Provenance Data Model Status

Mireille Louys , CDS & ICube , Strasbourg University
François Bonnarel, CDS
Mathieu Servillat, LUTH, Paris
and the IVOA provenance team
within the IVOA DM working group

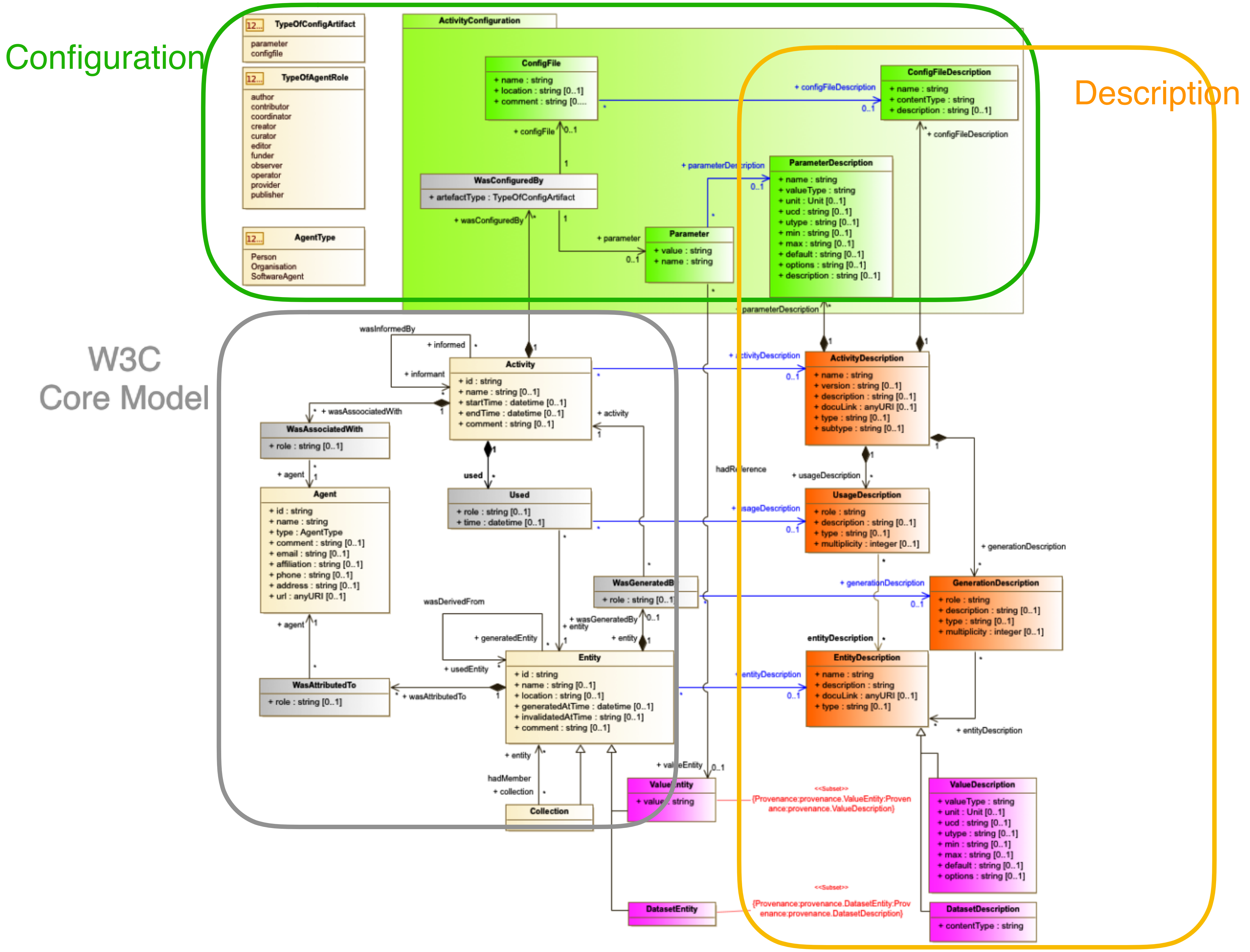
CENTRE DE DONNÉES ASTRONOMIQUES DE STRASBOURG

New Data Model Document

https://volute.g-vo.org/svn/trunk/projects/dm/provenance/ProvDM/doc/

- Text updated according to suggestions when appropriate
- RFC comments answered
- Vocabulary terms to be studied with the semantics group
    - during the current review of Vocabularies 2.0 WD
    - poll various projects for feedback
- To do :
- Minor changes to the Modelio project
- Reprocess VODML documents
- VODML XML reference doc—> Provenance.xml
- Data model description —> Provenance.html

Data Model Overview



What do we have now?

- A data model
- Experience with various use-cases
  - Working examples : ProvFocusAstericsExamples
- A list of implementations
  - CTA pipeline / OPUS jobs submission (end 2017)
  - Pollux theoretical spectra (2017)
  - RAVE prototype (07/2018)
  - PROV of Images prototype in Triplestore (2018)
  - MuseWise PROV prototype (02/2019)
  - Applause on line DB (05/2019)
  - CDS PROV-HiPS Image database (10/2019)
  - Prov-TAP prototype (10/2019)
  - CTA/HESS gammapy provenance tracking mechanism (10/2019)
- a library for translating serialization formats `voprov` (Michèle Sanguillon)
- Connexions to on going projects CWL Common Workflows Langage (M. Crusoe's talk @ADASS)

gammapy provenance tracking

from Mathieu Servillat

- Define the activity template.   YAML file
- Feed the template during execution for each step   Activitydescription tree(orange classes)
- Record provenance in a log file
- Extract provenance tags from the log —> Prov DB
- https://wiki.ivoa.net/internal/IVOA/InterOpOct2019DM/gammapy-prov.pdf

A best effort strategy

- This DM tries to cover all possible features of provenance in order to meet our use-cases. It is a rich model.
- The distribution of provenance metadata comes with a best effort strategy.
- On the data provider's side, the cost in implementing these features needs to be balanced with
  - an understandable content exportable outside the project
  - columns clearly mapped means better queries prepared by the user or by the wrapping API
  - enhance data search with provenance flavor selection
  - maintenance benefits to better monitor the archive collections
- On the client side, an application querying several data centers will have to deal with the various level of completeness chosen by the data centers.

Heterogeneous provenance coverage

- Some of our implementations services do not trace all classes.
- Some data products are fully traced, and some not within one project.

This is OK

- In order to stimulate the uptake of provenance metadata into collections:
- Encourage best effort

DM Class Feature coverage

DM Feature/ Project	Institute	Core DM	ValueEntity/ DatasetEntity	Parameter and configFile	DataFlow wasDerived From	Taskflow wasInformed By	ActivityDescrip tion
CTA Opus	LUTH paris	x	x	x	x	?	tree
ProvHiPS	CDS	x	x	x	x	x	tree
Image Triplestore	CDS	x	x	x	x	no	tree
Pollux	LUPM Montpellier	x	no	no	no	no	Activity/Entity
Dirac CTA reduction	LUPM Montpellier	x	x	x	no	no	tree
RAVE Prov	AIP Postdam	x	generic entity	no	no	no	Activity only
MuseWise prov	AIP Postdam	x	?	no	no	no	tree
Applause	AIP Postdam	x	no	no	no	no	tree

Accommodate differences and evolution

- Should we just downgrade existing prototypes and ignore implementation experience ? no
- Endorse a simplified W3C profile? then why IVOA PROV?
- Features are used by some of us
  - ex. configuration, wasDerivedFrom, wasInformedBy

Define a **Provenance Feature profile** and interpretation rules

- Explain the implementations choices and strategies
- Describe profile implementations as IVOA notes
- Check usage and implementation from other projects not us …

Compliance

- Provenance services should mention their Feature profile
- A TAP service which does not serve queries based on wasDerivedFrom should return a message like « wasDerivedFrom not implemented »
- Is my service compliant to the model ? Too vague a question :
  - We need to specify for which feature profile
    - P1: Core+ ActivityDesc level 1
    - P2: Core+ Dataflow
    - P3: Core+ taskflow+dataflow
    - etc…
    - P6: Full model (all features)

For one profile, classes with mandatory attributes should be filled.

Relations in the feature profile should be instantiated as in the model.

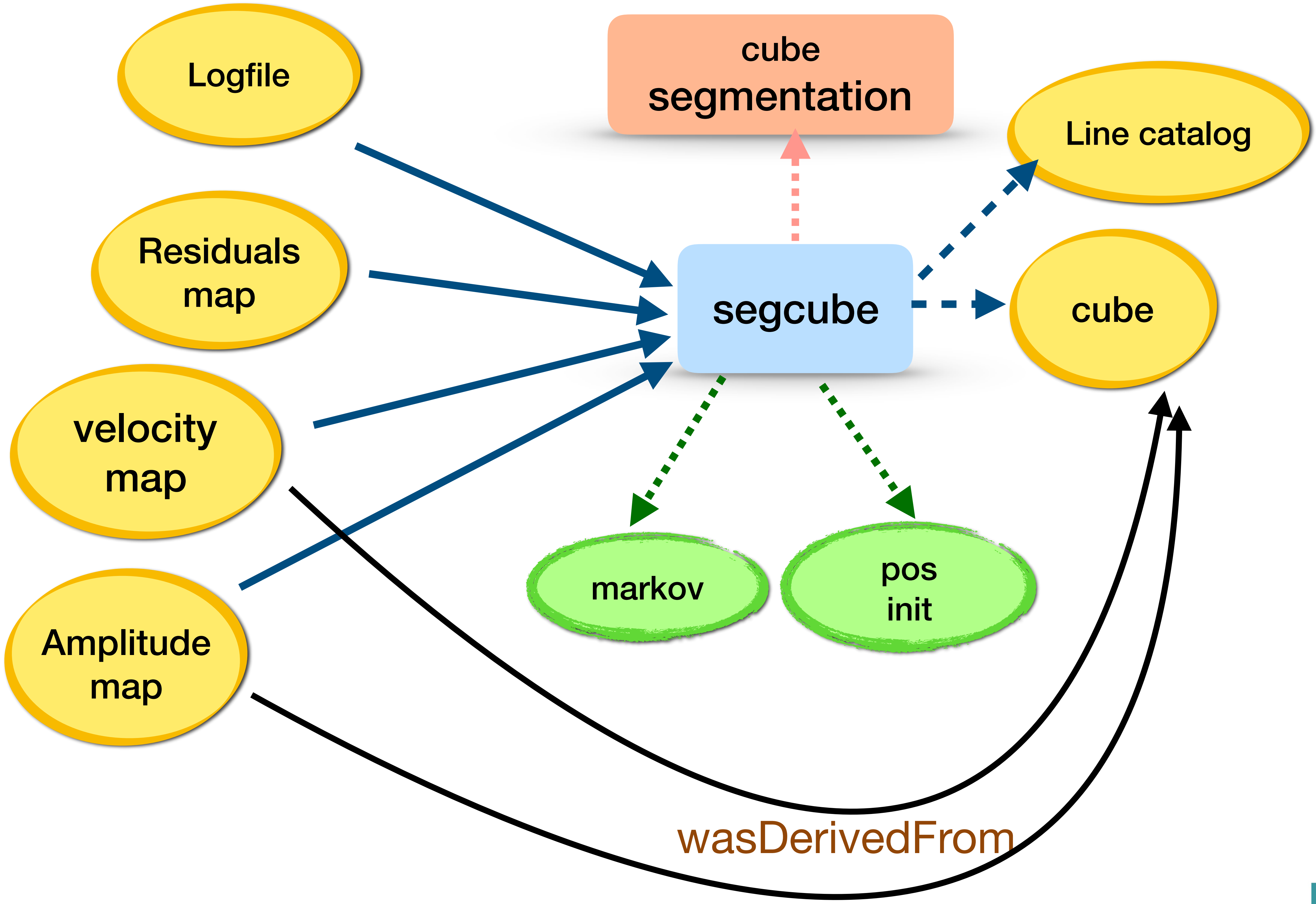
Feature profiles

- **Core Model** : the basic loop : an Activity with consumed and produced Entities and their links to Agents.
- **ActivityConfiguration** feature
  - to allow a focus on parameter/configuration search by specialised classes
  - the types and descriptions of Parameters and ConfigFiles are exposed as well
- **ActivityDescription feature**
  - level 1: an ActivityDescription instance is present and explains Activity instances bound to it
  - level 2: The full tree with root as an ActivityDescription instance and the Usage/generation description branches as well as the Configuration Description branch are present

Feature profiles (2)

- **Data flow** feature: To illustrate the processing dependencies between datasets
  - wasDerivedFrom encodes the progenitor link for one processing step.
  - it emphasizes data dependency through time
  - Not applicable to Parameter instances
- **Task chain** feature
  - to chain activities along the time line and show their execution dependencies
  - wasInformedBy encodes the progenitor
- Both relations can be added on top of the Core Feature. —> this is added semantics
  - These are emphasized relations between data or activities that the data provider think relevant to highlight
  - All results of an Activity instance are not necessarily exposed as derived products from all inputs
  - When parameter values are encoded as entities with Used.role=setup, they are part of the used Entity set, however not traced as progenitor data for the results of this particular activity

wasDerivedFrom semantics



Serialisation Formats

- Ready :
  - Gammapy Provenance embedded VOTable, PROV-N, PROV-XML
  - CTA Pipe/DIRAC text JSON
  - OPUS job submission and execution (LUTH) VOTable, JSON
  - Image database prototype in Triplestore (CDS) RDF/ttl
  - HiPS Image database (CDS) with PROV-TAP VOTable
  - Applause VOTable
  - RAVE implementation (AIP, Postdam)
    - Simple access (Prov-SAP) prototype Prov-N, PROV-JSON
  - Provenance for Pollux DB & *voprov* library (LUPM) VOTable, Prov-N, PROV-JSON
  - Under study :
  - SVOM pipeline execution tracking JSON FITS embedded

Access Protocols

- DAL protocols to serve provenance metadata
- Provenance TAP protocol PROV-TAP (WD in progress)
    - cf talk by François Bonnarel @Apps&DAL session
      https://wiki.ivoa.net/twiki/bin/view/IVOA/InterOpOct2019DAL#DAL
    - PROV-TAP working draft issued in the Working Group
- Provenance simple access protocol Prov-SAP
    - Many Implementations in RAVE, CTA, Pollux
- Both waiting for the DM being approved

Review comments are welcome on the PROV-TAP page

Thanks

Questions ?
Comments ?