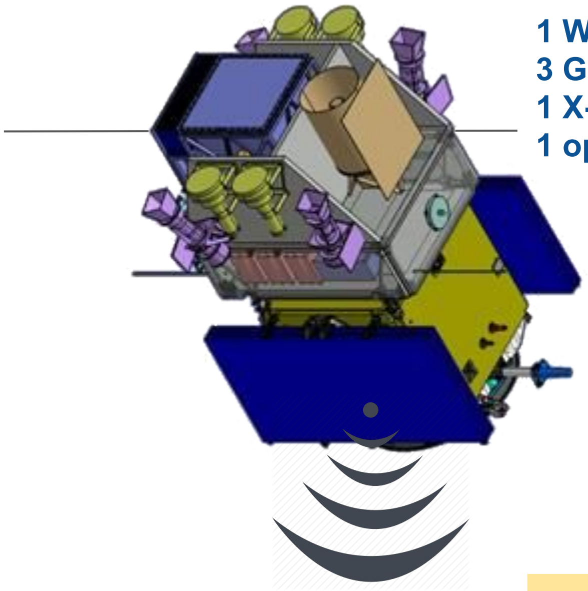
Annotating FITS Files with VO tags *SVOM case*

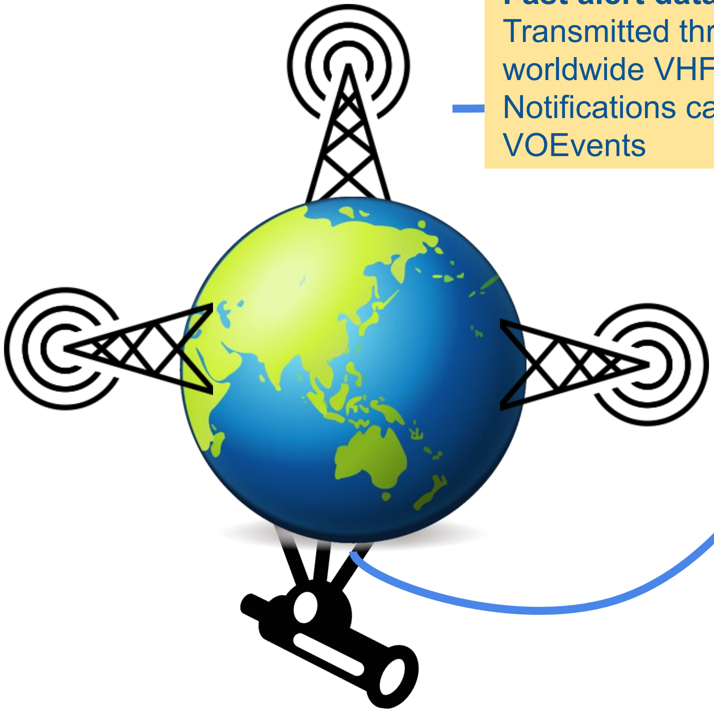
Laurent Michel - Mireille Louys
Strasbourg Observatory

The SVOM mission



1 Wide field Gamma Ray camera
3 Gamma Ray monitors
1 X-ray camera
1 optical telescope

X-Band data
Downlink 6 times a day



Fast alert data
Transmitted through a worldwide VHF network
Notifications carried out by VOEvents

Follow-up Data
2 ground IR and visible telescopes
China and Mexico
Transmitted through a dedicated link

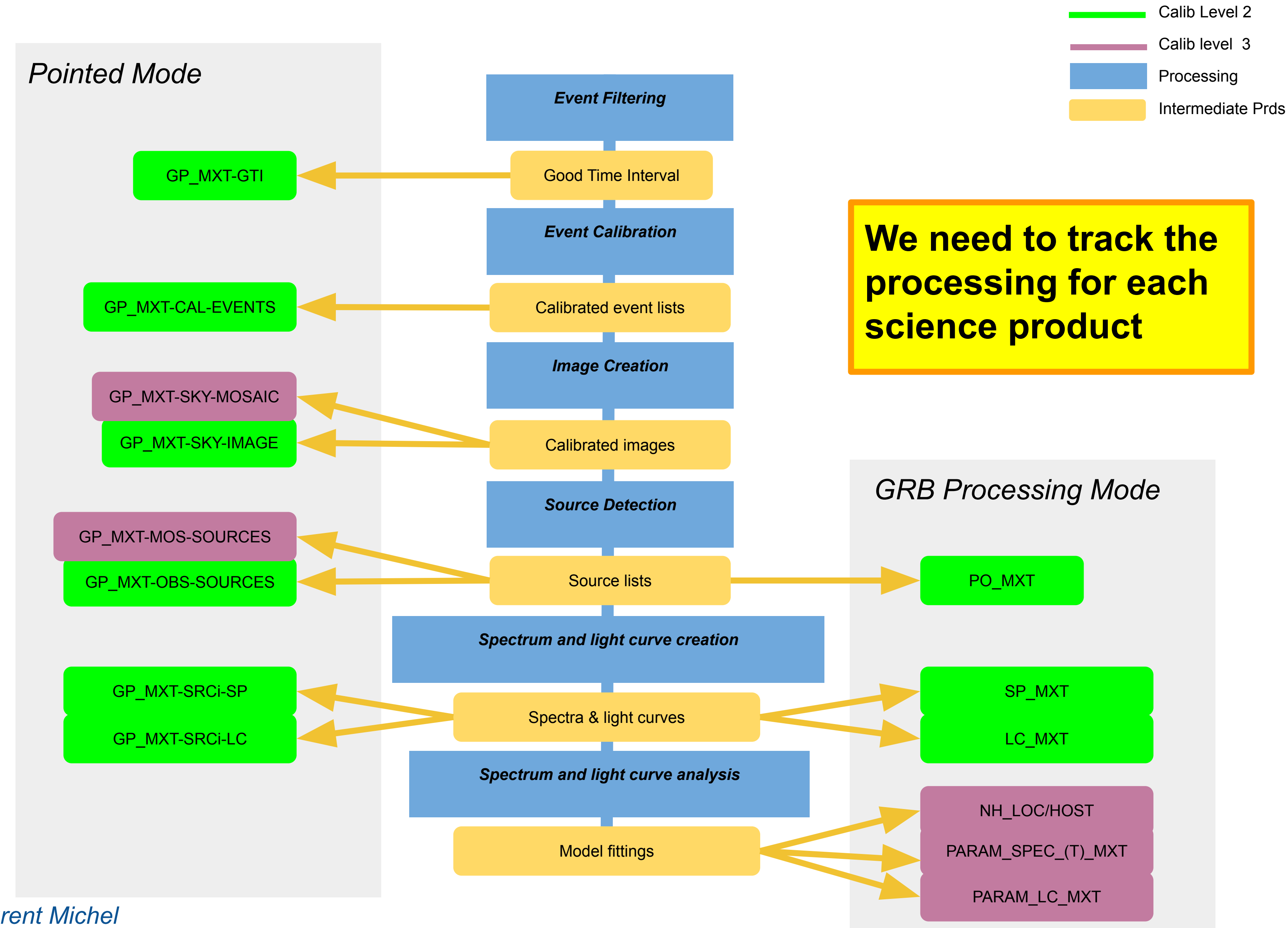
Ground segment

China and France (FSC)

VO in FITS at a Glance

- **All SVOM science products are in FITS format**
  - Mission requirement
- **Why VO tags in FITS files?**
  - OBSCORE: Facilitate the publishing in VO collections
  - PROVENANCE: Facilitate the reprocessing with a different setup
- **Guideline**
  - Clear separation between native data (OGIP kws, Mission data, science data) and VO stuff
    - One FITS extension for the VO: VO-TAGS
  - Obscore as a set ok keywords
  - Provenance: JSON serialization in a 1x1 ASCII table
- **Tools**
  - A python module to write and read data annotations
    - N ot public yet

Provenance: Pipeline Workflow



🟩 Calib Level 2
🟪 Calib level 3
🟦 Processing
🟨 Intermediate Prds

Pointed Mode

Event Filtering

Good Time Interval → GP_MXT-GTI

Event Calibration

Calibrated event lists → GP_MXT-CAL-EVENTS

Image Creation

Calibrated images → GP_MXT-SKY-MOSAIC, GP_MXT-SKY-IMAGE

Source Detection

Source lists → GP_MXT-MOS-SOURCES, GP_MXT-OBS-SOURCES, PO_MXT

Spectrum and light curve creation

Spectra & light curves → GP_MXT-SRCi-SP, GP_MXT-SRCi-LC, SP_MXT, LC_MXT

Spectrum and light curve analysis

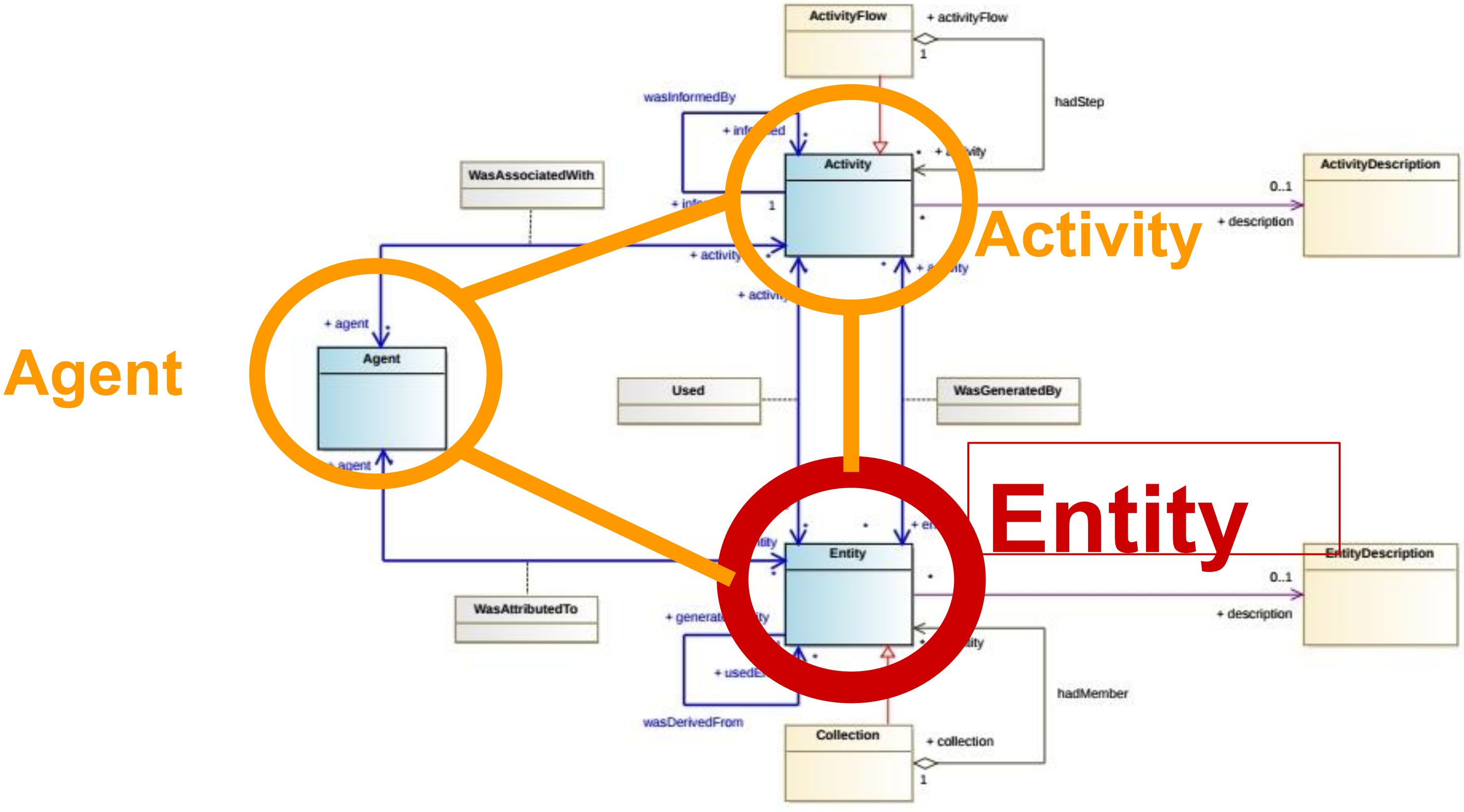
Model fittings → NH_LOC/HOST, PARAM_SPEC_(T)_MXT, PARAM_LC_MXT

GRB Processing Mode

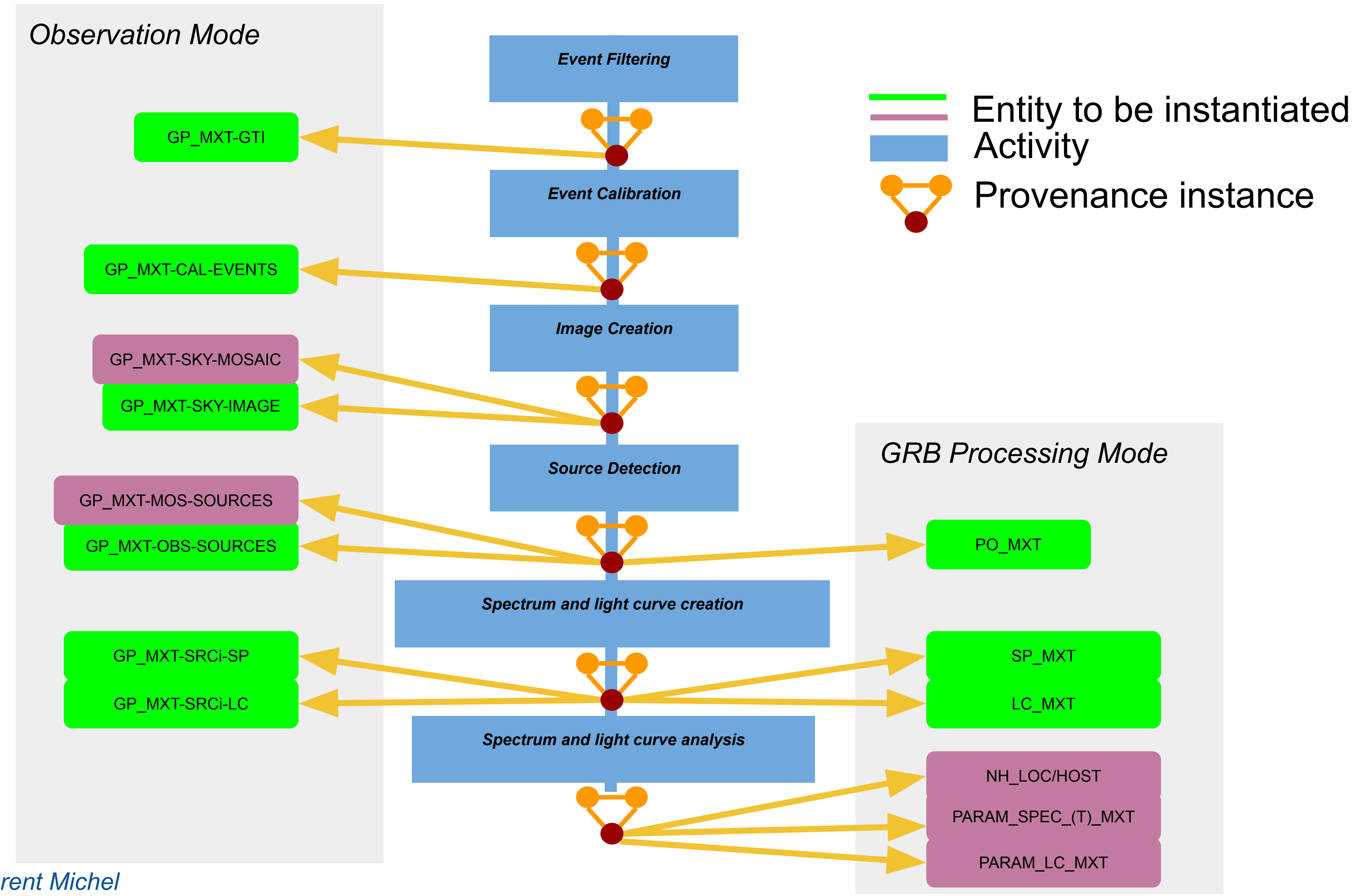
We need to track the processing for each science product

Provenance DM at a Glance

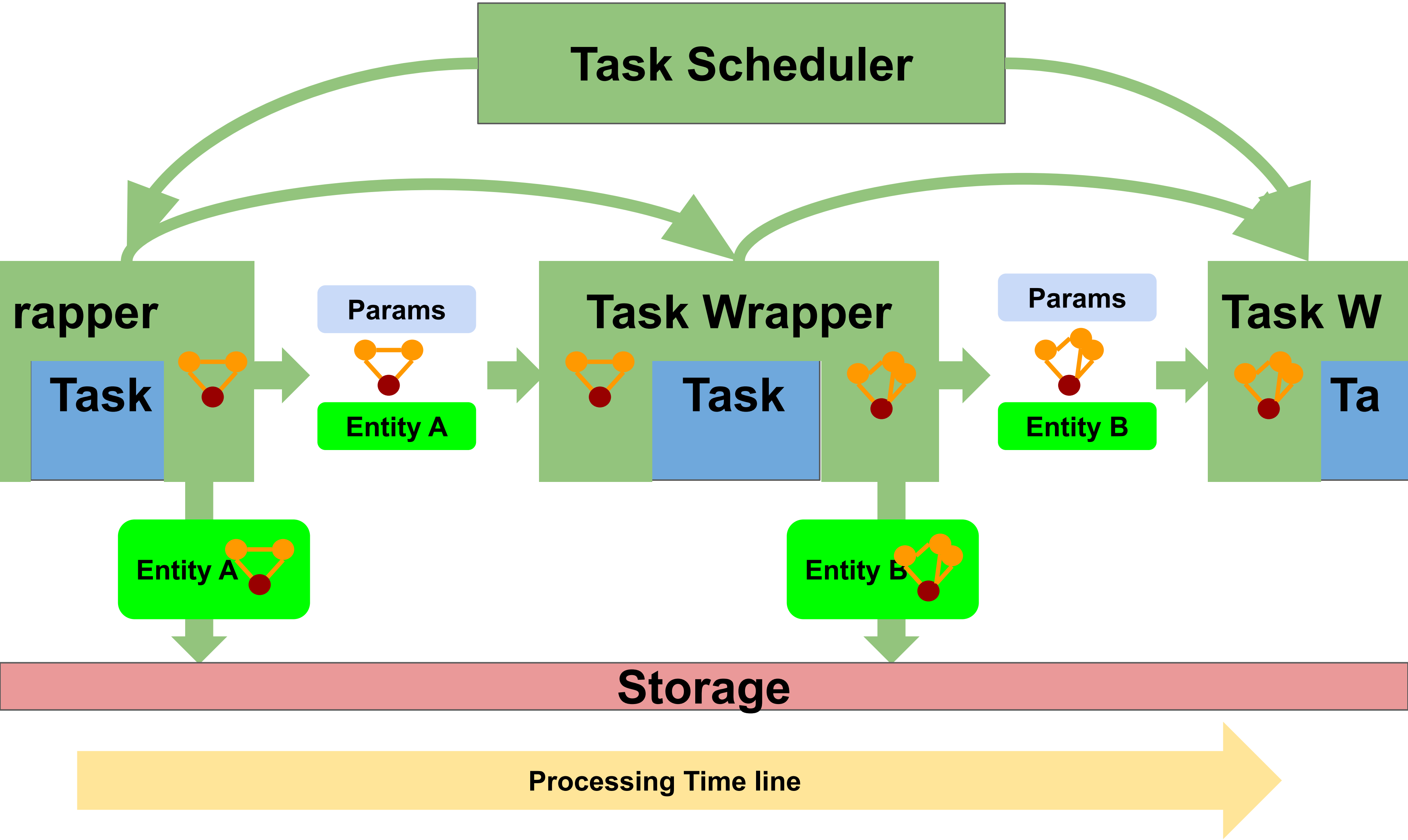
- **Designed around 3 poles** *(WARNING: the model has evolved since 2017)*
  - Activity, Agent and Entity
- ***Prov Speaking:*** **We want to describe the activities leading to our entities**
  - Entities are final science data files



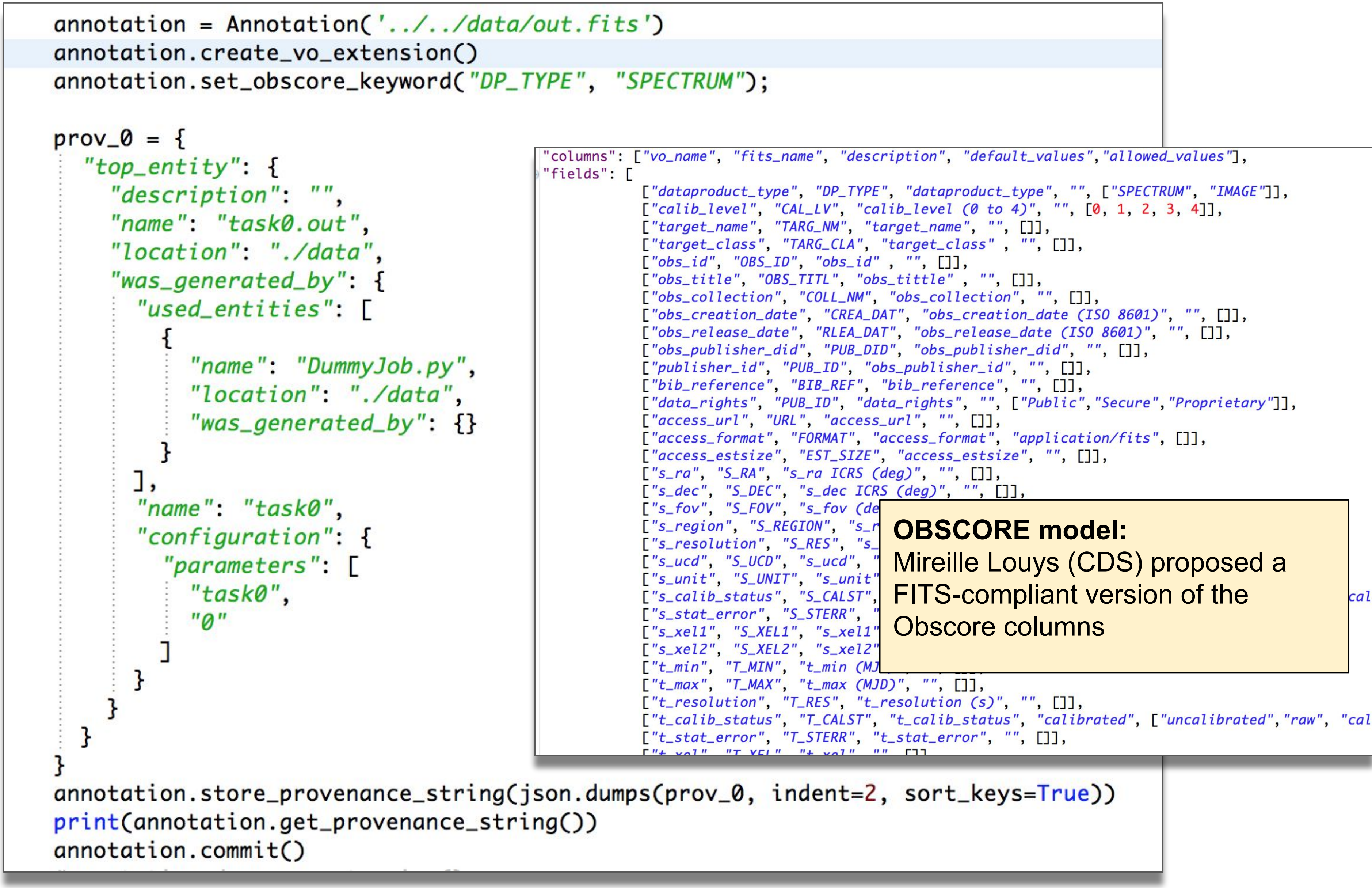
Provenance View



Incremental Provenance Construction



Python Code Snippet



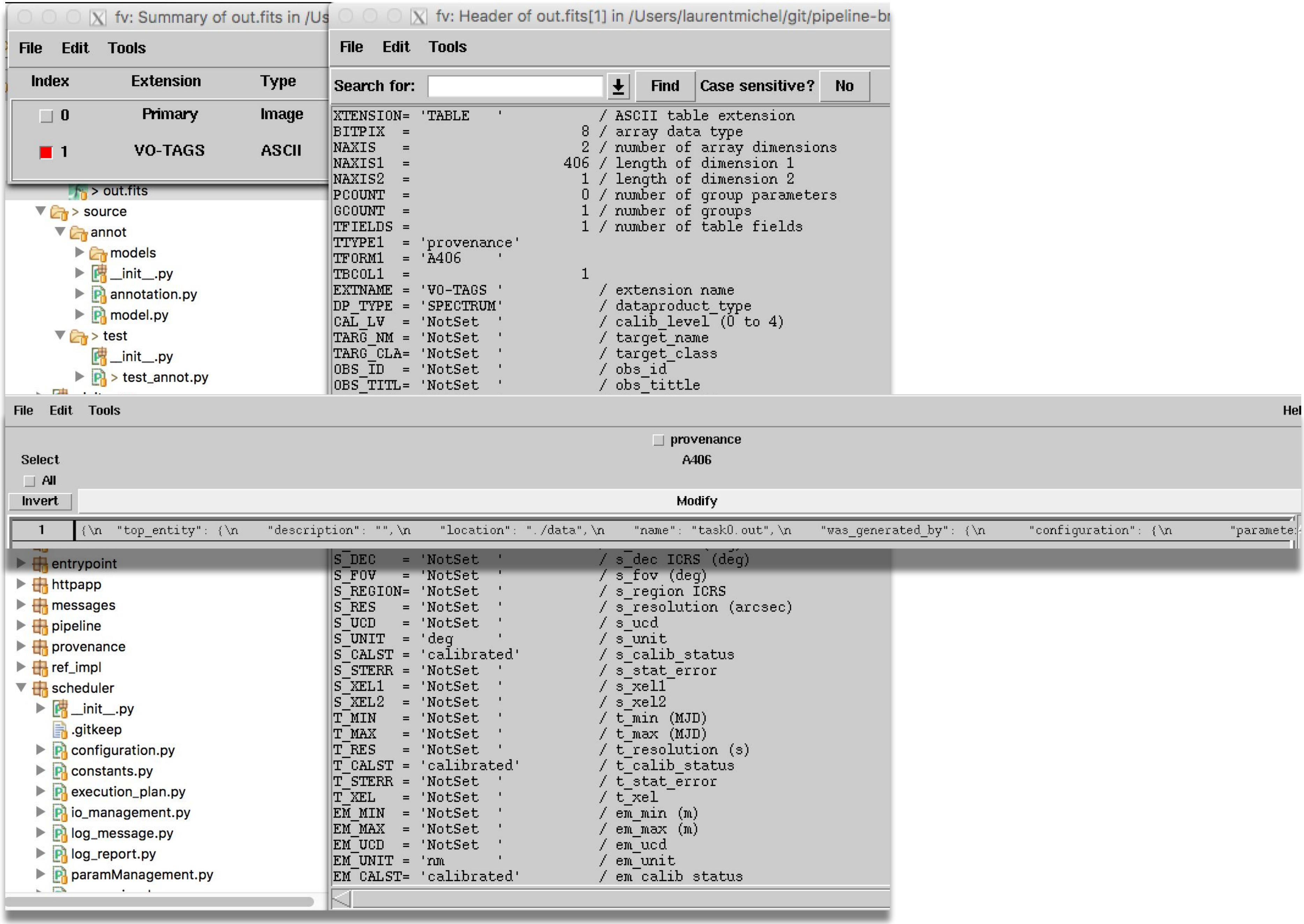
```
annotation = Annotation('../../data/out.fits')
annotation.create_vo_extension()
annotation.set_obscore_keyword("DP_TYPE", "SPECTRUM");

prov_0 = {
  "top_entity": {
    "description": "",
    "name": "task0.out",
    "location": "./data",
    "was_generated_by": {
      "used_entities": [
        {
          "name": "DummyJob.py",
          "location": "./data",
          "was_generated_by": {}
        }
      ],
      "name": "task0",
      "configuration": {
        "parameters": [
          "task0",
          "0"
        ]
      }
    }
  }
}
annotation.store_provenance_string(json.dumps(prov_0, indent=2, sort_keys=True))
print(annotation.get_provenance_string())
annotation.commit()
```

```
"columns": ["vo_name", "fits_name", "description", "default_values","allowed_values"],
"fields": [
            ["dataproduct_type", "DP_TYPE", "dataproduct_type", "", ["SPECTRUM", "IMAGE"]],
            ["calib_level", "CAL_LV", "calib_level (0 to 4)", "", [0, 1, 2, 3, 4]],
            ["target_name", "TARG_NM", "target_name", "", []],
            ["target_class", "TARG_CLA", "target_class" , "", []],
            ["obs_id", "OBS_ID", "obs_id" , "", []],
            ["obs_title", "OBS_TITL", "obs_tittle" , "", []],
            ["obs_collection", "COLL_NM", "obs_collection", "", []],
            ["obs_creation_date", "CREA_DAT", "obs_creation_date (ISO 8601)", "", []],
            ["obs_release_date", "RLEA_DAT", "obs_release_date (ISO 8601)", "", []],
            ["obs_publisher_did", "PUB_DID", "obs_publisher_did", "", []],
            ["publisher_id", "PUB_ID", "obs_publisher_id", "", []],
            ["bib_reference", "BIB_REF", "bib_reference", "", []],
            ["data_rights", "PUB_ID", "data_rights", "", ["Public","Secure","Proprietary"]],
            ["access_url", "URL", "access_url", "", []],
            ["access_format", "FORMAT", "access_format", "application/fits", []],
            ["access_estsize", "EST_SIZE", "access_estsize", "", []],
            ["s_ra", "S_RA", "s_ra ICRS (deg)", "", []],
            ["s_dec", "S_DEC", "s_dec ICRS (deg)", "", []],
            ["s_fov", "S_FOV", "s_fov (de
            ["s_region", "S_REGION", "s_r
            ["s_resolution", "S_RES", "s_
            ["s_ucd", "S_UCD", "s_ucd", "
            ["s_unit", "S_UNIT", "s_unit"
            ["s_calib_status", "S_CALST",                                    cal
            ["s_stat_error", "S_STERR", "
            ["s_xel1", "S_XEL1", "s_xel1"
            ["s_xel2", "S_XEL2", "s_xel2"
            ["t_min", "T_MIN", "t_min (MJ
            ["t_max", "T_MAX", "t_max (MJD)", "", []],
            ["t_resolution", "T_RES", "t_resolution (s)", "", []],
            ["t_calib_status", "T_CALST", "t_calib_status", "calibrated", ["uncalibrated","raw", "cal
            ["t_stat_error", "T_STERR", "t_stat_error", "", []],
            ["t_xel", "T_XEL", "t_xel", "", []]
```

OBSCORE model:
Mireille Louys (CDS) proposed a FITS-compliant version of the Obscore columns

Laurent Michel

VO Stuff with FV



Index	Extension	Type
□ 0	Primary	Image
🟥 1	VO-TAGS	ASCII

```
XTENSION= 'TABLE   '           / ASCII table extension
BITPIX  =                    8 / array data type
NAXIS   =                    2 / number of array dimensions
NAXIS1  =                  406 / length of dimension 1
NAXIS2  =                    1 / length of dimension 2
PCOUNT  =                    0 / number of group parameters
GCOUNT  =                    1 / number of groups
TFIELDS =                    1 / number of table fields
TTYPE1  = 'provenance'
TFORM1  = 'A406    '
TBCOL1  =                    1
EXTNAME = 'VO-TAGS '           / extension name
DP_TYPE = 'SPECTRUM'           / dataproduct_type
CAL_LV  = 'NotSet  '           / calib_level (0 to 4)
TARG_NM = 'NotSet  '           / target_name
TARG_CLA= 'NotSet  '           / target_class
OBS_ID  = 'NotSet  '           / obs_id
OBS_TITL= 'NotSet  '           / obs_tittle
```

□ provenance
A406

```
1   {\n  "top_entity": {\n    "description": "",\n    "location": "./data",\n    "name": "task0.out",\n    "was_generated_by": {\n      "configuration": {\n        "paramete
```

```
S_DEC   = 'NotSet  '           / s_dec ICRS (deg)
S_FOV   = 'NotSet  '           / s_fov (deg)
S_REGION= 'NotSet  '           / s_region ICRS
S_RES   = 'NotSet  '           / s_resolution (arcsec)
S_UCD   = 'NotSet  '           / s_ucd
S_UNIT  = 'deg     '           / s_unit
S_CALST = 'calibrated'         / s_calib_status
S_STERR = 'NotSet  '           / s_stat_error
S_XEL1  = 'NotSet  '           / s_xel1
S_XEL2  = 'NotSet  '           / s_xel2
T_MIN   = 'NotSet  '           / t_min (MJD)
T_MAX   = 'NotSet  '           / t_max (MJD)
T_RES   = 'NotSet  '           / t_resolution (s)
T_CALST = 'calibrated'         / t_calib_status
T_STERR = 'NotSet  '           / t_stat_error
T_XEL   = 'NotSet  '           / t_xel
EM_MIN  = 'NotSet  '           / em_min (m)
EM_MAX  = 'NotSet  '           / em_max (m)
EM_UCD  = 'NotSet  '           / em_ucd
EM_UNIT = 'nm      '           / em_unit
EM_CALST= 'calibrated'         / em_calib_status
```

```
> out.fits
▼ > source
  ▼ annot
    ▶ models
    ▶ __init__.py
    ▶ annotation.py
    ▶ model.py
  ▼ > test
      __init__.py
    ▶ > test_annot.py
▶ entrypoint
▶ httpapp
▶ messages
▶ pipeline
▶ provenance
▶ ref_impl
▼ scheduler
  ▶ __init__.py
    .gitkeep
  ▶ configuration.py
  ▶ constants.py
  ▶ execution_plan.py
  ▶ io_management.py
  ▶ log_message.py
  ▶ log_report.py
  ▶ paramManagement.py
```