Enhancing Machine-Actionable DMPs with Policy Entities Expressed in ODRL

Elli Papadopoulou[1,2], Maria Kontopidi[1], George Konstantinidis[3], Christopher Maidens[3], Dimosthenis Natsos[4], Georgios Kakaletris[5], Natalia Manola[2], Diamantis Tziotzios[5], and Evdokimos Konstantinidis[6]

[1] Athena Research Center, Athens, Greece
{elli.p, m.kontopidi}@athenarc.gr
[2] OpenAIRE AMKE, Athens, Greece
{elli.papadopoulou, natalia.manola}@openaire.eu
[3] University of Southampton, Southampton, UK
{G.Konstantinidis, C.Maidens}@soton.ac.uk
[4] Cyclopt, Thessaloniki, Greece
dnatsos@cyclopt.com
[5] CITE, Athens, Greece
{gkakas@cite-sa.gr, dtziotzios}@cite.gr
[6] Aristotle University of Thessaloniki, Thessaloniki, Greece
evdokimosk@gmail.com

Abstract. Machine-actionable Data Management Plans (maDMPs) structured according to the RDA DMP Common Standard (DCS) provide a machine-readable foundation for research data governance, yet the standard offers no mechanism to express governance intent as formally executable policy. The Open Digital Rights Language (ODRL) provides a W3C-standardised ontology for expressing permissions, prohibitions, and duties over digital assets, offering exactly this missing capability. This paper presents a systematic mapping of the RDA DCS to ODRL constructs, motivated by empirical evidence from the OSTrails project that policy compliance is among the most demanded yet least automatable dimensions of maDMP evaluation. The mapping is anchored to the `policies` entity introduced in the OSTrails maDMP Application Profile—proposed on the basis of surveys with 16 national funders and a review of 100 Horizon Europe DMPs—which maps directly to an ODRL Policy object. We make four contributions: (1) an entity-level mapping of 113 DCS fields to ODRL classes and properties, establishing that the DMP and, in a second instance of the same pattern, the License functions simultaneously as a governed Asset and a governing Policy; (2) a design for anchoring ODRL policy expressions in maDMPs via a `policies` entity at DMP root and dataset/distribution level, linking ARGOS-authored plans to RAISE Suite components through `related_identifiers`; (3) a coverage analysis and end-to-end worked example evaluating the mapping in practice, classifying all 113 fields by mapping degree and tracing a governance scenario from maDMP declaration to a runtime compliance decision; and (4) an identification of structural mapping boundaries, showing that Constraints and Rules require predefined rule-based inference from

DCS field values but can be formalised through ODRL's standardised, extensible ecosystem.

Keywords: Data Management Plans · machine-actionable DMPs · ODRL · data governance · FAIR · OSTrails · RAISE Suite · research data management · policy expression

1 Introduction

The RDA DMP Common Standard has established a machine-readable foundation for research data planning, but structured plans that cannot express policy are only half the solution. The DCS [1] provides a JSON-based application profile enabling DMPs to be processed programmatically across platforms, and subsequent work showed how structured plans reduce administrative burden and improve information quality [2]. The OSTrails project has further advanced automated assessment of maDMPs [3], establishing a community-grounded catalogue of evaluation metrics anchored to DCS schema fields. Yet the standard still offers no mechanism to express the governance intent that funders, data stewards, and institutions embed in their policies—and without that, machine-actionability stops at the wrong place.

The problem is structural. Even where a DMP declares governance intent in a structured, machine-readable field, the DCS provides no mechanism to express that intent as an executable rule—one that can be evaluated against an action actually performed on data at runtime. This gap predates and persists independently of any particular field's format: a systematic review of 100 Horizon Europe DMPs conducted within the OSTrails project [4] traced it to the DCS's original lack of any dedicated policy entity at all, motivating the `policies` entity now part of the OSTrails maDMP Application Profile (Section 2.3). But populating that entity does not by itself close the gap this paper addresses: a declared policy remains descriptive until it can be checked against runtime behaviour, and DCS—even extended—has no native mechanism for that. Surveys with 16 national funders across 13 countries confirmed that policy compliance is among the most demanded evaluation dimensions, yet remains the least automatable under the current DCS schema [4]. This gap is made more pressing by the evolving legal landscape: the General Data Protection Regulation (GDPR) and the EU AI Act impose concrete, enforceable obligations on data handling that current maDMP structures cannot represent in machine-executable form.

The urgency of this gap is underscored by the European Commission's revised Charter for Access to Research Infrastructures [14], which requires research infrastructures and their users to agree on a data management plan and mandates a research data management policy ensuring FAIR-compliant handling of research data. Meeting both requirements demands not only that governance intent be captured in a structured DMP, but that it be expressed in a form that systems can enforce. RAISE Suite—a Horizon Europe research infrastructure—is addressing the former. This paper addresses the latter.

The Open Digital Rights Language (ODRL) [6] offers a principled solution. As an OWL2 ontology for expressing permissions, prohibitions, and duties over digital assets, ODRL provides exactly the semantic layer that maDMPs lack: a formal vocabulary that bridges design-time declarations and runtime enforcement. A compliance engine can compare the actual sequence of actions performed on data against policies expressed in ODRL, returning machine-readable results without human intervention. ODRL has been applied to express licensing in scholarly publishing and FAIR data contexts, but its application to the structured governance of research data management plans has not been systematically formalised.

This paper addresses that gap. We present a systematic mapping of the RDA DCS to ODRL constructs, motivated by the OSTrails project's community evidence and grounded in an operational implementation through the RAISE Suite research infrastructure [17]. Our work is directly informed by the OSTrails maDMP Application Profile under co-development with the RDA DMP Common Standard Working Group, which introduces a dedicated `policies` entity—a structured extension precisely motivated by the absence of policy expression in the current DCS [4].

We make four contributions:

1. A **systematic entity-level mapping** of 113 DCS fields to ODRL classes and properties, revealing the dual identity of the DMP and, in a second instance of the same structural pattern, the License, as simultaneously an ODRL AssetCollection/Asset and a Policy, and establishing which DCS constructs map fully, partially, or require inference to align with ODRL.
2. A **design for anchoring ODRL policy expressions in maDMPs** via a `policies` entity at both DMP root and dataset/distribution level, linking ARGOS-authored plans to RAISE Suite components through `related_identifiers`, and supported by an initial policy expression generation mechanism developed at the University of Southampton.
3. A **coverage analysis and worked demonstration** of the mapping (Sections 3.4 and 3.5): a classification of all 113 fields by mapping degree, and an end-to-end trace of a single governance scenario from maDMP declaration through generated ODRL policy and a runtime event log to a compliance decision.
4. An identification of **structural mapping boundaries**, showing where DCS extensions are required to support ODRL's Constraint and Rule layers, and outlining the enrichment needed in the RAISE Suite logging infrastructure to support runtime ODRL action capture—with direct implications for the OSTrails maDMP Application Profile.

2 Background and Related Work

2.1 Machine-Actionable DMPs and the RDA DCS

The RDA DMP Common Standard [1] defines a JSON application profile covering datasets, distributions, licences, contributors, funding, and hosting ar-

rangements. Miksa et al. [2] articulated ten principles for maDMPs, treating DMPs as active, machine-processable governance documents. Papadopoulou et al. [5] demonstrated that DCS-conformant maDMP content can be published as linked open data within the OpenAIRE Research Graph via ARGOS, establishing DMPs as first-class research objects cross-referenced against live knowledge graph records. The companion paper in this volume [3] presents a catalogue of 94 evaluation metrics for maDMPs, several concerning policy and compliance dimensions that the present work formalises at the level of machine-executable rules.

2.2 Machine Processable Policy Frameworks

We considered several frameworks before selecting ODRL. Access-control frameworks—XACML [16], Rego/OPA [18], Cedar [19]—address authorisation but lack the stateful Duty concept governance requires. Legal and privacy-specific languages—LegalRuleML [20], Fides Lang [21], SPECIAL [22]—each cover part of the problem but lack a general-purpose, standardised model with an actively maintained ecosystem. ODRL [6] uniquely combines W3C Recommendation status, deontic logic (permissions, prohibitions, conditional Duties), and the extensibility needed for evolving legislation such as GDPR and the EU AI Act.

2.3 Policy Gaps in Current maDMP Practice

The OSTrails D4.2 Horizon Europe Report [4] provides the empirical foundation for this work. As part of its desk research, D4.2 manually reviewed 100 Horizon Europe DMPs—the standard template required of all Horizon Europe-funded projects—and traced a structural gap to its source: the DCS had no dedicated entity through which a DMP could declare which policy applied, under what jurisdiction, or with what enforceable terms, so policy-relevant decisions are consistently expressed in narrative prose rather than structured, machine-parseable declarations. In response, D4.2 proposed a dedicated `policies` entity as a nested extension of the DMP root, with structured fields for title, type, jurisdiction, and qualified references aligned with FAIR principles [15]. This entity, now part of the preliminary OSTrails maDMP Application Profile, closes the *structural* gap: a DMP can now declare that a policy exists and identify it.

It does not close the *executability* gap this paper addresses. A populated `policies` entity is still descriptive: it names a policy but provides no mechanism to check whether an action performed on data actually complied with it—that requires expressing the policy's permissions, prohibitions, and duties in a form a compliance engine can evaluate against a runtime event log, which is precisely what ODRL provides and DCS does not. This paper takes D4.2's `policies` entity as its starting point and anchor (Section 3.2), and goes one step further: our mapping identifies which *additional* DCS fields are required, beyond the `policies` entity itself, to express policy at the granularity ODRL demands—e.g. fields capturing spatial and temporal constraints, or a controlled relation type such as `isEnforcedBy` linking a declared policy to its enforcing

infrastructure (Sections 3.2 and 5). Surveys with 16 national funders across 13 countries confirmed that policy compliance is among the most demanded evaluation dimensions yet remains the least automatable under the current DCS schema, extending the quality dimension taxonomy of Arnhold et al. [8], which identified policy compliance as a core dimension but stopped short of a machine-executable formalisation.

2.4 ODRL: From Descriptive to Enforceable Policies

The Open Digital Rights Language [6] is a W3C Recommendation providing a vocabulary for expressing permissions, prohibitions, and duties over digital assets, with an action vocabulary [7] of 50 standardised operations. At its core, an ODRL Policy is a set of Rules governing one or more Assets. Each Rule is a Permission, Prohibition, or Duty (obligation) associated with an Action performed by a Party, optionally refined by Constraints. This structure—Policy → Rule → Asset—provides the formal backbone for the maDMP mapping presented in Section 3. ODRL has been adopted across data-sharing domains: the FAIR-IMPACT project demonstrated its application to machine-actionable research data access conditions [13], and European Data Spaces initiatives have adopted it as the standard for expressing usage control policies [11,12].

A critical limitation is well-documented: ODRL is descriptive but not inherently enforceable. Cimmino et al. [11] introduced ODRE, integrating ODRL's ontology terms with behaviour specification languages to produce an open-source enforcement algorithm; Salas et al. [9] provided formal query-answering semantics for ODRL evaluation and comparison; and Petrova-Antonova et al. [10] demonstrated knowledge-graph-driven policy enforcement in a distributed urban dataspace—work by the University of Southampton team informing the RAISE Suite implementation described here.

To our knowledge, no prior work has applied ODRL to the maDMP domain or proposed a systematic mapping between the RDA DCS and ODRL constructs.

2.5 The RAISE Suite

RAISE Suite is a Horizon Europe research infrastructure project extending EOSC-RAISE with maDMP-guided data collection and streaming. Data are collected from research sources and streamed to RAISE Streaming Nodes (RSNs), curated and progressively assembled into versioned datasets; completed datasets are registered through the RAISE Central Hub in RAISE Certified Nodes (RCNs) [17]. Its architecture includes the ARGOS-based Machine Actionable DMP component, the DMP Compliance Report engine, collecting and streaming data to RSNs, (pseudo)anonymisation engines, the Data Collection PID resolver, and the RAISE Suite portal; at the RSN level, brokering and curation, dataset staging and versioning, quality/bias monitoring, and RCN registration operationalise the declared data-management intent. Building on prior work on ODRL formal semantics [9] and knowledge-graph-driven policy enforcement [10], the University of Southampton's contribution includes an initial implementation generating

ODRL policy expressions that reference both maDMP fields and RAISE Suite components. Runtime logging is intended to associate observed actions with the applicable dataset, maDMP, and policy versions, providing the evidence needed for compliance evaluation.

3 Mapping the RDA DCS to ODRL

The mapping was developed through an iterative, expert-driven process involving the RAISE Suite maDMP Task Force—technical partners, pilot representatives, and domain experts in maDMPs, ODRL, and research data management—who performed a bidirectional conceptual mapping between the DCS and ODRL at entity and field level, starting from the RDA DCS, the OSTrails AP, and the ODRL Information Model and Vocabulary. Resulting mappings were iteratively reviewed against RAISE Suite's architectural requirements (policy generation, runtime compliance evaluation, enforcement); where no direct correspondence existed, discussions identified the need for predefined inference rules, DCS extensions, or domain-specific ODRL concepts. The final mapping classifies all 113 DCS fields as fully mappable, partially mappable, or requiring inference, per the criteria below.

The mapping was conducted in two stages: the first established entity- and field-level correspondences between the DCS schema and ODRL constructs (drawing on [1,4,6,7]); the second, reported in Section 4, grounds the mapping in concrete governance scenarios using ODRL Actions linked to RAISE Suite components. The full field-level mapping of 113 DCS fields is available as a supplementary artefact; this section presents the entity-level findings and the governance-relevant subset.

ODRL's core model comprises six constructs: *Policy* (a set of Rules governing an Asset); *Asset* (a resource subject to a Rule); *Party* (an entity undertaking a role); *Action* (an operation on an Asset); *Rule* (covering Permissions, Prohibitions, and Duties); and *Constraint* (a boolean expression refining a Rule). Mappability is assessed across three degrees: *full*, where a clean, direct correspondence exists (e.g. `contributor` → `odrl:Party`); *partial*, where a correspondence exists but requires inference or extension (e.g. `dataset.rights` → `odrl:Policy`, which is underspecified in the DCS); and *requires inference*, where no direct DCS field expresses the ODRL construct but it can be derived from field values (e.g. an ODRL Constraint from `host/geo_location`).

3.1 Entity-Level Mapping

Table 1 presents the entity-level mapping.

The most consequential structural finding is the **DMP's dual identity**. More precisely, the DMP is best understood as an expression of data management *policy*—requiring ODRL extension for complete expressibility—that governs an AssetCollection of one or more datasets, each of which requires its own asset-level rights expression. The DMP document may itself be included in that

Table 1. Entity-level mapping between DCS entities and ODRL constructs.

ODRL Construct	DCS Entities	Degree	Notes
Policy	DMP	Partial	Dual identity
	License	Partial	Dual identity
	Security and Privacy	Partial	—
Asset	Dataset, Distribution, Host, Metadata, Technical Resource	Full	—
	DMP	Partial	Dual identity
	License	Partial	Dual identity
	Project	Partial	—
	Cost	Partial	Via `pay` action
	Alternate Identifier	Full	—
	Related Identifier	Partial	Vocab. narrower
Party	Contact, Contributor, Affiliation	Full	Maps to `assignee`, `informedParty`
	Funding	Partial	—
Action	Cost (`pay`); else field-level within Asset entities	Partial	Implied
Constraint	(none)	Req. inference	Derived from field values
Rule (Perm., Prohib., Duty)	(none)	Req. inference	Implicit

AssetCollection, since nothing in the ODRL model prevents a Policy from referencing itself as an Asset—a valid modelling choice where the DMP's own access and reuse conditions need to be governed. The OSTrails AP `policies` entity [4] surfaces this structure by introducing a nested Policy object within the DMP root. The practical implication is significant: a DMP-consuming system must decide whether to treat the DMP as a data container to be processed, a governance instrument to be enforced, or both simultaneously.

License exhibits the same duality: it can be modelled either as the Asset being governed (a licensed digital object) or as the Policy expressing the terms of governance itself, and the choice is not resolved by the DCS schema. Project, Funding, Cost, and Security and Privacy map only partially to ODRL for distinct reasons: Project extends the Asset construct beyond its intended scope; Funding's status and Cost lack any ODRL-native representation; Security and Privacy is ambiguous between Policy and an annotation on Dataset/Distribution. Action's only direct entity-level source is Cost, via the core `pay` action; every other Action trigger operates at the field level within entities already classified as Asset above, and is detailed in Table 4.

3.2 A Design for Anchoring ODRL Policy Expressions in maDMPs

Anchoring ODRL policy expressions inside a maDMP requires resolving two general problems, independent of any specific infrastructure: at what structural granularity policies are declared (plan-wide, per-asset, or both), and how a declared policy is linked to whatever component ultimately enforces it, since ODRL itself is descriptive and enforcement is necessarily external to the language. We propose a two-level answer to the first: policies are anchored via the OSTrails AP `policies` entity at both the DMP root (funder mandates, institutional policies, jurisdictional constraints) and the dataset/distribution level (access conditions, embargo rules, licence terms, sensitive data handling). To illustrate: a DMP covering an openly-licensed dataset and one restricted under GDPR carries a root-level `policies` entry for the applicable funder and GDPR policies, while each dataset carries its own entry expressing the specific Permission (distribute under CC BY) or Prohibition (distribute without anonymisation).

For the second problem, we use `related_identifiers` and a proposed `isEnforcedBy` relation type (an extension to the DCS controlled vocabulary, not yet standardised) connecting a policy reference to the identifier of whatever system executes it—a mechanism any DMP-consuming system needs, not only RAISE Suite. In RAISE Suite, root-level `related_identifiers` point to the ARGOS-authored plan identifier and RAISE Suite component identifiers; dataset-level entries point to the specific workflow operationalising the declared intent (staging and versioning, anonymisation, dataset registration and storage). A dataset policy entry expressing a Duty to anonymise before distribution carries an `isEnforcedBy` entry pointing to the anonymisation service.

We instantiate this design with an initial implementation, developed at the University of Southampton, that takes maDMP field values as input and produces ODRL-serialised policy objects referencing the corresponding infrastructure components [10,9]. Two further components extend this instantiation but are not yet built: an ODRL evaluation engine based on the formal semantics of Salas et al. [9], and a domain-specific ODRL vocabulary extending the core action set to cover maDMP-specific governance concepts. RAISE Suite's existing event logging infrastructure provides the runtime record needed to compare declared intent against observed execution—a design-time to runtime bridge any implementation of this pattern requires.

Policy evolution is handled through versioned supersession rather than in-place replacement: a policy-relevant maDMP change creates a new maDMP and ODRL Policy revision with an explicit effective point. Dataset versions produced before that point remain linked to the previous policy; a change affecting processing starts a new dataset version, and runtime events are evaluated against the policy active when each action occurred. Retroactive effects must be explicitly declared.

The `policies` Entity and ODRL The OSTrails AP `policies` entity maps naturally to an ODRL Policy object. Table 2 details the field-level correspondences.

Table 2. Mapping between the OSTrails AP `policies` entity and ODRL.

OSTrails AP Field	ODRL Construct	Notes
`policies/title`	`odrl:Policy` + `dc:title`	Direct mapping
`policies/policy_type`	`odrl:Policy/@type`	Agreement, Offer, Privacy, Set
`policies/jurisdiction`	`odrl:Constraint` (`odrl:spatial`)	Requires Constraint layer
`policies/related_identifiers`	`odrl:Policy/uid`; `dct:relation`	PID-based policy linkage
`policies/description`	`dc:description`	Informational only

The `policies/jurisdiction` field is the single entry that *requires* ODRL Constraints to be expressed correctly. A value such as "European Union" maps to `odrl:Constraint` with `odrl:leftOperand = odrl:spatial` and `odrl:rightOperand = dbpedia:European_Union`. This is precisely the kind of conditional refinement that makes ODRL policy expressions machine-executable rather than merely machine-readable—and one that requires the inference step described in Section 5 to be formalised.

3.3 Selected Field-Level Mappings

Of the 113 DCS fields examined, the majority map to ODRL Assets (resource description fields), Parties (contributor fields), or Actions (lifecycle fields). Table 3 presents the governance-relevant subset; the full mapping is available as a supplementary artefact.

Table 3. Selected governance-relevant field-level mappings between DCS properties and ODRL. Full mapping available as supplementary artefact.

DCS Field	ODRL Construct	ODRL Property	Notes
`dmp/dmp_id/identifier`	AssetCollection	`uid`	DMP identity
`dataset/sensitive_data`	Action	`use/transfer`	Anonymisation duty
`dataset/preservation_statement`	Action	`use/archive`	Archive obligation
`host/geo_location`	Action	`use/transfer`	Residency constraint
`distribution/data_access`	Policy	—	Access condition
`distribution/license/license_ref`	Policy	—	Licence as ODRL Policy
`related_identifiers/relation_type`	Relation	`partOf`; `hasPolicy`	Inter-resource links

Three observations about the action-level mapping warrant attention.

Actions as governance triggers. Several DCS fields do not themselves express actions but *trigger* them when specific values are present. When `dataset/sensitive_data = yes`, the RAISE Suite is expected to invoke its anonymisation pipeline before any distribution action—an `odrl:Duty` of `odrl:anonymize` attached to the `odrl:Permission` to distribute. Formalising this trigger–duty relationship is what ODRL makes possible but the current DCS cannot express. The generated ODRL policy expression captures this Duty explicitly, and the logging infrastructure records whether anonymisation was executed prior to distribution, verifying

runtime compliance against design-time intent. The policy expression is created at DMP initialisation and updated alongside DMP changes; evaluation runs during lifecycle execution against a "state of the world" recorded via structured, timestamped logging, supporting both access-control and monitoring scenarios.

Embargo as a temporal Constraint. The Distribute scenario illustrates how ODRL Constraints operationalise time-bound governance. `license/start_date` implies a temporal boundary before which distribution is prohibited, corresponding to `odrl:Constraint` with `odrl:leftOperand = odrl:dateTime` and `odrl:operator = odrl:gteq`; the Permission activates once satisfied. This is governance intent that ODRL can formalise only once the inference step described in Section 5 is applied.

The DMP as its own governed object. The Read scenario surfaces a recursive structure: the maDMP is simultaneously the instrument governing its datasets *and* a digital object subject to governance in its own right—aligning with the D4.2 proposal for a `dmp_license` field [4] and the companion paper's analogy to FAIR assessment [3]. In the two-level design, the DMP-level `policies` entry governing access to the plan is a distinct ODRL Policy object, separate from the dataset-level policies it contains.

3.4 Coverage of the Mapping

Classifying all 113 DCS fields against the criteria defined at the start of this section yields 71 fields (62.8%) mapping fully, 40 (35.4%) partially, and 2 (1.8%) requiring inference. Full mappings dominate the Asset and Party layers, where the majority of DCS's descriptive metadata fields (titles, descriptions, identifiers) correspond directly to informational ODRL/Dublin Core properties; Partial mappings cluster around lifecycle and trigger fields that imply an Action without declaring one, and around entities with no clean single-construct target (License, Project, Funding, Cost, Security and Privacy). Requires-inference cases are numerically small (2 of 113) but structurally significant: both are the sole DCS-side sources for the Constraint construct, which Section 5 identifies as a genuine gap in the current schema.

3.5 A Worked Example: Anonymise-Before-Distribute

We instantiate the Anonymize scenario from Table 4 end-to-end, tracing a single dataset from its maDMP declaration through generated ODRL policy, runtime execution, and compliance evaluation.

Step 1—maDMP fragment (ARGOS). A dataset with sensitive personal data, to be distributed under CC BY only after anonymisation:

```
{ "dataset": [{ "dataset_id": {"identifier":"10.5281/zenodo.99001","type":"
    DOI"},
    "personal_data": "yes", "sensitive_data": "yes",
    "distribution": [{ "data_access": "open",
      "license": [{"license_ref":"https://creativecommons.org/licenses/by/4.
          0/","start_date":"2026-09-01"}] }],
    "policies": [{ "title": "GDPR Article 89 Research Exemption", "
        policy_type": "regulatory",
```

```
      "jurisdiction": "European Union",
      "related_identifiers": [{"identifier":"raise:anonymisation-service","
          identifierType":"URI","relationType":"isEnforcedBy"}] }]
} ] }
```

Step 2—generated ODRL policy (translation layer). `sensitive_data=yes` triggers a Duty to anonymise, precondition to the Permission to distribute; `jurisdiction` and `start_date` compile into Constraints (Section 5):

```
{ "@context": "http://www.w3.org/ns/odrl.jsonld", "@type": "Policy", "uid":
    "urn:raise:policy:zenodo.99001",
  "permission": [{ "target": "urn:raise:asset:zenodo.99001", "action": "
      distribute",
    "constraint": [
      {"leftOperand":"dateTime","operator":"gteq","rightOperand":"2026-09-01
          "},
      {"leftOperand":"spatial","operator":"eq","rightOperand":"dbpedia:
          European_Union"}],
    "duty": [{"action":"anonymize","target":"urn:raise:asset:zenodo.99001","
        consequence":[]}]
} ] }
```

Steps 3–4—runtime event log and compliance decision. RSN nodes emit events consumed by the Compliance Engine; the anonymize Duty (08:12:03) precedes the distribute Permission (08:14:51), and both Constraints hold at execution time:

```
[ {"event":"anonymize","node":"anonymisation-service","timestamp":"2026-09-0
    1T08:12:03Z","status":"completed"},
  {"event":"distribute","node":"storage-service","timestamp":"2026-09-01T08:
      14:51Z","status":"completed"} ]
{ "policy": "urn:raise:policy:zenodo.99001", "decision": "COMPLIANT",
  "checked": ["duty:anonymize precedes permission:distribute","constraint:
      dateTime","constraint:spatial"] }
```

Had the distribute event been logged *before* the anonymize event, or had it occurred before 2026-09-01, the Compliance Engine would return `NON_COMPLIANT` with the specific violated Duty or Constraint named—this is the enforcement capability that a purely descriptive DCS field cannot provide. If the data owner later requires anonymisation of an additional field X, the revised maDMP generates a new ODRL Policy revision. At its effective point, the current dataset version closes and a new one begins with X anonymised; earlier data remain linked to the policy under which they were produced.

4 ODRL Actions as Data Governance Triggers

Deriving ODRL Actions from a maDMP requires a general three-stage pipeline, independent of any specific infrastructure: field values must be interpreted as triggers for specific Actions; declared Actions must be linked to whatever component executes or enforces them; and the actual sequence of executed actions must be logged and compared against what was declared. The scenarios below instantiate that pipeline using the two-level anchoring design of Section 3.2. In RAISE Suite, this resolves concretely: a researcher authors a maDMP in ARGOS, exported as DCS-conformant JSON; the translation layer reads field values and generates ODRL policy expressions referencing the relevant components; the Compliance Engine evaluates whether declared Actions are permitted,

prohibited, or subject to a Duty; and the logging infrastructure records which actions are actually executed, enabling compliance verification. This is an initial, illustrative mapping rather than an exhaustive coverage of the ODRL action vocabulary. Table 4 presents ten scenarios—one per ODRL action in the initial governance vocabulary—each one representative case drawn from a fuller internal mapping of 22 governance scenarios, available as a supplementary artefact alongside the field-level mapping described in Section 3.

Table 4. Initial ODRL action scenarios for maDMP governance. Each scenario shows the triggering DCS fields and an illustrative RAISE Suite component path through which the corresponding action may be operationalised.

Action	Scenario	DCS Fields	Illustrative RAISE Suite path
Move	Secure storage for GDPR residency	`dataset.personal_data`; `dataset.sensitive_data`; `security_and_privacy.description`; `host.storage_type`; `host.geo_location`; `policies.jurisdiction`	Collection→Anon.→Broker.→Staging→Reg.
Read	Assess DMP funder compliance	`dmp_license.access_rights`; `dmp_license.license_ref`; `dmp.dmp_id`	ARGOS→Compl. Eng.→Portal
Distribute	Release dataset under CC BY post-embargo	`license.start_date`; `license.license_ref`; `distribution.data_access`	Portal→Policy Eval.→RCN→Portal
Modify	Apply a policy-relevant revision	`dmp.modified`; `dmp_status`; `funding.funding_status`	ARGOS→Policy Gen.→Staging→Compl. Eng.
Transform	Convert dataset to open format	`distribution.format`; `dataset.preservation_statement`; `host.storage_type`	Staging & Transform.
Anonymize	Anonymise before licensed sharing	`dataset.personal_data`; `dataset.sensitive_data`; `distribution.data_access`; `license.license_ref`	Anonymisation Engine
Archive	Archive per preservation statement	`dataset.preservation_statement`; `host.certified_with`; `dataset.dataset_status`	Staging→Reg.
Reproduce	Reproduce via scheduled backup	`host.backup_frequency`; `host.backup_type`; `host.availability`	Storage & Backup
Transfer	Transfer restricted dataset to committee	`distribution.data_access`; `distribution.restriction_explanation`; `contributor.role`	Portal→Policy Eval.→RCN
Aggregate	Aggregate metadata into one DMP record	`dmp.dmp_id`; `dataset.dataset_id`; `funding.grant_id`	Metadata Comp.→PID Resolver→Portal

5 Structural Mapping Boundaries and Implications

This section reports three structural observations at the boundary of our initial DCS–ODRL mapping, with concrete implications for the OSTrails Application Profile and the RDA DMP Common Standard Working Group. These are boundaries of the current DCS schema requiring inference, extension, or explicit modelling choices—not definitive limits of what is mappable in principle.

These boundaries are not artefacts of RAISE Suite's implementation choices, nor specific to the current DCS version. They follow from a structural mismatch

between what any DCS-based AP is designed to do and what ODRL requires: a DCS entity records declared intent as descriptive metadata, while an ODRL construct must be executable—a Rule checkable against a logged event, a Constraint evaluable as true or false. Any system aligning a DCS-based AP with ODRL will encounter the same three boundaries: dual identity (entities that are simultaneously governed objects and governance documents), governance intent implicit in field values rather than asserted as a Rule (requires inference), and Constraints with no natural DCS field to serialise into (also requires inference). We report these using RAISE Suite as the implementation context in which they were identified, not as findings specific to it.

Constraints require predefined rule-based inference. The DCS has no construct that directly serialises an ODRL Constraint expression. However, several fields carry values from which Constraints can be derived through known, predefined mappings: `host/geo_location` implies a spatial constraint (`odrl:spatial`); `license/start_date` implies a temporal one (`odrl:dateTime`); `policies/jurisdiction` implies a jurisdictional one. The translation layer formalises a defined subset of these patterns; extending coverage to the full range of governance-relevant DCS fields, potentially via a more flexible reasoning model, is a key direction for future work.

Rules are implicit, not declared—and ODRL is the right formalisation layer. The DCS carries governance intent implicitly: `distribution/data_access = restricted` implies a Prohibition, but no Rule object is ever asserted in the schema. Introducing ODRL is justified by what it uniquely provides: a standardised, actively developing tool ecosystem—evaluation engines [9], enforcement frameworks [11], reasoning infrastructure [10]—that would otherwise need to be built from scratch, and extensibility to integrate evolving legal frameworks (GDPR, the EU AI Act) and custom negotiation requirements. The translation layer performs this inference step, taking field values as input and producing ODRL-serialised Rule objects; formalising the complete field-value-to-Rule-type mapping is the most substantial remaining technical contribution, and a natural candidate for OSTrails AP standardisation.

The dual identity requires an explicit serialisation strategy—and an extensibility choice. As established in Section 3.2, the DMP functions simultaneously as an ODRL AssetCollection and a governance instrument containing Policy objects. License exhibits the identical tension (Table 1), suggesting this is a recurring pattern, not an idiosyncrasy of the DMP. This duality requires an explicit serialisation choice—policies nested within the DMP structure, or ODRL Policy objects published separately and referenced via `related_identifiers`—and raises a deeper question about DCS extension: extending the core model, as the OSTrails AP does with the `policies` entity, preserves compatibility with the broader DCS tool ecosystem, whereas custom extensions departing from the core model gain expressivity at the cost of standard tooling support. Defining a serialisation convention supporting both nested and referenced policies is a concrete next step for the RDA DMP Common Standard Working Group.

6 Conclusion

This paper has presented, to our knowledge, the first systematic mapping between the RDA DMP Common Standard and the ODRL policy expression vocabulary, motivated by empirical evidence from the OSTrails project that policy compliance is among the most demanded yet least automatable dimensions of maDMP evaluation [4], and grounded in the operational context of the RAISE Suite research infrastructure.

Four contributions were made. First, an entity- and field-level mapping of 113 DCS fields to ODRL constructs shows that the majority of DCS entities translate naturally to ODRL's Asset, Party, Action, and Relation layers, while establishing that the DMP—and, in a second instance of the same pattern, the License—is simultaneously a governed Asset and a governing Policy, an ambiguity the DCS does not resolve on its own. Second, a design anchors ODRL policy expressions at two structural levels (DMP root and dataset/distribution), using the OSTrails AP `policies` entity as anchor, `related_identifiers` as the wiring to RAISE Suite components, and an initial policy-generation mechanism developed at the University of Southampton, with policy and rights changes handled through versioned supersession and explicit effective points. Third, a coverage analysis and worked example evaluate the mapping in practice: all 113 fields are classified by degree (71 full, 40 partial, 2 requiring inference), and a governance scenario is traced end-to-end from maDMP declaration to a runtime compliance decision. Fourth, three structural boundaries are identified: Constraints and Rules both require predefined rule-based inference from DCS field values, and the dual identity exhibited by both the DMP and the License requires an explicit serialisation strategy within the OSTrails Application Profile.

The most consequential of these observations is also the most tractable: Constraints and Rules are present in the maDMP domain but implicit, and the translation layer demonstrates that they can be surfaced through inference from existing DCS field values—extending this inference and formalising it within the OSTrails AP and RAISE Suite's domain-specific ODRL vocabulary is the natural next step. The mapping itself was developed through iterative expert review within the RAISE Suite maDMP Task Force and refined against RAISE Suite's implementation requirements; future work will complement this with broader empirical evaluation across real machine-actionable DMPs and additional governance scenarios, negotiation of third-party policies, treatment of explicitly retroactive changes, and richer constraint-inference mechanisms.

The RDA DMP Common Standard has established a machine-readable foundation for research data planning. Making that foundation policy-aware and runtime-enforceable is the other half of the solution. This paper takes a first step in that direction.

Acknowledgments. This work has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101130187 (OSTrails—Open Science Plan-Track-Assess Pathways) and grant agreement No. 101188337 (RAISE Suite).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Miksa, T., Walk, P., Neish, P.: RDA DMP Common Standard for Machine-Actionable Data Management Plans (2020). https://doi.org/10.15497/rda00039
2. Miksa, T., Simms, S., Mietchen, D., Jones, S.: Ten principles for machine-actionable data management plans. PLOS Comput. Biol. 15(3), e1006750 (2019). https://doi.org/10.1371/journal.pcbi.1006750
3. Miksa, T., Tabima, A., Kontopidi, M., Papadopoulou, E.: Community Metrics and Automated Assessment of Machine-Actionable Data Management Plans. In: Proceedings of TPDL 2026 (2026)
4. Manola, N., Papadopoulou, E., Kontopidi, M., Grypari, I., Spichtinger, D., Kakaletris, G., Tziotzios, D.: D4.2: Horizon Europe Report. Tech. rep., OSTrails Project (2025). https://doi.org/10.5281/zenodo.16753141
5. Papadopoulou, E., Bardi, A., Kakaletris, G., Tziotzios, D., Manghi, P., Manola, N.: Data management plans as linked open data: exploiting ARGOS FAIR and machine-actionable outputs in the OpenAIRE Research Graph. J. Biomed. Semant. 14, 17 (2023). https://doi.org/10.1186/s13326-023-00297-5
6. Iannella, R., Villata, S. (eds.): ODRL Information Model 2.2. W3C Recommendation (2018). https://www.w3.org/TR/odrl-model/
7. Iannella, R., Villata, S. (eds.): ODRL Vocabulary & Expression 2.2. W3C Recommendation (2018). https://www.w3.org/TR/odrl-vocab/
8. Arnhold, L., Miksa, T., Staudinger, M.: Quality dimensions and evaluation framework for machine-actionable data management plans. ACM J. Data Inf. Qual. 17(4), Article 29 (2025). https://doi.org/10.1145/3776555
9. Salas, J.O., Pareti, P., Yumusak, S., Gheisari, S., Ibáñez, L.D., Konstantinidis, G.: Evaluation and comparison semantics for ODRL. arXiv:2509.05139 (2025)
10. Petrova-Antonova, D., Pareti, P., Tomov, P., Yumusak, S., Maidens, C., Jiang, S., Konstantinidis, G., Roman, D.: Knowledge graph-driven policy enforcement in urban dataspace. In: Proceedings of ESWC 2025 (2025)
11. Cimmino, A., Cano-Benito, J., García-Castro, R.: Open Digital Rights Enforcement Framework (ODRE): from descriptive to enforceable policies. arXiv:2409.17602 (2024)
12. Dam, T., Krimbacher, A.: Policy patterns for usage control in data spaces. In: Sem4Tra Workshop, ISWC 2023 (2023)
13. FAIR-IMPACT Project: Advancing access interoperability with ODRL (2024). https://fair-impact.eu/use-cases/advancing-access-interoperability-odrl
14. European Commission: Revised Charter for Access to Research Infrastructures (2024). https://research-and-innovation.ec.europa.eu/news/all-research-and-innovation-news/revised-charter-access-research-infrastructures-foster-open-science-innovation-and-research-security-2024-11-27_en
15. Wilkinson, M.D., et al.: The FAIR guiding principles for scientific data management and stewardship. Sci. Data 3, 160018 (2016). https://doi.org/10.1038/sdata.2016.18
16. A Complete Guide to XACML and How It Helps You Protect Your Data (2024). https://heimdalsecurity.com/blog/the-complete-guide-to-xacml/
17. Konstantinidis, E., Epelde, G., Natsos, D., Petsani, D., Valtopoulou, A., Papadopoulou, E., Hernandez, M., Bamidis, D., Sarigiannidis, P., Symeonidis, A., Chatzigeorgiou, A., Bamidis, P.: FAIR Data Management from Collection to Exploitation: The RAISE Suite Project. In: Proceedings of the 15th International Conference on Data Science, Technology and Applications – Volume 2, pp. 969–976. ISBN 978-989-758-854-9, ISSN 2184-285X (2026)
18. Open Policy Agent: Policy Language (2024). https://www.openpolicyagent.org/docs/policy-language
19. Amazon Web Services: Cedar overview – AWS Prescriptive Guidance (2024). https://docs.aws.amazon.com/prescriptive-guidance/latest/saas-multitenant-api-access-authorization/cedar.html
20. Athan, T., Governatori, G., Palmirani, M., Paschke, A., Wyner, A.: OASIS LegalRuleML. In: Proceedings of the 14th International Conference on Artificial Intelligence and Law (ICAIL 2013) (2014)
21. Ethyca: What is Fideslang? – Fides Language (2024). https://ethyca.github.io/fideslang/
22. Bonatti, P., Kirrane, S., Polleres, A., Wenning, R.: Transparent Personal Data Processing: The Road Ahead. In: Proceedings of SPECIAL Usage Policy Language, W3C Community Group (2019). https://ai.wu.ac.at/policies/policylanguage/