VAMPIRA: Provenance Generation

Michael Johnson, David J. Champion, Marta Dembska, Hans-Rainer Klöckner, Kristen Lackeos,
Albina Muzafarova, Marcus Paradies, Sirko Schindler

MAX-PLANCK-GESELLSCHAFT

DLR

Provenance

Originated in the art world to ascribe value to a work

- Documents the record of authorship and ownership

When applied to data processing it records:

- Metadata on initial, final and/or intermediate data
- Processes applied
- Parameters parsed
- Software versioning
- User data

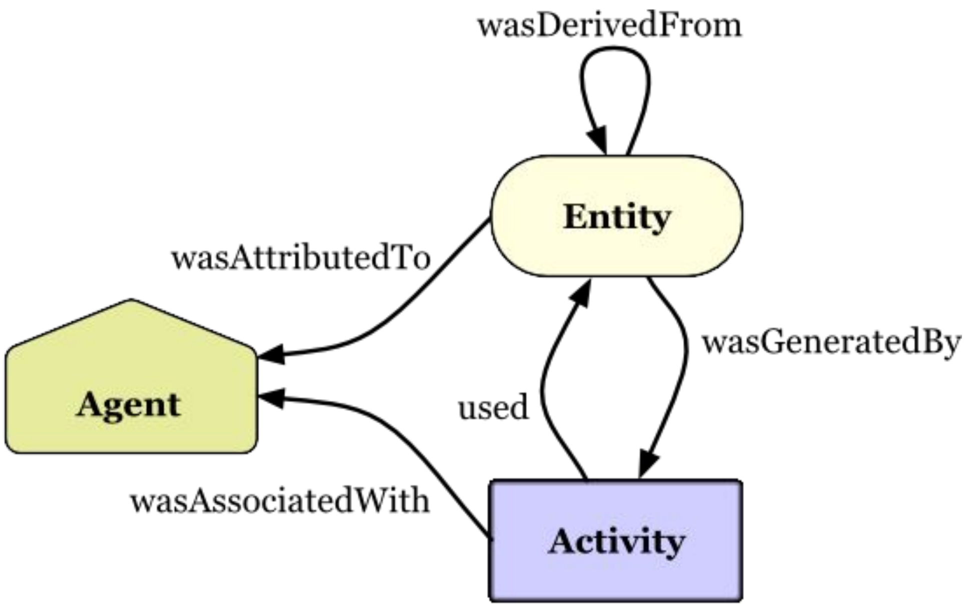
PROV

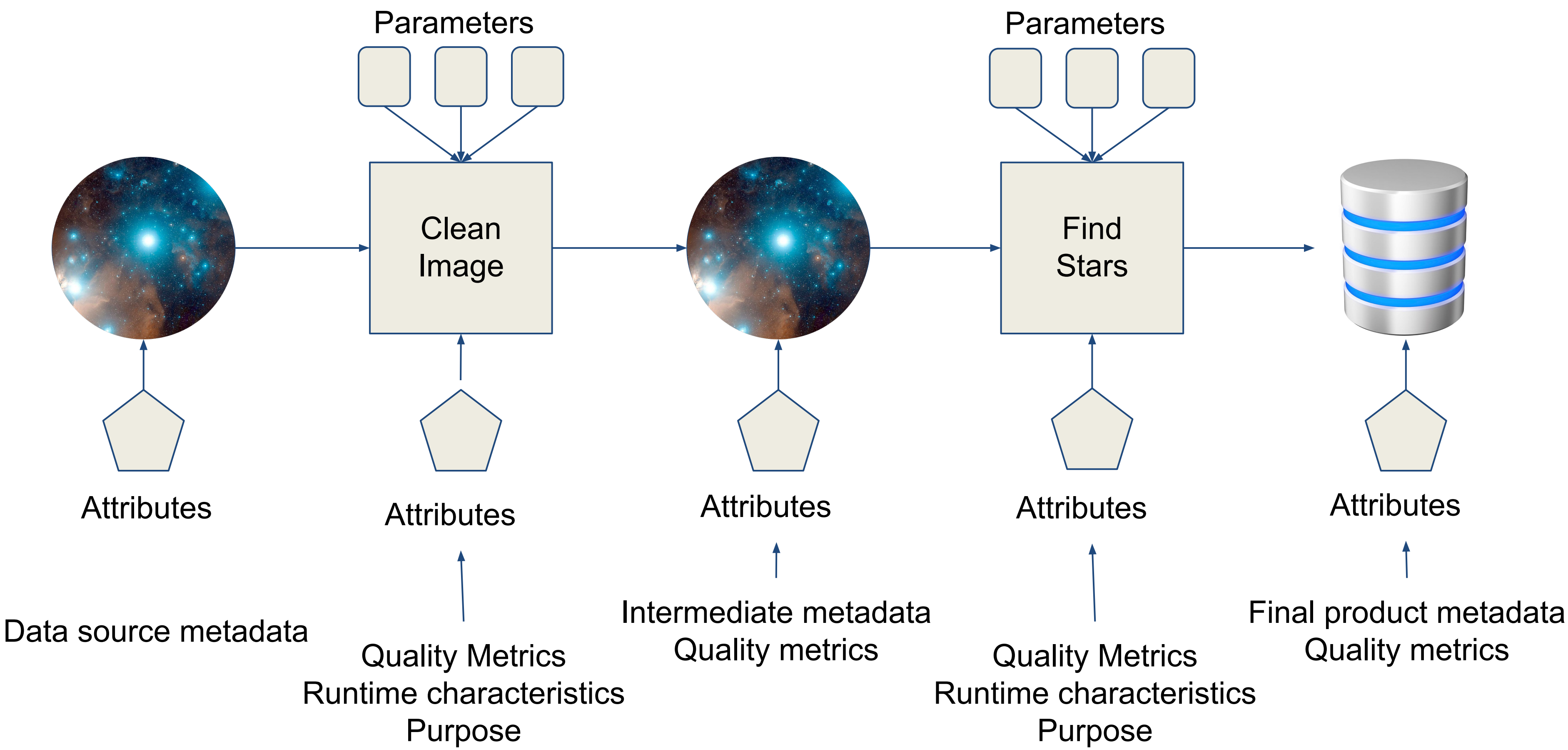
General purpose provenance model

Entities - things (e.g. data, telescopes)

Activities - processes (e.g. algorithms)

Agent - responsible for an activity
(e.g. people, institutions)




Parameters
Parameters
Clean Image
Find Stars
Attributes
Attributes
Attributes
Attributes
Attributes
Data source metadata
Quality Metrics
Runtime characteristics
Purpose
Intermediate metadata
Quality metrics
Quality Metrics
Runtime characteristics
Purpose
Final product metadata
Quality metrics

Motivation

Trust and Reproducibility

a) Re-execution of an existing pipeline
b) Impact of errors

Prediction

a) Prediction of pipeline performance
b) Storage vs recomputation analysis

Recommendation

a) Finding similar pipelines
b) Recommendation of pipeline components
c) Recommendation of parameter configurations

Anomaly Detection

a) Determining differences between pipeline executions
b) Anomaly detection in repeated pipeline runs

		Group A		Group B		Group C			Group D	
		UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7	UC 8	UC 9
Identifiers	Pipelines	○	○	◐	○	○	○	◐	●	●
	Pipeline Runs	●	○	◐	○	○	○	○	●	●
	Components	●	●	●	●	●	●	●	●	◐
	Data Sources	●	●	○	●	◐	◐	◐	●	○
	Data Products	○	●	○	○	○	○	○	○	○
	Intermediate Results	○	○	○	○	○	○	○	○	○
Attributes	Parameters	●	●	◐	●	◐	●	◐	●	●
	Runtime Environment	◐	○	◐	●	○	○	○	●	●
	Resource Consumption	○	○	●$_a$	●	○	◐	○	○	●
	Data Source Metadata	○	◐	◐	○	◐	◐	◐	○	○
	Data Product Metadata	○	○	○	○	○	○	○	○	●
	Interm. Result Metadata	○	○	○	●	◐	◐	○	○	◐
	Quality Metrics	○	○	●$_a$	○	◐	◐	○	○	○
Connections	Data Flow	●	●	◐	●	●	◐	●	●	○
	Pipeline Version	○	○	◐	○	○	○	◐	●	●
Prov. Records	same Pipeline	●	○	○	○	○	○	○	○	○
	other Pipelines	○	●	●	●	●	●	●	●	●
Other UCA	Miles et. al. [9]	●	●	●	○	○	●	○	●	●
	Bowers et al. [3]	●	○	○	○	○	○	○	○	●
	Ram et al. [12]	●	●	○	○	○	○	○	○	○

Table 1: Summary of requirements per use case. ● … mandatory; ◐ … optional; ○ … not required. For entries with the same subscript, at least one requirement has to be fulfilled. The "Other UCA" rows denote whether each use case was included in other analyses (further discussion in Section 4). ● … included; ○ … not included.

VAMPIRA Provenance Granularity

Function level provenance

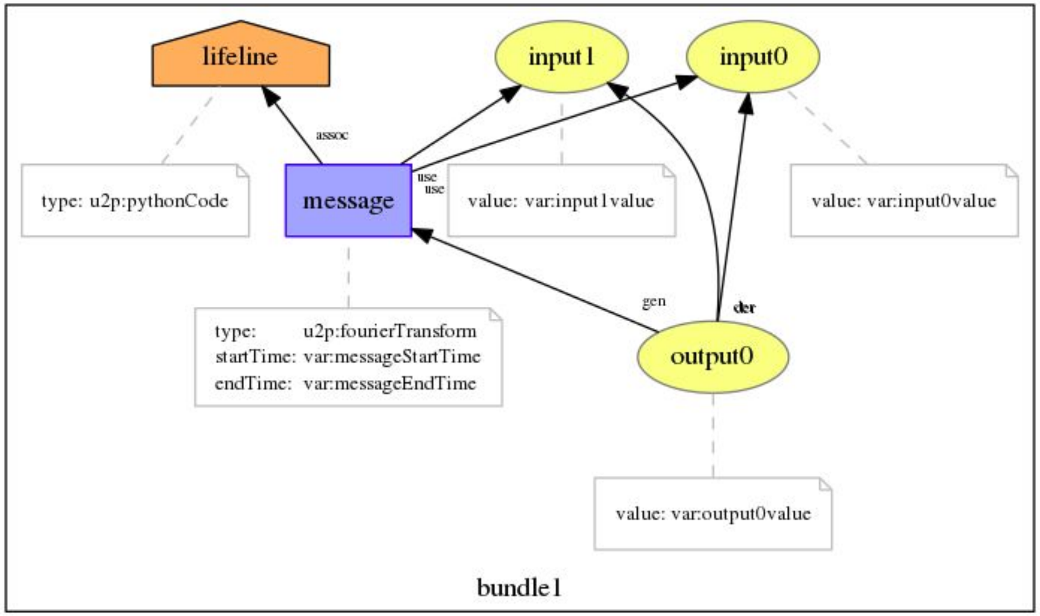
Includes:

- Inputs/Outputs
- Names/versions
- Run time
- Memory Usage
- Size of inputs
- User metadata

Applied to:

- Functions
- Methods
- Classes
- Modules

```python
def fourierTransform(imageName,std=2.5):
    hdu_list = fits.open(imageName)
    data = hdu_list[0].data
    kernel = Gaussian2DKernel(stddev=std)
    fftData = convolve_fft(data,kernel)
    hdu_list[0].data = fftData
    hdu_list.writeto(fourierImageName,overwrite=True)
    return fourierImageName
```



Collecting the Provenance

Vampira_run.py <your_python_scipt.py>

Output

PROV standard provenance describing the data processing

Implementations for provenance storage:

- Files (JSON, PROVN, etc.)
- MongoDB
- RDF
- Neo4j

Interfaceable via the VAMPIRA UI

— Function Overview

Function ↑ ↓	Occurrence ↑ ↓	Input
fourierTransform	2	imageName, std
generateHistogram	4	table, histogramName
sextractor	4	imageName

DLR

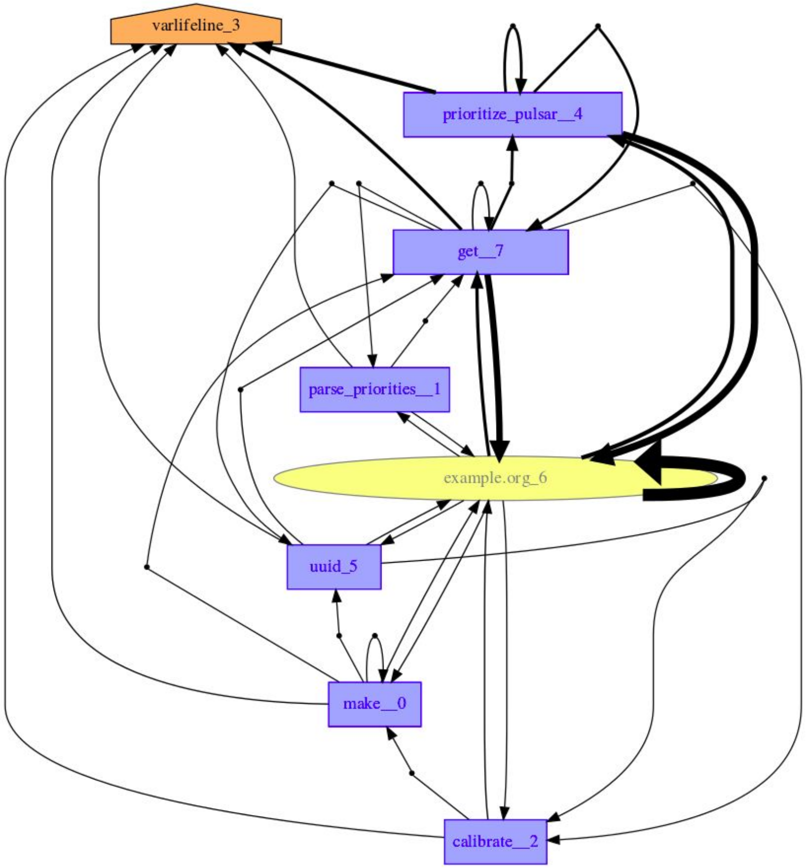
Demystifying the Black Box

Astronomical workflow (especially machine learning) can make lots of independent decisions

Path taken to the result may not always be clear

Detailed documentation of these pipelines helps with understanding through:

- Diagrams of the path taken
- Analysis of resources per function
- Identification of anomalies

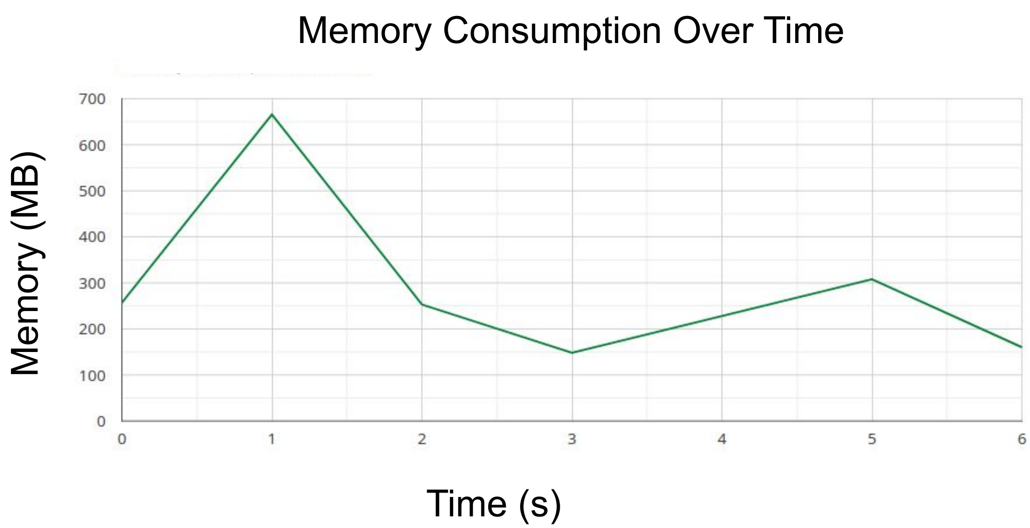


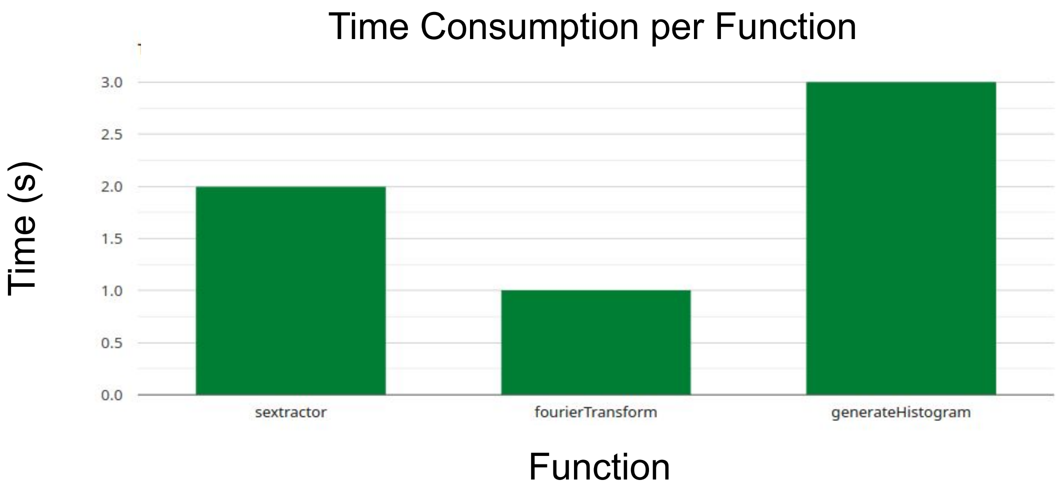
Prediction

Summary includes:
- Memory consumption
- Time spent on each function

Storage vs recomputation
- Should any intermediate data product be stored or remade?

Prediction of Pipeline Performance
- How will new pipelines perform?
- Is it worth recomputation of old data?

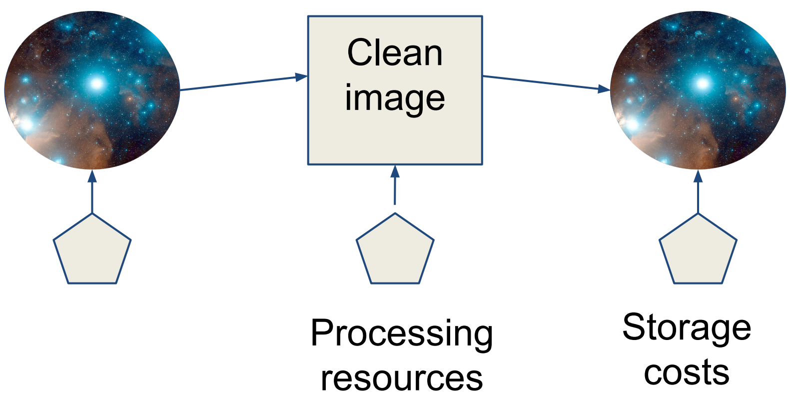


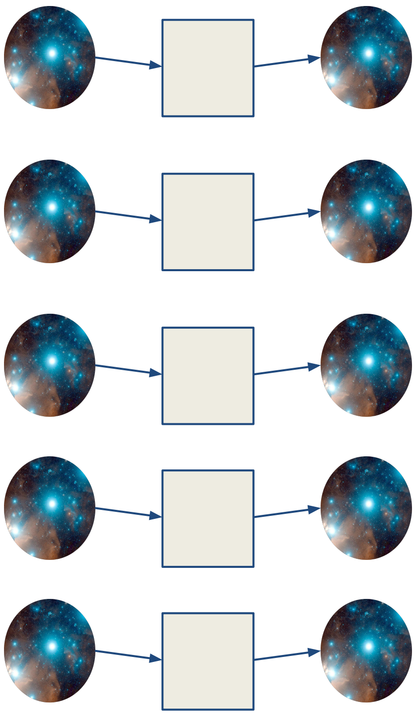


Example - Storage vs Recomputation

Determine the cost to store and to recompute:

- Memory of the object
- Computational resources to generate
- Number of times accessed





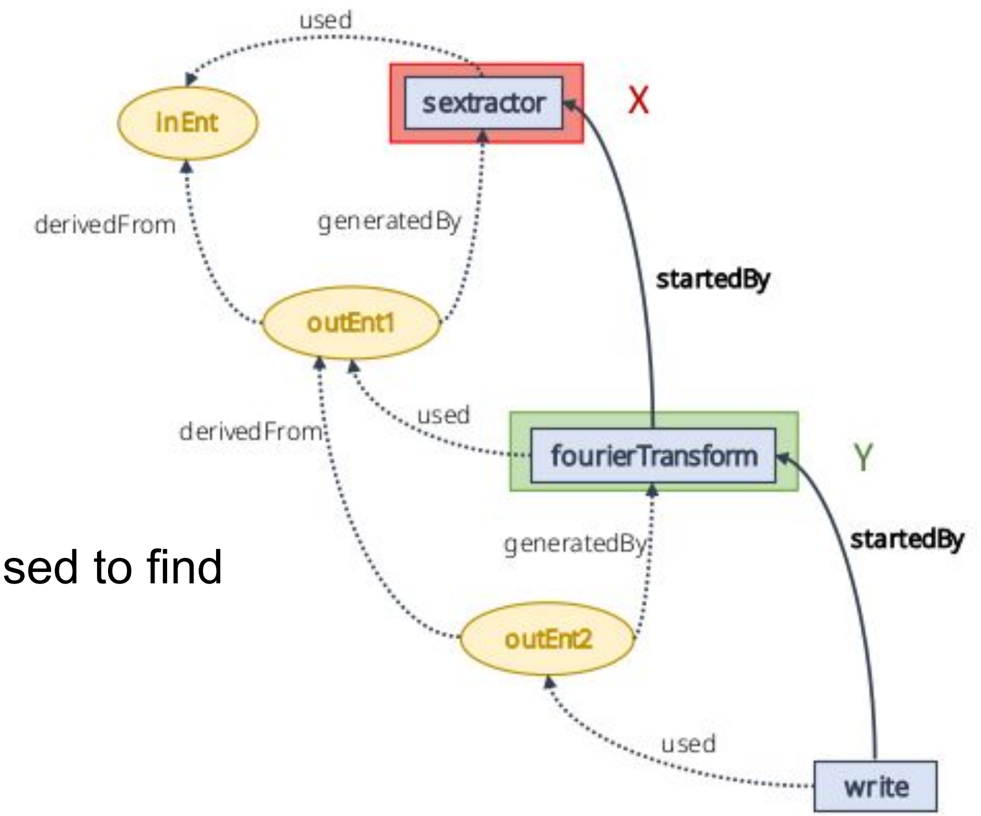
Recommendation

Recommending:

- Relevant pipelines
- Alternative function/modules
- Suitable parameters

Sequential association rule mining used to find example subsequent functions

Pipeline comparisons

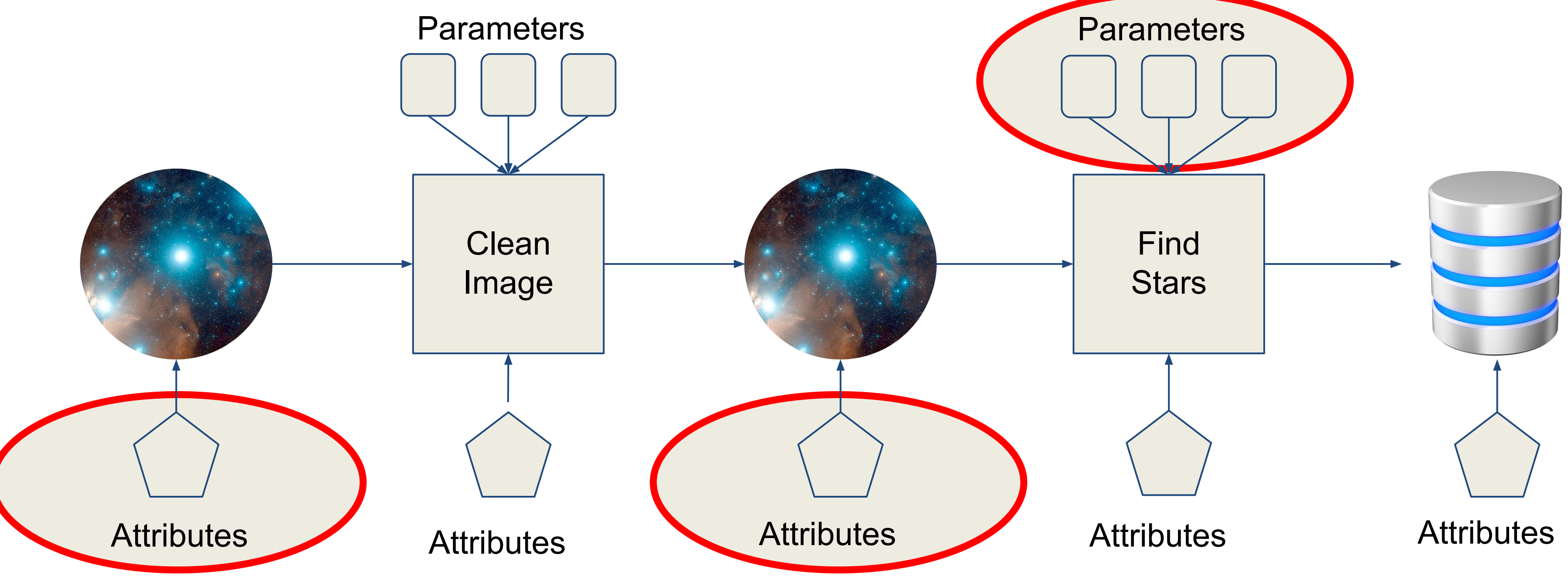


Functions by name

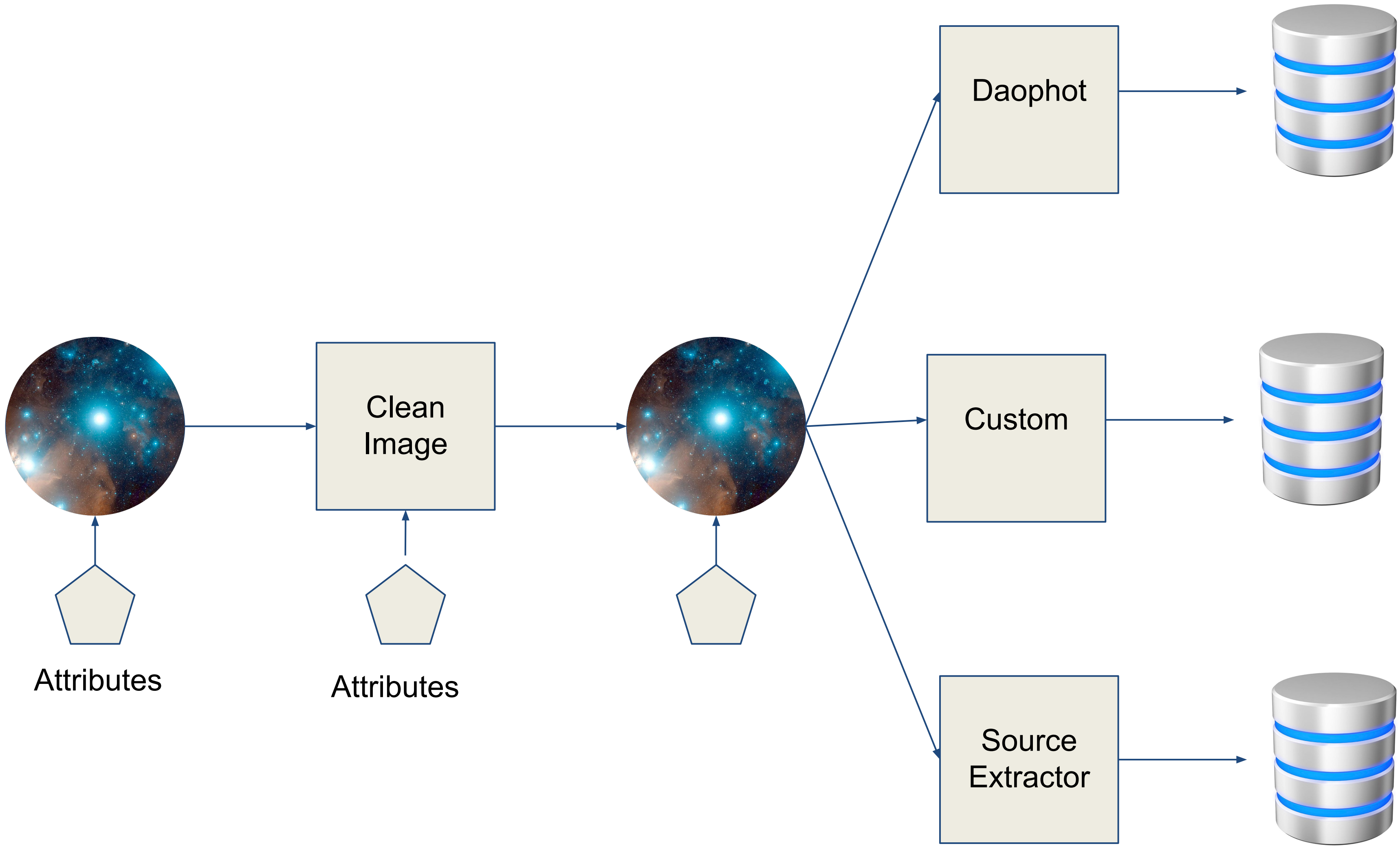
Stackers_LSST ↑ ↓	Function ↑ ↓	PullLightCurves_LSST ↑ ↓
5 (20.0%)	nside2npix	3 (42.9%)
4 (16.0%)	npix2nside	3 (42.9%)
3 (12.0	nside2pixarea	

DLR

Recommendation of Parameters



Recommendation of Functions



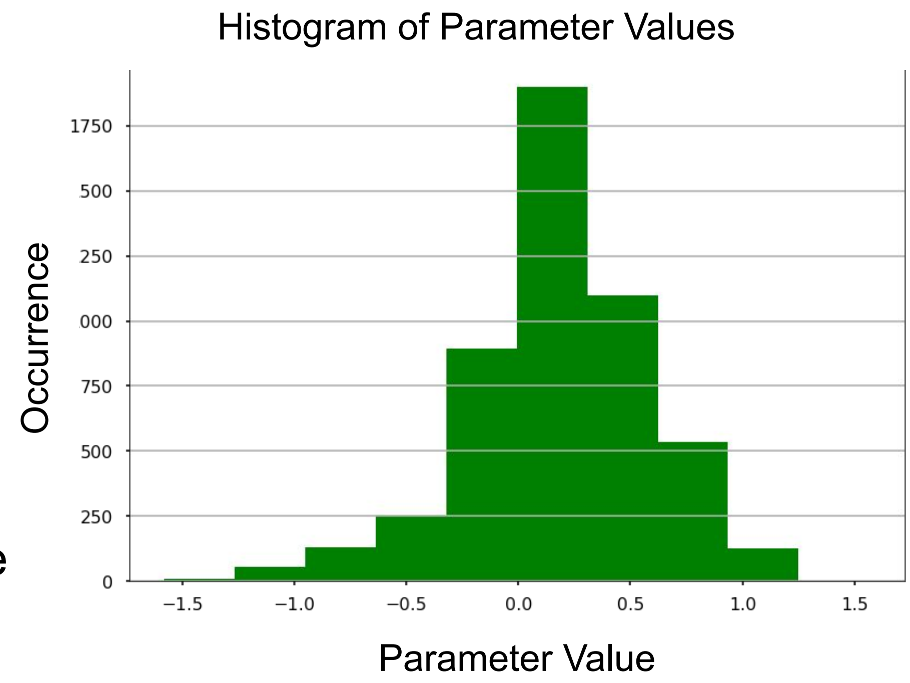
Anomaly Detection

Find results or parameters with unusual values

Determine the source of these anomalies

Execute a single pipeline multiple times

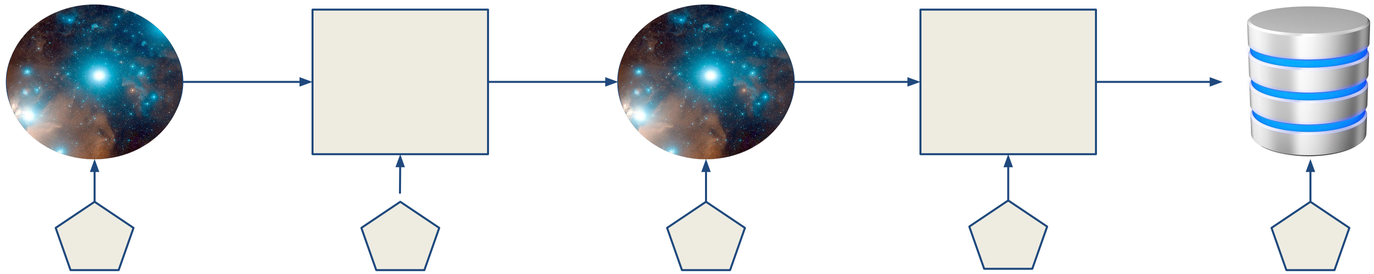
Then determine differences between these pipeline runs



swep_prov_py3 ↑ ↓	Function	swep_prov_py2 ↑ ↓
40.0%	generateHistogram	40.0%
40.0%	sextractor	40.0%
20.0%	fourierTransform	20.0%

DLR

Extra Science - Information on Unseen Objects



Combine the total observed objects with the quality:

- The completeness (unrecovered fraction)
- The accuracy (false positive rate)

Take into account:

- Image metadata
- Parameters of detection algorithm
- Version of detection algorithm
- Aggregate over all past pipelines

Determine the number of objects expected within the image that were not observed

Summary

VAMPIRA automatically generates provenance for python scripts

The output is PROV standard and function level

We also developed tools for the accessibility and dissemination of provenance

Provenance can be used to:

- Establish trust
- Predict performance
- Recommend components
- Detect anomalies
- Increase understanding

Contact us: Michael.Johnson@dlr.de