Facilitating Scientific Reproducibility and Interoperability through CWL Integration in the Dirac Middleware

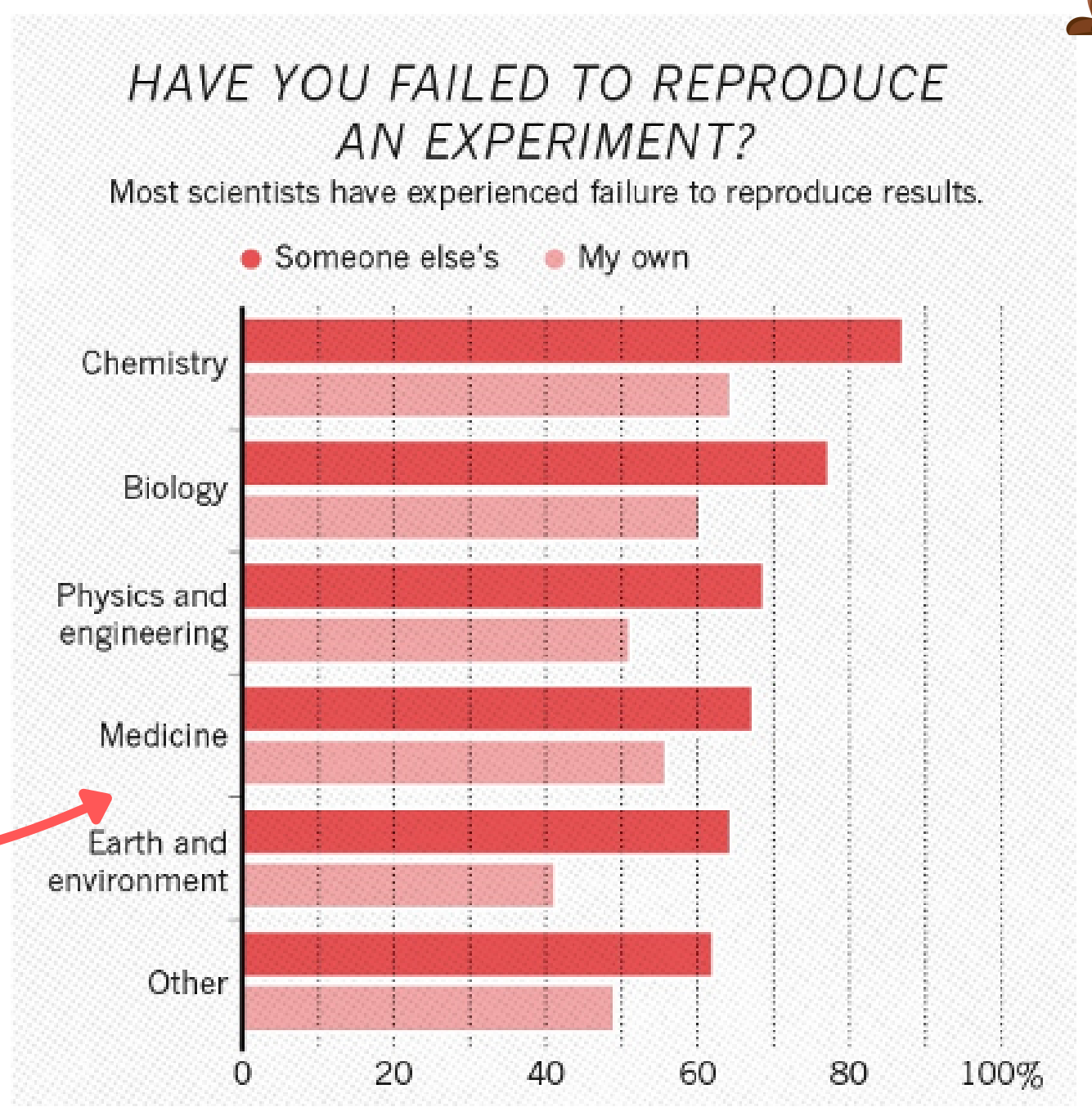
Alexandre Boyer[1], Natthan Pigoux[2], Luisa Arrabito[2]

1. CERN, EP Department, Geneva, Switzerland

2. Laboratoire Univers et Particules de Montpellier, CNRS/IN2P3, University of Montpellier, France

Reproducibility Crisis

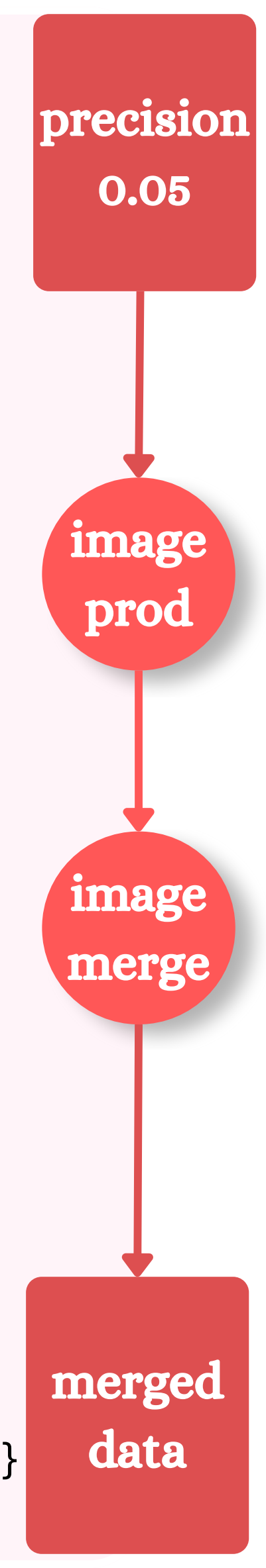
- Reproducibility is fundamental to scientific research, ensuring that results can be independently validated and generalized.
- Computers enabled complex calculations but also introduced issues, often taken for granted (e.g., software and hardware variations).
- A Nature survey (Baker, 2016) shows a widespread reproducibility crisis across scientific fields.
- Inconsistencies in software environments and computational workflows are major contributors (Antunes et al., 2024).



Writing reproducible workflows with CWL

- Computational Workflows automate complex data processes but face challenges like vendor lock-in and poor portability.
- The Common Workflow Language (CWL) (Crusoe et al., 2022) provides a standardized and vendor-neutral approach to defining workflows.
- CWL is widely adopted across diverse fields, with a growing number of workflow management systems supporting it.

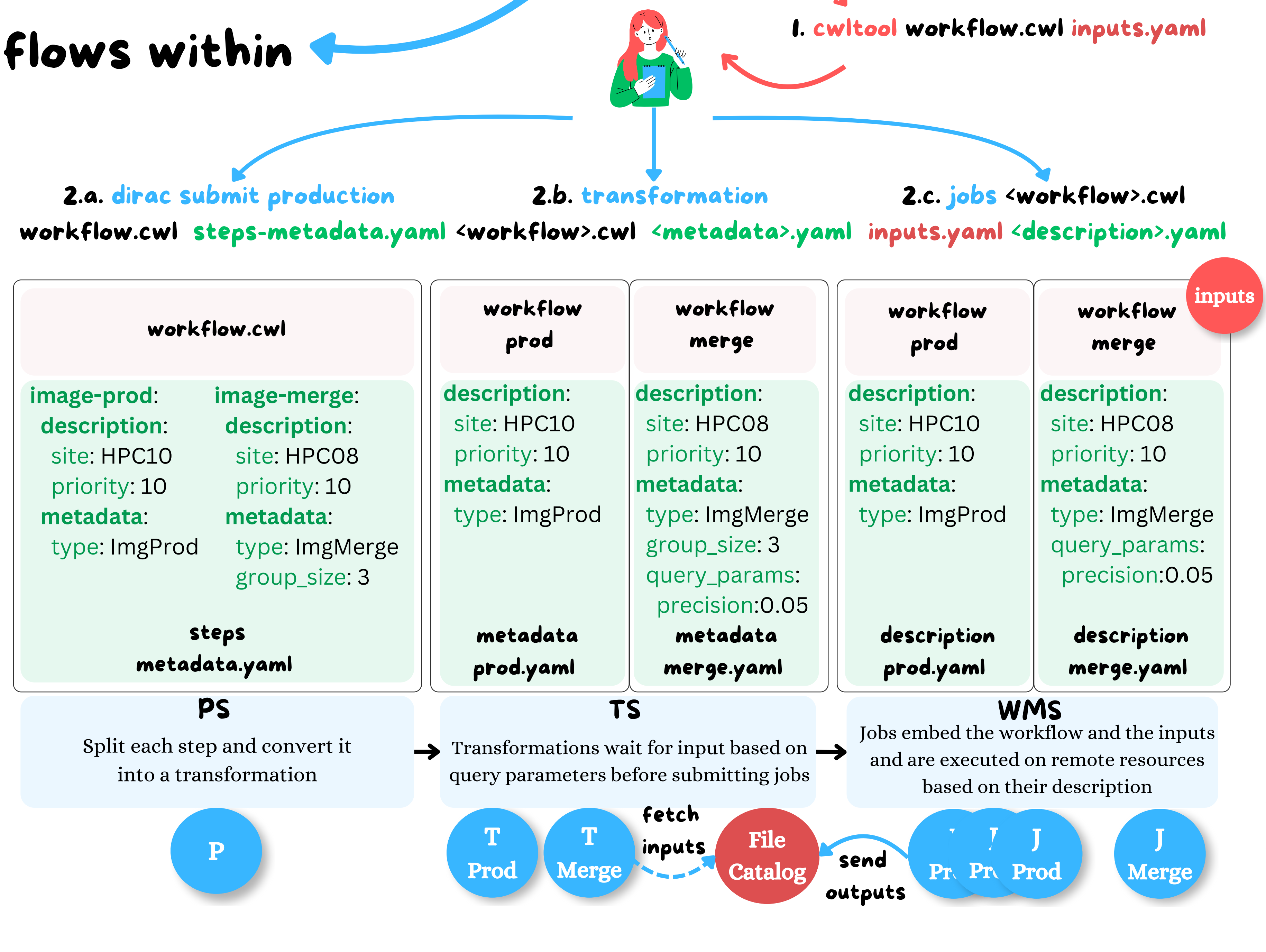
```
cwlVersion: v1.2
class: Workflow

inputs: {precision: {type: float, default: 0.05}}
outputs: {merged-data: {type: File, outputSource: image-merge/result}
requirements: {ResourceRequirement: {coresMin: 4, ramMin: 2048}}

steps:
 image-prod:
  in: {precision: precision}
  out: [data]
  run:
   class: CommandLineTool
   inputs: {precision: {type: float, inputBinding: {prefix: --precision}}}
   outputs: {data: {type: File[], outputBinding: {glob: "data*.txt"}}}
   baseCommand: [run-mandelbrot]

 image-merge:
  in: {data: image-prod/data}
  out: [result]
  run:
   class: CommandLineTool
   inputs: {data: {type: File[]}}
   outputs: {result: {type: File, outputBinding: {glob: "data-merged.txt"}}}
   baseCommand: [merge-mandelbrot]
```

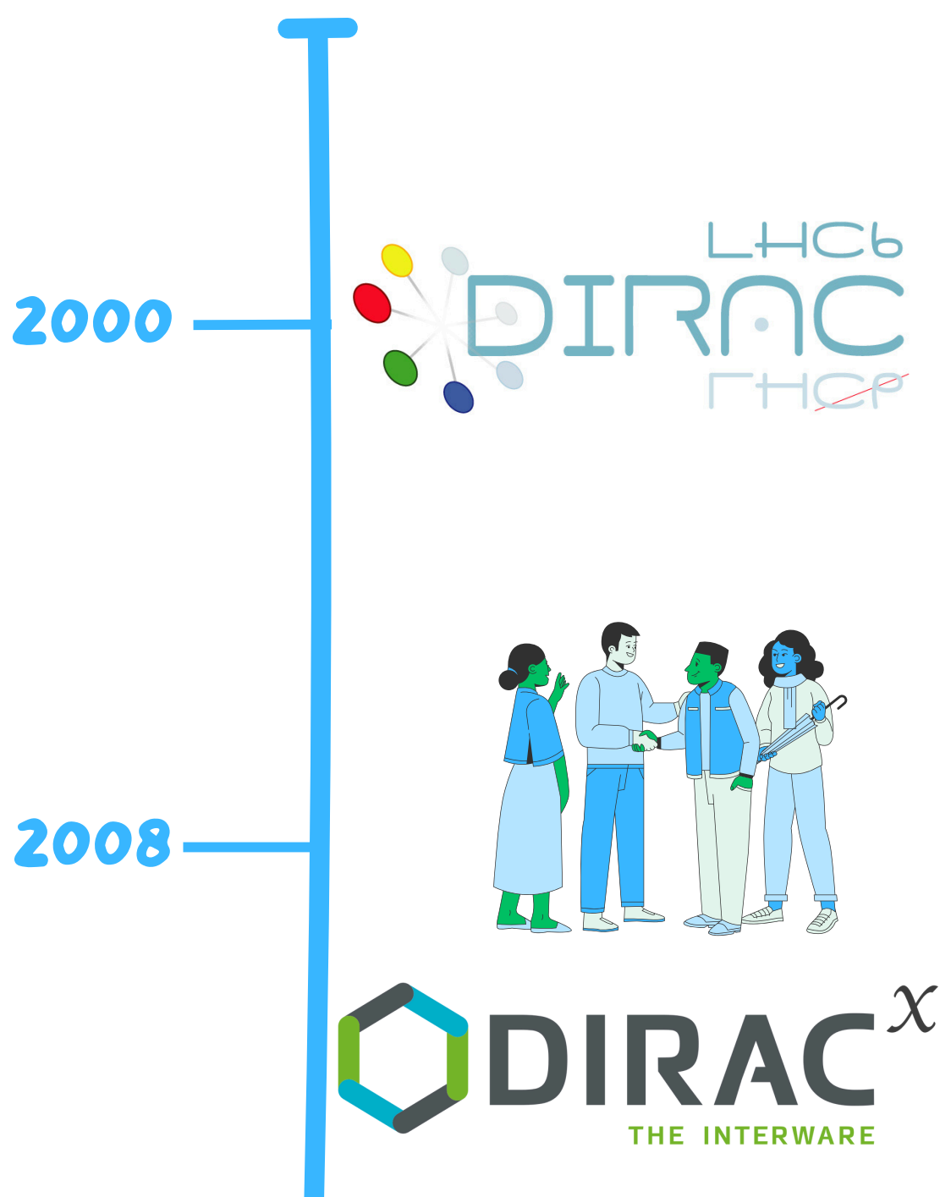


Definition of a 2-step workflow in CWL

Handling CWL workflows within Dirac

- CWL workflows can be tested locally using community tools like *cwltool* and then submitted to Dirac along with Dirac-specific attributes (i.e. target resources).
- In the PS, each CWL step is treated as a separate transformation, with sub-workflows handled as independent CWL workflows.
- The TS still uses metadata-based input queries to trigger job submissions when there are enough outputs from parent transformations available.
- The WMS extracts necessary CWL attributes from jobs for scheduling and executes the workflows on remote resources using *cwltool*. Outputs are then sent to a remote storage resource to be fetched by the TS.



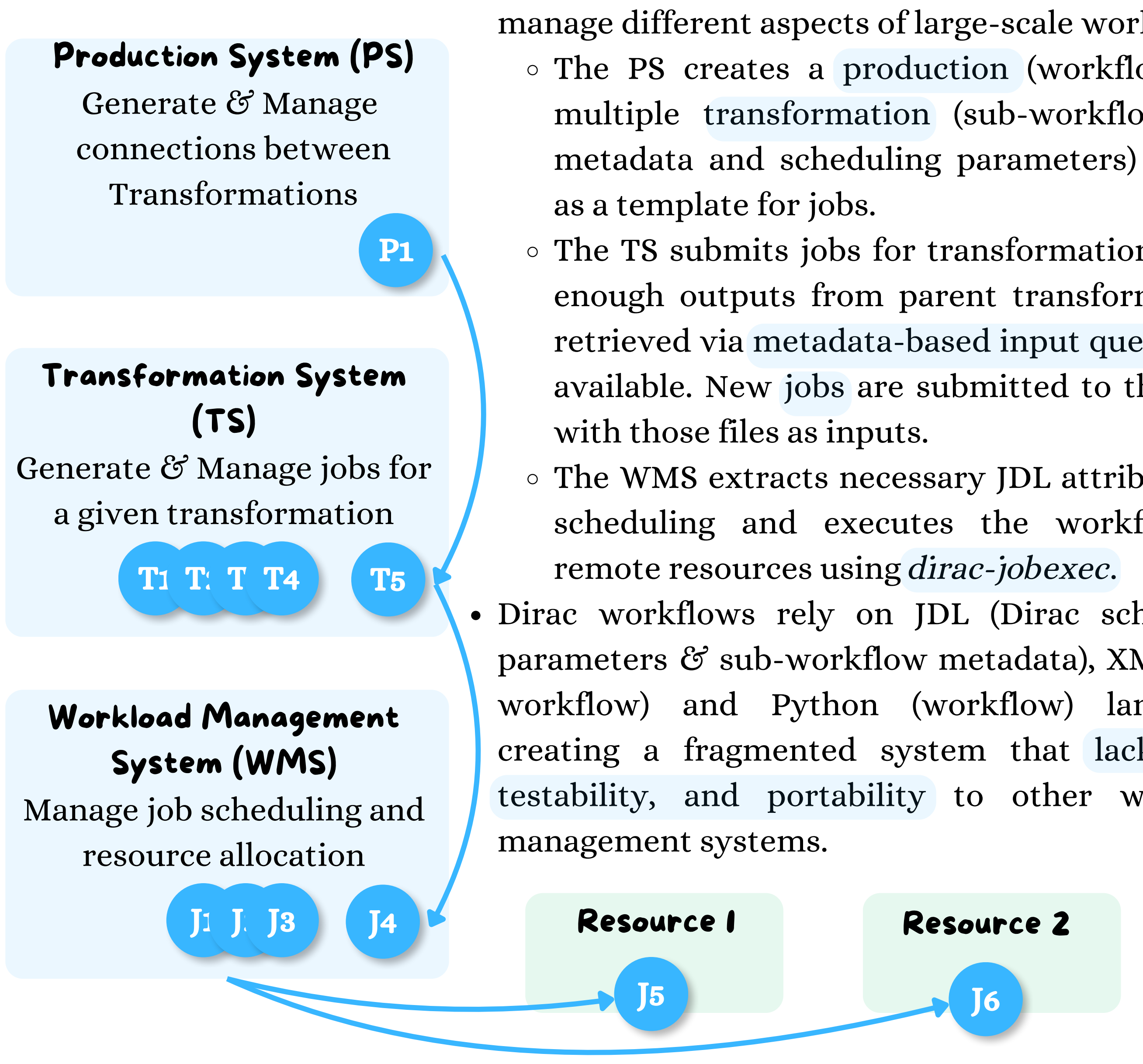


Dirac

- Dirac (Stagni et al., 2024) is a comprehensive framework for managing large-scale scientific workflows on distributed computing resources.
- Originally developed by, and for, the LHCb collaboration at CERN in 2000, now an experiment-agnostic and open source community project adopted across high-energy physics (HEP) and non-HEP experiments.

Managing workflows in Dirac

- Dirac is composed of interconnected systems that manage different aspects of large-scale workflows.
  - The PS creates a production (workflow) and multiple transformation (sub-workflow with metadata and scheduling parameters) serving as a template for jobs.
  - The TS submits jobs for transformations when enough outputs from parent transformations, retrieved via metadata-based input queries, are available. New jobs are submitted to the WMS with those files as inputs.
  - The WMS extracts necessary JDL attributes for scheduling and executes the workflow on remote resources using *dirac-jobexec*.
- Dirac workflows rely on JDL (Dirac scheduling parameters & sub-workflow metadata), XML (sub-workflow) and Python (workflow) languages, creating a fragmented system that lacks local testability, and portability to other workflow management systems.


Production System (PS)
Generate & Manage connections between Transformations
P1
Transformation System (TS)
Generate & Manage jobs for a given transformation
T1
T
T
T4
T5
Workload Management System (WMS)
Manage job scheduling and resource allocation
J1
J
J3
J4
Resource 1
J5
Resource 2
J6

```
# Step image-prod:
output_query_step1 = {"image_width": 7680}
prod_step1 = create_production_step(
   type="image-prod",
   output_query=outputquery
)
prod_step1.Body = description_prod
prod_client.addProductionStep(prod_step1)

# Step image-merge:
output_query_step2 = {"image_width": 7680}
prod_step2 = create_production_step(
   type="image-merge",
   input_query=output_query_step1,
   output_query=output_query_step2,
)
prod_step2.Body = description_merge
prod_step2.ParentStep = prod_step1
prod_client.addProductionStep(prod_step2)
```

Definition of a "2-transformation" production

```
<Workflow>
 <Param name="precision"></Param>
 <StepDefinition>
  <ModuleInstance>
   <name>ImageProd</name>
   <type>ImageProd</type>
  </ModuleInstance>
 </StepDefinition>
</Workflow>
```

Description.xml (Transformation)

```
TransformationID = 240415;
JobID = 1234;
JobName = image-prod-11104080;
Executable = "dirac-jobexec";
Args = "Description.xml -p
precision=0.05";
Platform = "x86_64-el9";
OutputData = "/mandelbrot/img/";
Site = Resource1;
```

JDL (Job description)

Results

- CWL is part of Dirac's dependencies, but is not yet the default method for submission.
- While users can already submit a JDL with a CWL executable to the WMS, this feature remains experimental.
- Prototype systems for Production, Transformation, and Job Management fully supporting CWL are under development.
- A transition plan for Dirac has been established to adopt CWL across all systems and phase out JDL and job description XML, though this process will take several years to complete.
- The integration of CWL into Dirac offers a promising solution to enhance workflow reproducibility and interoperability for large-scale scientific research.

References

B.A. Antunes, D.R.C. Hill, Reproducibility, replicability, and repeatability: A survey of reproducible research with a focus on high performance computing (2024), 2402.07530 https://arxiv.org/abs/2402.07530 / M. Baker, Nature 533, 452 (2016) / M.R. Crusoe, S. Abeln, A. Iosup, P. Amstutz, J. Chilton, N. Tijani´c, H. Ménager, S. Soiland-Reyes, B. Gavrilovi´c, C. Goble et al., Commun. ACM 65, 54–63 (2022) / Stagni, Federico, Boyer, Alexandre, Tsaregorodtsev, Andrei, Lytovchenko, Andrii, Sailer, André, Haen, Christophe, Burr, Christopher, Bauer, Daniela, Fayer, Simon, Martyniak, Janusz et al., EPJ Web of Conf. 295, 04018 (2024)