Beyond Permit and Prohibit: Provenance and Conflict Resolution Extensions for Generalising ODRL for Machine Readable Privacy and Data Management Policy Representation

- **Joseph K Iype, Senior Data Scientist, HSBC**

Abstract

ODRL 2.2 offers a strong foundation for expressing policy rules by specifying who can do what with which assets. As it is increasingly used beyond Digital Rights Management (DRM) across regulatory compliance, cross-border operations, and supply chains - key limitations are becoming clear. In particular, ODRL's basic three-outcome conflict handling (permit, prohibit, invalid) and its lack of built-in provenance make it harder to manage complex, real-world policy decisions.

Building on W3C Community Group work and academic consensus, this paper recommends enhancing ODRL with more sophisticated conflict combining approaches aligned to legal principles and XACML 3.0 patterns, alongside a first-class provenance layer that links rules to their sources and derivation history. It proposes two lightweight, backward-compatible W3C profiles - odrl-prov and odrl-cr - aligned with PROV-O and LegalRuleML, and sets out a roadmap to incorporate them into ODRL 3.0 to support stronger auditability and defeasible reasoning for modern policy engines.

Introduction

The W3C Open Digital Rights Language (ODRL) 2.2 has achieved substantial success as an interoperable vocabulary for expressing machine-readable policies. Originally rooted in digital rights management, ODRL's core abstractions (Policy, Rule, Permission, Prohibition, Duty, Constraint, Party, Asset) are highly suited for expressing granular access and usage control. Consequently, industrial use cases beyond privacy now dominate the ODRL adoption curve. Notable domains include IPTC RightsML 2.0 for media licensing, JPEG Trust for image provenance, the International Data Spaces (IDS) Usage Control Language, Gaia-X Verifiable Credentials, the W3C Text and Data Mining (TDM) Reservation Protocol, and various financial trading licences.

The appeal of ODRL in these diverse environments stems from its simplicity and its foundation in semantic web technologies, which allow policies to be easily serialized (e.g., as JSON-LD or Turtle) and integrated into existing knowledge graphs. However, this simplicity comes at a cost when applied to highly regulated, multi-jurisdictional environments. The rules governing a media asset in IPTC RightsML are fundamentally different in complexity from the rules governing the cross-border transfer of sensitive financial data. The former often involves straightforward commercial agreements; the latter requires reconciling overlapping national laws, sector-specific regulations, and corporate contracts.

Despite this broad traction, the rules-engine ecosystem is increasingly routing around ODRL's normative semantics. Production deployments such as the ODRL-PAP compiler routinely translate ODRL syntax into execution environments like Open Policy Agent (OPA)/Rego. Each proprietary execution layer injects its own conflict-resolution semantics and provenance tracking atop ODRL's underspecified base. For instance, when an ODRL policy is compiled into Rego, the compiler must make arbitrary decisions about how to handle conflicts that ODRL itself cannot express natively. As a result, ODRL is currently providing an interoperable syntax, but not interoperable policy semantics. If two different policy engines evaluate the same ODRL policy and arrive at different conclusions because they employ different, unstandardised conflict-resolution logic, the promise of semantic interoperability is broken.

In this paper we isolate two primary structural blockers - conflict resolution and provenance - and propose two modular, backward-compatible extension profiles (odrl-prov and odrl-cr) to resolve them. By addressing these gaps, we aim to ensure that ODRL can serve not just as a descriptive format, but as a robust foundation for automated, legally sound policy enforcement.

Structural Gaps in ODRL 2.2

The Conflict Resolution Deficit

The W3C ODRL Information Model 2.2 defines the conflict property as taking exactly one of three ConflictTerm values: perm, prohibit, or invalid. This mechanism fires only when a Permission and a Prohibition target the exact same Action on the exact same Asset.

This design is structurally insufficient for complex regulatory reasoning. It is strictly weaker than XACML 3.0's combining algorithms (e.g., deny-overrides, permit-unless-deny, first-applicable), which support ordered evaluation and indeterminate verdicts, allowing for a much more nuanced approach to resolving competing directives. Furthermore, ODRL provides no native mechanism for resolving:

Conflicts between two competing Permissions or two Prohibitions across merged policies. When multiple policies are combined (a common scenario in federated data spaces), ODRL cannot determine which permission takes precedence if they offer overlapping but slightly different rights.

Conflicts among Duties or Obligations. If one policy mandates data deletion after 30 days and another mandates retention for 7 years for tax purposes, ODRL provides no semantic vocabulary to resolve the conflicting obligations.

Conflicts arising from implicit action subsumption. (e.g., permit display versus prohibit print). As Steyskal & Polleres identified, ODRL operates under the limiting assumption that actions are strictly disjoint, ignoring the real-world reality that granting permission to "edit" might implicitly conflict with a prohibition to "delete."

Jurisdictional, temporal, or specificity-based precedence. ODRL lacks the vocabulary to express that rules drawn from different legal instruments might have different weights based on their source (e.g., federal law vs. state law) or their enactment date.

Anatomy of the Conflict Gap: A Worked Example

Consider a request to transfer a customer dataset across borders, a common scenario in international banking. This action is governed concurrently by three policies:

- An EU GDPR-derived Policy (P1) prohibiting transfer to non-adequate countries.
- A US CLOUD-Act-derived Policy (P2) permitting disclosure upon lawful subpoena.
- A corporate contract Policy (P3) permitting transfer with explicit user consent.

All three target the identical Asset and Action (transfer).

Under ODRL 2.2, if we merge these into a single policy set (P1 ∪ P2 ∪ P3):

- Setting conflict=invalid (the default): This voids the entire merged policy upon detecting a conflict. The result is a "default deny" outcome that blocks perfectly legitimate, compliant transfers (e.g., a transfer where the user has consented).
- Setting conflict=prohibit: This ensures the prohibition (GDPR) always overrides the permissions. This incorrectly blocks transfers where a user has explicitly consented, violating the intended operation of GDPR Article 49 derogations.
- Setting conflict=perm: This ensures the most permissive rule always wins. This is a catastrophically incorrect setting for a regulated financial entity, as it would allow a corporate contract to override a statutory prohibition.

Currently, ODRL offers no normative vocabulary for these strategies, forcing systems to offload the reasoning to unstandardised external engines, thereby losing the benefits of a shared, semantic policy language.

The Provenance Deficit

The ODRL Information Model does not define any normative properties to support rule derivation traceability. As a result, when an auditor asks questions such as: “Why was this transfer blocked at 14:07 GMT on 12 May 2026, under which clause of which regulation, and which engineer approved the derivation?” ODRL 2.2 cannot provide a native, standards-based answer.

This absence of end-to-end traceability represents a significant barrier to adoption in regulated industries, where organisations must be able to evidence compliance as rigorously as they enforce it.

Implementers often attempt to retrofit Dublin Core terms (dc:creator, dct:source) or Provenance, Authoring and Versioning (PAV) properties onto their ODRL instances. However, these are generic metadata tags. They lack the relational expressivity required to model complex derivation chains and fail entirely to provide audit-trail provenance over runtime policy evaluation. The W3C ODRL Temporal Model Working Draft attempts to address versioning but remains unfinished and notably omits derivation chains.

LegalRuleML demonstrates a superior, purpose-built pattern for this domain. Using dedicated metadata blocks (lrml:LegalSources, lrml:Authorities, lrml:Context), it links each rule explicitly to the exact textual provision it derives from, the authoring agent, its governing, and its temporal efficacy (e.g., when the rule enters into force versus when it becomes applicable). Without adopting a similar Linked Data approach, ODRL remains unsuitable for high-stakes compliance and smart-contract anchoring, where the exact provenance of a rule must be undeniable.

Proposed Architecture: Extension Profiles

Following the Data Privacy Vocabulary (DPV) 2.0 pattern of layering domain-agnostic extensions atop a core ontology, we propose two thin, backward-compatible profiles. This avoids overloading the ODRL core with complexity that might not be needed for simpler DRM use cases, while providing essential industrial capabilities for regulatory environments.

The odrl-prov Profile (Provenance)

The odrl-prov profile imports concepts from prov: and pav:, mapping ODRL constructs directly into the PROV-O hierarchy. We declare **odrl:Policy⊑prov:Entity** and **odrl:Rule⊑prov:Entity**.

The profile adopts the following as standard ODRL properties, providing a normative vocabulary for traceability:

- **prov:wasDerivedFrom**: Maps the specific ODRL rule back to its legal or contractual source document.
- **prov:wasAttributedTo**: Identifies the authoring agent, system, or knowledge engineer responsible for formalising the rule.
- **prov:hadPrimarySource**: Provides the original legal text URI, ideally using stable identifiers like an Akoma Ntoso or ELI (European Legislation Identifier).
- **pav:version and pav:previousVersion**: Provides strict version control, essential for tracing changes in policy over time.

Crucially, the profile introduces **odrl:EvaluationRecord⊑prov:Entity** to record every Policy Decision Point (PDP) execution. This is a vital addition, as it unifies design-time metadata (where the rule came from) with runtime decision provenance (when and why the rule was applied to a specific request).

Provenance Fragment Example

A PROV-O-aligned extension natively represents the traceability chain in a machine-readable format:

ex:RulePolicy123 a odrl:Policy, prov:Entity ;

prov:wasDerivedFrom legis:GDPR-Art-44 ;

prov:wasGeneratedBy ex:DerivationActivity-789 ;

prov:wasAttributedTo ex:DataArchitectAlice ;

prov:hadPrimarySource https://eur-lex.europa.eu/eli/reg/2016/679 ;

pav:version "1.2.0" ;

pav:previousVersion ex:RulePolicy122 ; dct:issued "2026-04-30T09:00:00Z"^^xsd:dateTime .

ex:DerivationActivity-789 a prov:Activity ;

prov:startedAtTime "2026-04-30T08:42:00Z"^^xsd:dateTime ; prov:wasAssociatedWith ex:LegislationConverter-v2.3 ; prov:used https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng/pdf .

Runtime evaluation provenance

ex:Decision-2026-05-12-1407 a prov:Entity, odrl:EvaluationRecord ; prov:wasGeneratedBy ex:Evaluation-2026-05-12-1407 ; prov:wasInfluencedBy ex:RulePolicy123 .

This fragment delivers rule-to-source traceability at the article level, agent attribution, derivation timestamps, and runtime decision provenance. This is the exact unbroken chain of evidence required by legal auditors to verify automated compliance decisions.

The odrl-cr Profile (Conflict Resolution)

The odrl-cr profile fundamentally restructures how ODRL handles conflict. It generalises **odrl:ConflictTerm** from a closed enumeration of three simple values into an extensible class, permitting a rich hierarchy of conflict resolution strategies:

- **XACML equivalents:** cr:DenyOverrides,cr:PermitOverrides,cr:FirstApplicable, andcr:OnlyOneApplicable. These are mapped directly to the ODRL deontic vocabulary, enabling ODRL to achieve parity with the expressive capability of the industry-standard XACML access control language.
- **cr:PriorityWeighted:** Adds cr:hasWeight(xsd:decimal) attribute to each Rule, providing a clear, deterministic numerical ordering to support straightforward priority-based conflict resolution.

To guarantee auditability, the profile dictates that each strategy MUST emit a cr:ResolutionReport carrying the properties cr:winningRule, cr:overriddenRules, and cr:strategyApplied. This aligns perfectly with the Compliance Report Model recently introduced to make the output of ODRL evaluations formally interoperable and transparent.

Integration and Comparative Position

Comparative Position against Legacy Languages

A comparison with predecessor policy languages such as Rei and KAoS is highly instructive. Both utilised semantic-web technologies (RDF-S and OWL-DL, respectively) for policy expression and, notably, offered significantly richer conflict-handling mechanisms than ODRL 2.2. KAoS, for example, pioneered "policy deconfliction" via an online description logic theorem prover to automatically detect and resolve conflicts across different abstraction levels.

However, neither language natively addressed rule provenance, nor did they achieve anything approaching ODRL's massive industry adoption. Therefore, rather than abandoning ODRL for a theoretically purer alternative, the optimal path is to fold the defeasible logic strengths of Rei and KAoS, and the rich metadata models of LegalRuleML, into ODRL via our proposed profiles. This allows the community to leverage the vast existing ecosystem of ODRL tooling while upgrading its inferential capacity to meet modern demands.

Conclusion and Standardisation

The W3C ODRL standard has successfully transcended its DRM origins, but its normative semantics have not scaled to meet the rigorous requirements of its new, highly regulated domains. Because ODRL currently lacks robust conflict resolution and provenance models, developers are forced to hardcode essential regulatory logic into external execution engines, effectively destroying the semantic interoperability the language was designed to provide.

References

[1] W3C ODRL Community Group. ODRL Information Model 2.2. W3C Recommendation, 2018. Available: https://www.w3.org/TR/odrl-model/

[2] "ODRL-PAP: An ODRL Policy Administration Point Compiler," GitHub Repository, 2024. [Online]. Available: https://github.com/wistefan/odrl-pap

[3] "ODRL Policy Formulation and Execution via InstAL," in Proc. of the 3rd International Joint Conference on Rules and Reasoning (RuleML+RR),

[4"W3C Standard ODRL Policy gaining industry adoption," W3C Blog, Oct. 2025.

[5] OASIS. eXtensible Access Control Markup Language (XACML) Version 3.0, OASIS Standard, 2013.

[6] "Towards Formal Semantics for ODRL Policies," in RuleML, LNCS 9202, pp. 360-375, 2015.

[7] "A Defeasible Deontic Calculus for Resolving Norm Conflicts," arXiv preprint arXiv:2407.04869, 2024.

[8] OASIS. LegalRuleML Core Specification Version 1.0, OASIS Standard, Aug. 2021.

[9] "DPV 2.0: A Modular Ontology for Data Privacy," in Proc. of the 23rd International Semantic Web Conference (ISWC 2024)

[10] W3C. PROV-O: The PROV Ontology. W3C Recommendation, 2013.

[11] One License to Compose Them All," in Proc. of the 12th International Semantic Web Conference (ISWC 2013)

[12] A Policy Language for a Pervasive Computing Environment," in Proc. of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY), pp. 63-74, 2003.

[13] "New Developments in Ontology-Based Policy Management: Increasing the Practicality and Comprehensiveness of KAoS," in Proc. of the IEEE POLICY, 2008.

[14] "ODRE: ODRL Enforcement Framework," Computers & Security, vol. 150, 104282, 2025.