



**cmcc**

Centro Euro-Mediterraneo  
sui Cambiamenti Climatici

[www.cmcc.it](http://www.cmcc.it)

# **yProv: a Cloud-enabled Service for Multi-level Provenance Management And Exploration in Climate Workflows**

EGI2024

*Session: Reproducible Open Science: making research  
reliable, transparent and credible*

3 October 2024

***F. Antonio<sup>1</sup>, M. Rampazzo<sup>2</sup>, J. Clocchiatti<sup>2</sup>, G. Tabarelli De  
Fatis<sup>2</sup>, L. Sacco<sup>2</sup>, S. Fiore<sup>2</sup>***

<sup>1</sup> *Advanced Scientific Computing Division, CMCC Foundation*

<sup>2</sup> *Department of Information Engineering and Computer Science, University of Trento*

# Provenance introduction

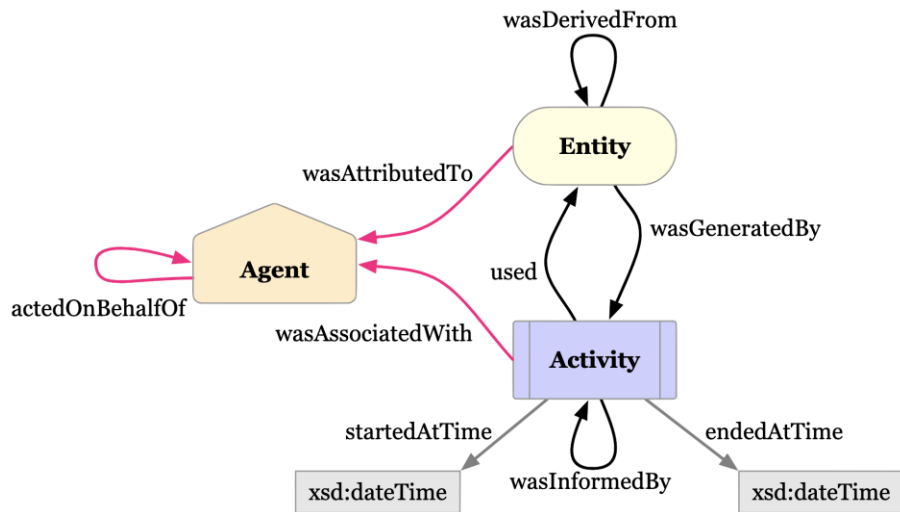
- **Provenance**: the **historical record** of data from its **original sources**
- **Provenance** and **reproducibility** are key requirements for **analytics workflows** in **Open Science** contexts
- **Provenance management** is crucial for **large-scale experiments**
  - Lots of data from the **modelling** and **observational climate communities**
  - Need for managing **lineage information** at different levels of granularity
- Complete **provenance** record can enable **reproducibility** scenarios
- Reproducibility fosters **re-usability** → **FAIR** guiding data principles

# Multi-level provenance management

- The increasing **complexity** of data analysis workflows leads to different needs with respect to provenance management:
  - **Coarse-grained**, regarding the overall set of tasks in a workflow
  - **Fine-grained**, regarding in-depth details of a specific task (*micro-provenance*)
- **Multi-level provenance** management addresses the challenge of navigating and exploring the **provenance space** across multiple axes
  - **Multi-level navigation:** from one level to another, drilling-down into a specific task
  - Provides and limits provenance information to the requested level

# W3C PROV (family of) standards

«**Provenance** is information about **entities**, **activities**, and **people** involved in producing a piece of data or thing, which can be used to form assessments about its **quality**, **reliability** or **trustworthiness**»



<https://www.w3.org/TR/prov-overview/>

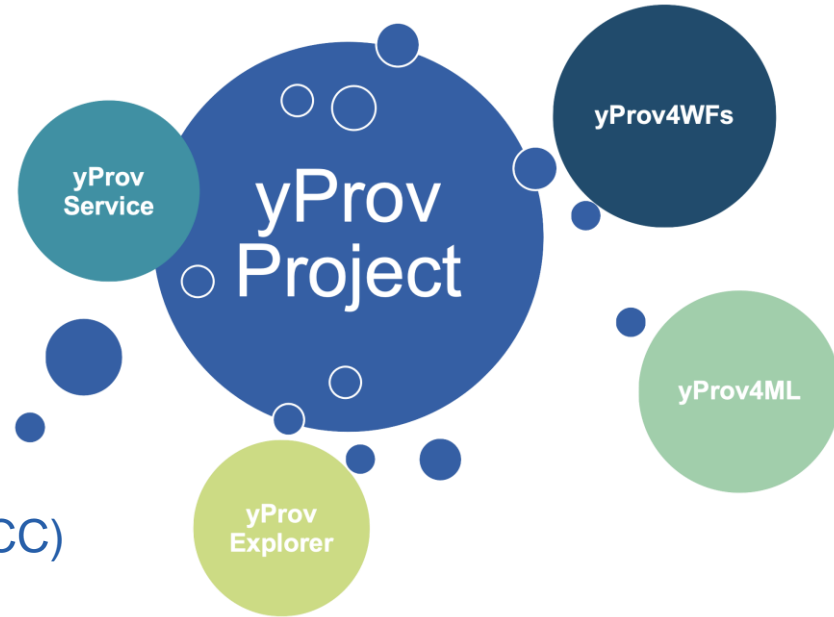
**PROV-DM** is the conceptual data model forming a basis for the W3C PROV family of specifications.

# yProv project

An **interoperable service** and a rich ecosystem of **tools** and **libraries**

- Tracking provenance metadata in complex **AI-based scientific workflows**
- Expand **documentation** of experiments
- Foster **provenance exploration** and **analysis** opportunities

**Co-PIs:** S. Fiore (UniTrento) and F. Antonio (CMCC)

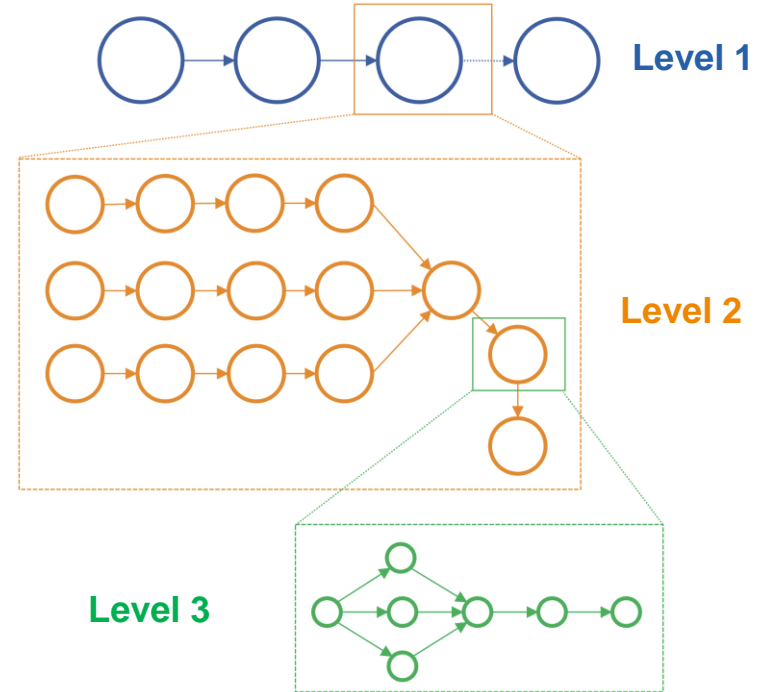


# yProv Service

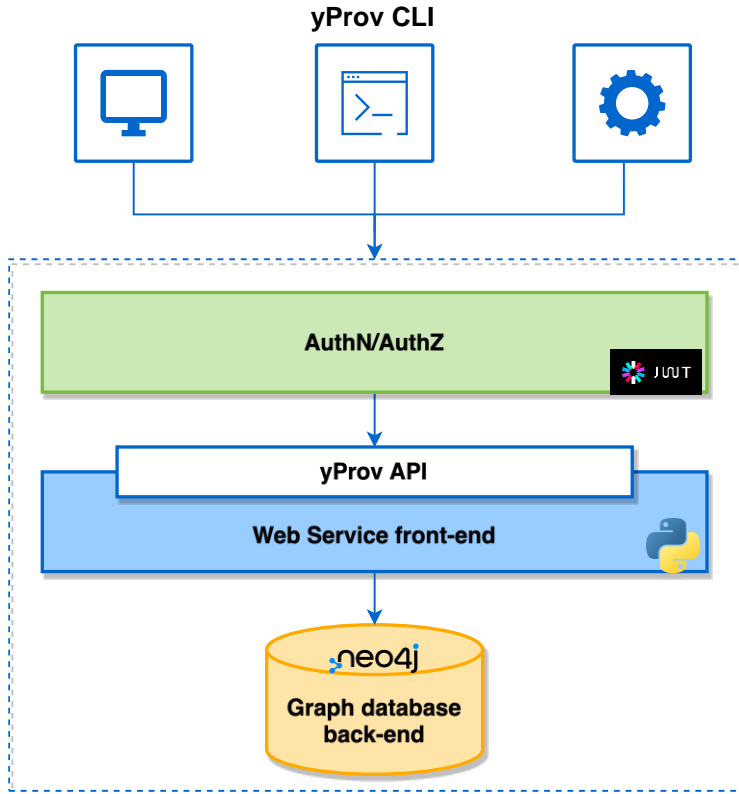
A lightweight and interoperable service for provenance management within end-to-end scientific workflows

- **Multi-level provenance** management support
- Back-end based on **graph data model**
- **Interoperable interface** and **W3C PROV compliance**

<https://github.com/HPCI-Lab/yProv>



# yProv architecture



- **3 components**
  - **Web Service** front-end
  - **Graph database engine** back-end (Neo4J)
  - **Command Line Interface**
- **Authentication/Authorization**
  - Based on **JSON Web Token (JWT)**
- **RESTful API**
  - Easy way to interact with the service and manage PROV information

# yProv API

- 5 resource classes:
  - **document, entity, activity, agent, relation**
- One-to-one association between **document** and **graph database**
- **Isolation** with respect to the provenance management of **different experiments**
- Main operations allowed:
  - **CRUD**: Create, Read, Update, Delete
  - **User registration & authentication**
  - **Permissions management**

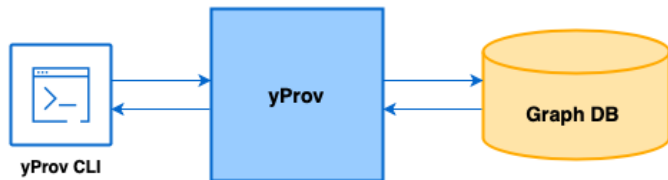
	GET	PUT	POST	DELETE
/documents	Get all documents available			
/documents/<doc.id>	Get a json representation of document <doc.id>	Create or Update document <doc.id>		Delete document <doc.id>
/documents/<doc.id>/subgraph?id=<e.id>	Get the subgraph of entity <e.id> of document <doc.id>			
/documents/<doc.id>/permissions		Manage users permissions of document <doc.id>		
/documents/<doc.id>/entities	Get all entities in document <doc.id>		Create a new entity in document <doc.id>	
/documents/<doc.id>/entities/<e.id>	Get a json representation of entity <e.id> of document <doc.id>	Create or Replace entity <e.id> of document <doc.id>		Delete entity <e.id> and its relations from document <doc.id>
/documents/<doc.id>/activities	Get all activities in document <doc.id>		Create a new activity in document <doc.id>	
/documents/<doc.id>/activities/<ac.id>	Get a json representation of activity <ac.id> of document <doc.id>	Create or Replace activity <ac.id> of document <doc.id>		Delete activity <ac.id> and its relations from document <doc.id>
/documents/<doc.id>/agents	Get all agents in document <doc.id>		Create a new agent in document <doc.id>	
/documents/<doc.id>/agents/<ag.id>	Get a json representation of agent <ag.id> of document <doc.id>	Create or Replace agent <ag.id> of document <doc.id>		Delete agent <ag.id> and its relations from document <doc.id>
/documents/<doc.id>/relations	Get all relations in document <doc.id>		Create a new relation in document <doc.id>	
/auth/register			Register to the service by username and password	
/auth/login			Log in to the service to get a valid token	

yProv API and mapping with HTTP verbs



# yProv CLI

- An easy-to-use tool for interacting with the **yProv Web Service** front-end
- **Python wrappers** to the RESTful API calls



<https://github.com/HPCI-Lab/yProv-CLI>

```
export YPROV_ADDR=<yProv service address>
export YPROV_PORT=<yProv service port>
```

```
yprov-cli auth register --user <username>
--password <password>
```

```
yprov-cli auth login --user <username>
--password <password> → TOKEN
```

```
export YPROV_TOKEN=<token>
```

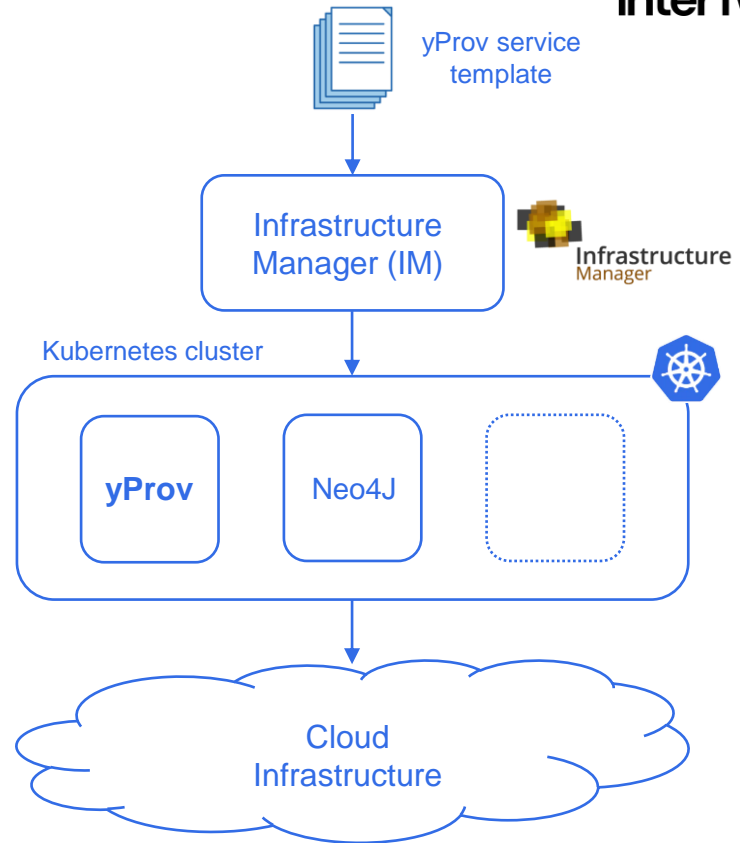
```
yprov-cli documents create --doc-id pta
--file pta.json
```

```
yprov-cli documents subgraph --doc-id pta
--e-id <node id>
```

# Cloud-based yProv version

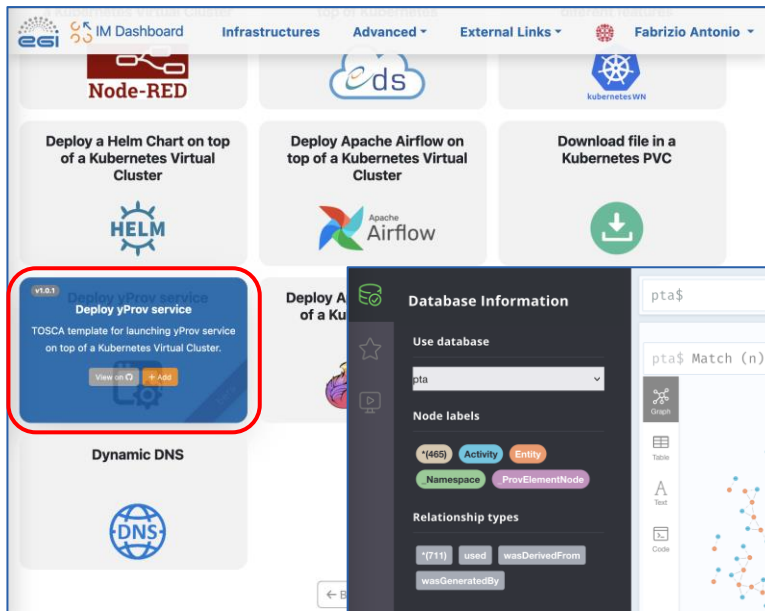


- **yProv components** handled as **Docker containers** for easy deployment and orchestration on a cloud infrastructure
- **Kubernetes** for managing containerized workloads and services
- **IM** (Infrastructure Manager) tool for provisioning and configuring resources
- **TOSCA template** for describing the service, its components and the orchestration process
- Towards an integration into the **EOSC** ecosystem

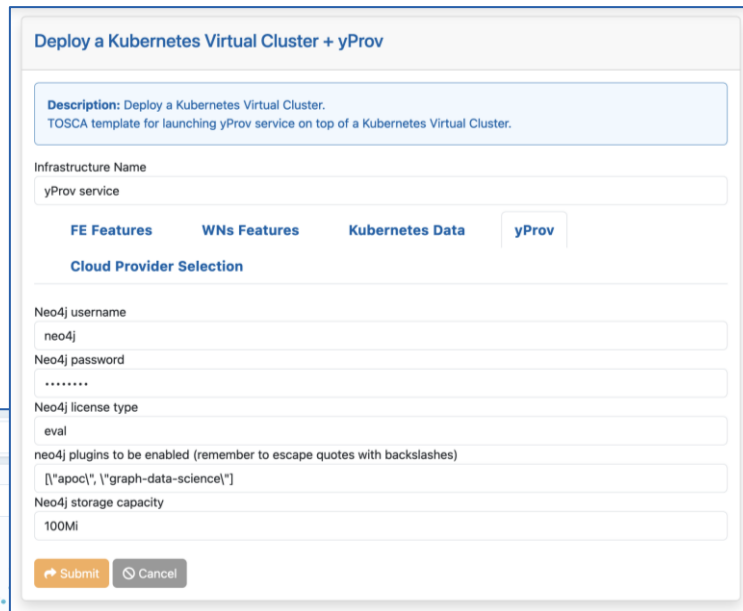


# yProv & IM dashboard

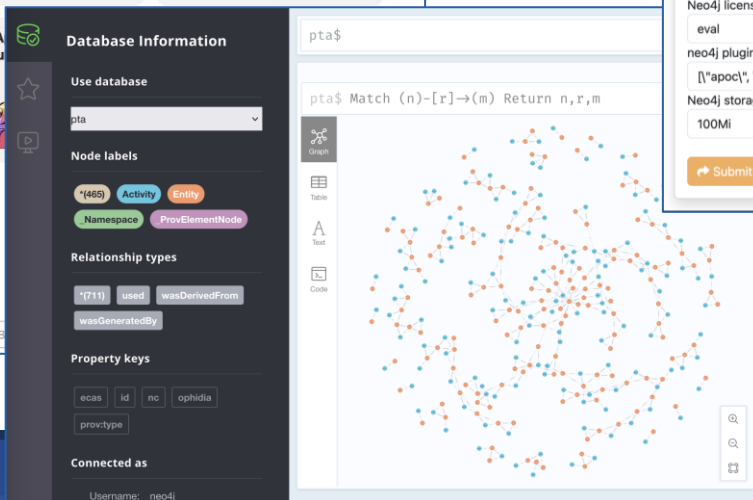
<https://im.egi.eu/im-dashboard/>



The screenshot shows the IM Dashboard interface. At the top, there is a navigation bar with 'egi IM Dashboard', 'Infrastructures', 'Advanced', 'External Links', and a user profile 'Fabrizio Antonio'. Below the navigation bar, there are several service cards: 'Node-RED', 'eds', 'kubernetes WN', 'Deploy a Helm Chart on top of a Kubernetes Virtual Cluster', 'Deploy Apache Airflow on top of a Kubernetes Virtual Cluster', and 'Download file in a Kubernetes PVC'. A red box highlights a card for 'Deploy yProv service' with a 'View on G' button.



The screenshot shows a form titled 'Deploy a Kubernetes Virtual Cluster + yProv'. It includes a description: 'Deploy a Kubernetes Virtual Cluster. TOSCA template for launching yProv service on top of a Kubernetes Virtual Cluster.' Below the description, there is a field for 'Infrastructure Name' with the value 'yProv service'. There are tabs for 'FE Features', 'WNs Features', 'Kubernetes Data', and 'yProv'. Under 'Cloud Provider Selection', there are several input fields: 'Neo4j username' (value: neo4j), 'Neo4j password' (value: .....), 'Neo4j license type' (value: eval), 'Neo4j plugins to be enabled' (value: ["apoc", "graph-data-science"]), and 'Neo4j storage capacity' (value: 100Mi). At the bottom, there are 'Submit' and 'Cancel' buttons.



The screenshot shows the Neo4j database interface. It includes a 'Database Information' section with 'Use database' set to 'pta'. Below that, there are 'Node labels' (Activity, Entity, Namespace, ProvElementNode) and 'Relationship types' (used, wasDerivedFrom, wasGeneratedBy). There is also a 'Property keys' section with 'ecas', 'id', 'nc', 'ophidia', and 'prov.type'. At the bottom, it shows 'Connected as' with the username 'neo4j'. On the right, there is a Cypher query editor with the query 'pta\$ Match (n)-[r]->(m) Return n,r,m' and a graph visualization showing a network of nodes and relationships.

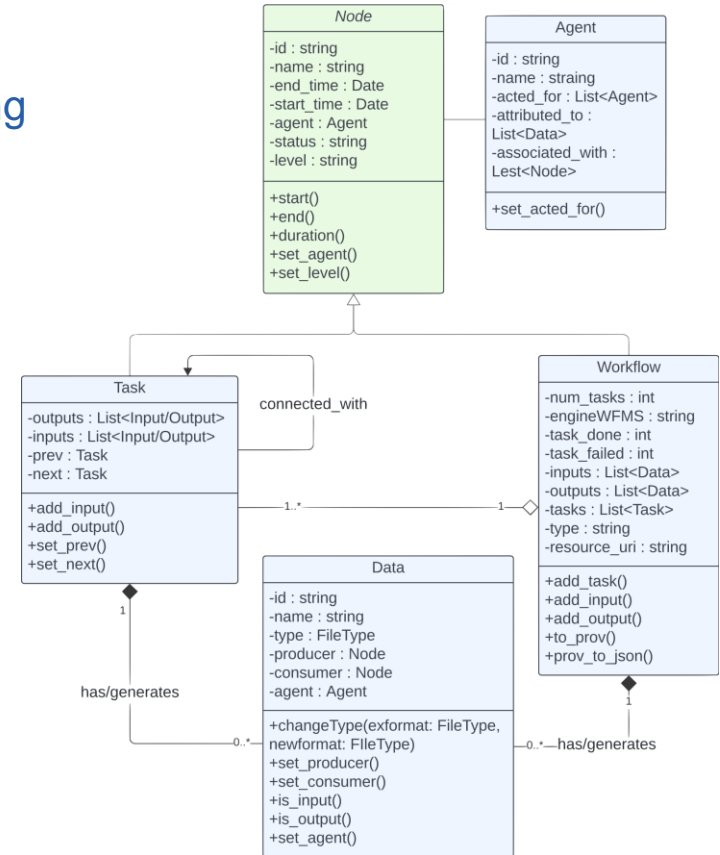


Acknowledgements  
M. Caballer and I. Blanquer

# yProv4WFs

A Python library for standard workflow provenance tracking

- Data model defining 5 different concepts:
  - **Node, Task, Workflow, Data, Agent**
- Provenance information collected at runtime
  - **Overall workflow metrics**
  - **Metrics related to ach specific task**
- Support to different Workflow Management Systems:  
e.g., **Streamflow, Cylc, ecFlow** (in progress)

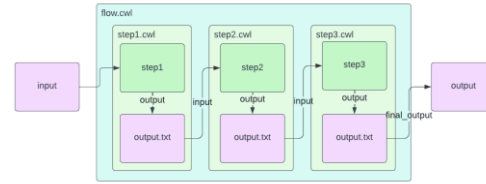


# Prov4WFs: a simple test case with multi-level outputs

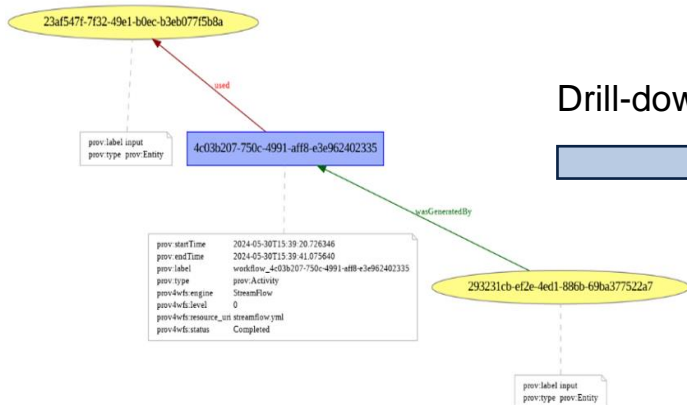


Level 0

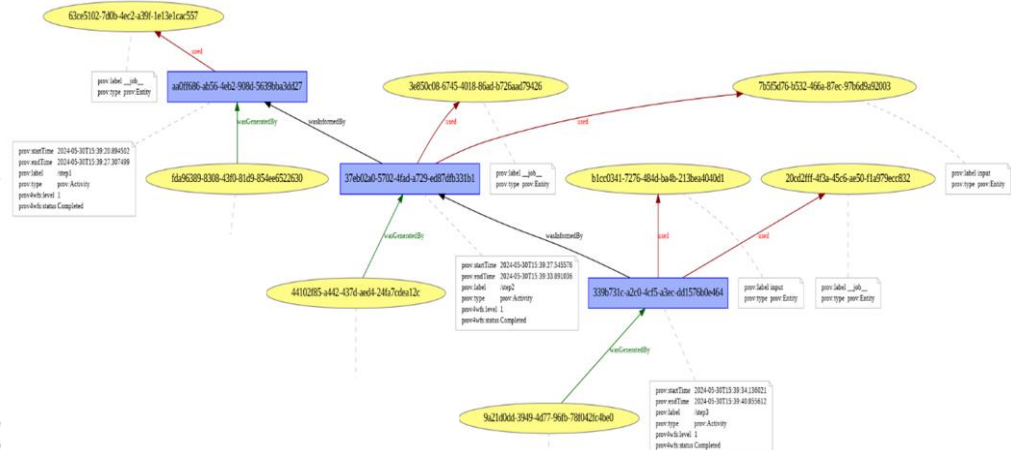
Drill-down



Level 1

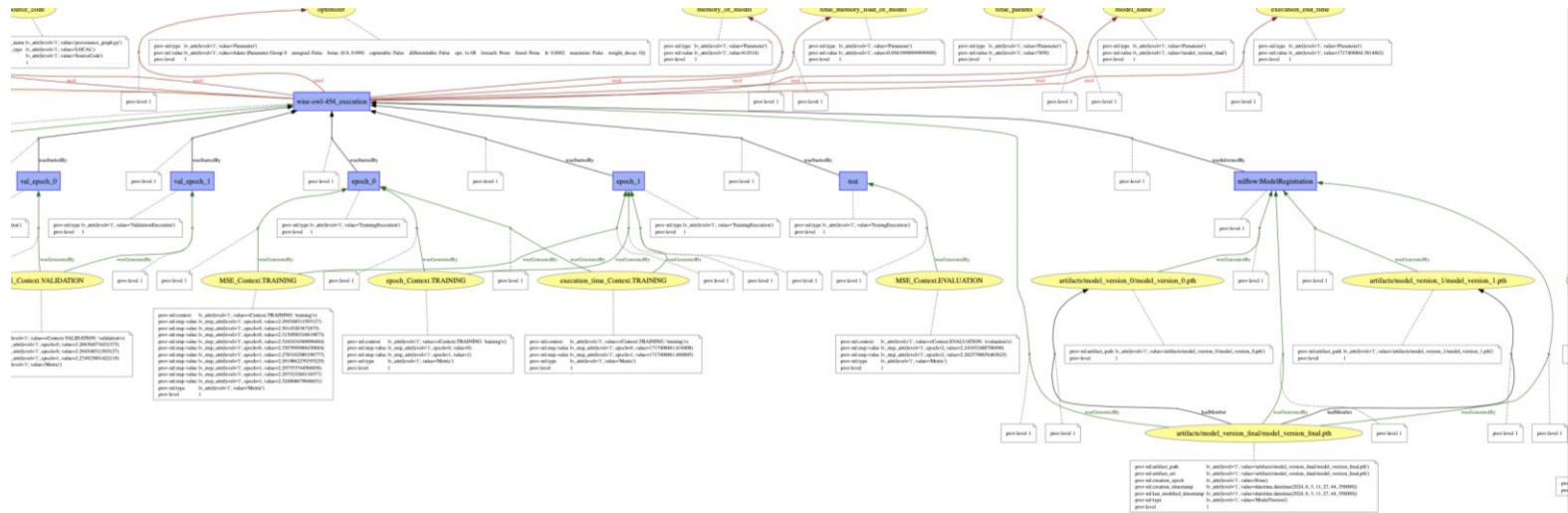


Drill-down



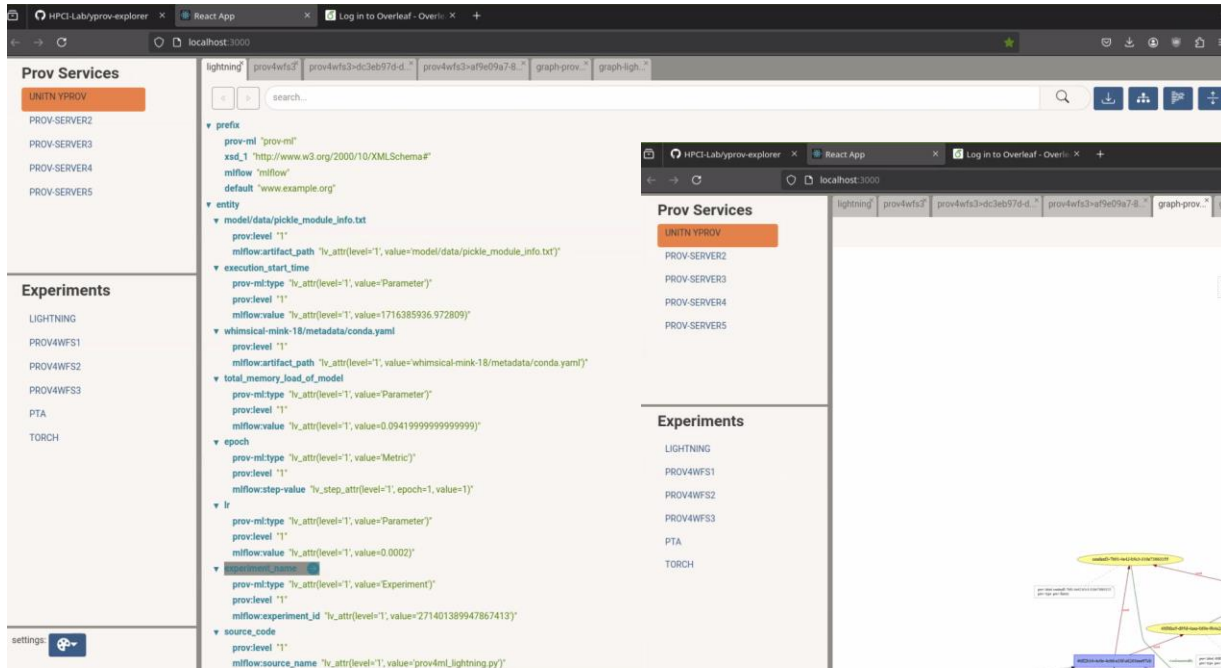
# yProv4ML

A Python library for tracking provenance in ML processes



**Goal:** tracking W3C PROV-compliant provenance within AI processes jointly with a set of key metrics, across runs and epochs

# yProv Explorer (alpha version)



The screenshot displays the yProv Explorer interface with a textual view of provenance. The left sidebar lists 'Prov Services' (LUNITN YPROV, PROV-SERVER2 to 5) and 'Experiments' (LIGHTNING, PROV4WFS1-3, PTA, TORCH). The main area shows a JSON-like structure for a provenance record:

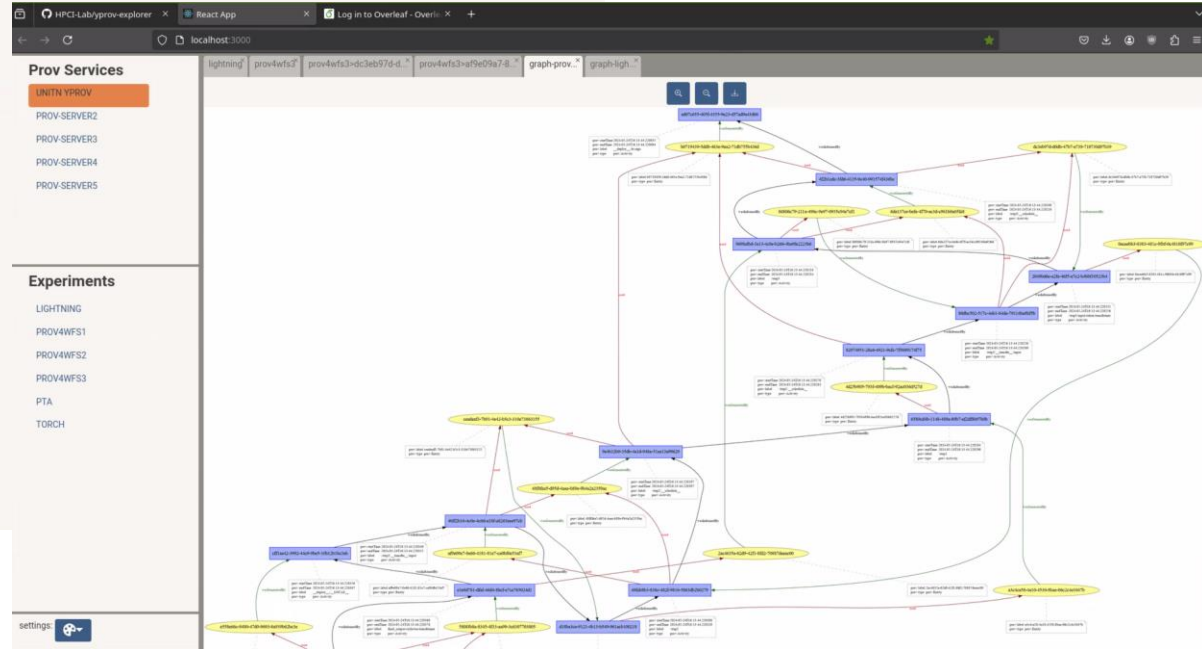
```
prefix
  prov:ml "prov:ml"
  xsd_1 "http://www.w3.org/2000/10/XMLSchema#"
  miflow "miflow"
  default "www.example.org"

entity
  model/data/pickle_module_info.txt
  prov:level "1"
  miflow:artifact_path "lv_attr(level=1, value=model/data/pickle_module_info.txt)"
  execution_start_time
  prov:ml:type "lv_attr(level=1, value=Parameter)"
  prov:level "1"
  miflow:value "lv_attr(level=1, value=1716385936.972809)"
  whimsical-mink-18/metadata/conda.yaml
  prov:level "1"
  miflow:artifact_path "lv_attr(level=1, value=whimsical-mink-18/metadata/conda.yaml)"
  total_memory_load_of_model
  prov:ml:type "lv_attr(level=1, value=Parameter)"
  prov:level "1"
  miflow:value "lv_attr(level=1, value=0.094199999999999999)"
  epoch
  prov:ml:type "lv_attr(level=1, value=Metric)"
  prov:level "1"
  miflow:step-value "lv_attr(level=1, epoch=1, value=1)"
  lr
  prov:ml:type "lv_attr(level=1, value=Parameter)"
  prov:level "1"
  miflow:value "lv_attr(level=1, value=0.0002)"
  experiment_id
  prov:ml:type "lv_attr(level=1, value=Experiment)"
  prov:level "1"
  miflow:experiment_id "lv_attr(level=1, value=Z7140138947867413)"
  source_code
  prov:level "1"
  miflow:source_name "lv_attr(level=1, value=prov4ml_lightning.py)"
```



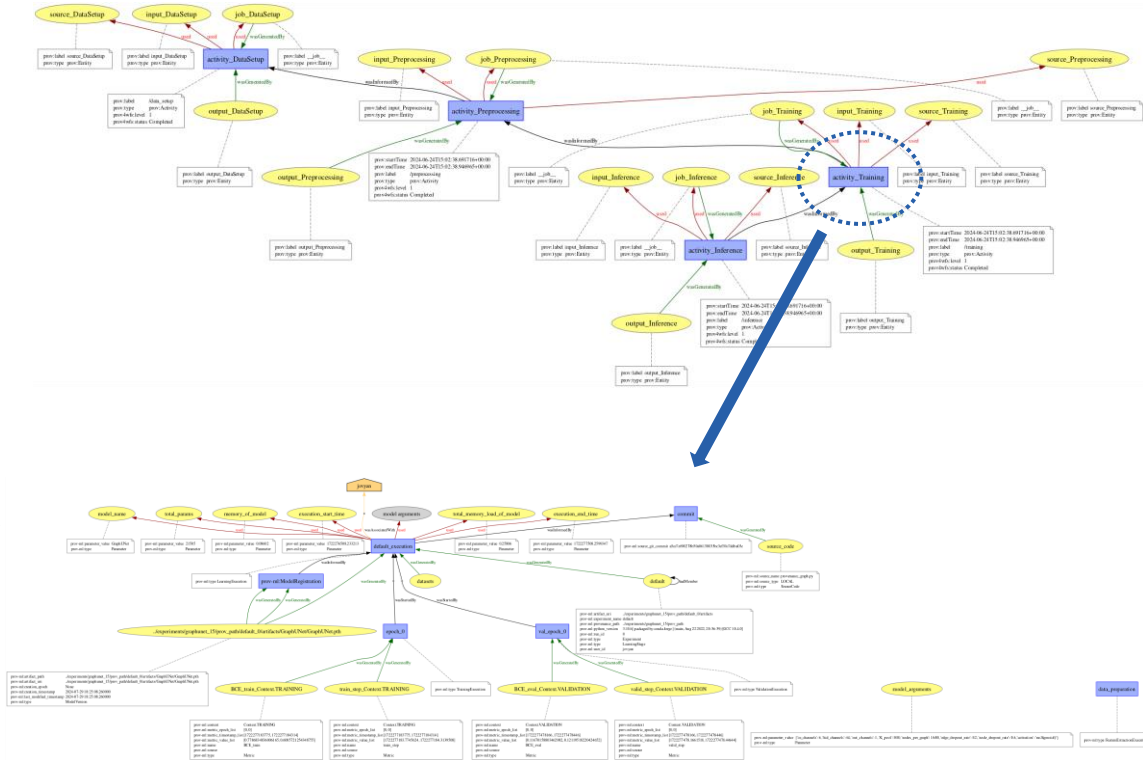
Textual prov exploration

Graphical prov exploration



# AI-based pipeline for Climate Extremes

- Integration within **Tropical Cyclones DT** from interTwin
- **ENES Data Space** used as development platform
- **Provenance tracking** performed throughout the **overall pipeline** as well as the specific **AI training task**
- Valuable in terms of **documentation** and **energy-related metrics**





# Conclusions and future work

## Conclusions

- **Multi-level approach** for a more structured/modular provenance management in climate workflows
- **Interoperable service and a rich ecosystem of tools and libraries** (yProv4ML, yProv4WFs, yProvExplorer)

## Future work

- Broader exploitation within **EOSC-related projects** (e.g., interTwin, EOSC Beyond)
- **Service enhancement** to include data-driven scenarios, new metrics and advanced use cases
- Enhancement of the **UI** for **searching, navigating** and **exploring** provenance graphs
- **Sustainable approach** with new proposals submitted or under preparation



[www.cmcc.it](http://www.cmcc.it)



[fabrizio.antonio@cmcc.it](mailto:fabrizio.antonio@cmcc.it)