From Enterprise Knowledge Graphs to Policy-Compliant Agents: Governing Data, Models, and Autonomous Curation

Chris Hokamp[1], Jack Boylan[1], Jerome Bryssinck[1], Deniz Kuey[1] and John Murray[1]

[1]*Quantexa*

Abstract

The governance surface for knowledge graphs in highly regulated domains such as financial crime now extends to model risk management, regulatory obligation traceability, and the guiding and auditing of LLM-backed agent behaviors. Importantly, different use cases for enterprise knowledge graphs require adherence to different policies, so frameworks must be flexible and easy to extend; otherwise, adding compliance harnesses for new policies or regulation can be prohibitively expensive. In this paper, we describe the governance architecture of the Quantexa Metagraph, an enterprise knowledge graph that encodes its own governance rules as first-class graph content, alongside the domain knowledge those rules govern: governance and policy ingestion protocols aligned with open standards, regulatory policies decomposed into machine-traversable obligation→control chains, and behavioral policies that curator agents read from the same graph they write to, under write-mode controls and a per-mutation, agent-attributed audit log. New policies can be curated into the graph, and because agents read and validate their in-graph instructions as the work, they can be dynamically configured to follow new or customized policies.

We map these mechanisms onto ODRL, DPV, and SHACL, identifying what existing standards do not yet cover, and argue that next-generation data governance should treat policies for agents as data in the graph rather than as compliance overlays around it.

Keywords

data governance, knowledge graphs, autonomous agents, ODRL, model risk management, regulatory compliance, provenance

1. Introduction

Knowledge graphs are critical infrastructure in financial services, underpinning entity resolution, risk scoring, and investigative workflows across anti-money-laundering (AML), know-your-customer (KYC), and fraud use cases [1, 2], among others. Two recent shifts change what it takes to govern them. First, LLM-backed agents now automate substantial parts of the knowledge graph lifecycle — construction, enrichment, deduplication, and documentation — which has made enterprise knowledge graphs tractable to build and maintain for many organisations. Second, knowledge graphs increasingly serve as the memory and coordination substrate for those same agents [3, 4]. The graph is therefore no longer only an asset to be protected: it is also the medium through which autonomous systems decide what to do next.

This paper describes the governance architecture of the Quantexa Metagraph, an enterprise knowledge graph whose deployment domain currently targets financial crime detection and investigation, and regulatory compliance. The Metagraph holds the domain knowledge from which AML, KYC, and fraud deployments are assembled, and it is used and continuously curated by human experts and by a fleet of autonomous LLM curators working through the same interfaces. That combination raises governance questions with regulatory weight: who may write what, on what authority, with what audit trail, and in satisfaction of which obligations under regimes such as SR 11-7, SS1/23, and the EU AI Act.

3rd NeXt-generation Data Governance workshop (NXDG 2026), co-located with the 22nd SEMANTiCS conference, 15 September 2026, Ghent, Belgium

chrishokamp@quantexa.com (C. Hokamp); jackboylan@quantexa.com (J. Boylan); jeromebryssinck@quantexa.com (J. Bryssinck); denizkuey@quantexa.com (D. Kuey); johnmurray@quantexa.com (J. Murray)

0000-0002-7850-9398 (C. Hokamp)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-13

CEUR-WS.org/Vol-4327/paper1.pdf

Two ideas are easily conflated here, and we distinguish them explicitly. *Governance of the graph* is the requirement: controlling and accounting for how the graph changes and how it is used. *In-graph governance* is the design position this paper argues for: representing the governance rules themselves — regulatory obligations, model documentation, and the behavioural policies of curator agents — as first-class graph content in the same store as the domain knowledge they govern. Mature standards address individual pieces of this problem, and access and usage control for linked data is an active research area, particularly around European data spaces (Section 2). The combination that an agent-curated enterprise graph needs has received far less attention: prescriptive behavioural policies for autonomous curators, graduated trust between them, and per-mutation audit, delivered so that the governance layer can evolve as quickly as the graph it governs.

Our contributions are:

1. a **regulatory knowledge model** that decomposes regulations into machine-traversable obligation→control→risk-factor chains, connected to the detection logic that implements them (Section 4.1);
2. a **model governance layer** that represents model risk management artifacts required by SR 11-7[1], SS1/23[2], and the EU AI Act as graph entities linked to the components they document (Section 4.2);
3. a **policy-compliant agent architecture** in which autonomous curators operate under behavioural policies stored in the graph they curate, with write-mode controls, structural shape constraints, and a per-mutation, agent-attributed audit log (Section 4.3);
4. an **alignment analysis** mapping these mechanisms onto ODRL, DPV, and SHACL, identifying the governance requirements that current standards do not cover (Section 5).

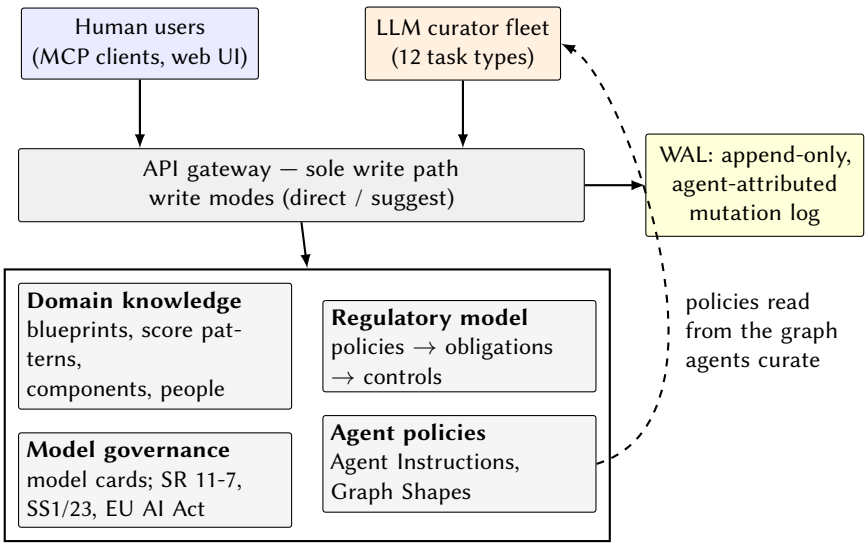


Figure 1: Governance architecture of the Metagraph. All writes pass through one gateway and are logged with attribution before execution. Behavioural policies (Agent Instructions) and structural constraints (Graph Shapes) are entities in the same store, so the curator fleet reads its governance from the graph it maintains.

2. Related Work

The semantic web community has produced mature standards for various governance concerns. The Open Digital Rights Language (ODRL) expresses access- and usage-control policies [5]. The language itself is domain-agnostic; ODRL profiles extend it with the domain-specific concepts a particular setting requires, and ODRL-based policies already govern data sharing in dataspace architectures such as

[1] https://www.federalreserve.gov/boarddocs/srletters/2011/sr1107.pdf
[2] https://www.bankofengland.co.uk/prudential-regulation/publication/2023/may/model-risk-management-principles-for-banks-ss

Gaia-X and IDS [6, 7]. Recent work has moved ODRL policies from descriptive to enforceable [8, 9] and encoded regulatory instruments themselves as semantic patterns [10]. The Data Privacy Vocabulary (DPV) [11, 12] provides a jurisdiction-agnostic core vocabulary for privacy, data protection, and AI metadata; extensions cover specific regulations, among them the GDPR [13], the EU AI Act [14], and the Data Governance Act [15]. SHACL validates graph structure [16], and the AI Risk Ontology (AIRO) maps the EU AI Act into an ontology designed to aid risk modeling [17].

Policy evaluation and enforcement have a longer history outside RDF. XACML [18] standardizes an attribute-based access-control architecture, and the Open Policy Agent [19] is widely deployed for infrastructure policy. Within the ODRL ecosystem, the W3C ODRL Community Group's formal-semantics effort [20] defines an evaluation output model that goes beyond a binary permitted/prohibited outcome, and interoperable evaluators are emerging around it [21]. Data governance and AI governance also meet on the documentation side: model cards [22], structured documentation of a model's purpose, training, validation, and limitations, have become the artifact that model-risk and AI regimes effectively mandate; they reappear as graph content in Section 4.2.

These standards primarily govern *data*: who may access, process, or share it. Agent participation changes the governance problem in kind, not only in degree. A policy that is unambiguous in the mind of a compliance officer may be ambiguous under an LLM's interpretation, so behavioral rules cannot simply be written down and assumed to hold; they must be made explicit, attributed, validated structurally where possible, and audited per mutation. At the same time, the semantics of the graph itself are partly emergent: because many use cases require agents to extend the schema as new data arrives (including the ingestion and curation of new policies or user-specific governance protocols), the governance layer must evolve with the graph rather than being fixed outside it. In this paper we describe how these requirements are met in the Quantexa Metagraph.

3. The Metagraph

The Metagraph is a Wikidata-style [23] enterprise knowledge graph framework: entities and properties are the top-level domain items, and the schema itself is graph content, so new classes and properties can be introduced at runtime without migration. Our internal instantiation of the metagraph currently holds approximately 300,000 triples describing roughly 16,000 entities, spanning solution blueprints for AML, KYC, and fraud use cases, score patterns across risk typologies, platform components, personnel, documentation, and the regulatory and governance content this paper describes. Importantly, the AI agents that interact with the graph, and their instructions, are also stored in the graph. The Metagraph is stored in a SPARQL-queryable graph database behind an API gateway that is the sole write path, and is consumed by human users through Model Context Protocol tools [4], and a CLI interface accessible by coding assistants, by a web interface, and by a fleet of autonomous LLM curators, spanning twelve registered task types, that runs continuously.

Three architectural features carry the governance load (Figure 1). First, every mutation passes through a write-ahead log (WAL): an append-only record of the full SPARQL operation, timestamp, target graph, and agent identity, retained in version control and sufficient to reconstruct the store by replay. Second, the gateway enforces hard policy at write time — malformed SPARQL is rejected, a collision guard prevents conflicting entity and property creation. Third, agents are governed by content *in* the graph: behavioral instructions and structural shape constraints that curators fetch at startup and monitor for changes, updates, and messages as they work. This last loop — agents reading their policies from the artifact they maintain — is the pattern the rest of the paper develops. The loop extends to curating the *policies themselves*: ingesting a new policy is itself a curation task, so agents can be tasked with adding policies to the graph, and with evaluating how well other agents follow the policies already in it.

The four store layers in Figure 1 interact through ordinary typed properties rather than bespoke machinery. The regulatory model reaches into domain knowledge: obligation and control entities connect, through dedicated properties, to the score patterns and solution blueprints that implement them (Section 4.1). The model-governance layer documents the platform itself: model cards and software

cards attach to the components they describe, and the regulatory standards that mandate them are entities in the same store (Section 4.2). Agent policies close the loop drawn as the dashed edge: the curator fleet fetches its Agent Instructions and Graph Shapes from the store at startup and writes back through the same gateway those policies govern (Section 4.3).

4. Governance as Graph Content

4.1. The Regulatory Knowledge Model

Financial institutions must demonstrate that deployed detection logic addresses specific regulatory requirements. To make that demonstration a query rather than a documentation exercise, the Metagraph decomposes regulations into graph structure: we currently have ingested policies and regulations from 17 regulatory bodies (FATF, MAS, FCA, EBA, FinCEN, and others across nine jurisdictions), 15 regulatory policies, and 37 policy sections, connected through dedicated properties to 12 obligations, 8 control measures, and 8 risk factors. The properties *imposesObligation* (P473), *requiresControl* (P474), *identifiesRiskFactor* (P476), and *riskLevel* (P477) make obligation chains traversable in both directions. For example, for Singapore's MAS Notice 626 [24], decomposed most deeply to date:

$$\texttt{MAS 626 §8.2 (PEPs)} \xrightarrow{\text{P473}} \texttt{apply EDD for PEPs} \xrightarrow{\text{P474}} \texttt{\{PEP screening; source-of-wealth verification; senior-management approval\}}$$

A compliance officer — or an agent — can ask either "what controls implement this obligation?" or "which obligations does this control satisfy?"

The same pattern connects regulation to detection. A four-property *configuration chain* links each score pattern in the Quantexa software to its risk typology (P468), required entity types (P469), graph-traversal requirements (P470), and applicable solution blueprints (P471), so that "which score patterns address trade-based money laundering, require Business entities, and apply to the trade-AML blueprint?" is a single SPARQL query. In practice we find the value of this model is less multi-jurisdictional comparison than *use-case-specific assembly*: given a deployment's blueprint and jurisdictions, an agent bootstraps the applicable requirement set — obligations, controls, score patterns, thresholds — from the graph instead of re-reading regulatory text. A synthetic simulation layer grounds this model end-to-end: narrative financial-crime scenarios (cross-border layering, trade-based laundering, luxury-asset integration) are mapped to real graph identifiers for typologies, score patterns, and regulatory agencies, providing a testbed for validating that obligation chains support complete compliance narratives without exposing client data.

4.2. Model Governance as Graph Content

Model risk management regimes — SR 11-7 [25] in the US, SS1/23 [26] in the UK, and the horizontal requirements of the EU AI Act [14] — require model cards, validation evidence, and lifecycle documentation. When these artifacts live outside the knowledge graph, they cannot participate in the queries that govern everything else. The Metagraph therefore encodes an company-internal model-risk-management framework as a 31-entity subgraph: model cards [22] and component *software cards* (the same documentation pattern applied to non-model software components), a four-pillar testing framework (technical validation, conceptual soundness, performance evaluation, ongoing monitoring), a six-way classification of AI system artifacts, and red/yellow/green license-risk assessment for third-party assets. A dedicated property (*hasGovernanceDocumentation*, P423) links more than 50 platform components to their governance artifacts, so "which components lack model cards?" and "which model cards claim SR 11-7 compliance?" are queries, not audits. The three regulatory standards are themselves entities, linked to the artifacts they mandate; concepts from the AI Risk Ontology (AIRO) [17] — risk, risk source, consequence, control — are included as alignment points with the wider AI governance vocabulary landscape. The same layer tracks governance documentation across a portfolio of client deployments (global custody and retail banks, a securities-market utility, and professional-services firms), which

makes *governance maturity* — the completeness of the governance artifacts each deployment requires — itself queryable across the portfolio.

4.3. Policy-Compliant Autonomous Curation

The Metagraph is curated continuously by autonomous LLM agents — twelve registered task types covering discovery, enrichment, deduplication, synchronization, and documentation. Governing them requires attribution, authorization, behavioural constraint, human review where trust is limited, structural compliance, and full auditability. To the best of our knowledge, no existing semantic governance standard addresses this combination.

In the Metagraph, four mechanisms do the work.

1. *Write modes*: trusted curators write directly while untrusted or experimental agents can be configured to operate in `suggest` mode, where assertions become pending suggestions that a human reviewer approves or rejects — approval materializes the triple with full attribution.
2. *Agent Instructions*: behavioral policies are entities in the graph, prioritized (critical / high / normal), categorized, and optionally targeted at specific agent types; agents fetch them at startup via SPARQL and receive them in their system prompts.
3. *Graph Shapes*: SHACL constraints specify required, optional, and banned properties for key classes, and a per-entity endpoint computes constraint-by-constraint compliance on demand. Shapes themselves are curatable graph citizens rather than external documents.
4. *Audit and graduated trust*: the WAL attributes every mutation to an agent identity, and an authority score (currently derived from the reliability of an agent's sources) implements *graduated trust*: a continuum between suggest-only and direct-write authority, rather than a binary trusted/untrusted split.

Some of the features above were implemented after iteration with earlier prototypes. For example, strict attribution was added in the first week of autonomous curation, after concurrent agents produced conflicting edits that could not be assigned; write modes arrived with multi-user deployment; instructions were introduced when recurring semantic errors (generic properties, duplicates, inconsistent dates) proved impossible to prevent with structural validation alone; shapes then caught the structural remainder that natural-language instructions addressed only probabilistically. Our experience is that agent governance is necessarily iterative: each new operational context (autonomous writes, concurrency, multiple users) forces a mechanism that was not evident from first principles, and we expect our governance and policy harness to continue to evolve over time.

4.4. An Example Workflow

Consider an agent tasked with connecting score patterns to the PEP obligations of Section 4.1. At startup it fetches its Agent Instructions from the graph, including a critical-priority instruction to verify claims against an authoritative source before writing. Its authority score does not yet permit direct writes, so when it asserts a link between a screening score pattern and the *apply EDD for PEPs* obligation, the gateway holds the assertion as a pending suggestion. The proposed triple is checked against the Graph Shapes for score patterns; a human reviewer approves it; approval materializes the triple; and the WAL records the mutation, attributed to the agent. Every step in that chain — regulation, obligation, instruction, write-mode decision, shape check, approval, audit entry — is graph content, so the compliance narrative from MAS 626 regulation to the approved graph mutation is a single traversal.

5. Alignment with Existing Standards

The mechanisms above map partially onto existing standards. In ODRL terms [5], `direct` write mode is a permission, property and shape-based bans are prohibitions, and `suggest` mode a permission carrying

an approval duty; formalizing our write policies as an ODRL profile would make them interoperable with dataspace policy frameworks such as Gaia-X and IDS [6, 7], and enforceable with emerging ODRL engines [8, 9]. Agent Instructions sit at the edge of ODRL's scope. The community group's formal semantics [20] and the evaluators built on it [21] already produce evaluation outputs richer than a binary permitted/prohibited verdict, so we mark agent behavioural governance as partially within ODRL's reach in Table 1. What remains outside the language's remit is prescriptive guidance — which properties to prefer, what to verify before writing — and the authority score's graduated trust, a continuum that permission/prohibition semantics do not express. DPV 2.0's AI Act extension [12] aligns naturally with the model-governance layer (AI system classification, risks, controls), with the difference that the Metagraph operationalizes these concepts in live queries driving blueprint navigation and gap analysis rather than as static annotations. The Graph Shape system is functionally SHACL [16], with two deliberate departures: shapes are first-class entities subject to the same provenance and curation as domain content, and compliance is computed continuously through an API rather than in batch validation runs. Table 1 summarizes the coverage.

Table 1
Governance requirements of agent-curated enterprise KGs versus the scope of existing semantic governance standards (• = covered; ○ = partial; — = out of scope).

Requirement	ODRL	DPV	SHACL	Metagraph mechanism
Data access control	•	○	—	write modes
Privacy metadata	—	•	—	attribution properties
Structural validation	—	—	•	Graph Shapes
Agent behavioural governance	○	—	—	Agent Instructions
Obligation traceability	—	○	—	P473/P474/P476 chains
Model governance artifacts	—	○	—	P423 + framework entities
Per-mutation audit trail	—	—	—	agent-attributed WAL
Graduated trust	—	—	—	authority scoring
Review workflow	○	—	—	suggestion pipeline

6. Discussion

In-graph governance Keeping governance rules, regulatory obligations, model documentation, and agent policies in the graph — rather than in systems around it — yields one query surface: a single SPARQL traversal can run from a regulatory obligation to the score pattern that implements it, the model card that documents that component, and the agent that last curated any of them. It also means governance content is maintained by the same curation machinery as domain content, and that provenance is self-describing in the sense of PROV [27]: "who changed this obligation?" and "what does this agent typically change?" are the same kind of query. Critically, storing the governance framework inside the graph allows straightforward curation and extension of the framework itself, meaning that our design allows us to rapidly ingest new policies and governance documents, and our agents adapt their behavior on the fly to changes in the graph.

Limitations Agent Instructions are currently natural language interpreted by LLMs, not rules executed by a reasoner; compliance is probabilistic, which is precisely the gap noted in Section 1 (a policy unambiguous to its human author may admit multiple machine interpretations). We believe that this compromise is necessary to make real-world use of current AI models, because much of the power of these models is in their ability to generalize from natural language instructions. We mitigate risk with hard enforcement where rules are expressible in code and shapes where they are structural.

Next steps The natural continuation is formal alignment: expressing write-mode policy as an ODRL profile with conflict detection over the emerging formal semantics [20]; mapping the model-governance

layer onto DPV 2.0's AI Act extension for interoperable compliance reporting; exporting Graph Shapes as standard SHACL while retaining the in-graph representation as authoritative; and formalizing or reviewing agent instructions enough to detect contradictions unambiguously before agents encounter them. Two further extensions are on our roadmap: explicit roles for humans and agents (RBAC, federation, and multitenancy across deployments), and closing the governance loop experimentally — one agent following a policy while others evaluate, score, and curate improved policy-following behaviour.

7. Conclusion

Enterprise knowledge graphs in regulated industries need governance that spans data access, regulatory obligation, model documentation, and agent behaviour. The Quantexa Metagraph demonstrates a working implementation of a holistic, AI-native framework in which policies, obligations, governance artifacts, and behavioral rules live as first-class content in the graph they govern, enforced at a single write path with per-mutation attribution.

Resource Availability

The Metagraph is a proprietary production system; its content includes client-deployment and internal platform data, so neither the graph nor the agent harness can be released. What this paper offers for reuse is the design: the standards alignment of Table 1, the property patterns of the regulatory model (obligation→control chains and configuration chains), and the four agent-governance mechanisms of Section 4.3, described at a level intended to support independent re-implementation. The synthetic simulation scenarios of Section 4.1 contain no client data.

Declaration on Generative AI

During the preparation of this work, the authors used Claude Code (Anthropic) for drafting assistance and manuscript preparation. The authors contributed all material and take full responsibility for the publication's content.

References

[1] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. Labra Gayo, R. Navigli, S. Neumaier, A.-C. Ngonga Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Computing Surveys 54 (2021) 71:1–71:37. doi:10.1145/3447772.

[2] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor, Industry-scale knowledge graphs: Lessons and challenges, Communications of the ACM 62 (2019) 36–43. doi:10.1145/3331166.

[3] P. Rasmussen, P. Paliychuk, T. Beauvais, J. Ryan, D. Chalef, Zep: A temporal knowledge graph architecture for agent memory, arXiv:2501.13956, 2025.

[4] Anthropic, Model context protocol, Specification, 2024. URL: https://modelcontextprotocol.io/.

[5] R. Iannella, S. Villata, ODRL information model 2.2, W3C Recommendation, 2018. URL: https://www.w3.org/TR/odrl-model/.

[6] Gaia-X European Association for Data and Cloud, Gaia-x architecture document, Release 22.10, 2022. URL: https://gaia-x.eu/.

[7] B. Otto, S. Steinbuß, A. Teuscher, S. Lohmann, IDS reference architecture model 4, International Data Spaces Association, 2022. URL: https://docs.internationaldataspaces.org/ids-knowledgebase/ids-ram-4.

[8] A. Cimmino, J. Cano-Benito, R. García-Castro, Open digital rights enforcement framework (ODRE): From descriptive to enforceable policies, Computers & Security 150 (2025) 104282. doi:10.1016/j.cose.2024.104282.
[9] W. Slabbinck, J. A. Rojas Meléndez, B. Esteves, R. Verborgh, P. Colpaert, May the FORCE be with you? A framework for ODRL rule compliance through evaluation, in: Posters, Demos, Workshops, and Tutorials at SEMANTiCS 2025 (NXDG), volume 4064 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4064/.
[10] D. Golpayegani, H. J. Pandit, D. Lewis, Semantic patterns of prohibited AI systems in the EU AI Act, in: Posters, Demos, Workshops, and Tutorials at SEMANTiCS 2025 (NXDG), volume 4064 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4064/.
[11] H. J. Pandit, A. Polleres, B. Bos, R. Brennan, B. Bruegger, F. J. Ekaputra, J. D. Fernández, R. Gachpaz Hamed, E. Kiesling, M. Lizar, E. Schlehahn, S. Steyskal, R. Wenning, Creating a vocabulary for data privacy, in: On the Move to Meaningful Internet Systems: OTM 2019 Conferences (ODBASE), volume 11877 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 714–730. doi:10.1007/978-3-030-33246-4_44.
[12] H. J. Pandit, B. Esteves, G. P. Krog, P. Ryan, D. Golpayegani, J. Flake, Data privacy vocabulary (DPV) — version 2.0, in: The Semantic Web – ISWC 2024, volume 15233 of *Lecture Notes in Computer Science*, Springer, 2024. doi:10.1007/978-3-031-77847-6_10.
[13] European Parliament and Council of the European Union, Regulation (EU) 2016/679 of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), Official Journal of the European Union, L 119, pp. 1–88, 2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.
[14] European Parliament and Council of the European Union, Regulation (EU) 2024/1689 — artificial intelligence act, Official Journal of the European Union, L series, 2024.
[15] European Parliament and Council of the European Union, Regulation (EU) 2022/868 — data governance act, Official Journal of the European Union, L 152, 2022.
[16] H. Knublauch, D. Kontokostas, Shapes constraint language (SHACL), W3C Recommendation, 2017. URL: https://www.w3.org/TR/shacl/.
[17] D. Golpayegani, H. J. Pandit, D. Lewis, AIRO: An ontology for representing AI risks based on the proposed EU AI Act and ISO risk management standards, in: Towards a Knowledge-Aware AI (Proc. SEMANTiCS 2022), IOS Press, 2022. doi:10.3233/SSW220008.
[18] OASIS, eXtensible Access Control Markup Language (XACML) version 3.0, OASIS Standard, 2013. URL: https://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.
[19] Open Policy Agent Project, Open Policy Agent, Cloud Native Computing Foundation, 2024. URL: https://www.openpolicyagent.org/.
[20] W3C ODRL Community Group, ODRL formal semantics, W3C Community Group Draft Report, 2025. URL: https://w3c.github.io/odrl/formal-semantics/.
[21] W. Slabbinck, J. A. Rojas Meléndez, B. Esteves, P. Colpaert, R. Verborgh, Interoperable interpretation and evaluation of ODRL policies, in: The Semantic Web – ESWC 2025, Lecture Notes in Computer Science, Springer, 2025, pp. 192–209. doi:10.1007/978-3-031-94578-6_11.
[22] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, T. Gebru, Model cards for model reporting, in: Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19), ACM, 2019, pp. 220–229. doi:10.1145/3287560.3287596.
[23] D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledgebase, Communications of the ACM 57 (2014) 78–85. doi:10.1145/2629489.
[24] Monetary Authority of Singapore, Guidelines to MAS notice 626 on prevention of money laundering and countering the financing of terrorism, 2015. As amended.
[25] Board of Governors of the Federal Reserve System and Office of the Comptroller of the Currency, Supervisory guidance on model risk management, SR Letter 11-7, 2011.
[26] Bank of England Prudential Regulation Authority, Model risk management principles for banks, Supervisory Statement SS1/23, 2023.

[27] T. Lebo, S. Sahoo, D. McGuinness, PROV-O: The PROV ontology, W3C Recommendation, 2013. URL: https://www.w3.org/TR/prov-o/.