

Provenance **Capture** Python Implementation

J. Enrique Ruiz (IAA – CSIC), Mathieu Servillat (Obs. Paris-Meudon)

French Weekly Provenance Meeting
03/06/2020



Software candidates



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>



Python **pipeline** for the **On-site Analysis of low-level data** captured by the Large Size Telescope (Cherenkov Telescope Array precursor in La Palma)

<https://contrera.gitlab.io/lstosa>

Istchain

Python **library** for the **processing of low-level data** captured by the Large Size Telescope (Cherenkov Telescope Array precursor in La Palma)

<https://github.com/cta-observatory/cta-istchain>



Python **framework** for prototyping the **low-level data processing** algorithms for the Cherenkov Telescope Array.

<https://github.com/cta-observatory/ctapipe>



Requirements

How?

Capture provenance info in text files using standard automatic logging mechanism.

Implementation non-intrusive in already existing code with function/class decoration.

Configuration set in independent config files.

Which info?

Defined in a file model following IVOA Prov recommendation.

What do we get?

Processing of log files to produce filtered provenance products using Prov syntax

JSON files and PDF graphs





Gammapy High-level Interface API

A collection of **command-line tools** that may be used within Python scripts or in IPython sessions/notebooks.

Used and generated entities, as well as input parameters, for each tool are well known and can be described in the **provenance model file**.

Only provenance info defined in the model file is **transparently and automatically logged** in a text file during the analysis session.

After session ends the log file can be post-processed to produce filtered (i.e. time-range or agent) **provenance products like graphs** for a basic inspection and analysis.





Gammapy High-level Interface API

All code is in an independent package/folder `gammapy.utis.provenance`

Responsible class providing High-Level Tools is decorated with `@provenance`

Execution environment is captured and stored in a `session provenance entity`

Code initially forked from `ctapipe.core.Provenance`

Branch: prov `gammapy / gammapy / utis / provenance /` Create new file Upload files Find file History

This branch is 111 commits ahead, 169 commits behind gammapy:master. Pull request Compare

Bultako remove unused export log_file_generation function Latest commit bd28787 on 21 Apr

..

config

<code>__init__.py</code>	refactor functions in new scripts	last month
<code>capture.py</code>	remove unused export log_file_generation function	last month
<code>io.py</code>	clean up and make graphs in png format	last month

- `definition.yaml`
- `environment.yaml`
- `logger.yaml`

config/definition.yaml

```
activities:

get_observations:
  description:
    "Fetch observations from datastore and list of identifiers filtering by optional spatial/temporal criteria."
  parameters:
    - name: datastore_path
      description: "DataStore file path as string"
      value: config.observations.datastore.__str__()
    - name: obslist_path
      description: "Observations list file path as string"
      value: config.observations.obs_file
    - name: obs_ids
      description: "Observations list of identifiers"
      value: config.observations.obs_ids
    - name: obs_cone
      description: "Conesearch criteria to select observations"
      value: config.observations.obs_cone.json()
    - name: obs_time
      description: "Time criteria to select observations"
      value: config.observations.obs_time.json()
  usage:
    - role: datastore
      description: "DataStore object file"
      entityName: DataStore
      value: config.observations.datastore
    - role: obslist_file
      description: "Observations list file"
      entityName: DataStore
      value: config.observations.obs_file
  generation:
    - role: observations_selected
      description: "Observations selected"
      entityName: Observations
      value: observations
      has_members:
        list: observations._observations
        entityName: Observation
        id: obs_id
        location: _events_hdu.base_dir
        namespace: gamma-events
```

config/environment.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

# 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
```

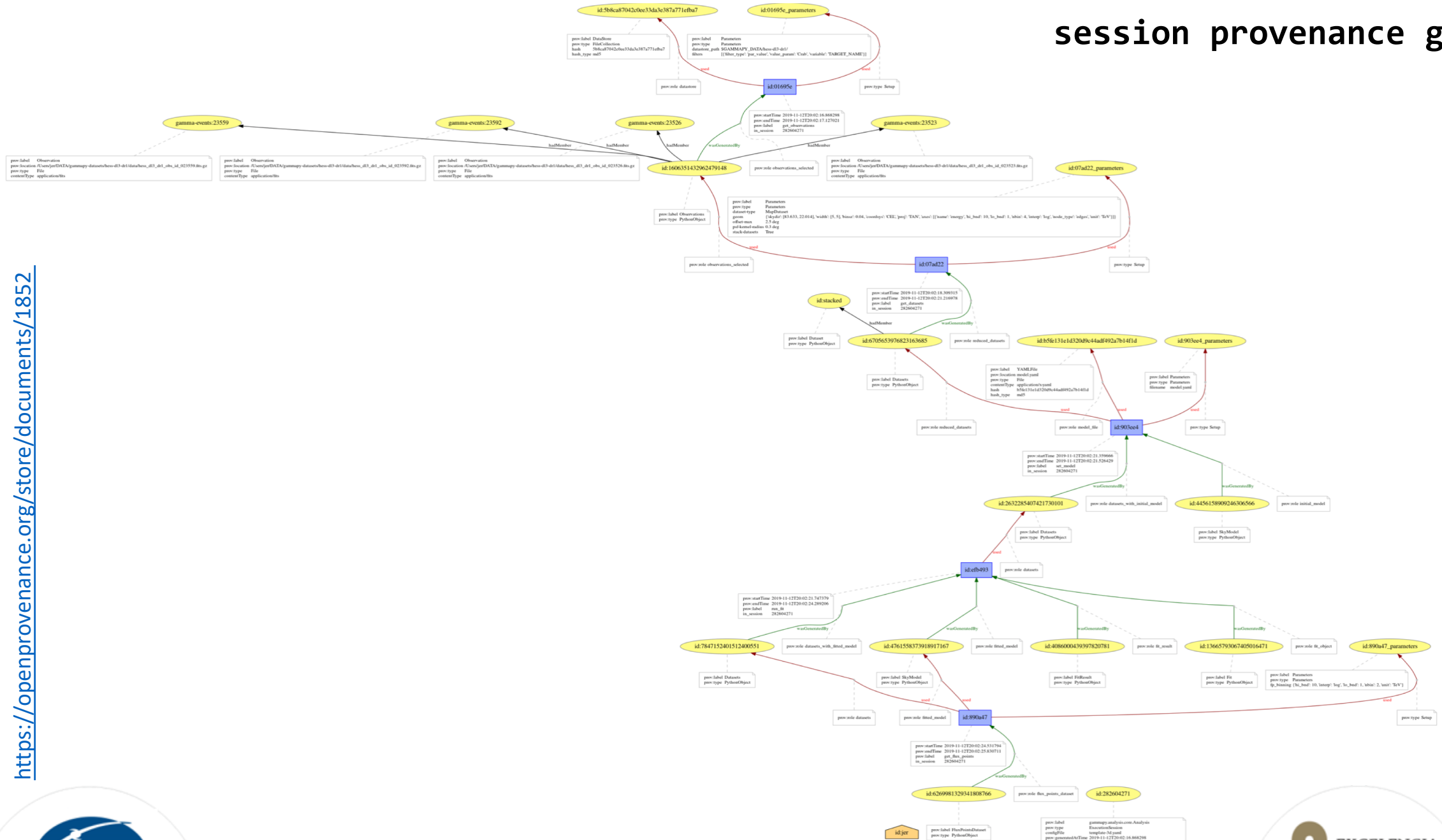


prov.log

```
INFO provLogger _PROV_2019-11-18T10:48:20.428982_PROV_{'entity_id': 519318473740404787, 'name': 'Observations', 'type': 'PythonObject'}
INFO provLogger _PROV_2019-11-18T10:48:20.429298_PROV_{'activity_id': '722a28', 'generated_id': 519318473740404787, 'generated_role':
'observations_selected'}
INFO provLogger _PROV_2019-11-18T10:48:20.430010_PROV_{'entity_id': 'gamma-events:23523', 'name': 'Observation', 'location': '/Users/test/DATA/
gammapy-datasets/hess-dl3-dr1/data/hess_dl3_dr1_obs_id_023523.fits.gz', 'type': 'File', 'contentType': 'application/fits'}
INFO provLogger _PROV_2019-11-18T10:48:20.430217_PROV_{'entity_id': 519318473740404787, 'member_id': 'gamma-events:23523'}
INFO provLogger _PROV_2019-11-18T10:48:20.431010_PROV_{'entity_id': 'gamma-events:23526', 'name': 'Observation', 'location': '/Users/test/DATA/
gammapy-datasets/hess-dl3-dr1/data/hess_dl3_dr1_obs_id_023526.fits.gz', 'type': 'File', 'contentType': 'application/fits'}
INFO provLogger _PROV_2019-11-18T10:48:20.431250_PROV_{'entity_id': 519318473740404787, 'member_id': 'gamma-events:23526'}
INFO provLogger _PROV_2019-11-18T10:48:20.432081_PROV_{'entity_id': 'gamma-events:23559', 'name': 'Observation', 'location': '/Users/test/DATA/
gammapy-datasets/hess-dl3-dr1/data/hess_dl3_dr1_obs_id_023559.fits.gz', 'type': 'File', 'contentType': 'application/fits'}
INFO provLogger _PROV_2019-11-18T10:48:20.432216_PROV_{'entity_id': 519318473740404787, 'member_id': 'gamma-events:23559'}
INFO provLogger _PROV_2019-11-18T10:48:20.433059_PROV_{'entity_id': 'gamma-events:23592', 'name': 'Observation', 'location': '/Users/test/DATA/
gammapy-datasets/hess-dl3-dr1/data/hess_dl3_dr1_obs_id_023592.fits.gz', 'type': 'File', 'contentType': 'application/fits'}
INFO provLogger _PROV_2019-11-18T10:48:20.433252_PROV_{'entity_id': 519318473740404787, 'member_id': 'gamma-events:23592'}
INFO provLogger _PROV_2019-11-18T10:48:20.433418_PROV_{'activity_id': '722a28', 'endTime': '2019-11-18T10:48:18.995966'}
INFO provLogger _PROV_2019-11-18T10:48:25.940142_PROV_{'activity_id': '665a0c', 'name': 'get_datasets', 'startTime':
'2019-11-18T10:48:21.012240', 'in_session': 9223372036581382375, 'agent_name': 'test'}
INFO provLogger _PROV_2019-11-18T10:48:25.940709_PROV_{'activity_id': '665a0c', 'parameters': {'stack-datasets': True, 'dataset-type':
'MapDataset', 'geom': {'skydir': [83.633, 22.014], 'width': [5, 5], 'binsz': 0.04, 'coordsys': 'CEL', 'proj': 'TAN', 'axes': [{'name': 'energy',
'hi_bnd': 10, 'lo_bnd': 1, 'nbin': 4, 'interp': 'log', 'node_type': 'edges', 'unit': 'TeV']}}, 'offset-max': '2.5 deg', 'psf-kernel-radius': '0.3
deg'}}
INFO provLogger _PROV_2019-11-18T10:48:25.941046_PROV_{'entity_id': 519318473740404787, 'name': 'Observations', 'type': 'PythonObject'}
INFO provLogger _PROV_2019-11-18T10:48:25.941174_PROV_{'activity_id': '665a0c', 'used_id': 519318473740404787, 'used_role':
'observations_selected'}
INFO provLogger _PROV_2019-11-18T10:48:25.941341_PROV_{'entity_id': 15337330674484208277, 'name': 'Datasets', 'type': 'PythonObject'}
INFO provLogger _PROV_2019-11-18T10:48:25.941449_PROV_{'activity_id': '665a0c', 'generated_id': 15337330674484208277, 'generated_role':
'reduced_datasets'}
INFO provLogger _PROV_2019-11-18T10:48:25.941574_PROV_{'entity_id': 'stacked', 'name': 'Dataset', 'type': 'PythonObject'}
INFO provLogger _PROV_2019-11-18T10:48:25.941668_PROV_{'entity_id': 15337330674484208277, 'member_id': 'stacked'}
INFO provLogger _PROV_2019-11-18T10:48:25.941835_PROV_{'activity_id': '665a0c', 'endTime': '2019-11-18T10:48:25.940069'}
INFO provLogger _PROV_2019-11-18T10:48:26.495235_PROV_{'activity_id': '1a1ace', 'name': 'set_model', 'startTime': '2019-11-18T10:48:26.245433',
'in_session': 9223372036581382375, 'agent_name': 'test'}
INFO provLogger _PROV_2019-11-18T10:48:26.495513_PROV_{'activity_id': '1a1ace', 'parameters': {'filename': 'model.yaml'}}
INFO provLogger _PROV_2019-11-18T10:48:26.495759_PROV_{'entity_id': 'b5fe131e1d320d9c44adf492a7b14f1d', 'name': 'YAMLFile', 'location':
'model.yaml', 'hash': 'b5fe131e1d320d9c44adf492a7b14f1d', 'hash_type': 'md5', 'type': 'File', 'contentType': 'application/x-yaml'}
INFO provLogger _PROV_2019-11-18T10:48:26.495967_PROV_{'activity_id': '1a1ace', 'used_id': 'b5fe131e1d320d9c44adf492a7b14f1d', 'used_role':
```



session provenance graph



<https://openprovenance.org/store/documents/1852.png>

<https://openprovenance.org/store/documents/1852>





LST On-site Analysis pipeline

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

Used and generated entities, as well as input parameters, for each **function** in a script are well known and can be described in the **provenance model file**.

Only provenance info defined in the model file is **transparently and automatically logged** in a text file during the pipeline execution.

After **data processing of a run** ends the log file can be post-processed to produce filtered (i.e. time range, agent, activity, etc..) **provenance products like graphs** for a basic inspection and analysis.





LST On-site Analysis pipeline

Code is in a **restricted access** Gitlab repository curated/developed by [GAE-UCM](#)

All code is in an independent package/folder `osa.provenance`

Responsible **functions** providing data processing are decorated with `@trace`

Execution environment is captured and stored in a `session provenance` entity

Post-processing of provenance logs may produce **different levels of granularity**

An observation may be a list of runs

A run is a list of subruns

Most of the info is *hidden* in **small configuration files** that are compared with hash-content algorithm and **copied for reproducibility purposes**



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"
        description: "Pointing filename for DL1"
        entityName: PointingFile
        value: PointingFile
```

config/environment.yaml 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

# 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
```

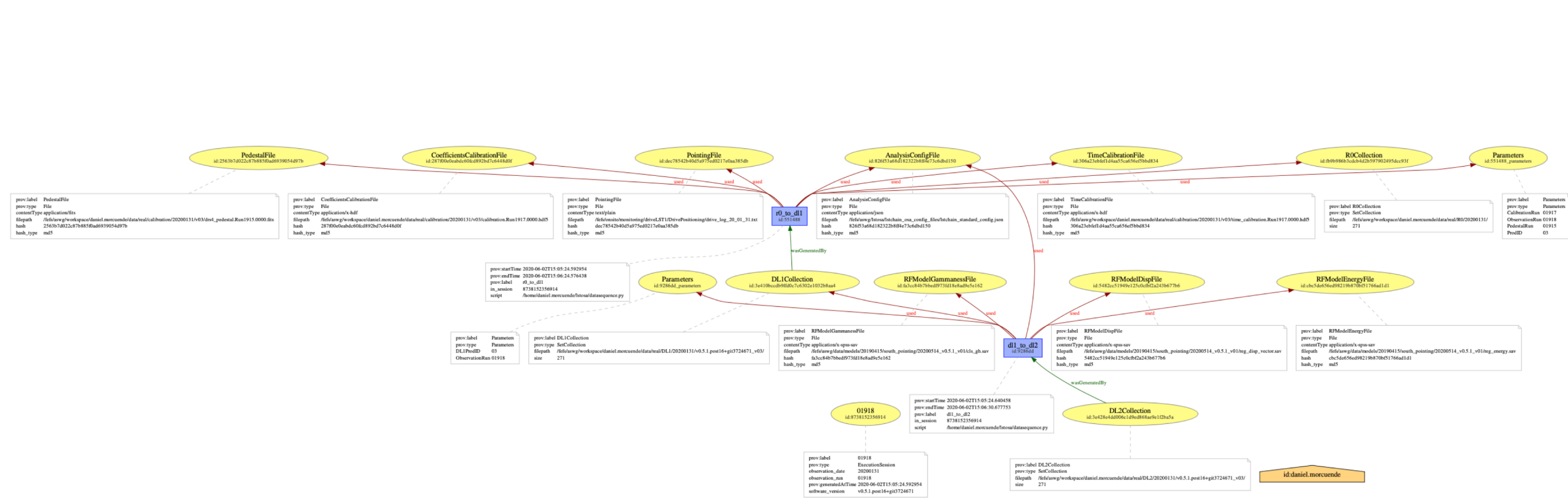


prov.log

```
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/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': 'd8077e3bbdbc371f688ae65b0972e212', 'name': 'CoefficientsCalibrationFile', 'filepath': '/fefs/aswg/data/real/calibration/20191123/v03/calibration.Run1614.0000.hdf5', 'hash': 'd8077e3bbdbc371f688ae65b0972e212', '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/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': 'd8077e3bbdbc371f688ae65b0972e212', '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



Lessons learnt

The **importance of a model** to capture **interlinked info** among activities.

Structured logging may be a solution for small session provenance storage.

Need of a **provenance query mechanism** for detailed analysis and inspection.

This is easier if provenance info is not stored in log files but in a RDBMS or noSQL **database**.

Independent capture from different dependent software packages is possible/desirable?

LSTOSA requires Isthain

Isthain requires ctapipe

Post-processing of captured provenance info may be needed to filter raw provenance according to specific needs and/or artificially produce different **levels of granularity**.