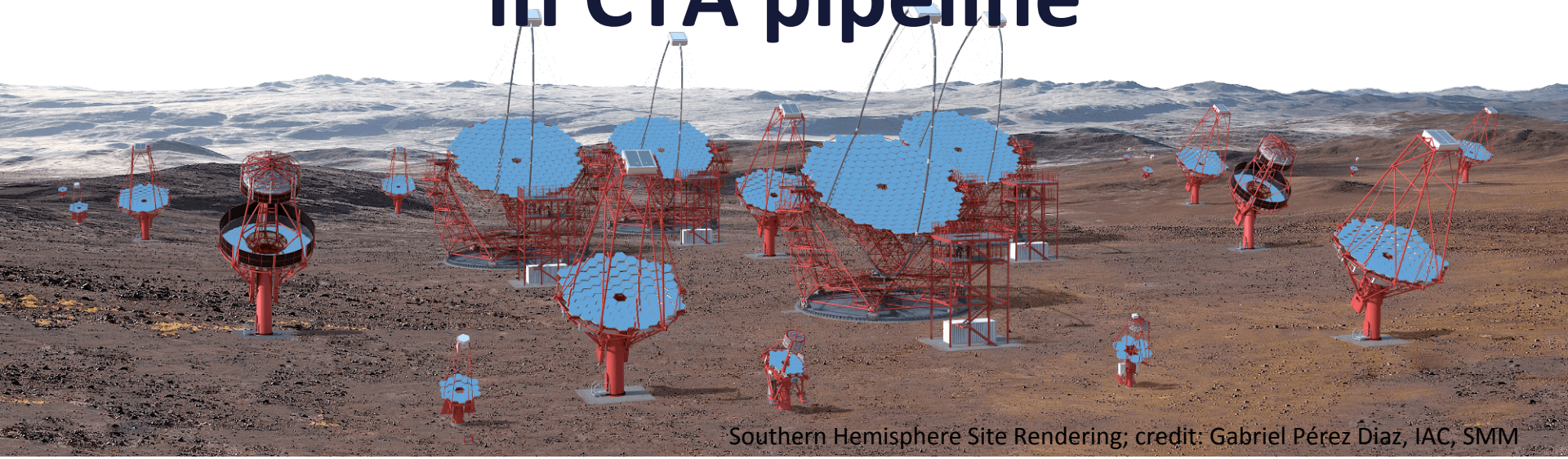
Provenance implementation in CTA pipeline



Southern Hemisphere Site Rendering; credit: Gabriel Pérez Diaz, IAC, SMM

Michèle Sanguillon[(1)], Catherine Boisson[(2)], Johan Bregeon[(1)], Karl Kosack[(3)], Nicolas Renault-Tonacci[(2)], Mathieu Servillat[(2)]

[(1)] LUPM, Montpellier, France [(2)] LUTH, Meudon, France [(3)] CEA, Paris, France

Context

- Next Generation Gamma-Ray Astronomy (succeeding H.E.S.S., MAGIC and VERITAS)
- Two arrays of 99 (South) and 19 (North) Cherenkov telescopes (4, 12 et 24 m in diametre)
- Observatory open to the Astronomy community
- Timeline:

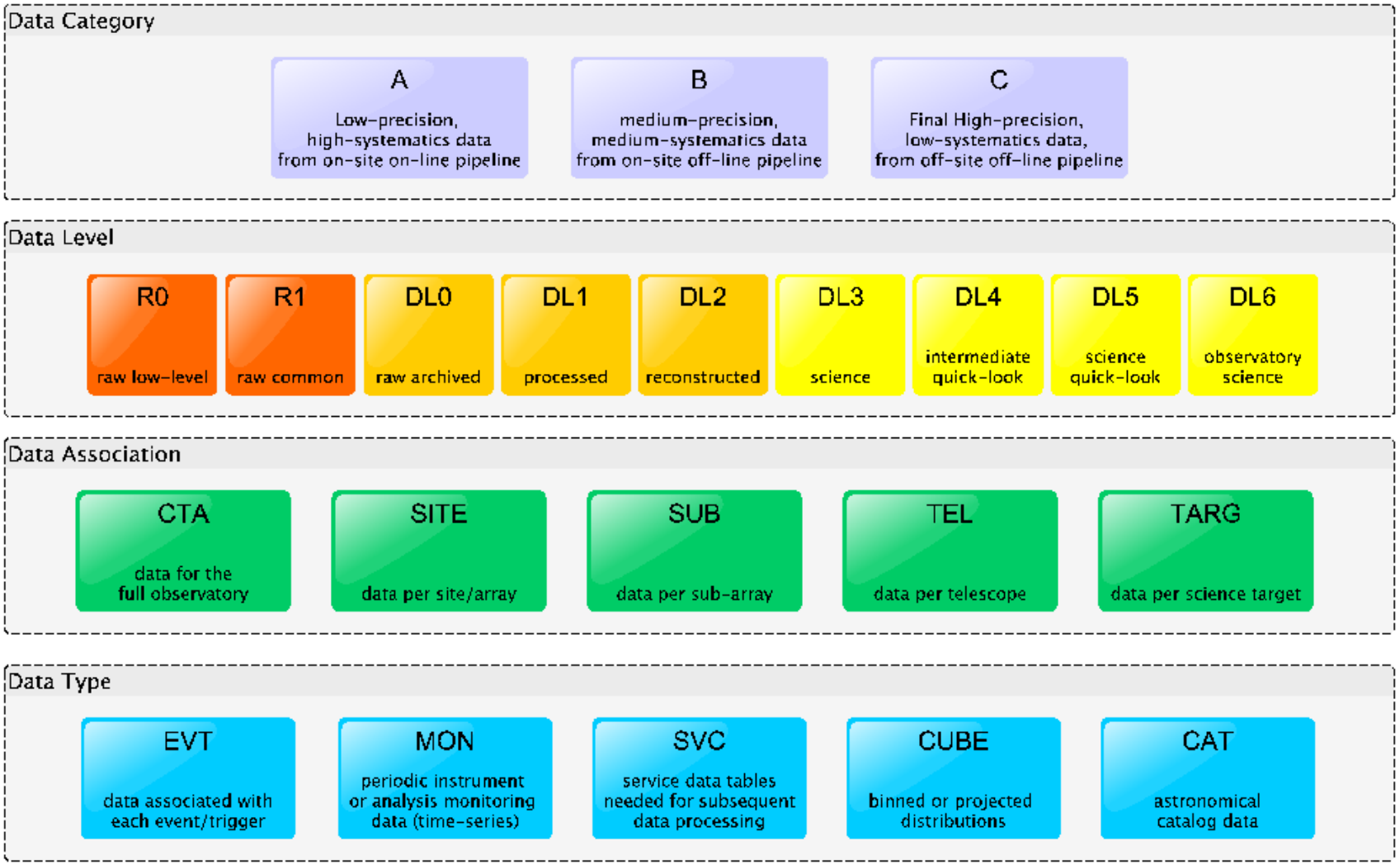
Pre-Construction phase:
- Simulations

Pre & Production phase:
- Data model definition
- Software development
  - Prototype OPUS
  - ctapipe project

Operation phase:
- Acquisitions
- High level data available

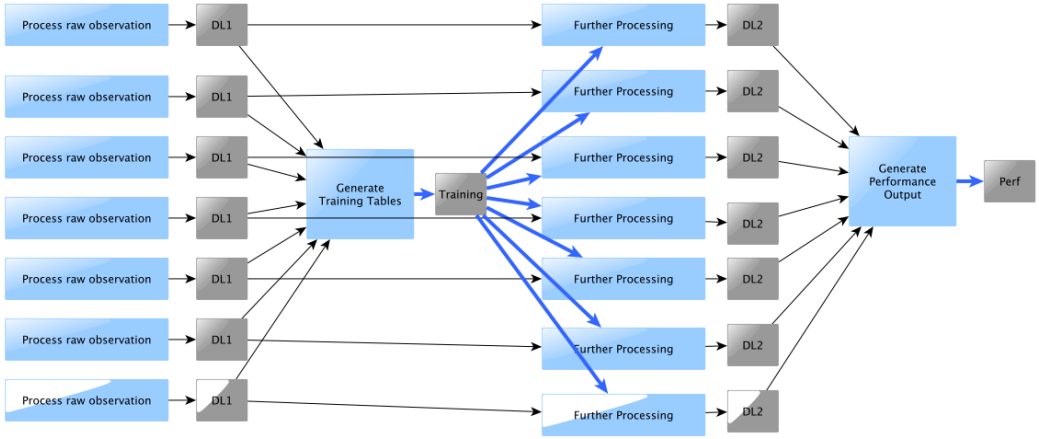
2005 — 2019 — 2022 — 2025

Data Model


Data Category
A
Low-precision, high-systematics data from on-site on-line pipeline
B
medium-precision, medium-systematics data from on-site off-line pipeline
C
Final High-precision, low-systematics data, from off-site off-line pipeline
Data Level
R0
raw low-level
R1
raw common
DL0
raw archived
DL1
processed
DL2
reconstructed
DL3
science
DL4
intermediate quick-look
DL5
science quick-look
DL6
observatory science
Data Association
CTA
data for the full observatory
SITE
data per site/array
SUB
data per sub-array
TEL
data per telescope
TARG
data per science target
Data Type
EVT
data associated with each event/trigger
MON
periodic instrument or analysis monitoring data (time-series)
SVC
service data tables needed for subsequent data processing
CUBE
binned or projected distributions
CAT
astronomical catalog data

Software

- Software tools are used or under development but not yet validated by the CTA Observatory
- Usage:
  - Workflow: Set of data “Transformations” that are parallelized into Jobs
  - Job: Set of Multiple tools running sequencially
  - Tool: simple task



- Ctapipe :
  - https://github.com/cta-observatory/ctapipe
  - Framework containing the different tools (Python)

What provenance?

- Implementation of Provenance includes to work on:
  - How to catch the information?
  - How to store the information?
    - In a database: ProvenanceDM!
    - In data products (header, etc.)
    - In files
  - How to retrieve the information?
    - ProvSAP: to request Provenance path
    - ProvTAP: to search data products based on Provenance

Capture

- The provenance of all data (low and high levels) must be carefully registered and easily available for further use.
- Provenance is captured *a priori* during a workflow execution at the level of each tool
- Provenance Tracking is integrated in the ctapipe framework => Provenance module (ctapipe.core.provenance)

ctapipe

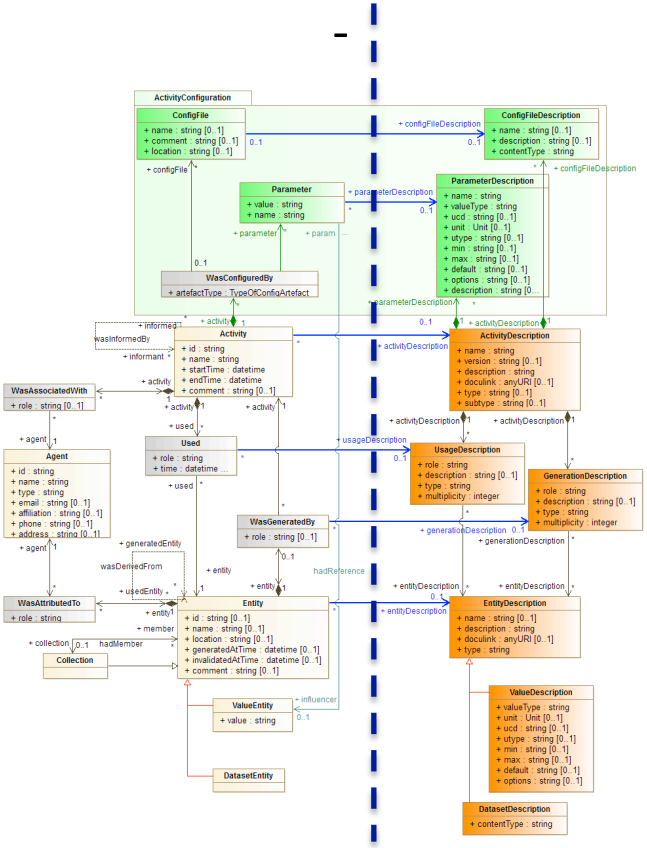
- Provenance module
  - ctapipe.core.provenance
  - Developed so that each tool automatically register provenance information
  - This module has been designed to return a dictionary with all provenance information


```
[{'activity_name': 'ctapipe-display-muons',
  'activity_uuid': '93fc2206-59de-4852-868d-ec045ec2
5d1d',
  'config': {'MuonDisplayerTool': {'events': 'proton
_20deg_180deg_run22___cta-prod3-demo-2147m-LaPalma-b
aseline.simtel.gz'}},
  'duration_min': 0.01649999999998819,
  'input': [{'role': 'dl0.sub.evt',
    'url': '/Users/bourgeat/Documents/CTA/Provenance
/ctasoft/ctapipe/tests/proton_20deg_180deg_run22___c
ta-prod3-demo-2147m-LaPalma-baseline.simtel.gz'}],
  'output': [{'role': 'dl1.tel.evt.muon',
    'url': '/Users/bourgeat/Documents/CTA/Provenance
/ctasoft/ctapipe/tests/muons.hdf5'}],
  'start': {'time_utc': '2019-05-07T06:28:44.605'},
  'status': 'completed',
  'stop': {'time_utc': '2019-05-07T06:28:45.595'},
  'system': {'arguments': ['/Users/bourgeat/anaconda
3/lib/python3.6/site-packages/ipykernel_launcher.py'
,
    '-f',
    '/Users/bourgeat/Library/Jupyter/runtime/kernel-
d793d0e4-abd3-4989-a71c-19b68194ea9e.json'],
   'ctapipe_resources_version': '0.2.15',
   'ctapipe_svc_path': None,
   'ctapipe_version': '0.6.1',
```

Fill the Prov DB

The Provenance Database will be fed
- initially with the tool description information
- then at the end of each run of tool with the information from the dictionary.

Information relative to each run of tool



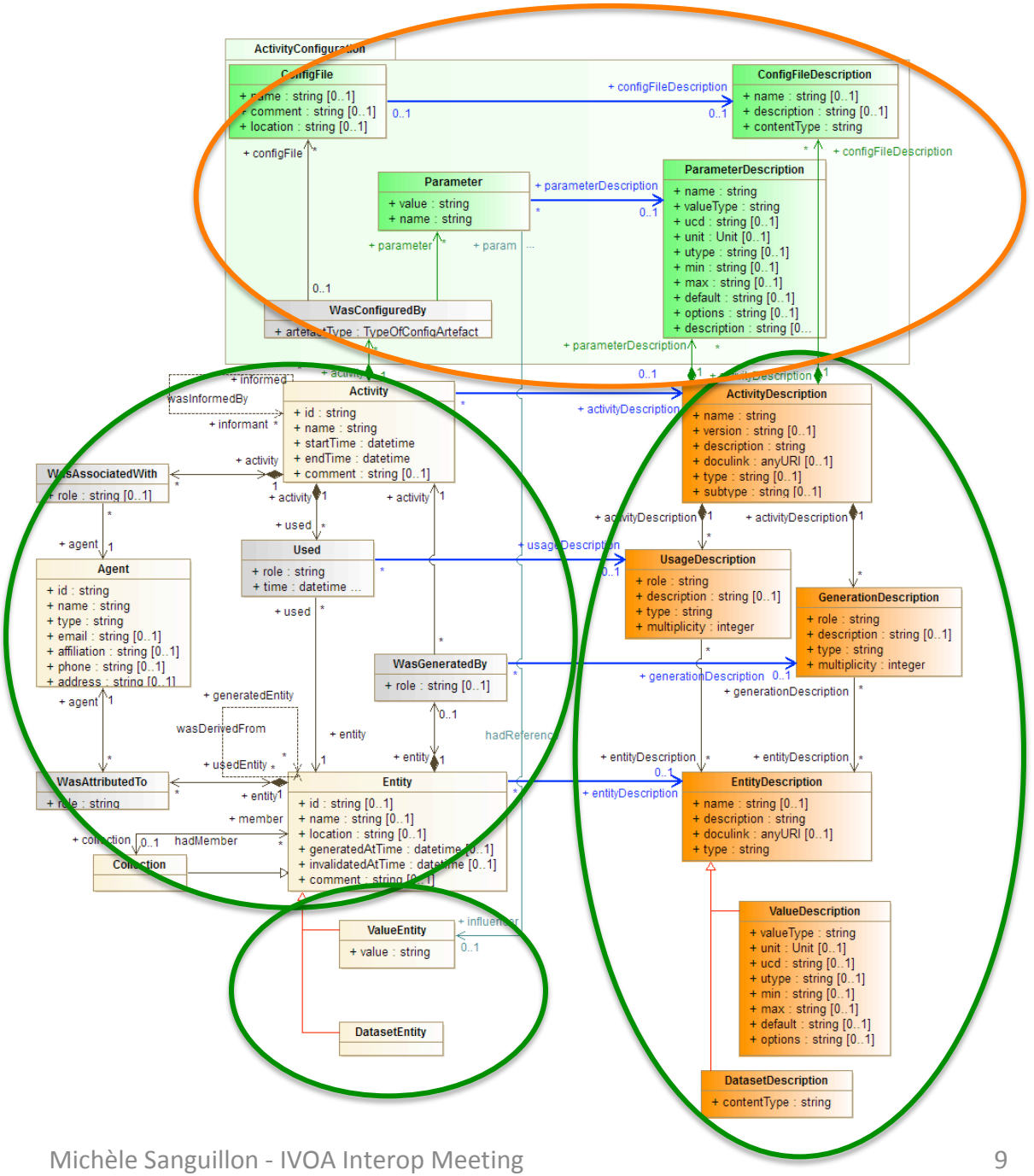
Descriptions relative to each type of tool

Storage in the Provenance DB

Dev steps:

1:

2:



Choices made

- Rules to associate an item with its description
  - Activity: activity_name + cta_pipe_version
  - Entity: activity description + role
- Unique Identifiers
  - Activity.id: uuid generated in the job
  - Entity.id: uuid depending on file(location+name), DIRAC lfn?
  - Relations Used.id, WasGeneratedBy.id, WasAttributedTo.id, WasAssociatedWith.id are intergers and autoincremented

I thank you for your attention

Any questions?