

RO-Crate

Capturing FAIR Research outputs for reproducibility across domains

Stian Soiland-Reyes

Munazah Andrabi

Finn Bacall

Eli Chadwick

Warren Del-Pinto

Alexander Hambley

Doug Lowe

Stuart Owen

Phil Reed

Oliver Woolland

Annie Zheng

Carole Goble

Abigail Miller

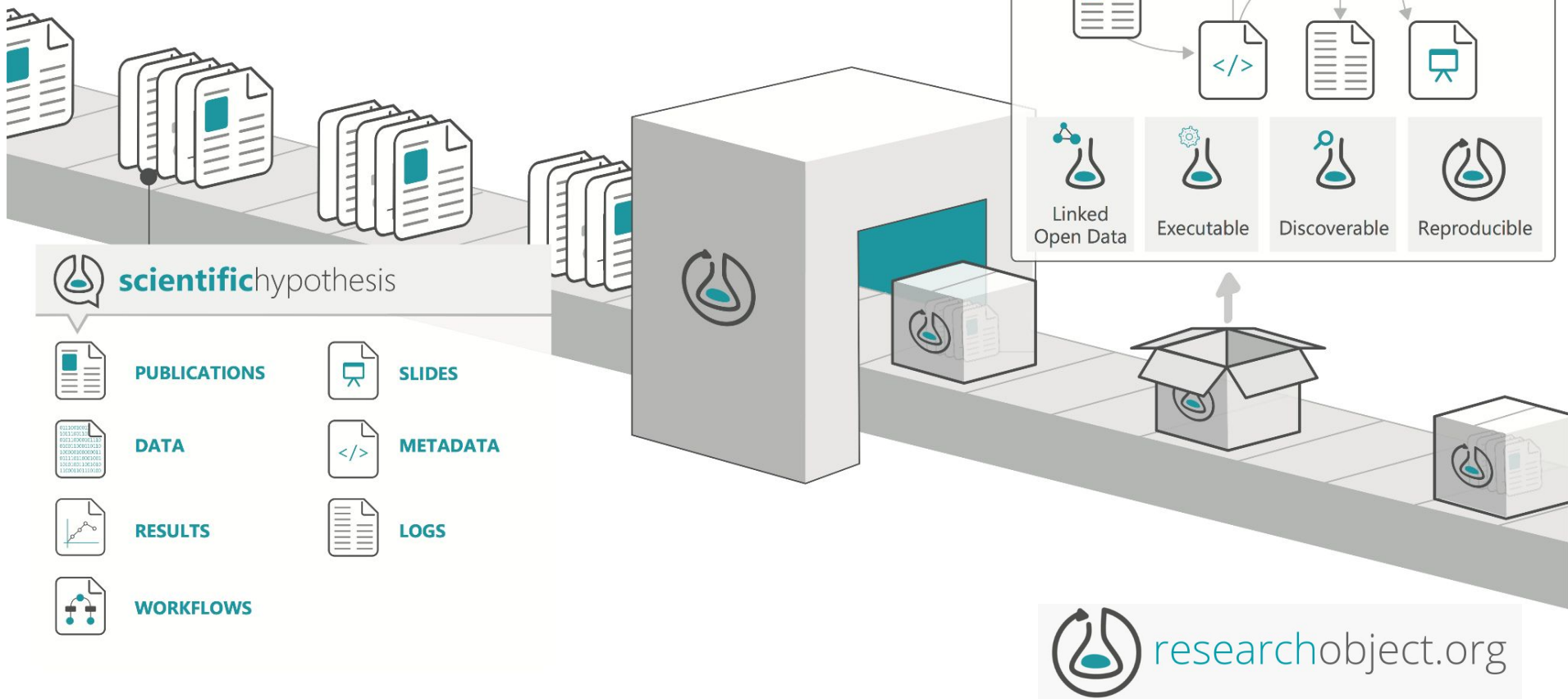
Peter Sefton

RO-Crate community



© 2025 [Stian Soiland-Reyes](https://orcid.org/0009-0001-9312-548X), et al, licensed under [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)
<https://doi.org/10.5281/zenodo.17801347>

 Enabling **reproducible**, transparent research.



RO-Crate: Practical and general purpose



Infrastructure independent – avoiding repository/service silos
Practical, lightweight, robust

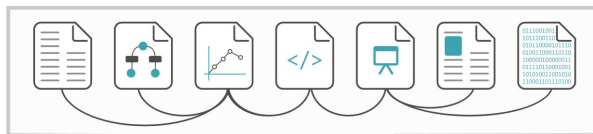


Familiar, developer friendly, web native, machine- and human-readable,
search-engine accessible
Adaptable Linked Data JSON and PIDs



Embrace diversity, legacy, unknowns, open-ended, multi-interpretation,
self-describing, interlingua
Adaptable Metadata Profiles

Aims of FAIR Research Objects



Challenge	Addressed by
Lack of standardized packaging for research outputs	Describing and packaging data collections , datasets, software etc. with their metadata
Limited interoperability across systems	Platform-independent object exchange between repositories and services using Linked Data and persistent identifiers
Barriers to reuse and reproducibility	Supporting reproducibility and analysis : link data with code and workflows
Loss of context during data sharing	Propagating provenance and existing metadata
Barrier to reuse and reproducibility	Making datasets self-describing and machine-actionable , enabling better discoverability and reuse.
Difficulty publishing heterogeneous data types together	Publishing and archiving mixed objects and references
Complexity of existing metadata standards	Reusing existing standards , but hiding their complexity

RO-Crate in software ecosystems

Standardizing data packaging with RO-Crate supports reliable **data consumption** within research environments

Sharing data across tools, services, and repositories is seamless

Data always **travel with their metadata**, which includes persistent identifiers, provenance, and ontology annotations

Reproducibility is enhanced, **linking** datasets with executable workflows

Enables **sensitive datasets** to be described and moved between **secure environments** with clear **governance**



RO-Crate Data Package Contents

RO-Crate Metadata

A single linked-data document describing data sets and their context



primary/raw data
processed/derived data
image/multimedia files



Local directories and files
within the package and/or links
to digital resources



context entities

license
software
equipment
contacts
publications

who created this data?
how was it created?
what study does it belong to?
who funded it?
how can I use it?

RO-Crate Preview

Human readable HTML file representing
the linked-data metadata



RO-Crate Files and Directories

These are the physical files and folders
referenced in the RO-Crate metadata



ro-crate-metadata.json

- Main metadata file located at the root of the RO-Crate package
- If this file is present, it is an RO-Crate!
- Uses JSON-LD format: simple and extensible for diverse use cases
- Describes the datasets and contextual metadata
- Includes metadata about authors, institutions, licenses, etc.
- Base vocabulary and types: schema.org

Contents of an RO-Crate

Example:

<https://by-covid.github.io/baseline-use-case-comparative-analysis/>
<https://doi.org/10.5281/zenodo.11209362>

RO-Crate Structure

@context: a list of **terms** that can be used in the rest of the metadata. This maps to **vocabularies**, such as schema.org, Bioschemas.

e.g. "license" -> <http://schema.org/license>

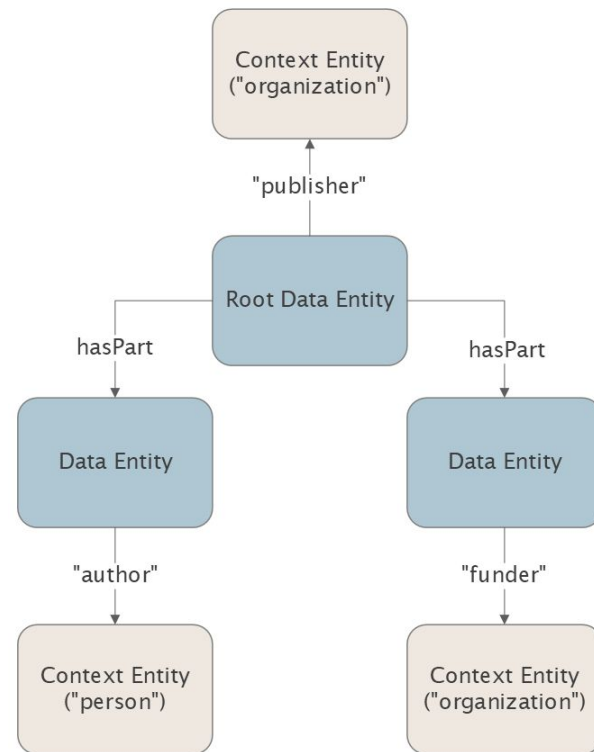
@graph: a list of "things in the crate" described with **terms** from the context.

e.g. "@id": "data.csv" describes the file data.csv

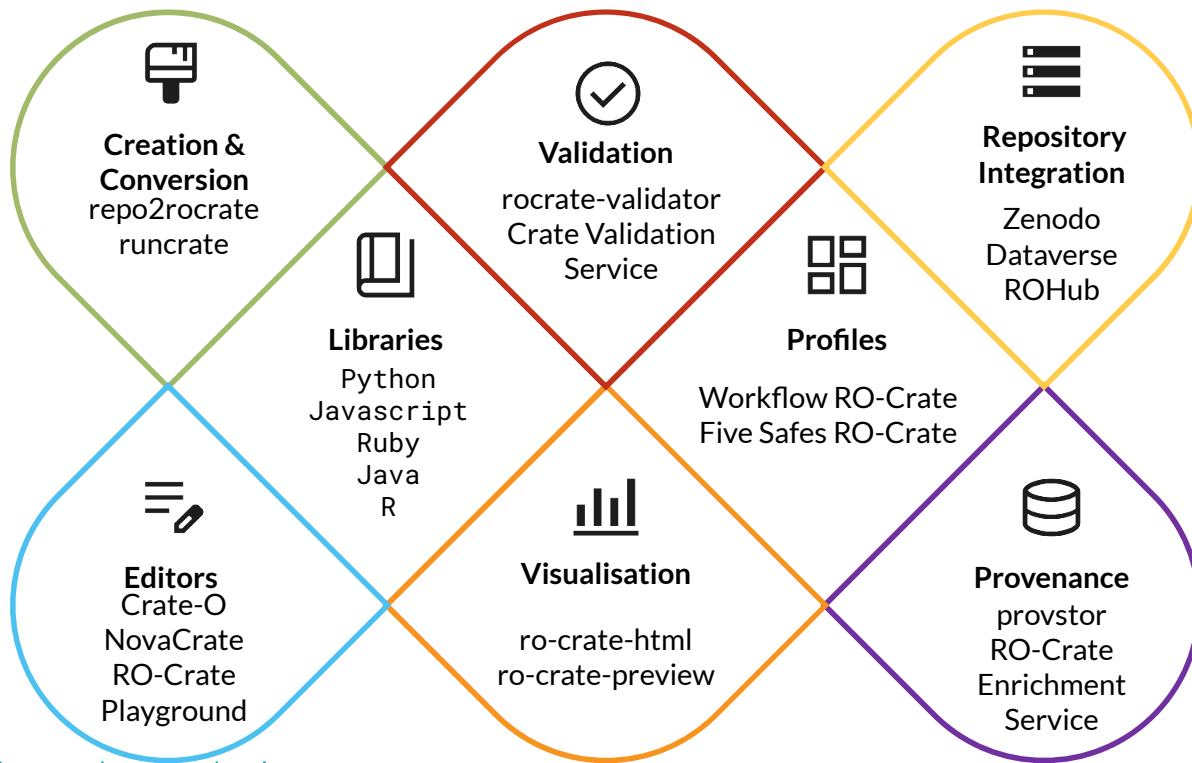
```
{
  "@context": "https://w3id.org/ro/crate/1.2/context",
  "@graph": [
    {
      "@id": "ro-crate-metadata.json",
      "@type": "CreativeWork",
      "conformsTo": {"@id": "https://w3id.org/ro/crate/1.2"},
      "about": {"@id": "./"}
    },
    {
      "@id": "./",
      "@type": ["Dataset", "LearningResource"],
      "hasPart": [
        {"@id": "data.csv"}
      ],
      "name": "Example dataset for RO-Crate specification",
      "description": "Official rainfall readings for Katoomba, NSW
2022, Australia",
      "datePublished": "2023-05-22",
      "license": {"@id": "http://spdx.org/licenses/CC0-1.0"},
      "author": { "@id": "https://orcid.org/0000-0002-1825-0097" },
      "publisher": {"@id": "https://ror.org/05gq02987"}
    },
    {
      "@id": "data.csv",
      "@type": "File",
      "name": "Rainfall Katoomba 2022-02",
      "description": "Rainfall data for Katoomba, NSW Australia
February 2022",
      "encodingFormat": "text/csv",
      "license": {"@id":
https://creativecommons.org/licenses/by-nc-sa/4.0/},
      "https://www.researchobject.org/packaging_data_with_ro-crate/
https://www.researchobject.org/packaging\_data\_with\_ro-crate/
}
```

RO-Crate metadata entities

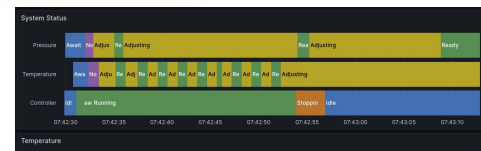
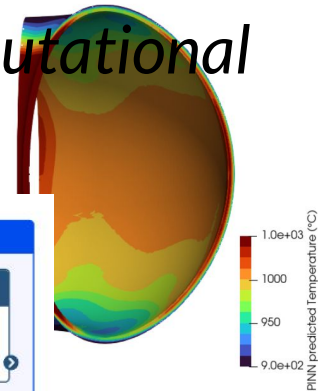
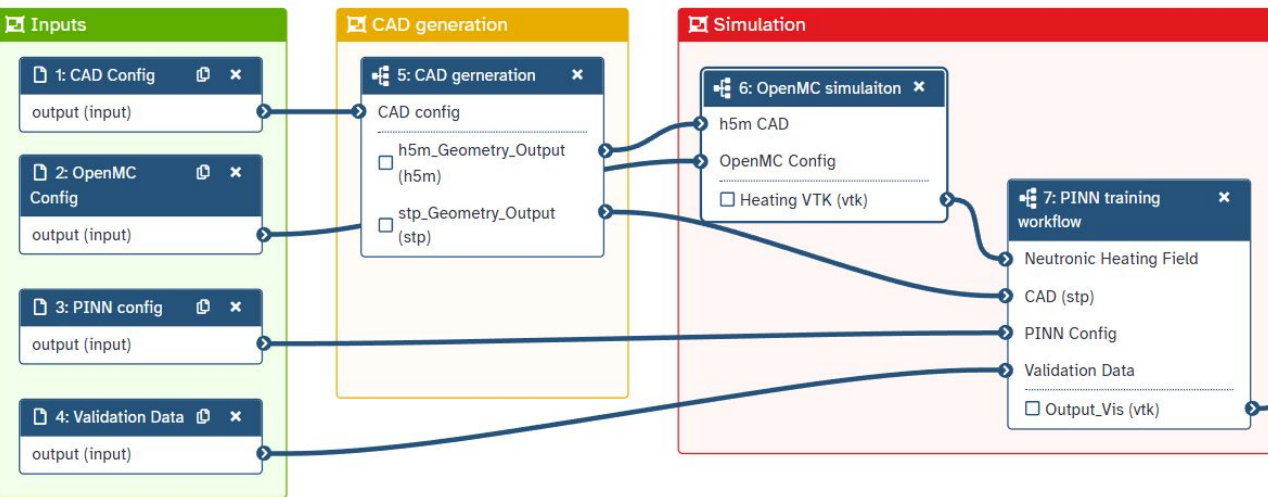
- **Root Data Entity:** The main dataset or digital object being described. Typically represents the folder or primary output.
- **Data Entity:** A file or directory included in the crate (.csv, .txt, images, etc.) that represents a research resource.
- **Context Entity:** Descriptive metadata such as Person, Organization, SoftwareApplication, etc., that provide context but aren't data files.
- Science-related entities include: equipment (instrument → IndividualProduct)



RO-Crate Ecosystem



Workflow systems: Capturing the computational method





The University of Manchester

Many workflow systems exist, with differences across reproducibility, scalability, capability, integration, domains, ...

<https://s.apache.org/existing-workflow-systems>

File Edit View History Bookmarks Tools Help

github.com/common-workflow-language/common-workflow-language/wiki/Existing-Workflow-systems

common-workflow-language / common-workflow-language

Type to search

< Code Issues 220 Pull requests 21 Discussions Actions Wiki Security 4 Insights Settings

Existing Workflow systems

Stian Soiland-Reyes edited this page on Mar 17 · [81 revisions](#)

Permalink: <https://s.apache.org/existing-workflow-systems>

Cite as (update dates):

Peter Amstutz, Maxim Mikheev, Michael R. Crusoe, Nebojša Tijanić, Samuel Lampa, et al. (2024): **Existing Workflow systems**. *Common Workflow Language wiki*, GitHub. <https://s.apache.org/existing-workflow-systems> updated 2024-08-18, accessed 2024-08-18.

Computational Data Analysis Workflow Systems

An incomplete list

Please add new entries at the bottom. Try to include links to website and/or source code and ideally DOI for publications.

In addition to this list, actively developed free/open-source systems should be registered at <https://workflows.community/systems>

See also: <https://github.com/pditommaso/awesome-pipeline>

1. Arvados - CWL-based distributed computing platform for data analysis on massive data sets. <https://arvados.org/> <https://github.com/arvados/arvados>
2. Apache Taverna <http://www.taverna.org.uk/> <https://taverna.incubator.apache.org/>
3. Galaxy <https://galaxyproject.org/>

Pages 23

Find a page or section...

- Home
- 2021 CWL Mini Conference
- Arvados
- Common Workflow Language != K...
- Conference Video Guidelines
- CWL Implementations
- CWL v1.2.1 "Barn Raising"
- cwl2script
- cwltool (reference implementation)
- Example workflow run provenance
- Existing CLI description languages
- Existing Workflow systems

Computational Data Analysis



A registry for describing, sharing and publishing **scientific computational workflows**

WorkflowHub aims to **facilitate discovery and re-use** of workflows in an accessible and interoperable way. This is achieved through extensive use of **open standards** and tools, including [CWL](#), [RO-Crate](#), [Bioschemas](#) and [GA4GH's TRS API](#), in accordance with the **FAIR principles**.

WorkflowHub **supports workflows of *any* type** in its native repository.

<https://workflowhub.eu/>

<https://workflowhub.eu/workflows/1838>

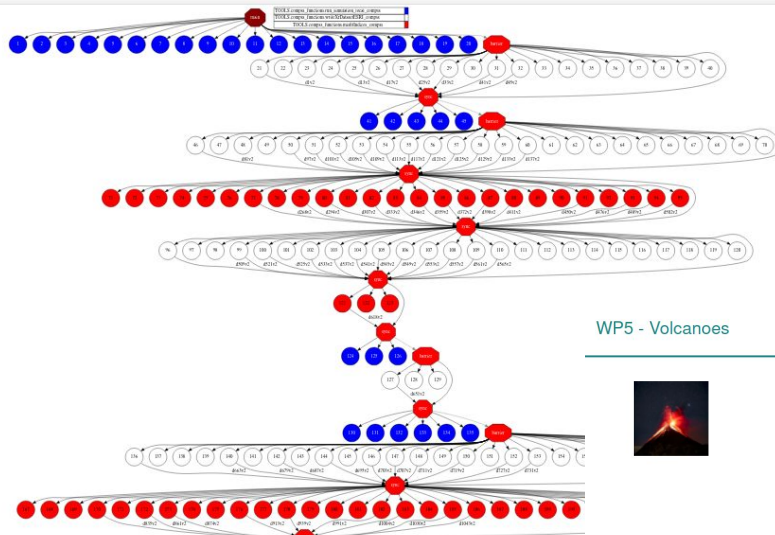
[DTC-V3] WF5301: PyCOMPSs DT-LAVA-WF - Etna 2006 lava test- case

Version 1

[Visit source](#)[Download RO-Crate](#)**DT-GEO**<https://dtgeo.eu/>[Overview](#)[Files](#)[Related items](#)**Workflow Type:** COMPSS

Work-in-progress

PyCOMPSs DT-LAVA-WF (Lava flow digital twin component DTCV3) run in linux, local laptop experiment test



WP5 - Volcanoes



Develop and implement 4 DTCs for volcano-related extremes: volcanic unrest (DTC-V1), forecast of volcanic ash clouds and fallout (DTC-V2), lava flows (DTC-V3), and volcanic gases (DTC-V4).

Test the 4 DTC-V through demonstrators at 3 relevant European sites: Mt. Etna in Italy (SD1), and Grímsvötn and Fagradalsfjall in Iceland (SD2 and SD3 respectively).

Space: A Digital Twin for GEOPhysical extremes (DT-GEO)**Public web page:** <https://dtgeo.eu/>**Organisms:** Not specified

Creators and Submitter

Creators

Louise Cordrie, Giovanni Macedonio, Antonio Costa Roberto Spina Francesco Zuccarello, Gaetana Ganci Annalisa Cappello

Submitter

Louise Cordrie

License

MIT No Attribution (MIT-0)

Extract +

< > 🏠 Location: 📁 /

Name

- application_sources
- App_Profile.json
- complete_graph.svg
- compss_submission_command_line.t
- DTCV3_Louise.yaml

ro-c

ro-c

[Browse files](#) PyCOMPSSs VLAVA - Etna 2006 lava test-case

@id	/
name [?]	PyCOMPSSs VLAVA - Etna 2006 lava test-case
@type	Dataset
description [?]	PyCOMPSSs VLAVA run in linux, local laptop experiment test
datePublished [?]	2025-07-24T11:43:57+00:00
author [?]	<ul style="list-style-type: none"> Louise Cordrie Giovanni Macedonno Antonio Costa Roberto Spina Francesco Zuccarelli Gaetana Ganci Annalisa Cappello

Download: WF5301.py

@id	application_sources/WF5301.py
name [?]	WF5301.py
@type	<ul style="list-style-type: none"> File SoftwareSourceCode ComputationalWorkflow
description [?]	Main file of the COMPSSs workflow source files
encodingFormat [?]	text/plain
contentSize [?]	3097
image [?]	complete_graph.svg
programmingLanguage [?]	COMPSSs Programming Model
softwareRequirements [?]	vlava
Items that reference this one	
hasPart [?]	PyCOMPSSs VLAVA - Etna 2006 lava test-case
mainEntity [?]	PyCOMPSSs VLAVA - Etna 2006 lava test-case
about [?]	complete_graph.svg
instrument [?]	COMPSSs WF5301.py execution at 7eeeb066ee94

COMPSSs WF5301.py execution at 7eeeb066ee94

@id	#COMPSSs_Workflow_Run_Crate_7eeeb066ee94_d7ed873c-a96f-4850-812c-33f
name [?]	COMPSSs WF5301.py execution at 7eeeb066ee94
@type	CreateAction
description [?]	Linux 7eeeb066ee94 6.11.0-29-generic #29~24.04.1-Ubuntu SMP PREEMPT_D x86_64 x86_64 GNU/Linux
actionStatus [?]	http://schema.org/CompletedActionStatus
agent [?]	Louise Cordrie
endTime [?]	2025-07-24T11:43:55+00:00
environment [?]	COMPSS_HOME: /opt/COMPSSs/
instrument [?]	WF5301.py
object [?]	<ul style="list-style-type: none"> DATA config.json DATA
resourceUsage [?]	<ul style="list-style-type: none"> maxTime: 10092 executions: 66 avgTime: 3332 minTime: 753 maxTime: 1667 executions: 135 avgTime: 269 minTime: 147 maxTime: 315441 executions: 41 avgTime: 34430 minTime: 3148 maxTime: 10092 executions: 66 avgTime: 3332 minTime: 753 maxTime: 1667 executions: 135 avgTime: 269 minTime: 147 maxTime: 315441 executions: 41 avgTime: 34430 minTime: 3148 executionTime: 722732
result [?]	<ul style="list-style-type: none"> RUN RUN_ASSI PyCOMPSSs VLAVA - Etna 2006 lava test-case

What's inside the RO-Crate?

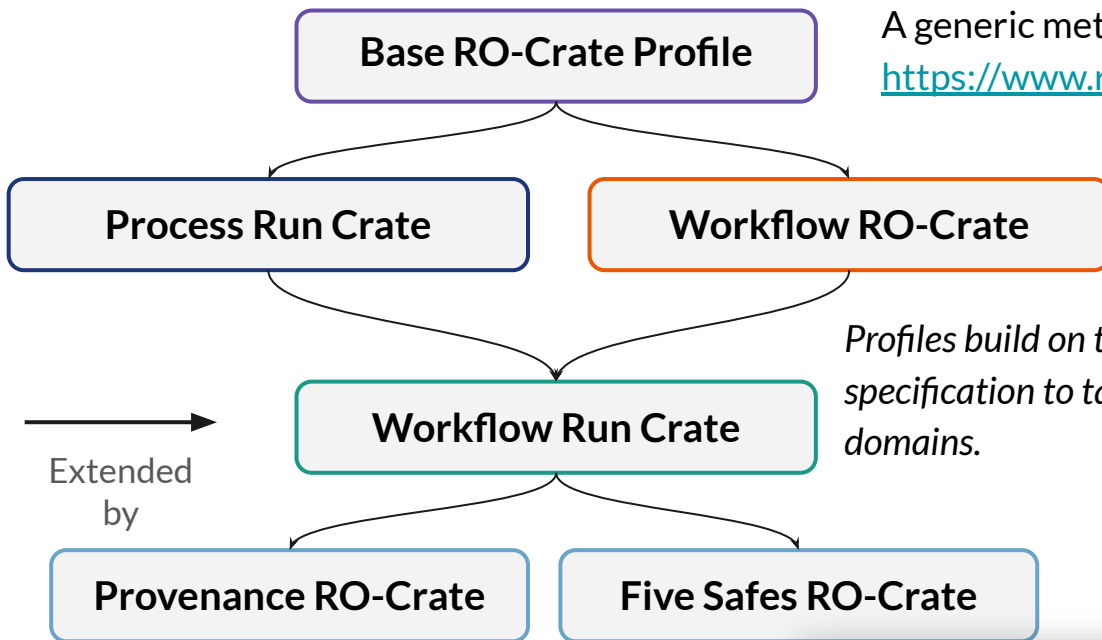
<https://www.researchobject.org/ro-crate/tools>

RO-Crate Profiles - Ducktyping

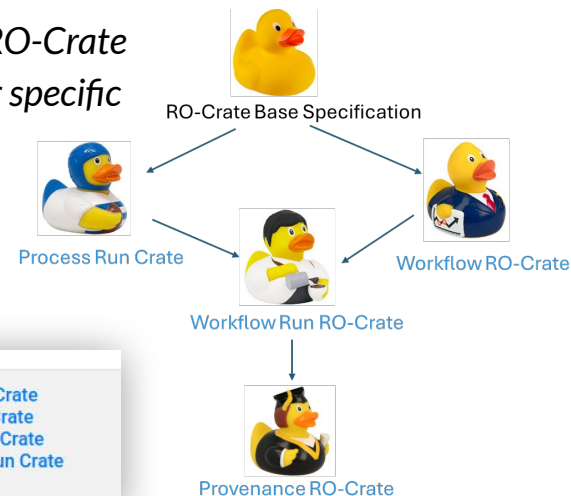
Base RO-Crate Profile

A generic metadata structure for RO-Crate

<https://www.researchobject.org/ro-crate/specification>



Profiles build on the core RO-Crate specification to tailor it for specific domains.



conformsTo [?]

- Workflow RO-Crate
- Process Run Crate
- Workflow Run Crate
- Provenance Run Crate

What are RO-Crate Profiles?

- A set of **rules and expectations** for the structure and metadata of an RO-Crate to ensure consistency and interoperability
- **Community defined**-allows for flexible **domain specific** conventions → minimally useful metadata model (MUST/SHOULD/MAY)
- Enables **reliable programmatic consumption** of RO-Crate content
- Define **constraints and extensions** – for validation and editing
- Profiles can be just human-readable instructions, but ideally published as a **Profile Crate**

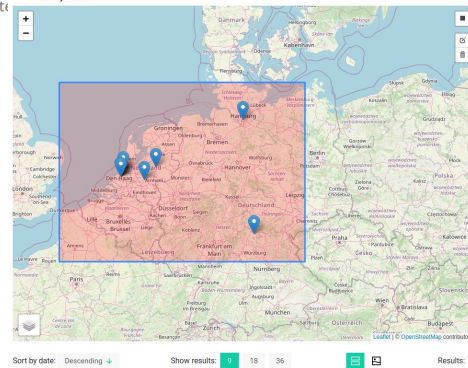
A DataCube Data Entity **MUST** include the following properties:

- `@reliance:spatialCoverage`: MUST be the id of a Contextual Entity that identifies the spatial coverage in detail, including `rel:extent` (expressed as a WKT text), `rel:spatial-step-resolution` and `rel:coordinate-reference-system`.
- `@reliance:temporalCoverage`: MUST be the id of a Contextual Entity that identifies the temporal coverage in detail, including `rel:beginningTime` (expressed as ISO8601), `rel:endTime` (expressed as ISO8601), `rel:temporal-step-resolution` and `rel:temporal-reference-system-unit`.

A DataCube Entity **MAY** include the following property:

- `@verticalCoverage`: MUST be the id of a Contextual Entity that identifies the vertical coverage in detail, including `rel:highestValue`, `rel:lowestValue` and `rel:vertical-reference-system-unit`.

Research objects list



Sort by date: Descending Show results: 1 18 36 Results: 11

PUBLIC **MANUAL** **LIVE** **BASE RESEARCH OBJECT** **TEMPORARY**
APPLIED SCIENCES, CLIMATOLOGY, ECONOMIC GEOGRAPHY, GEOGRAPHY, HUMAN GEOGRAPHY, POLITICAL
GEOGRAPHY, SOCIAL SCIENCES
FAIRification of a Climate Adaptation Strategy
Julia Bartsch, Stephanie Bligam
Created: 08 March 2025 (16:08)
Resources: 1
View

PUBLIC **MANUAL** **LIVE** **RESEARCH-OBJECT RESEARCH OBJECT** **RESEARCH**
APPLIED SCIENCES
City of Hamburg
Lale Mae Nüsse, Malte von Szombathely
Created: 08 March 2025 (15:29)
Resources: 4

Specification

RO-Crate 1.2 (newest release)

[Introduction](#)

[Terminology](#)

[RO-Crate Structure](#)

[Metadata of the RO-Crate](#)

[Root Data Entity](#)

[Data Entities](#)

[Contextual Entities](#)

[The focus of an RO-Crate](#)

[Provenance of entities](#)

[Profiles](#)

[Workflows and scripts](#)

[Appendix](#)

[RO-Crate 1.1](#)

[RO-Crate 1.0](#)

[RO-Crate 0.2](#)

RO-Crate Metadata Specification 1.2

- Permalink: <https://w3id.org/ro/crate/1.2>
- Published: 2025-06-04
- Publisher: researchobject.org community
- Status: Recommendation
- JSON-LD context: <https://w3id.org/ro/crate/1.2/context>
- This version: <https://w3id.org/ro/crate/1.2>
- Alternate formats: [Web pages](#), [single-page HTML](#), [PDF](#), [RO-Crate JSON-LD](#), [RO-Crate HTML](#)
- Previous version: <https://w3id.org/ro/crate/1.1>
- Cite as: <https://doi.org/10.21961/ro-crate.1.2>
- Editors: [Peter Sefton](#), [Stian Ståhl](#)
- Authors: [Peter Sefton](#), [Eoghan](#), [FERNÁNDEZ, Kyle Chard, Jo](#), [Gutierrez, Alasdair J. G. G.](#), [Paul Walk](#), [brandon white](#), [Kennedy, Mark Graves, Jas](#), [Miksa, Marco La Rosa, Ced](#), [Xuanqi Li, Sveinung Gund](#), [Abigail Miller, Jake Emers](#), [Bram Ulrichs, Michael Fa](#), [Bainglass, Balazs E. Patak](#)

Funding and grants

To associate a research project with a [Dataset](#), the *RO-Crate JSON-LD* SHOULD contain an entity for the project using type [Organization](#), referenced by a [funder](#) property. The project [Organization](#) SHOULD in turn reference any external [funder](#), either by using its URL as an [@id](#) or via a *Contextual Entity* describing the funder.

❑ **Tip:** To make it very clear where funding is coming from, the *Root Data Entity* SHOULD also reference funders directly, as well as via a chain of references.

```
{
  "@id": "https://doi.org/10.5281/zenodo.1009240",
  "@type": "Dataset",
  "funder": {
    "@id": "https://ror.org/038sjwq14"
  }
}

{
  "@id": "https://ersearch.uts.edu.au/projects/provisioner",
  "@type": "Organization",
  "description": "The University of Technology Sydney Provisioner project is ...",
  "funder": [
    {
      "@id": "https://ror.org/03f0f6041"
    }
  ]
}
```

See <https://w3id.org/ro/crate/1.2>

Copyright 2019-2020 University of

Copyright 2019-2025 The Universi

Copyright 2019-2025 RO-Crate contributors

<https://www.researchobject.org/ro-crate/specification>

RO-Crate Community

- Active community of developers, researchers, and data stewards
<https://www.researchobject.org/ro-crate/community>

Slack, Drop-in session, community governance calls

- Specification developed openly using GitHub
<https://www.researchobject.org/ro-crate>
- RO-Crate profiles developed in separate working groups or independently
<https://www.researchobject.org/ro-crate/profiles.html>

Tools and libraries:

- ro-crate-py (Python)
- ro-crate-js (JavaScript)
- ro-crate-ruby, ro-crate-java
- HTML generators and preview tools

<https://www.researchobject.org/ro-crate/tools>

Tutorials:

<https://www.researchobject.org/ro-crate/tutorials>

Frequently Asked Questions

<https://www.researchobject.org/ro-crate/frequently-asked-questions>