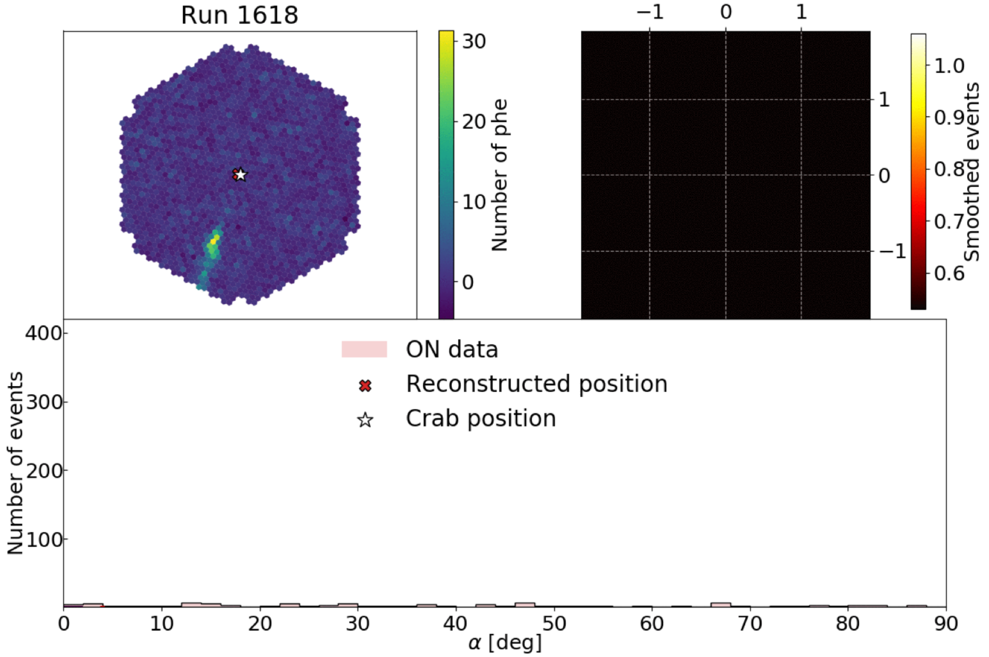
The MoU for the construction of 4 LSTs is prepared.

LST1 – The first Large Size Telescope

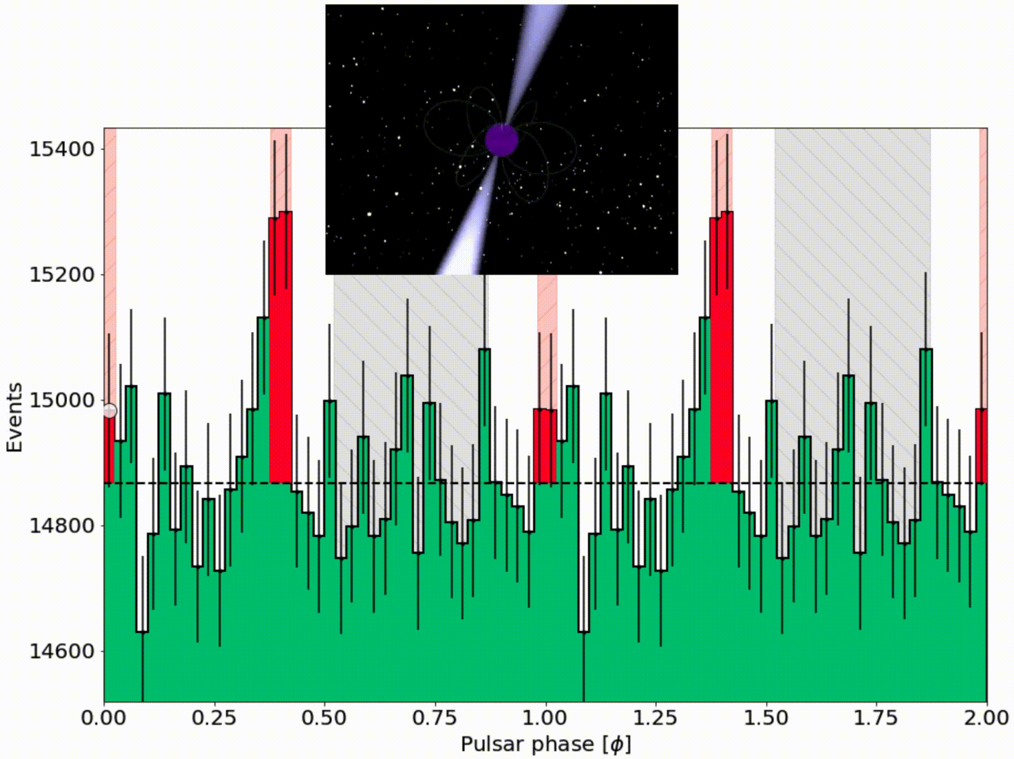
LST1 – The first Large Size Telescope

Inauguration 10th October 2018

2019 /2020
Critical Design Review
Deployment and commissioning
Crab Campaigns



Crab Nebula
R. López-Coto 2019



Crab Pulsar
R. López-Coto 2020

Provenance requirements

CTA General Requirement

A-USER-0110 The CTA Observatory must ensure that data processing is traceable and reproducible

Specific Requirements

Observatory

- Keep the traceability of the data products
- Data quality and reliability checks
- Reproduce or reprocess the data
- Debug a pipeline

Astronomer

- Provide more information to the final user
- Refine final results based on analysis of the provenance info

CTA software candidates

work-in-progress software

Python **tool** prototype for the Cherenkov Telescope Array Science Tools. Software for end-users to analyse, model and fit **science-ready data**.
https://gammapy.org

LSTOSA

Python **pipeline** for the **On-site Analysis of low-level** data observations from the LST1 curated and developed by GAE-UCM
https://contrera.gitlab.io/lstosa

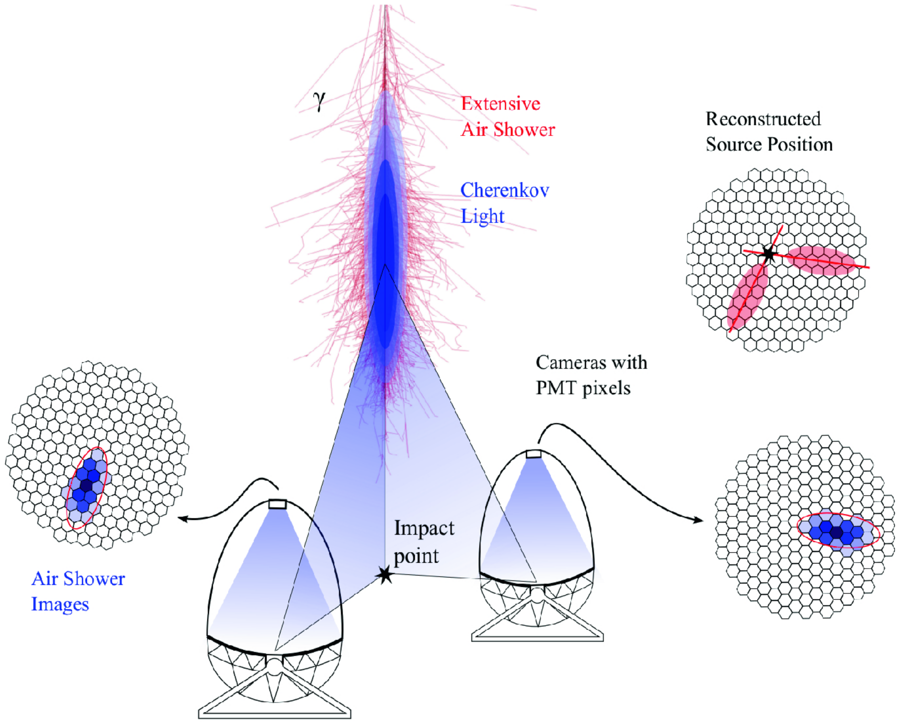
lstchain

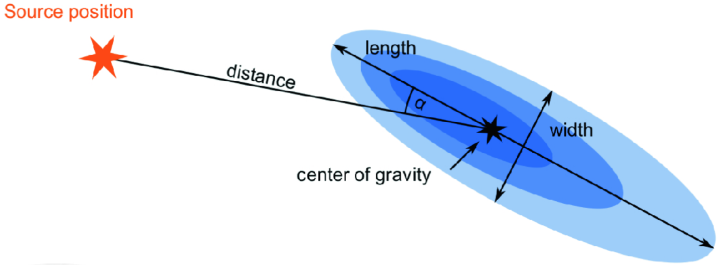
Python **library** for the **processing of low-level** data observations from the LST1 curated and developed by the LST Collaboration
https://github.com/cta-observatory/cta-lstchain

ctapipe

Python **framework** for prototyping the **low-level data processing** algorithms for the Cherenkov Telescope Array.
https://github.com/cta-observatory/ctapipe

Data Products in Cherenkov Imaging and Analysis





DL0	RAW	Digital signal from acquisition hardware	~ 10 GB
DL1	CALIBRATED	Real photons and times measured in each **telescope**	~ 10 GB
DL2	RECONSTRUCTED	Inferred direction, energy, gammaness for each **event**	~ GB
DL3	REDUCED	Selected events list and instrumental response	~ 10 MB
DL4	SCIENCE	Spectra, sky-maps, light-curves	~ 10^2 KB
DL5	ARCHIVE	Legacy observatory data (catalogs, surveys,..)	

rough file size numbers per observation in LST I
multiply by 10^2 observations to have an estimated total volume per per year

- *DL0 is actually decomposed in two levels R0 and R1*
  - *pulse charges integration in time windows and waveform corrections*
  - *no official DL0 data format yet*

protozfits/hdf5
json/text

- *DL0 -> DL1 is renamed to R0 -> DL1*
  - *photon count and timing calibration with cleaning levels*
  - *geometrical parametrization of events*
  - *muon analysis and data quality checks*

20 min

hdf5
json/text

- *DL1 -> DL2*
  - *low count gamma and very high background (gammaness cut)*
  - *direction/energy reconstruction via ML Random Forest algorithms with Montecarlo simulations*

60 min

hdf5
json/text and sav

- *DL2 -> DL3 process work-in-progress*

fits

EXCELENCIA
SEVERO
OCHOA

LST On-site Analysis pipeline

A collection of daily scheduled **scripts** that are **run in parallel in a grid environment**

Provenance capture

How?

- Using standard Python logging mechanism and a **provenance model defined in a YAML file**
- Non-intrusive implementation with **function/class decoration** in existing code
- Python **logging configuration** is set in an independent configuration file

Which info?

- Used and generated datasets, as well as input params and variables in decorated functions are well known and can be mapped and described in a **provenance model file** following W3C/IVOA Prov syntax

What do we get?

- **Post-processed** text log files as merged/filtered logs of W3C/IVOA Prov syntax info
- W3C provenance **JSON** files and **PDF** graphs as final provenance products

LST On-site Analysis pipeline

A collection of daily scheduled **scripts** that are **run in parallel in a grid environment**

Detailed considerations

- Provenance capture code is in an **independent package**/folder

provlog package could be used instead

- Execution **environment** is captured and stored as a session provenance entity
- Post-processing/merging of provenance logs may produce **different levels of granularity**

  A run is composed of a list of sub-runs

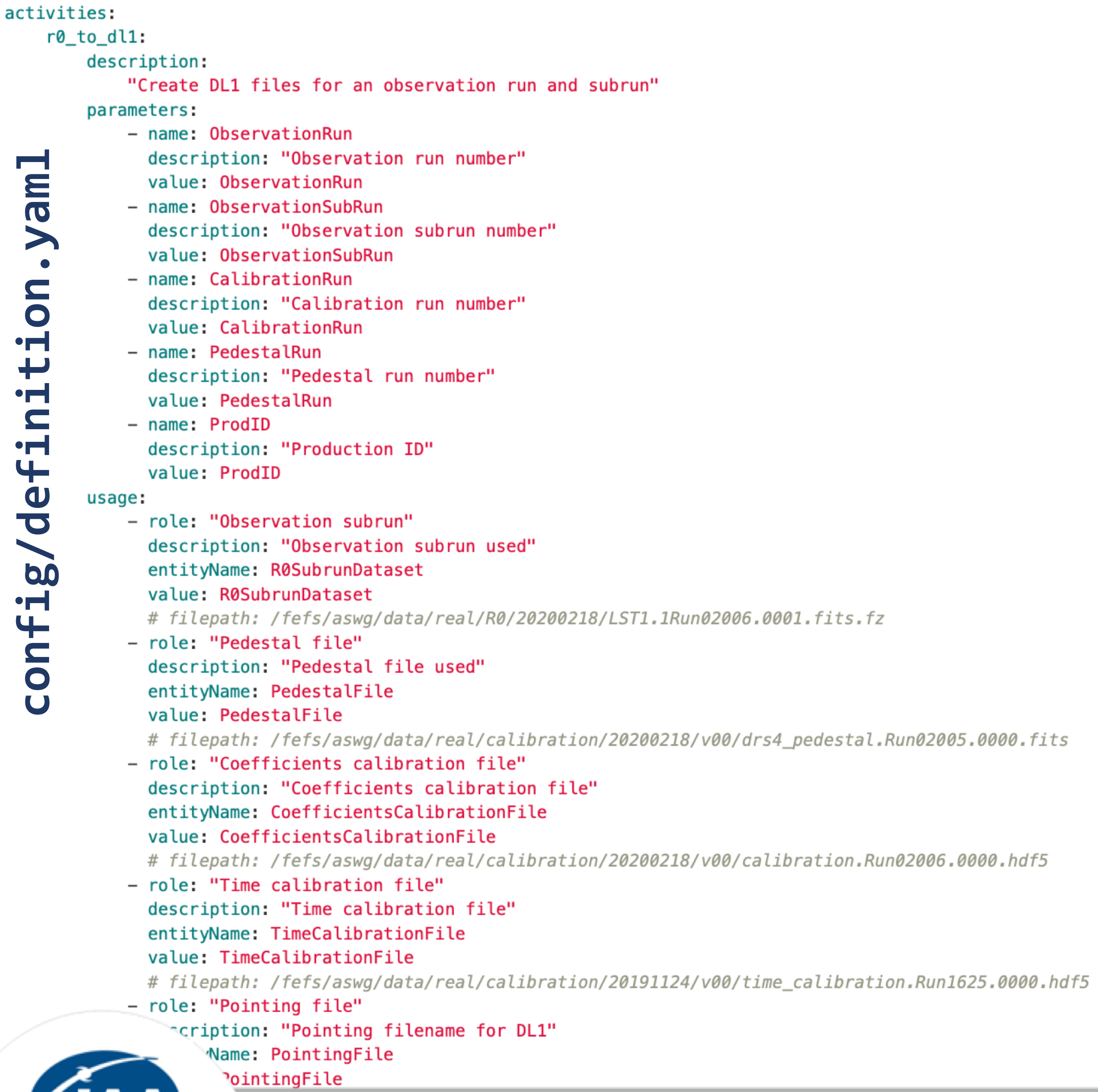
  An observation is composed of a list of runs

  A processed dataset is at a processing level

sub-run wise at different processing levels

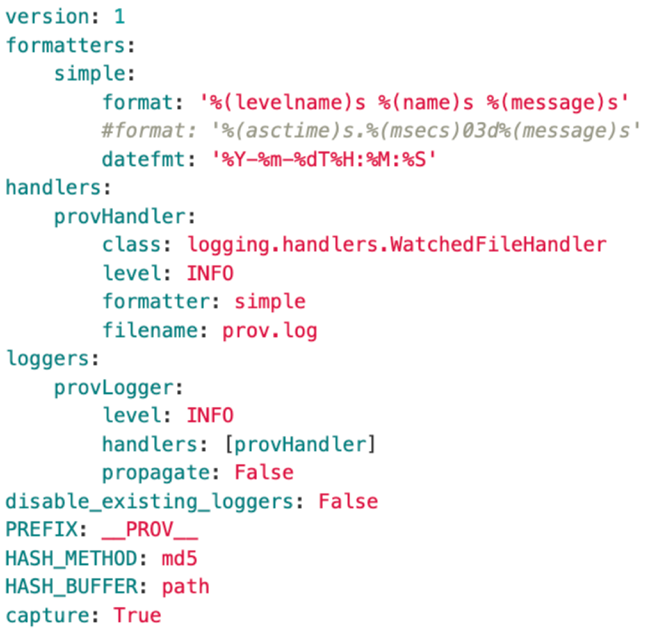
- Most of the info is *hidden* in **small configuration files** that are compared with hash-content algorithm and **copied** for reproducibility purposes
- Montecarlo simulated training datasets are not copied but **referenced**
- **Dry execution** mechanism allows provenance capture and merging avoiding data processing

config/definition.yaml



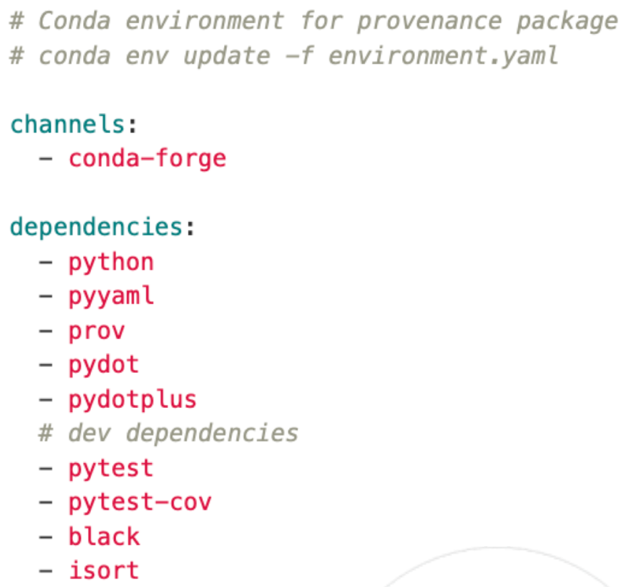
```
activities:
    r0_to_dl1:
        description:
            "Create DL1 files for an observation run and subrun"
        parameters:
            - name: ObservationRun
              description: "Observation run number"
              value: ObservationRun
            - name: ObservationSubRun
              description: "Observation subrun number"
              value: ObservationSubRun
            - name: CalibrationRun
              description: "Calibration run number"
              value: CalibrationRun
            - name: PedestalRun
              description: "Pedestal run number"
              value: PedestalRun
            - name: ProdID
              description: "Production ID"
              value: ProdID
        usage:
            - role: "Observation subrun"
              description: "Observation subrun used"
              entityName: R0SubrunDataset
              value: R0SubrunDataset
              # filepath: /fefs/aswg/data/real/R0/20200218/LST1.1Run02006.0001.fits.fz
            - role: "Pedestal file"
              description: "Pedestal file used"
              entityName: PedestalFile
              value: PedestalFile
              # filepath: /fefs/aswg/data/real/calibration/20200218/v00/drs4_pedestal.Run02005.0000.fits
            - role: "Coefficients calibration file"
              description: "Coefficients calibration file"
              entityName: CoefficientsCalibrationFile
              value: CoefficientsCalibrationFile
              # filepath: /fefs/aswg/data/real/calibration/20200218/v00/calibration.Run02006.0000.hdf5
            - role: "Time calibration file"
              description: "Time calibration file"
              entityName: TimeCalibrationFile
              value: TimeCalibrationFile
              # filepath: /fefs/aswg/data/real/calibration/20191124/v00/time_calibration.Run1625.0000.hdf5
            - role: "Pointing file"
               cription: "Pointing filename for DL1"
                  Name: PointingFile
                   ointingFile
```

config/logger.yaml



```
version: 1
formatters:
    simple:
        format: '%(levelname)s %(name)s %(message)s'
        #format: '%(asctime)s.%(msecs)03d%(message)s'
        datefmt: '%Y-%m-%dT%H:%M:%S'
handlers:
    provHandler:
        class: logging.handlers.WatchedFileHandler
        level: INFO
        formatter: simple
        filename: prov.log
loggers:
    provLogger:
        level: INFO
        handlers: [provHandler]
        propagate: False
disable_existing_loggers: False
PREFIX: __PROV__
HASH_METHOD: md5
HASH_BUFFER: path
capture: True
```

config/environment.yaml

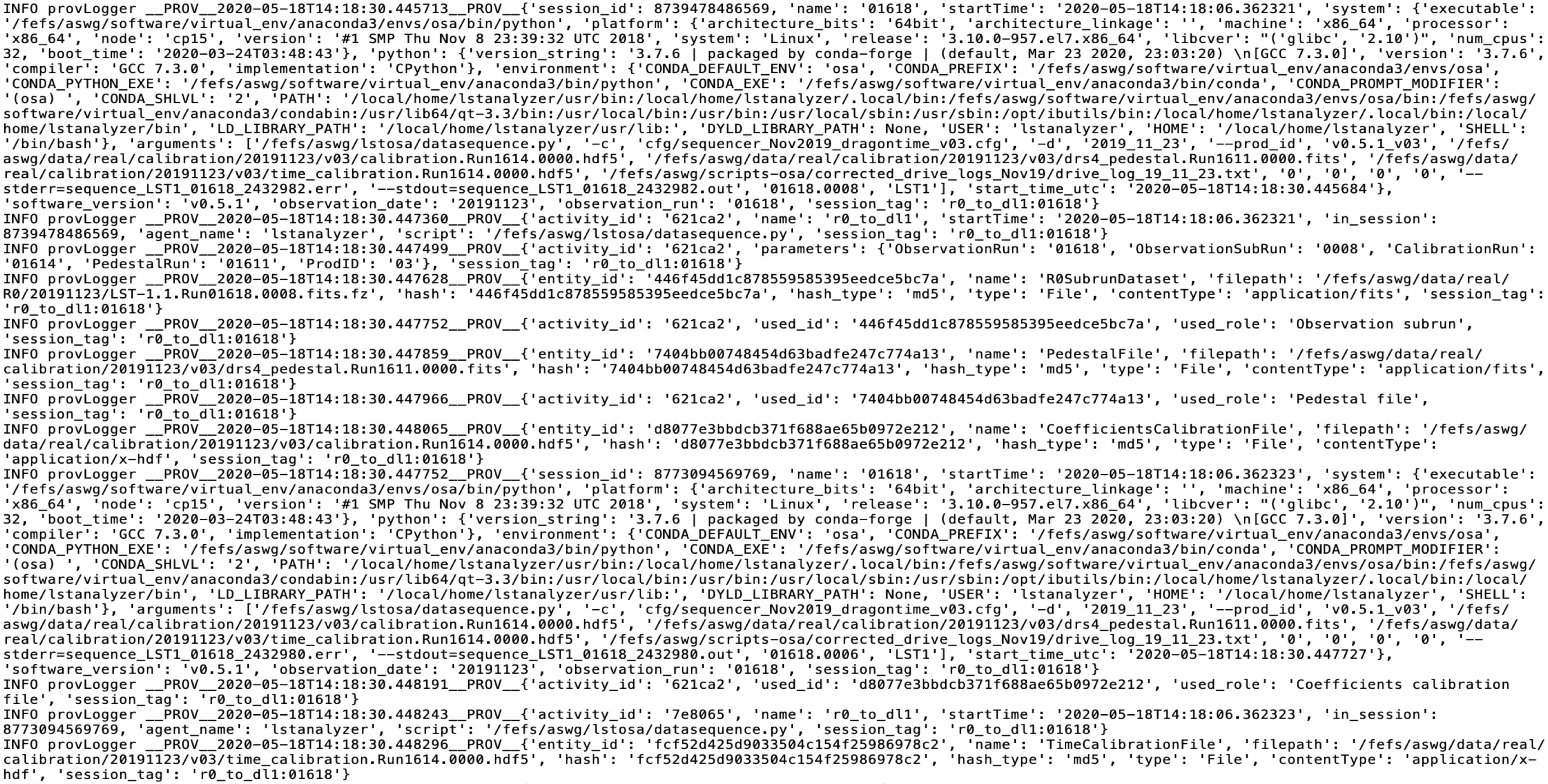


```
# Conda environment for provenance package
# conda env update -f environment.yaml

channels:
  - conda-forge

dependencies:
  - python
  - pyyaml
  - prov
  - pydot
  - pydotplus
  # dev dependencies
  - pytest
  - pytest-cov
  - black
  - isort
```

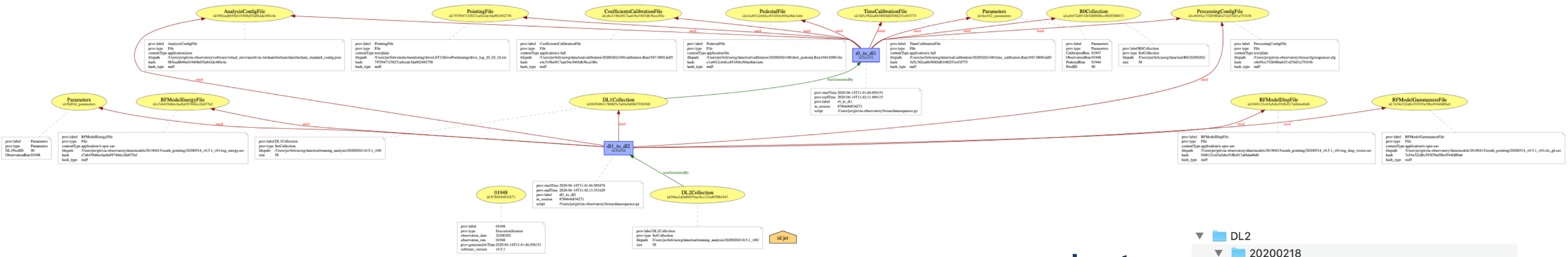
prov.log



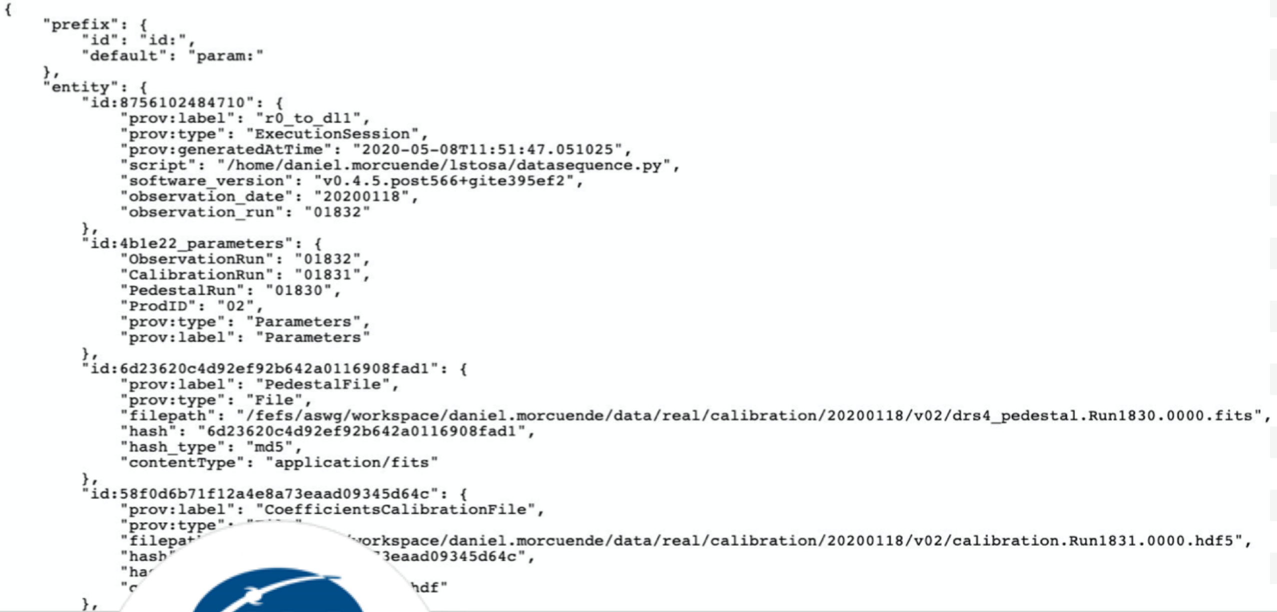
```
INFO provLogger __PROV__2020-05-18T14:18:30.445713__PROV__{'session_id': 8739478486569, 'name': '01618', 'startTime': '2020-05-18T14:18:06.362321', 'system': {'executable':
'/fefs/aswg/software/virtual_env/anaconda3/envs/osa/bin/python', 'platform': {'architecture_bits': '64bit', 'architecture_linkage': '', 'machine': 'x86_64', 'processor':
'x86_64', 'node': 'cp15', 'version': '#1 SMP Thu Nov 8 23:39:32 UTC 2018', 'system': 'Linux', 'release': '3.10.0-957.el7.x86_64', 'libcver': "('glibc', '2.10')", 'num_cpus':
32, 'boot_time': '2020-03-24T03:48:43'}, 'python': {'version_string': '3.7.6 | packaged by conda-forge | (default, Mar 23 2020, 23:03:20) \n[GCC 7.3.0]', 'version': '3.7.6',
'compiler': 'GCC 7.3.0', 'implementation': 'CPython'}, 'environment': {'CONDA_DEFAULT_ENV': 'osa', 'CONDA_PREFIX': '/fefs/aswg/software/virtual_env/anaconda3/envs/osa',
'CONDA_PYTHON_EXE': '/fefs/aswg/software/virtual_env/anaconda3/bin/python', 'CONDA_EXE': '/fefs/aswg/software/virtual_env/anaconda3/bin/conda', 'CONDA_PROMPT_MODIFIER':
'(osa) ', 'CONDA_SHLVL': '2', 'PATH': '/local/home/lstanalyzer/usr/bin:/local/home/lstanalyzer/.local/bin:/fefs/aswg/software/virtual_env/anaconda3/envs/osa/bin:/fefs/aswg/
software/virtual_env/anaconda3/condabin:/usr/lib64/qt-3.3/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/opt/ibutils/bin:/local/home/lstanalyzer/.local/bin:/local/
home/lstanalyzer/bin', 'LD_LIBRARY_PATH': '/local/home/lstanalyzer/usr/lib:', 'DYLD_LIBRARY_PATH': None, 'USER': 'lstanalyzer', 'HOME': '/local/home/lstanalyzer', 'SHELL':
'/bin/bash'}, 'arguments': ['/fefs/aswg/lstosa/datasequence.py', '-c', 'cfg/sequencer_Nov2019_dragontime_v03.cfg', '-d', '2019_11_23', '--prod_id', 'v0.5.1_v03', '/fefs/
aswg/data/real/calibration/20191123/v03/calibration.Run1614.0000.hdf5', '/fefs/aswg/data/real/calibration/20191123/v03/drs4_pedestal.Run1611.0000.fits', '/fefs/aswg/data/
real/calibration/20191123/v03/time_calibration.Run1614.0000.hdf5', '/fefs/aswg/scripts-osa/corrected_drive_logs_Nov19/drive_log_19_11_23.txt', '0', '0', '0', '0', '--
stderr=sequence_LST1_01618_2432982.err', '--stdout=sequence_LST1_01618_2432982.out', '01618.0008', 'LST1'], 'start_time_utc': '2020-05-18T14:18:30.445684'},
'software_version': 'v0.5.1', 'observation_date': '20191123', 'observation_run': '01618', 'session_tag': 'r0_to_dl1:01618'}
INFO provLogger __PROV__2020-05-18T14:18:30.447360__PROV__{'activity_id': '621ca2', 'name': 'r0_to_dl1', 'startTime': '2020-05-18T14:18:06.362321', 'in_session':
8739478486569, 'agent_name': 'lstanalyzer', 'script': '/fefs/aswg/lstosa/datasequence.py', 'session_tag': 'r0_to_dl1:01618'}
INFO provLogger __PROV__2020-05-18T14:18:30.447499__PROV__{'activity_id': '621ca2', 'parameters': {'ObservationRun': '01618', 'ObservationSubRun': '0008', 'CalibrationRun':
'01614', 'PedestalRun': '01611', 'ProdID': '03'}, 'session_tag': 'r0_to_dl1:01618'}
INFO provLogger __PROV__2020-05-18T14:18:30.447628__PROV__{'entity_id': '446f45dd1c878559585395eedce5bc7a', 'name': 'R0SubrunDataset', 'filepath': '/fefs/aswg/data/real/
R0/20191123/LST-1.1.Run01618.0008.fits.fz', 'hash': '446f45dd1c878559585395eedce5bc7a', 'hash_type': 'md5', 'type': 'File', 'contentType': 'application/fits', 'session_tag':
'r0_to_dl1:01618'}
INFO provLogger __PROV__2020-05-18T14:18:30.447752__PROV__{'activity_id': '621ca2', 'used_id': '446f45dd1c878559585395eedce5bc7a', 'used_role': 'Observation subrun',
'session_tag': 'r0_to_dl1:01618'}
INFO provLogger __PROV__2020-05-18T14:18:30.447859__PROV__{'entity_id': '7404bb00748454d63badfe247c774a13', 'name': 'PedestalFile', 'filepath': '/fefs/aswg/data/real/
calibration/20191123/v03/drs4_pedestal.Run1611.0000.fits', 'hash': '7404bb00748454d63badfe247c774a13', 'hash_type': 'md5', 'type': 'File', 'contentType': 'application/fits',
'session_tag': 'r0_to_dl1:01618'}
INFO provLogger __PROV__2020-05-18T14:18:30.447966__PROV__{'activity_id': '621ca2', 'used_id': '7404bb00748454d63badfe247c774a13', 'used_role': 'Pedestal file',
'session_tag': 'r0_to_dl1:01618'}
INFO provLogger __PROV__2020-05-18T14:18:30.448065__PROV__{'entity_id': 'd8077e3bbdcb371f688ae65b0972e212', 'name': 'CoefficientsCalibrationFile', 'filepath': '/fefs/aswg/
data/real/calibration/20191123/v03/calibration.Run1614.0000.hdf5', 'hash': 'd8077e3bbdcb371f688ae65b0972e212', 'hash_type': 'md5', 'type': 'File', 'contentType':
'application/x-hdf', 'session_tag': 'r0_to_dl1:01618'}
INFO provLogger __PROV__2020-05-18T14:18:30.447752__PROV__{'session_id': 8773094569769, 'name': '01618', 'startTime': '2020-05-18T14:18:06.362323', 'system': {'executable':
'/fefs/aswg/software/virtual_env/anaconda3/envs/osa/bin/python', 'platform': {'architecture_bits': '64bit', 'architecture_linkage': '', 'machine': 'x86_64', 'processor':
'x86_64', 'node': 'cp15', 'version': '#1 SMP Thu Nov 8 23:39:32 UTC 2018', 'system': 'Linux', 'release': '3.10.0-957.el7.x86_64', 'libcver': "('glibc', '2.10')", 'num_cpus':
32, 'boot_time': '2020-03-24T03:48:43'}, 'python': {'version_string': '3.7.6 | packaged by conda-forge | (default, Mar 23 2020, 23:03:20) \n[GCC 7.3.0]', 'version': '3.7.6',
'compiler': 'GCC 7.3.0', 'implementation': 'CPython'}, 'environment': {'CONDA_DEFAULT_ENV': 'osa', 'CONDA_PREFIX': '/fefs/aswg/software/virtual_env/anaconda3/envs/osa',
'CONDA_PYTHON_EXE': '/fefs/aswg/software/virtual_env/anaconda3/bin/python', 'CONDA_EXE': '/fefs/aswg/software/virtual_env/anaconda3/bin/conda', 'CONDA_PROMPT_MODIFIER':
'(osa) ', 'CONDA_SHLVL': '2', 'PATH': '/local/home/lstanalyzer/usr/bin:/local/home/lstanalyzer/.local/bin:/fefs/aswg/software/virtual_env/anaconda3/envs/osa/bin:/fefs/aswg/
software/virtual_env/anaconda3/condabin:/usr/lib64/qt-3.3/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/opt/ibutils/bin:/local/home/lstanalyzer/.local/bin:/local/
home/lstanalyzer/bin', 'LD_LIBRARY_PATH': '/local/home/lstanalyzer/usr/lib:', 'DYLD_LIBRARY_PATH': None, 'USER': 'lstanalyzer', 'HOME': '/local/home/lstanalyzer', 'SHELL':
'/bin/bash'}, 'arguments': ['/fefs/aswg/lstosa/datasequence.py', '-c', 'cfg/sequencer_Nov2019_dragontime_v03.cfg', '-d', '2019_11_23', '--prod_id', 'v0.5.1_v03', '/fefs/
aswg/data/real/calibration/20191123/v03/calibration.Run1614.0000.hdf5', '/fefs/aswg/data/real/calibration/20191123/v03/drs4_pedestal.Run1611.0000.fits', '/fefs/aswg/data/
real/calibration/20191123/v03/time_calibration.Run1614.0000.hdf5', '/fefs/aswg/scripts-osa/corrected_drive_logs_Nov19/drive_log_19_11_23.txt', '0', '0', '0', '0', '--
stderr=sequence_LST1_01618_2432980.err', '--stdout=sequence_LST1_01618_2432980.out', '01618.0006', 'LST1'], 'start_time_utc': '2020-05-18T14:18:30.447727'},
'software_version': 'v0.5.1', 'observation_date': '20191123', 'observation_run': '01618', 'session_tag': 'r0_to_dl1:01618'}
INFO provLogger __PROV__2020-05-18T14:18:30.448191__PROV__{'activity_id': '621ca2', 'used_id': 'd8077e3bbdcb371f688ae65b0972e212', 'used_role': 'Coefficients calibration
file', 'session_tag': 'r0_to_dl1:01618'}
INFO provLogger __PROV__2020-05-18T14:18:30.448243__PROV__{'activity_id': '7e8065', 'name': 'r0_to_dl1', 'startTime': '2020-05-18T14:18:06.362323', 'in_session':
8773094569769, 'agent_name': 'lstanalyzer', 'script': '/fefs/aswg/lstosa/datasequence.py', 'session_tag': 'r0_to_dl1:01618'}
INFO provLogger __PROV__2020-05-18T14:18:30.448296__PROV__{'entity_id': 'fcf52d425d9033504c154f25986978c2', 'name': 'TimeCalibrationFile', 'filepath': '/fefs/aswg/data/real/
calibration/20191123/v03/time_calibration.Run1614.0000.hdf5', 'hash': 'fcf52d425d9033504c154f25986978c2', 'hash_type': 'md5', 'type': 'File', 'contentType': 'application/x-
hdf', 'session_tag': 'r0_to_dl1:01618'}
```

data processing provenance graph

https://openprovenance.org/store/documents/3198

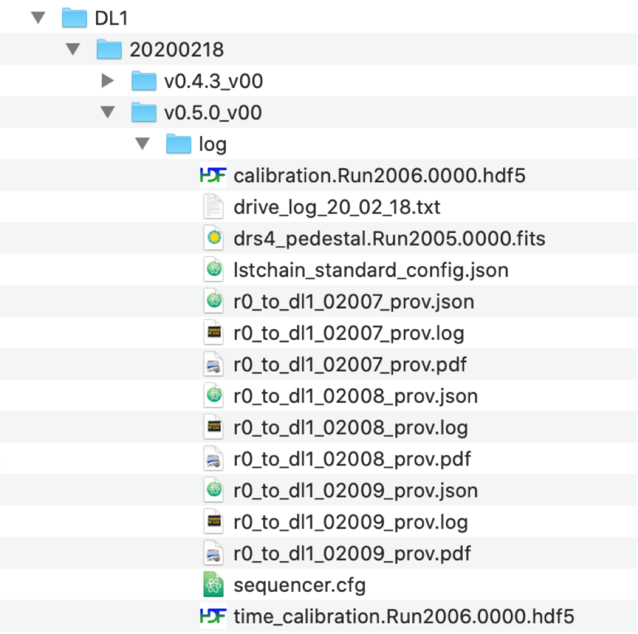

AnalysisConfigFile
id:985eadfb94e019408d55a842dc480c4e
PointingFile
id:797f9471f3827ca9cede3da902492758
CoefficientsCalibrationFile
id:e6c319be9f17aa636e3403db3bca180c
PedestalFile
id:e1a4012cb6fcc851f04cf69ae8de1a6e
TimeCalibrationFile
id:3d5c382ea8b5800fd0108f251e43f775
Parameters
id:0ec632_parameters
R0Collection
id:af6472d931fb5d0868bcc88f85ff6071
ProcessingConfigFile
id:c6b95ce752b98bab251d35df1a751b3b
r0_to_dl1
id:0ec632
used
wasGeneratedBy
prov:startTime 2020-06-14T11:41:46.056151
prov:endTime 2020-06-14T11:42:11.496115
prov:label r0_to_dl1
in_session 8786646834271
script /Users/jer/git/cta-observatory/lstosa/datasequence.py
Parameters
id:f0d3fd_parameters
RFModelEnergyFile
id:e7eb439d6ecfae8a99746fec2fa975ef
DL1Collection
id:8650404178888f5c3a04e0d0b07936560
RFModelDispFile
id:04f6121c63a5ebc91f8c017a6bda46d6
RFModelGammanessFile
id:7e54e322d8c593f39a5f8e4544fd80a6
dl1_to_dl2
id:f0d3fd
prov:startTime 2020-06-14T11:41:46.085476
prov:endTime 2020-06-14T11:42:13.351629
prov:label dl1_to_dl2
in_session 8786646834271
script /Users/jer/git/cta-observatory/lstosa/datasequence.py
01948
id:8786646834271
DL2Collection
id:04aa1a0afb0f3bac9cc123a4058bf443
prov:label 01948
prov:type ExecutionSession
observation_date 20200202
observation_run 01948
prov:generatedAtTime 2020-06-14T11:41:46.056151
software_version v0.5.1
prov:label DL2Collection
prov:type SetCollection
filepath /Users/jer/fefs/aswg/data/real/running_analysis/20200202/v0.5.1_v00/
size 58
id:jer

json serialization

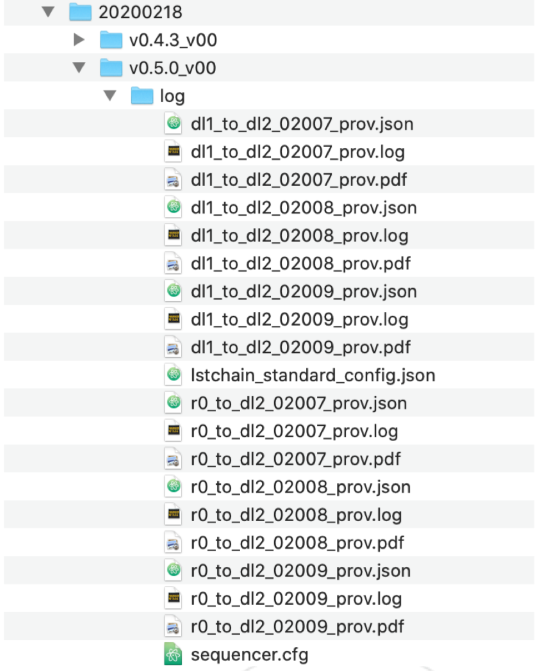


```
{
    "prefix": {
        "id": "id:",
        "default": "param:"
    },
    "entity": {
        "id:8756102484710": {
            "prov:label": "r0_to_dl1",
            "prov:type": "ExecutionSession",
            "prov:generatedAtTime": "2020-05-08T11:51:47.051025",
            "script": "/home/daniel.morcuende/lstosa/datasequence.py",
            "software_version": "v0.4.5.post566+gite395ef2",
            "observation_date": "20200118",
            "observation_run": "01832"
        },
        "id:4b1e22_parameters": {
            "ObservationRun": "01832",
            "CalibrationRun": "01831",
            "PedestalRun": "01830",
            "ProdID": "02",
            "prov:type": "Parameters",
            "prov:label": "Parameters"
        },
        "id:6d23620c4d92ef92b642a0116908fad1": {
            "prov:label": "PedestalFile",
            "prov:type": "File",
            "filepath": "/fefs/aswg/workspace/daniel.morcuende/data/real/calibration/20200118/v02/drs4_pedestal.Run1830.0000.fits",
            "hash": "6d23620c4d92ef92b642a0116908fad1",
            "hash_type": "md5",
            "contentType": "application/fits"
        },
        "id:58f0d6b71f12a4e8a73eaad09345d64c": {
            "prov:label": "CoefficientsCalibrationFile",
            "prov:type"
            "filepat                  workspace/daniel.morcuende/data/real/calibration/20200118/v02/calibration.Run1831.0000.hdf5",
            "hash                     3eaad09345d64c",
            "ha
            "c                        hdf"
        },
```

provenance products



```
▼ DL1
  ▼ 20200218
    ▶ v0.4.3_v00
    ▼ v0.5.0_v00
      ▼ log
          calibration.Run2006.0000.hdf5
          drive_log_20_02_18.txt
          drs4_pedestal.Run2005.0000.fits
          lstchain_standard_config.json
          r0_to_dl1_02007_prov.json
          r0_to_dl1_02007_prov.log
          r0_to_dl1_02007_prov.pdf
          r0_to_dl1_02008_prov.json
          r0_to_dl1_02008_prov.log
          r0_to_dl1_02008_prov.pdf
          r0_to_dl1_02009_prov.json
          r0_to_dl1_02009_prov.log
          r0_to_dl1_02009_prov.pdf
          sequencer.cfg
          time_calibration.Run2006.0000.hdf5
```



```
▼ DL2
  ▼ 20200218
    ▶ v0.4.3_v00
    ▼ v0.5.0_v00
      ▼ log
          dl1_to_dl2_02007_prov.json
          dl1_to_dl2_02007_prov.log
          dl1_to_dl2_02007_prov.pdf
          dl1_to_dl2_02008_prov.json
          dl1_to_dl2_02008_prov.log
          dl1_to_dl2_02008_prov.pdf
          dl1_to_dl2_02009_prov.json
          dl1_to_dl2_02009_prov.log
          dl1_to_dl2_02009_prov.pdf
          lstchain_standard_config.json
          r0_to_dl2_02007_prov.json
          r0_to_dl2_02007_prov.log
          r0_to_dl2_02007_prov.pdf
          r0_to_dl2_02008_prov.json
          r0_to_dl2_02008_prov.log
          r0_to_dl2_02008_prov.pdf
          r0_to_dl2_02009_prov.json
          r0_to_dl2_02009_prov.log
          r0_to_dl2_02009_prov.pdf
          sequencer.cfg
```

Lessons learnt

- Continuous update and addition of captured info – **flexible model and implementation**
- **Configuration values** may be stored as metadata attached to datasets (*fits headers/hdf5 attributes*)
- **Capturing relationships** among activities and entities (*datasets*) needs a provenance model
- Improvement of capture mechanism for **execution environment**/grid nodes configuration is needed
- **Post-processing of captured provenance** info may be needed to filter raw provenance according to specific needs and/or to artificially produce different **levels of granularity**
- **Independent capture from different dependent software** packages is possible/desirable

  LSTOSA requires lstchain

  lstchain requires ctapipe

  gammapy used independently for analysis

- **Structured logging** in text files may be a solution for small session provenance storage (*gammapy*)
- Considering storing provenance in a RDBMS or *better* in a noSQL **database** (i.e. *mongo + json*)
- Development of a **provenance query mechanism** for detailed analysis and inspection is needed

  …considering using url query params to produce on-the-fly SVG graphs with access-links in a browser