



International
Centre for
Radio
Astronomy
Research

The DALiuGE Ecosystem

A workflow editor and execution framework for
transforming existing code into scalable,
distributed workflows

Ryan Bunney, Professor Andreas Wicenec, Rodrigo Tobar, Nicholas
Pritchard, James Strauss, Moritz Wicenec



ICRAR is a joint venture between Curtin University and The University of Western Australia and receives support from the Western Australian and Australian Governments.



Outline

- Science Workflows
 - What
 - Why
 - How
- The software
 - EAGLE
 - Code 'palette generator'
 - DALiuGE
- Current Work
 - DINGO
 - WALLABY
 - CHILES
- Is DALiuGE right for you?
- Questions

Science Workflows

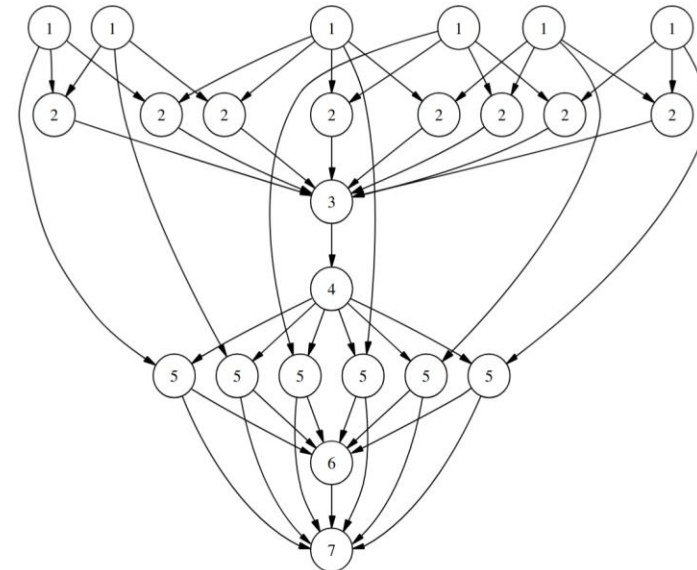
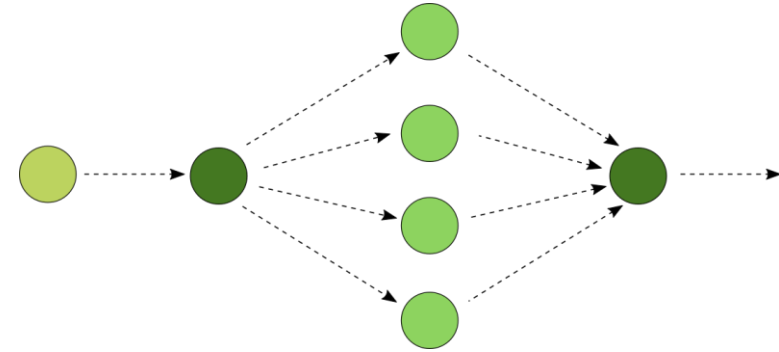
Science workflows

Workflow management systems

Existing systems

Science Workflows: What?

- Workflows are a common tool in HPC for grouping together interrelated computing jobs
- Jobs often have precedence constraints
- Typically use Directed Acyclic Graph (DAG) representation
 - Compute tasks are nodes
 - Data dependencies are edges



Montage workflow (Deelman *et al.*, 2005)



Science Workflows: Why?

Computational

- Scheduling & Resource Allocation are NP-Hard jobs
- DAG/stricter definitions can improve heuristic performance

Practical

- Separate researchers from optimisation and deployment
- Improve robustness
 - N people on N similar codes vs. N people on 1 shared workflow
- Improve reproducibility
 - Federated workflows allow for re-computation across multiple scientists/groups



Science Workflows Management Systems (SWFMS)



Apache Taverna
(Retired 2020)





DALiuGE

- Data-Activated Flow (Liu) Graph Engine
 - Deployment and execution of radio astronomy workflows
- Prototype workflow engine for the SKA Phase 1
- SC 2020 Gordon Bell Prize Finalist
- Demonstrated at scale:
 - run on 4,560 compute nodes
 - used 27,360 GPUs to generate 2.6 PB of data



What DALiuGE does differently

Feature	SFWMs	DALiuGE
Separation of concerns (Scientist vs. Developer)	<ul style="list-style-type: none">• Requires learning a DSLs• Time spent transitioning existing work to new tool	<ul style="list-style-type: none">• UI for drawing graphs• Re-use of existing Python/scripting code
Control flow	<ul style="list-style-type: none">• Fixed shell-like management of workflow execution	<ul style="list-style-type: none">• Support for conditional execution of the graph
Data management	<ul style="list-style-type: none">• Data is implicit (edge in DAG)	<ul style="list-style-type: none">• Data as first-class citizen• Controls its own runtime execution
Data types	<ul style="list-style-type: none">• Reliance on files for intermediary data storage	<ul style="list-style-type: none">• In-memory support between different types of applications (Python -> Bash -> Docker)



EAGLE

Logical Graph Editor

Repository and Palette Management



Team



Andreas Wicenec
Project Lead



James Strauss
Software Engineer

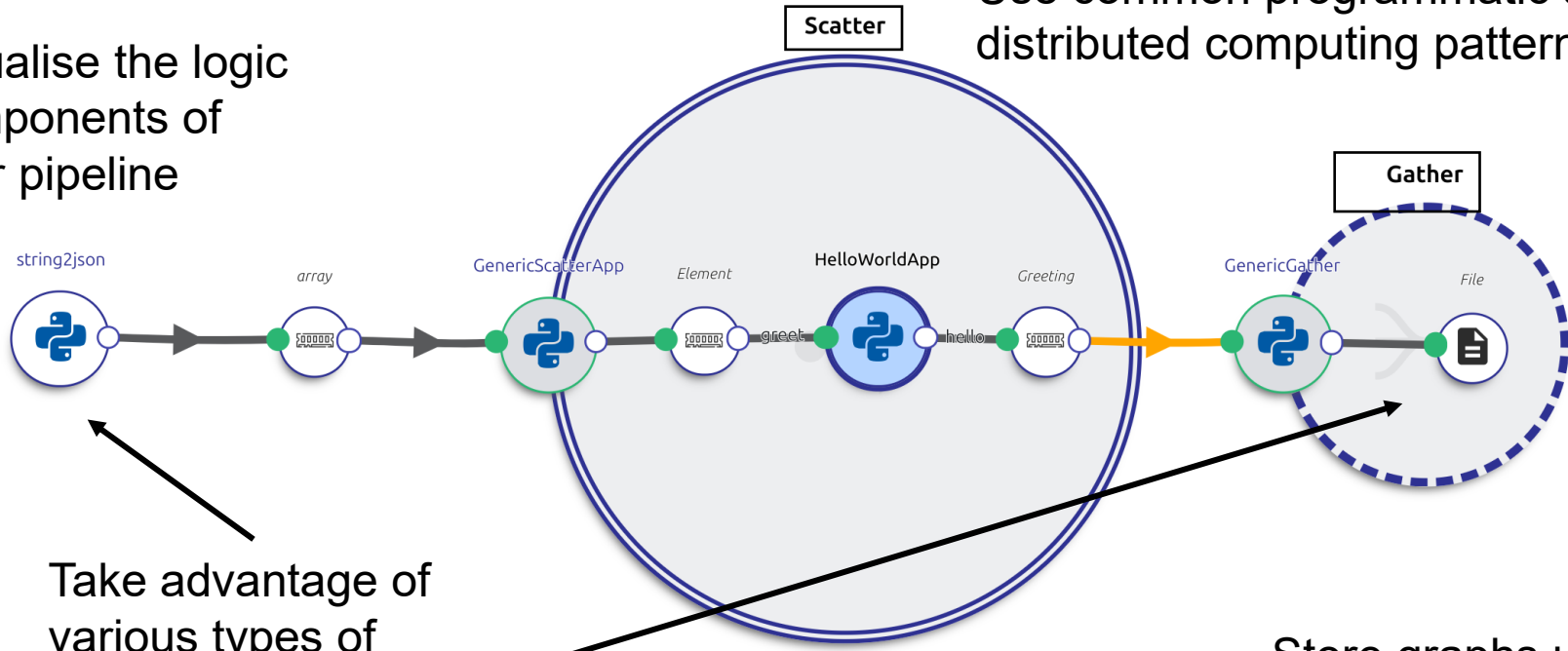


Moritz Wicenec
Software Engineer



Visualise the logic components of your pipeline

Use common programmatic and distributed computing patterns



Take advantage of various types of Application and Data components

Modify application parameters in the UI

Store graphs using VCS (GitHub/GitLab)

HelloWorldApp

Parent: Scatter

Port Multiplicity: 3

Category: PythonApp

Repositories Hierarchy Translation

- ICRAR/EAGLE-graph-repo (master)
 - EAGLE-1194
 - EAGLE-1302
 - EAGLE-856
 - SDP Pipelines
 - casa
 - chiles
 - dalluge
 - dalluge-translator
 - dingo
 - dummy
 - examples
 - local
 - FileCollector.palette
 - FITSplit.palette
 - ig.palette
 - ArrayIngest_Demo.graph
 - ArrayLoop.graph
 - astropy_SkyCoords_Constellations
 - astropyArrayLoop.graph
 - branchDemo.graph
 - compileExample.graph
 - ddExample.graph
 - ddExample_mixedoorts.graph
 - earth_location.graph
 - FileCollector.graph
 - FileCollectorScatter.graph
 - fits_load_save.graph
 - FITSplit.graph
 - HelloWorld-simple.graph
 - HelloWorld-Universe.graph
 - HelloWorld-Universe_testing_OA.g
 - logTest.graph
 - keys_from_paramset.graph
 - memory_with_data.graph
 - NGASLogProcess.graph
 - NGASLogProcess_unzip.graph
 - NGASLogProcessGather.graph
 - parallel_sleep.graph
 - parallelPI.graph
 - parameterSet.graph
 - pyfunc_glob_test.graph
 - pyfunc_test.graph
 - rascil_blockvis.graph
 - S3_download.graph
 - S3_upload.graph
 - S3_upload_GCS.graph
 - SkvCoord.graph

Node Fields Table: HelloWorldApp

Right Click on Objects in the graph to see more options. [D] duplicate selected objects. [Delete, Backspace] delete selected objects. [T] show fields table. [Alt+Y, Ctrl+T] display all graph configs.

Selection	Attribute Name	Value	Content	Default Value	Type	Flags	Parameter Type	Use As	Actions
	persist	<input type="checkbox"/>		<input type="checkbox"/>	Boolean	🔒	ComponentParameter	NoPort	🗑️
	streaming	<input type="checkbox"/>		<input type="checkbox"/>	Boolean	🔒	ComponentParameter	NoPort	🗑️
	dropclass	dlg.apps.simple>HelloWorldApp			String	🔒	ComponentParameter	NoPort	🗑️
	execution_time	5		0	Float	🔒	ConstraintParameter	NoPort	🗑️
	num_cpus	1		0	Integer	🔒	ConstraintParameter	NoPort	🗑️
	group_start	<input type="checkbox"/>		<input type="checkbox"/>	Boolean	🔒	ComponentParameter	NoPort	🗑️
	input_error_threshold	0		0	Integer	🔒	ComponentParameter	NoPort	🗑️

Palette Generator

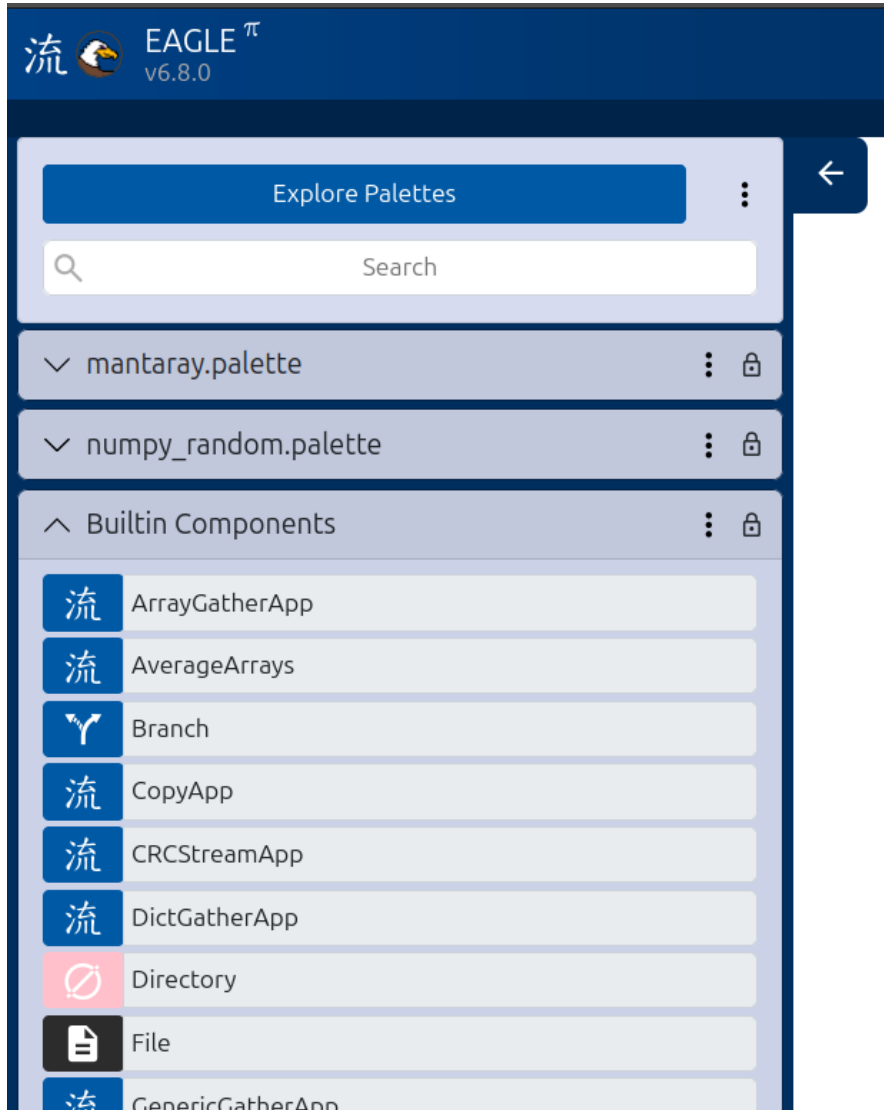
Point to existing code

Use palette in EAGLE

Run palette with DALiUGE



dlg_paletteGen



- EAGLE graphs draw components from their palettes
- There are built-in palettes that support generic applications
- dlg_paletteGen tool generate palettes from arbitrary Python modules in a fully automatic way
- The tool can extract component information from plain Python modules, but also C extensions and PyBind11 generated modules



dlg_paletteGen

Selection	Content			
Attribute Name	Value	Default Value	Description	Type
scale	1	1	The scale parameter, $\beta = 1/\lambda$. Must be	float
size			Output shape. If the given shape is, e.g., "(m, n, k)", then	int or tuple
func_name	numpy.random.exponential	numpy.random.exponential	Complete import path of function or just a function name which is also used in func_code below.	String

- random.chisquare
- random.choice
- random.default_rng
- random.dirichlet
- random.exponential
- random.f
- random.gamma
- random.geometric
- random.get_state
- random.gumbel

PyFuncApp : random.exponential

```

exponential(scale=1.0, size=None)
Draw samples from an exponential distribution.

Its probability density function is

.. math:: f(x; \frac{1}{\beta}) = \frac{1}{\beta} \exp(-\frac{x}{\beta})
for "x > 0" and 0 elsewhere. :math:\beta is the scale parameter
which is the inverse of the rate parameter :math:\lambda = 1/\beta
The rate parameter is an alternative, widely used parameterization
of the exponential distribution [3]_.

The exponential distribution is a continuous analog of the
geometric distribution. It describes many common
the size of raindrops measured over many rainstorms
between page requests to Wikipedia [2]_.

.. note::
    New code should use the ~numpy.random.Generator

```

- All in-line documentation (docstrings) and parameter descriptions are extracted

- (New feature!) No docstrings? Gemini will figure it out for you!

- mwa_client.get_job_list
- mwa_client.download_func
- mwa_client.enqueue_all_ready_to_download_jobs
- mwa_client.get_job_summary
- mwa_client.get_jobs_status
- mwa_client.get_pretty_version_string
- mwa_client.get_status_message
- mwa_client.init

PyFuncApp : mwa_client.get_job_list

This docstring has been generated by GenAI!!

```

python
Retrieves a list of jobs from the server.

Args:
    session: The session object used to communicate with the server

Returns:
    list: A list of job objects, or an empty list if no jobs are found

Raises:
    Exception: If an error occurs while retrieving the job list from the server.

```

DALiuGE

Scheduling and Partitioning

Runtime Management of Applications

Remote Deployment and Execution



Team (over the years)



Andreas Wicenec



Chen Wu



Rodrigo Tobar



Kevin Vinsen



Dave Pallot



Nicholas Pritchard

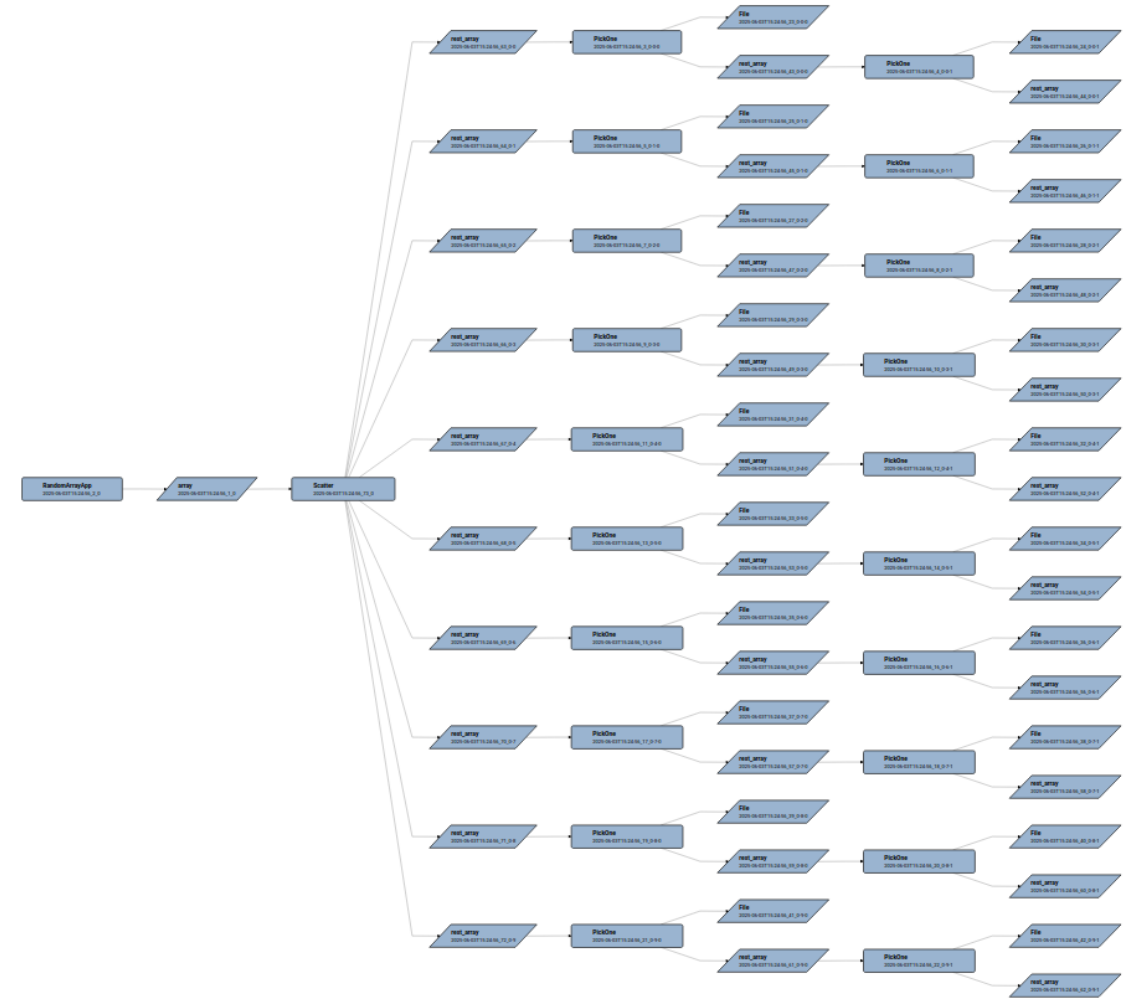
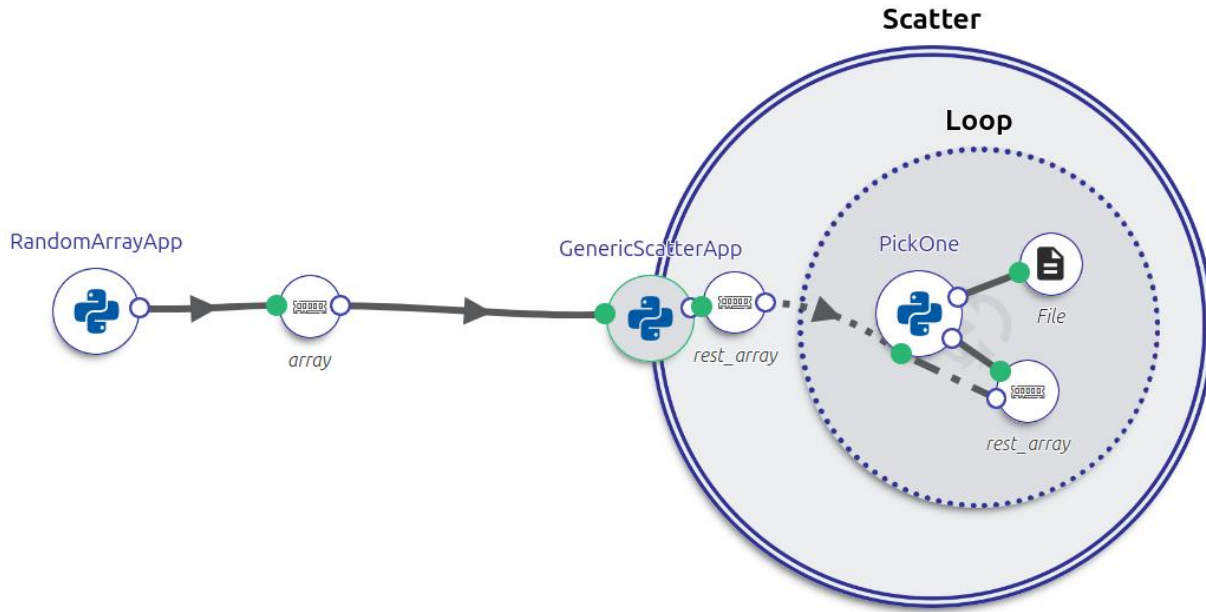


Callan Gray



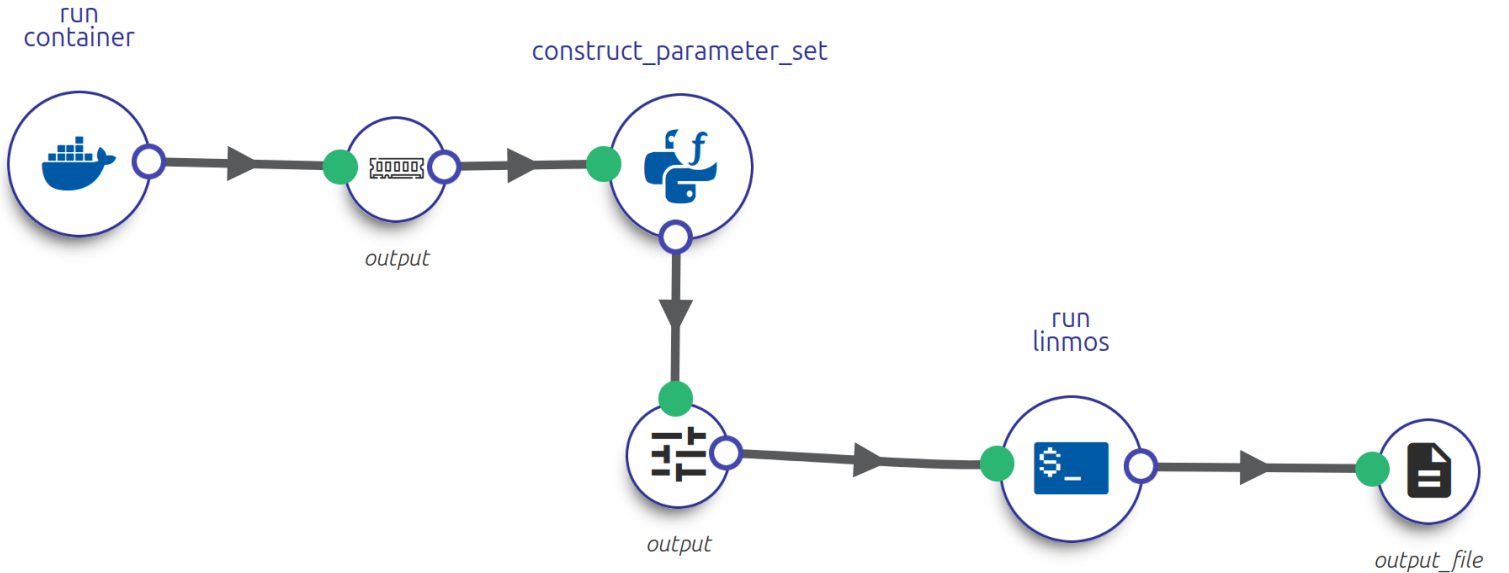
Julian Carravick

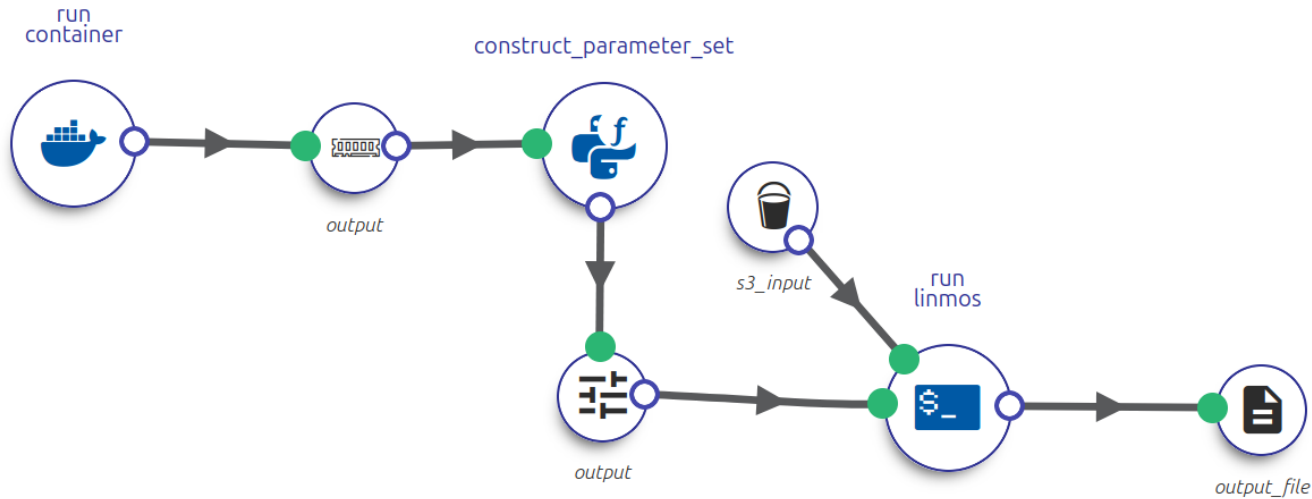
Translating, Partitioning, Scheduling



Built-in support for:

- BashApp
- DockerApp
- **PyFuncApp**
- DALiuGEApps, e.g.
 - CopyApp
 - ScpApp





Supported types include

- Memory
- File
- ParameterSet
- NGAS
- S3

- Data manages itself at runtime and triggers consumers
- In-memory data support across apps



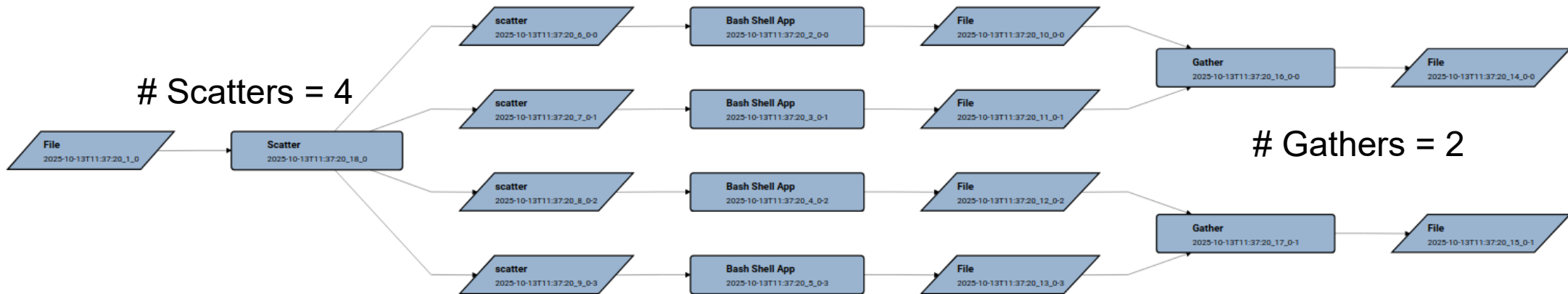
Constructs: Loop example



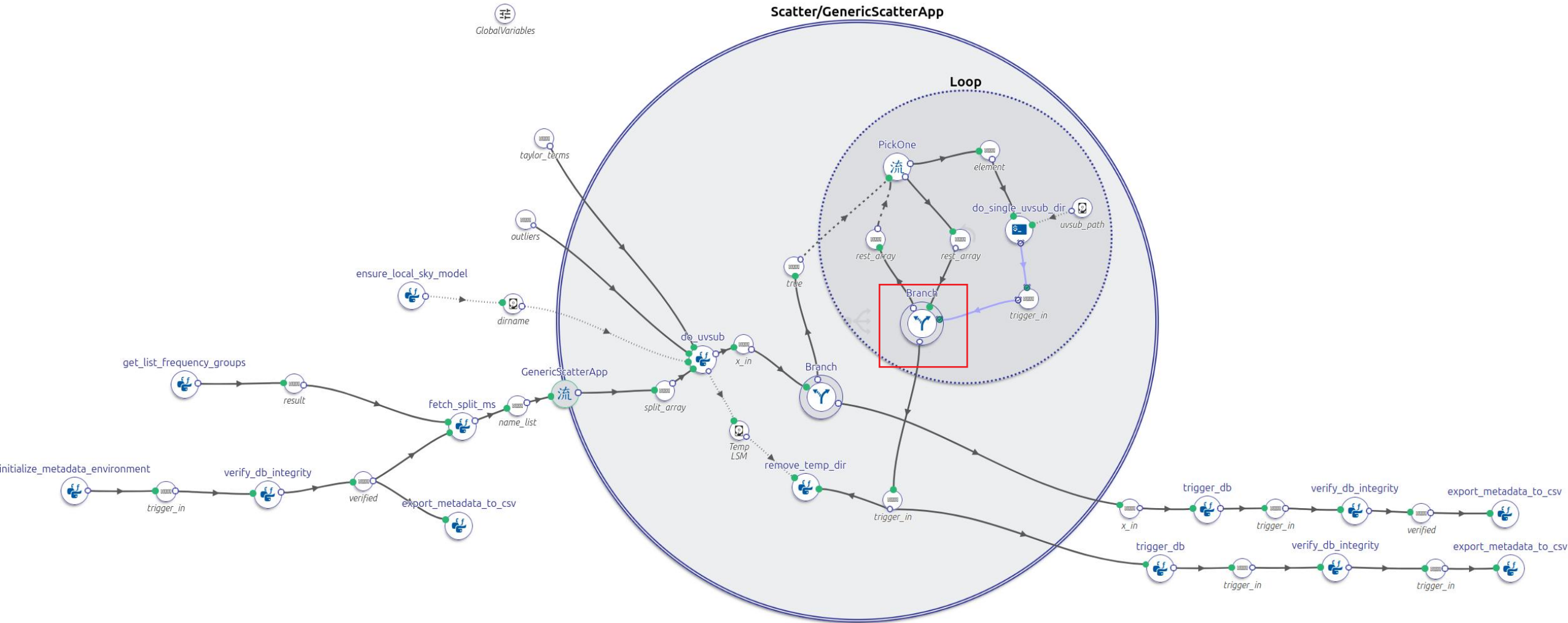
initial_heatmap_plot.png



Constructs: Scatter & Gather



Constructs: Branches example



- Iterate through dynamic number of files/data
- Apply convergence criteria to workflow execution



Reproducibility

- DALiuGE can grade a workflow on 7 Reproducibility 'tests'
 - Rerun
 - Repeat
 - Recompute
 - Reproduce
 - Scientific replication
 - Computational replication
 - Total Replication
- Implemented using BlockDAG and MerkleTrees to implement run hashes
- Further reading: *Pritchard, N. J., & Wicenec, A. (2024). Formal Definition and Implementation of Reproducibility Tenets for Computational Workflows*

Putting it all together

Explore Palettes

Search

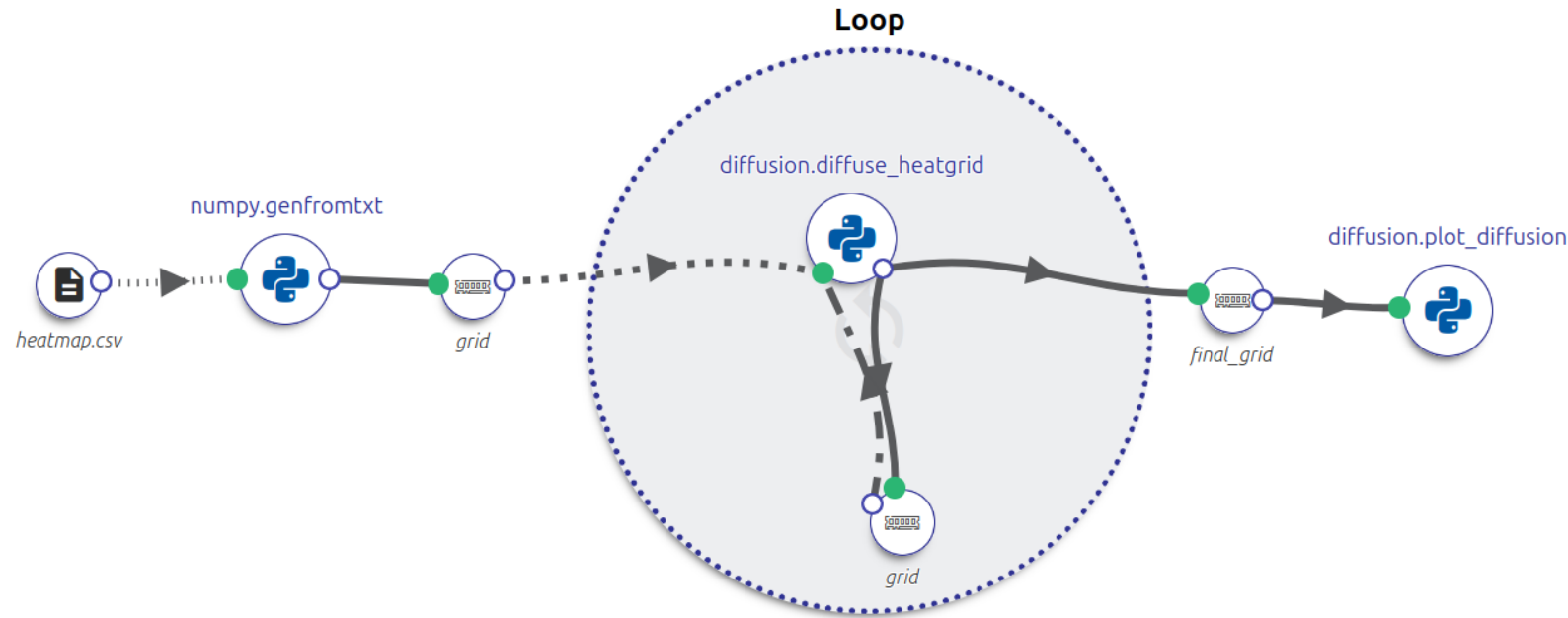
diffusion.palette

- diffusion.diffuse_heatgrid
- diffusion.plot_diffusion
- diffusion.animate_diffusion
- diffusion.diffuse_heatgrid_with_timesteps

heatmap.palette

- generate_heatmap.random_mean_from_range
- generate_heatmap.random_covariance_from_range
- generate_heatmap.generate_sources
- generate_heatmap.generate_heatmap
- generate_heatmap.heatmap_to_csv

Run `dlg_paletteGen` on local code & create a palette



Import palette into EAGLE

Build & run code from palette through DALiUGe

Recent & current work

DINGO

Wallaby Hires

CHILES

Yours?



Project Scientists & Engineers



Omar Anwar

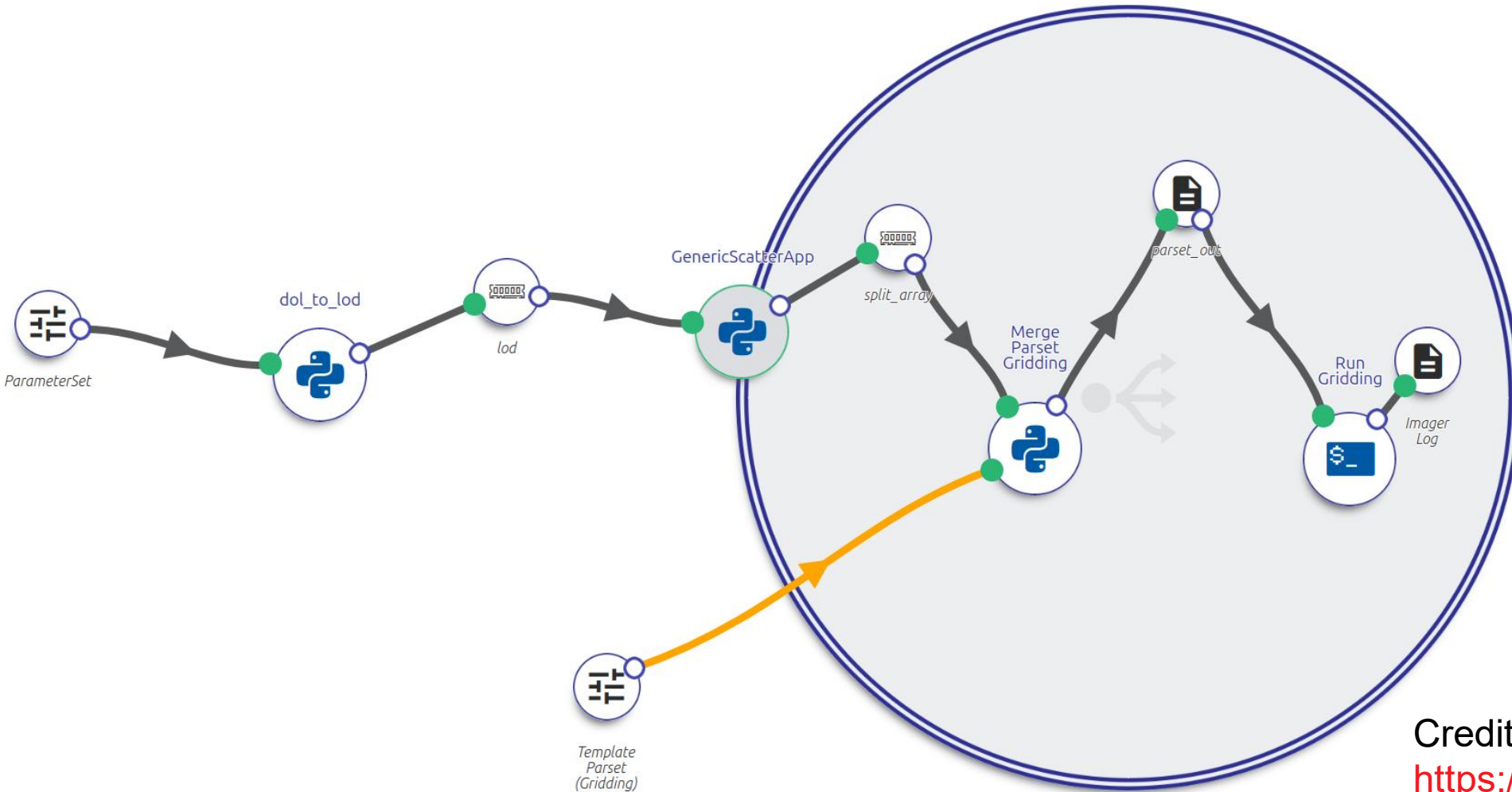


Alex Williamson



Gayatri Aniruddha

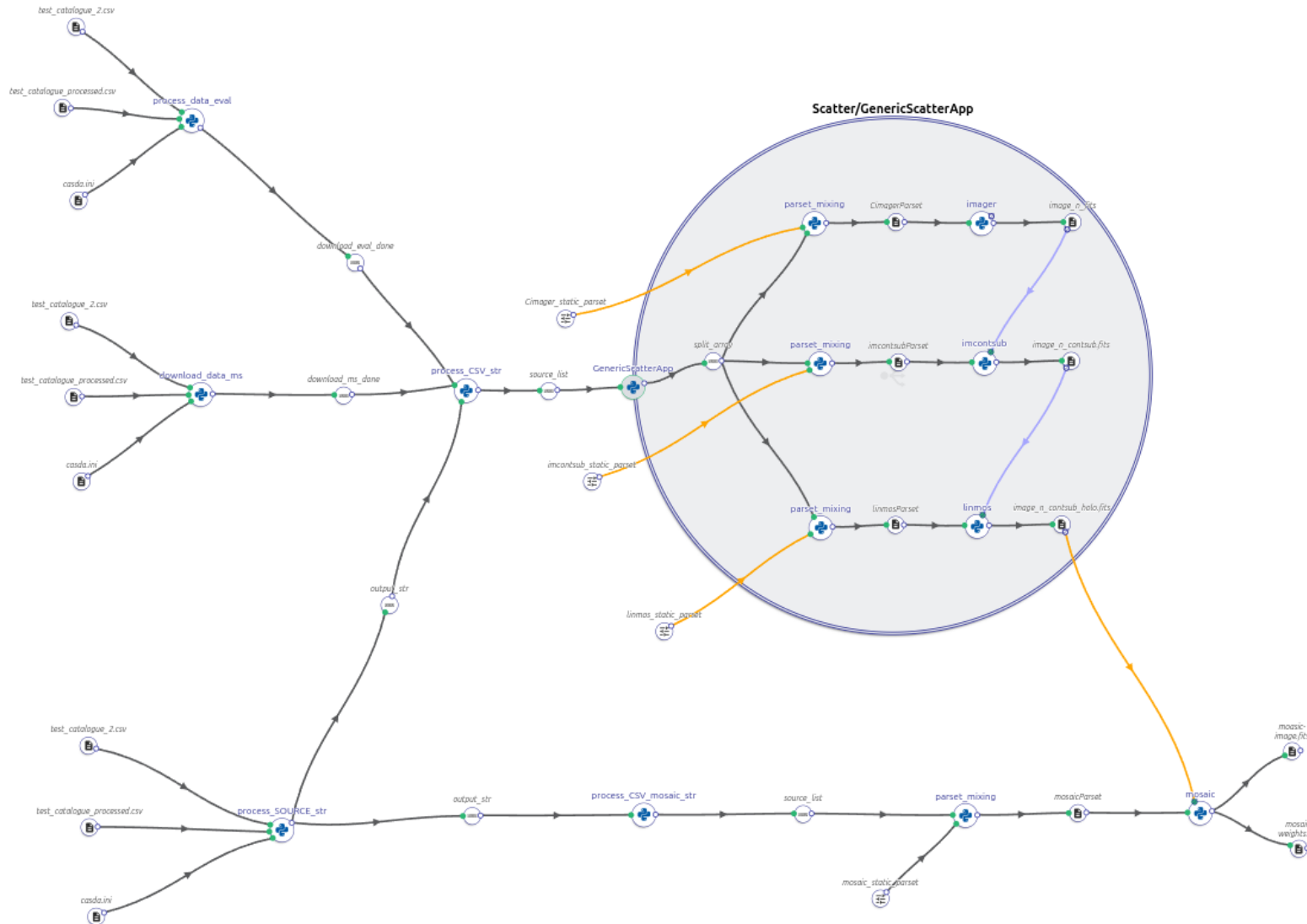
Scatter/
GenericScatterApp



Credit: Dr. Alex Williamson
<https://github.com/ICRAR/dingo>

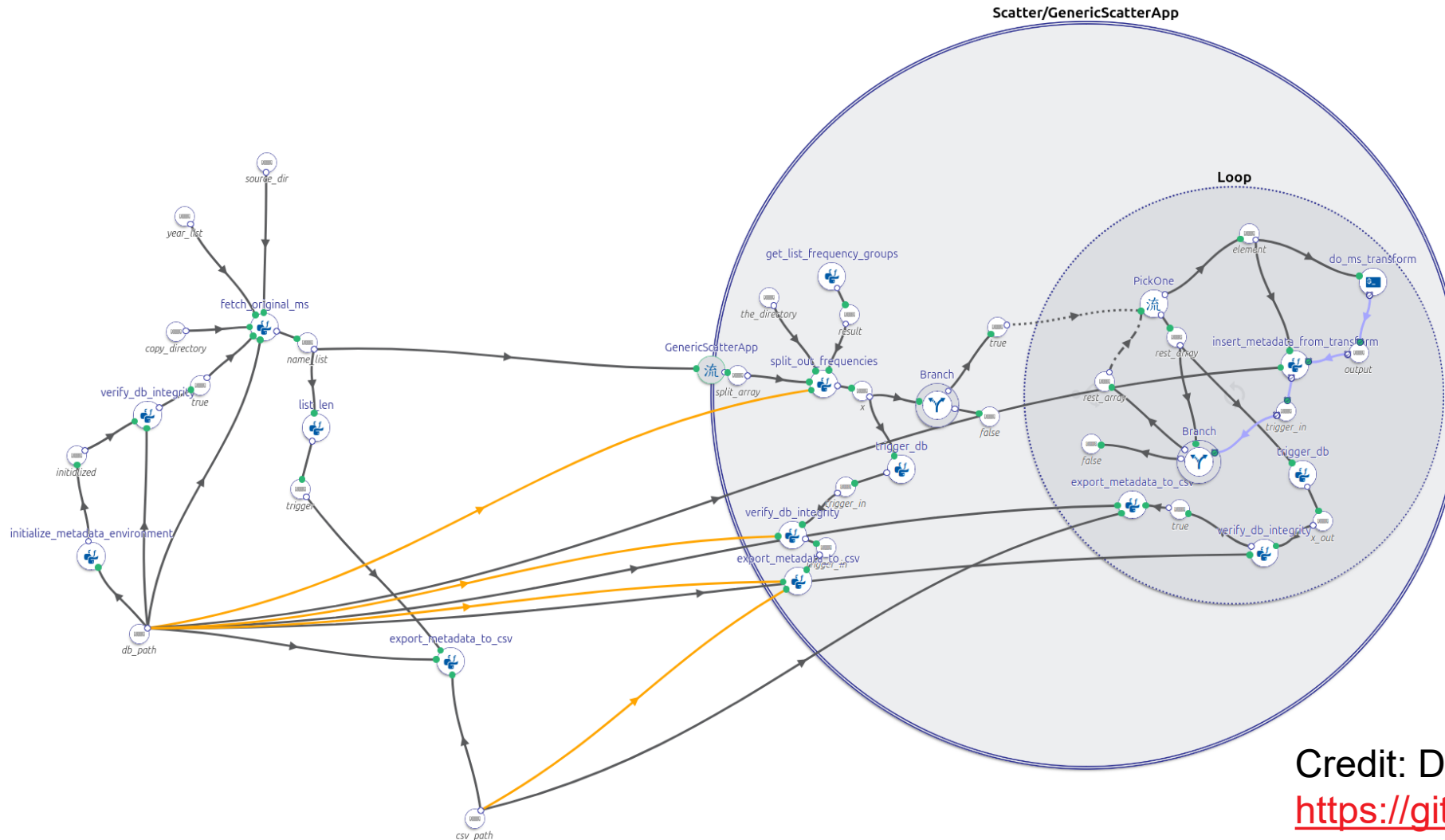
Dodson et al, 2025

Deployment: Setonix



Credit: Gayatri Aniruddha <https://github.com/ICRAR/wallaby-hires>

Deployment: Setonix
(In-progress!)



Credit: Dr. Omar Anwar
<https://github.com/ICRAR/chiles-daliuge>



The Pitch

- We would love to engage with non-Astronomy researchers and practitioners
- Access to in-house developers
- Quick(?) turn-around on bugs and features
- Let you focus on research rather than code

流 DALiuGE

EAGLE π



Questions?



ICRAR is a joint venture between Curtin University and The University of Western Australia and receives support from the Western Australian and Australian Governments.