



OpenREL

A machine-actionable rights and policy layer for Open Science

Why OpenREL? — what it is, what it solves, and how it works

EOSC Symposium- October 2026

Wim Hugo (KNAW-DANS, EUDAT) · Prodromos Tsiavos (OpenAIRE) · Melios Katsamakis (OpenAIRE)

THE PROBLEM

Rights in open science are written for people, not machines



A licence label is not enough

'CC BY' tells a human a lot, but it does not say who may access a sensitive dataset, under what identity, for how long, from where, or what must happen first or after the resource was used.



The real conditions live in prose

Access rules, ethics approvals, DMP obligations and institutional policy sit in repository settings and documents as free text, outside the licence.



So every decision needs a human

Adjudicating access by hand is slow and costly. Each request or integration re-implements the same logic, and automated agents risk acting outside terms they cannot read.

This becomes acute as EOSC moves toward federated data spaces and AI agents that request and process data on a researcher's behalf.

THE KEY IDEA

Machine-readable is not enough. We need machine-actionable.



Machine-readable

A system can parse it.

It can read a licence identifier from metadata, but still cannot decide whether a specific request should be allowed.



Machine-actionable

A system can evaluate it into a decision.

Actions, constraints, obligations and permissions are typed, identifier-based statements. An evaluator checks a request against them and returns an outcome.

OpenREL defines a deterministic evaluation convention on top of ODRL: every request resolves to exactly one outcome, always with a structured, auditable reason:



PERMIT

Allowed. Obligations such as attribution may still apply after access.



CONDITIONAL

Something must happen first — e.g. ethics approval or consent before access.



DENY

A prohibition applies or a mandatory condition cannot be met.

OpenREL in one sentence

An open, **machine-actionable** rights and policy framework that expresses licence terms, access conditions, data-subject requirements, ethical and legal constraints, and operational rules **together, in one structured representation** that systems can exchange and evaluate.

Built on open standards

W3C ODRL 2.2

the rights model it extends

JSON-LD, RDF TTL

the interchange syntax

DPV · SPDX · IDSC · GeoNames

and several more aligned vocabularies

IT IS

A horizontal, reusable policy layer that repositories, PID services, catalogues, data spaces, workflows and AI agents can all build on.

IT IS NOT

A repository, a licence registry, an identity provider, an access-control system or a legal reasoning engine. It evaluates conditions; the infrastructure enforces them.

Why profile and extend ODRL?

Generic ODRL can express both policies and licences, but does not by itself provide the domain-level typing needed to distinguish their role consistently across Open Science implementations, and to record specific properties that are not shared between policies:



odrl:Set

A CC BY licence



odrl:Set

Managed access to a sensitive dataset

Without a shared domain profile, their different roles may depend on implementation-specific interpretation. OpenREL profiles and extends ODRL for interoperable Open Science semantics — it does not replace it.

OpenREL adds explicit policy types

openrel:Licence

a rights licence — carries legal character, and protects property rights

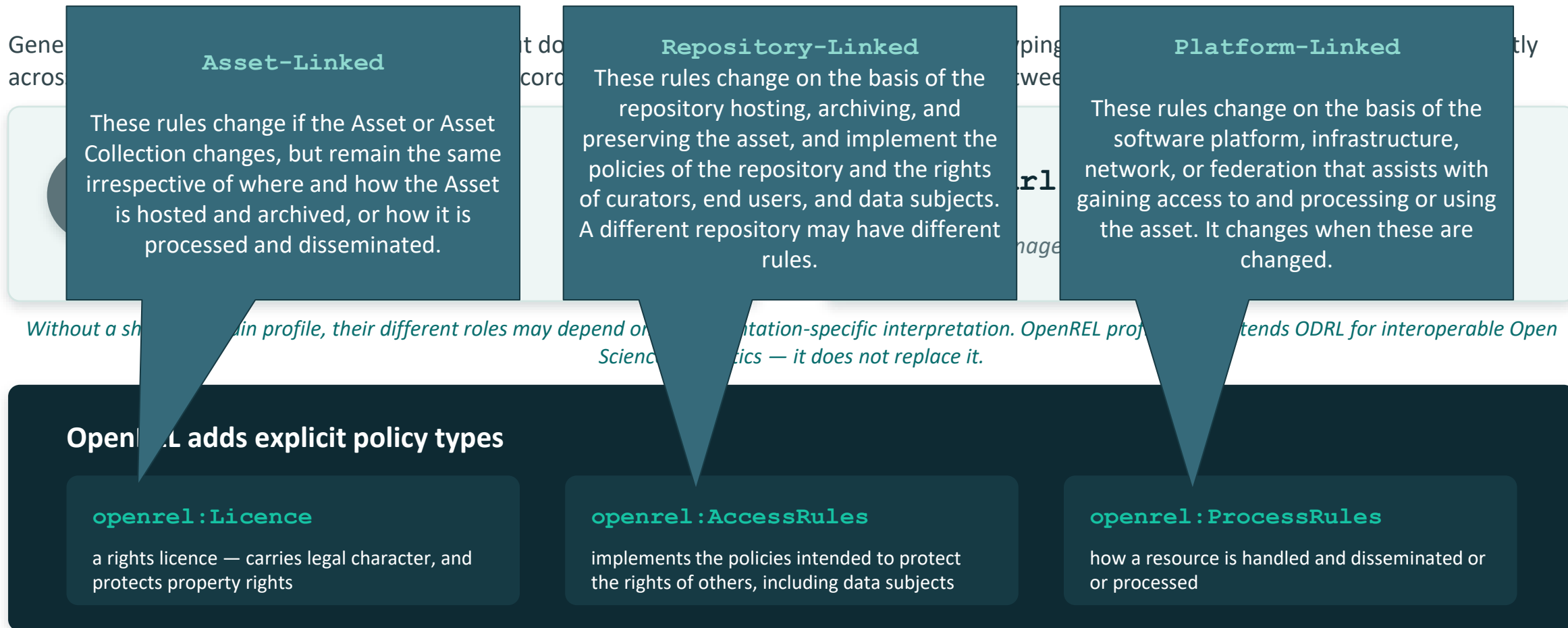
openrel:AccessRules

implements the policies intended to protect the rights of others, including data subjects

openrel:ProcessRules

how a resource is handled and disseminated or processed

Why profile and extend ODRL?



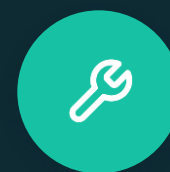
Knowledge Base and the Licence Wizard



Knowledge Base

Version-controlled, modular

- Vocabularies, Reusable templates and Named Constraints, referenceable by IRI
- Managed with an automated curation platform
- Designed to align with established vocabularies: DPV, SPDX, GeoNames, IDSC, Creative Commons
- Planned to be served under a W3ID persistent namespace



Licence Wizard

Author a policy without code

- Simple mode asks plain-language questions and assembles the policy
- Advanced mode exposes actions, constraints and agents for fine control
- Output is ODRL JSON-LD drawing on Knowledge Base terms by construction

Duplicate detection keeps the registry clean

These are current prototype and reference capabilities, demonstrated within EOSC Beyond — not production services.

Proposed Federated Infrastructure



OpenREL Vocabulary

Deployed in **one location**, accessible as a CURIE - `openrel:<concept>` and via API.



Federation of LFS APIs

The EOSC EDEN project created the code and example deployments for a federated set of **Licence Facade Servers**



SPDX Licence Registry

SPDX, operated by Linux Foundation, provides metadata descriptions for **mainstream** software and data licences

SPDX registers mainstream licences - but most custom licences and policies will not be accepted. Moreover, there may be substantial numbers of custom and specific agreements, collections, and the like. These are handled by LFS nodes.

An LFS node handles federation - synching with other LFS nodes, and submitting pull requests to SPDX as required. The local OpenREL registries host machine-actionable rules for all policies originating at the node.



OpenREL Registries

Registries of machine-actionable policies and licences that are private or custom, as well as encodings of mainstream licences



Individual Licence Texts

Many licences (e.g. CC) have human-readable and legal text available



What happens next



How to get involved

We invite the EOSC community to review the draft, contribute real policy and use cases, test implementation assumptions, and explore pilot adoption.



First Public Draft on Zenodo

The full specification — architecture, policy model, vocabulary, worked examples and conformance — targeted for publication in October 2026 under CC BY 4.0.



Toward Version 1.0

The normative v1.0 follows with EOSC Beyond Deliverable D14.3, incorporating community feedback and pilot experience.

Contact Details

Wim Hugo: wim.hugo@dans.knaw.nl

Melios Katsamakis: melioskatsamakis@gmail.com