Received: 3 May 2026 | Revised: 9 July 2026 | Accepted: 29 July 2026 | Published online: 2 September 2026

RESEARCH ARTICLE

Journal of Computational Law and Legal Technology
2026, Vol. 00(00) 1–12
DOI: 10.47852/bonviewJCLLT620210282

BON VIEW PUBLISHING

Operationalizing the EU Data Act for Automated Compliance

Sheyla Leyva-Sánchez[1,*], Fabian Linde[1], Meem Arafat Manab[1], María Poveda-Villalón[1] and Víctor Rodríguez-Doncel[1]

[1] *Universidad Politécnica de Madrid, Spain*

Abstract: The EU Data Act establishes comprehensive rules governing data access and sharing across business-to-consumer (B2C), business-to-business (B2B), and business-to-government (B2G) contexts. This paper presents a comprehensive ontology for the EU Data Act, enabling reasoning over data-sharing agreements through machine-readable representations. The DAOnt ontology reuses elements from three established ontologies—the Core Ontology of the Legal Knowledge Interchange Format, Open Digital Rights Language, and Data Privacy Vocabulary—to capture the normative structure of the Data Act. The ontology captures the main concepts and relationships in the Regulation, and it also operationalizes three articles to facilitate compliance checking: Article 4(1) (B2C user access rights), Article 8(6) (B2B trade secret exceptions), and Article 19(2)(a) (B2G competitive use prohibitions). The ontology supports compliance checking through SPARQL queries that return obligations, permissions, and prohibitions, allowing organizations to verify whether data-sharing agreements meet the requirements of the EU Data Act and to assess conditions such as FRAND obligations. By representing key legal concepts in Resource Description Framework (RDF), our work helps bridge the gap between the legal provisions of the Data Act and their computational interpretation. The complete ontology, along with example instances and queries, is available online.

Keywords: EU Data Act, legal ontology, compliance verification

1. Introduction

The EU Data Act[1] has become applicable on September 12, 2025, introducing a comprehensive regulatory framework for data access and sharing across the European Union. The regulation addresses three primary scenarios: business-to-consumer (B2C) data sharing where users gain access to data generated by their connected products (Chapter II, Articles 3–7); business-to-business (B2B) data sharing requiring user authorization and fair, reasonable, and nondiscriminatory (FRAND) terms (Chapter III, Articles 8–13); and business-to-government (B2G) data sharing under exceptional need circumstances (Chapter V, Articles 14–22).

Complying with the Regulation is not straightforward, and the difficulty has not gone unnoticed: legal scholars have already questioned what it will mean for competition, innovation, and everyday contracting [1]. Part of the problem is sheer intricacy. The Act runs to more than 50 articles that lean on one another—B2B sharing, for instance, needs the user's authorization under Article 8(4), has to meet the FRAND conditions of Articles 8–12, and may still be withheld for trade secrets under Article 8(6). Checking all of this by hand is both error-prone and hard to scale, and the strain falls hardest on small and medium enterprises (SMEs) with no legal department of their own. Data space architectures such as GAIA-X, meanwhile, need contracts that interoperate, and cloud providers need a way to enforce policy automatically.

Between the text of the law and its enforcement in software lies a gap that is awkward to close in practice. Because the Regulation is written in natural language, each obligation has to be read and interpreted by a person, and two organizations, or two jurisdictions, will not always read it the same way. As long as there is no machine-readable version to work from, checking compliance stays a slow, expensive exercise in which mistakes are easy to make. The tooling that does exist has so far concentrated on the General Data Protection Regulation (GDPR) and, more recently, the AI Act; the Data Act has been left almost untouched by automated solutions, even though its rules began to apply in September 2025.

Encoding a regulation in machine-readable form buys several things at once. A reasoner can work out which obligations bite in a given situation, a query can surface the clauses of a contract that break a rule, and a policy can be checked for compliance before it is ever put into production. For a small firm without an in-house legal team, this is the difference between an affordable tool and no tool at all, and it is also what the interoperability goals of the European data spaces quietly depend on.

This paper addresses the research question: *Can compliance checking with the EU Data Act requirements be automated through formal ontology and semantic reasoning?*

Our contribution is threefold. First, we present a comprehensive ontology for the EU Data Act, which integrates three

***Corresponding Author:** Sheyla Leyva-Sánchez, Universidad Politécnica de Madrid, Spain. Email: sheyla.leyva.sanchez@upm.es

[1] Regulation (EU) 2023/2854 of the European Parliament and of the Council of December 13, 2023, on harmonized rules on fair access to and use of data (Data Act).

established standards (the Core Ontology of the Legal Knowledge Interchange Format [LKIF-Core], Open Digital Rights Language [ODRL], and Data Privacy Vocabulary [DPV]) to formally represent regulatory requirements. Second, we demonstrate proof-of-concept formalizations for Articles 4(1), 8(6), and 19(2)(a), showing how different deontic modalities (mandatory obligations in B2C, permissive exceptions in B2B, and absolute prohibitions in B2G) can be captured and reasoned over. Third, we provide executable reasoning examples using SPARQL queries over RDF-formatted knowledge bases, demonstrating practical automated compliance-checking capabilities.

Paper Structure

The remainder of the paper is structured as follows. Section 2 reviews related work on legal ontologies, data governance and data-sharing ontologies, compliance approaches, and regulatory compliance ontologies, identifying the research gap that motivates our contribution. Section 3 presents the methodology, including the ontology engineering process and the integration of external vocabularies. Section 4 introduces the proof of concept, detailing the formalization of Articles 4(1), 8(6), and 19(2)(a) and demonstrating automated reasoning examples in B2C, B2B, and B2G compliance scenarios. Section 5 evaluates the ontology in terms of coverage, expressiveness, and interoperability. Finally, Section 6 concludes the article, and Section 7 outlines the directions for future work.

2. Related Work

The development of an ontology for the EU Data Act builds upon two complementary research lines: the use of ontologies to represent legal and regulatory knowledge and the formalization of compliance mechanisms enabling automated or semi-automated assessment of normative requirements. The first line concerns the design of conceptual models that capture legal concepts such as rights, obligations, and permissions, providing a shared semantic foundation for interoperability and reasoning. These models are typically expressed using Semantic Web standards such as RDF and Web Ontology Language (OWL), which ensure compatibility with existing ecosystems and support the representation of machine-readable normative structures. The second focuses on computational approaches to verify whether data processing or sharing activities comply with legal norms, using technologies such as SPARQL, Shapes Constraint Language (SHACL), logic programming, or rule-based systems. This section reviews previous work in both areas, highlighting their relevance to the modeling of data-sharing obligations and justifying the design choices adopted in our ontology.

2.1. Legal ontologies

Early legal ontologies were designed primarily as conceptual frameworks to represent the structure of legal knowledge rather than as operational data models. Their main purpose was to facilitate understanding, interoperability, and knowledge sharing within the legal domain, rather than to support good metadata or even machine-executable compliance checking. This focus on conceptual modeling distinguished them from later ontologies, such as those used in data governance and compliance, which aim to be directly operationalized through technologies like OWL reasoners, SHACL, or SPARQL.

A comprehensive survey by de Oliveira Rodrigues et al. [2] provides an overview of this evolution, building on earlier foundational work that emerged in the late 2000s and early 2010s and characterized legal ontologies as efforts to formalize legal concepts, case structures, and argumentation patterns, often within the contexts of knowledge management and document retrieval. The more recent literature emphasizes the shift toward lightweight, interoperable ontologies integrated with Semantic Web standards and designed to support automation, such as compliance checking and legal analytics.

Among the first and most influential initiatives, LKIF-Core [3] stands out as a foundational ontology to represent legal reasoning and normative structures. Developed within the ESTRELLA project, LKIF-Core provides an upper-level schema for capturing concepts such as norms, acts, and roles, inspired by legal theory and deontic logic. Although not originally intended as an operational compliance model, LKIF-Core established a reusable conceptual foundation that has informed subsequent vocabularies. Its modular and extensible design continues to make it a reference point for aligning new legal ontologies with established conceptual categories.

LegalRuleML [4] complements ontology-based representations with a formal markup language to express legal norms, rules, and reasoning structures. While LKIF-Core provides a conceptual vocabulary, LegalRuleML offers a syntactic and logical framework to encode the prescriptive aspects of law (obligations, permissions, prohibitions, and exceptions) using an XML-based serialization aligned with the broader RuleML family of rule-interchange languages. Its primary objective is to enable the interchange of machine-readable legal rules across systems, preserving their temporal and defeasible characteristics. However, unlike lightweight ontologies expressed in RDF, LegalRuleML was not conceived as a data model for operational use within Linked Data environments, but rather as a rule representation language supporting formal reasoning. For this reason, it bridges conceptual and computational layers, yet its adoption has remained limited outside specialized rule-based reasoning systems.

2.2. Ontologies for data governance and data sharing

The ODRL is a W3C specification, provided as two recommendations [5, 6], which provides a flexible and extensible policy expression language to represent permissions, prohibitions, and obligations associated with the use of digital assets. Originally conceived to manage usage rights in digital content distribution, ODRL has evolved into a general-purpose vocabulary to express policies that govern access, sharing, and usage of data and services. The activity in the domain of data spaces, closely related to our data act, is actually burgeoning. ODRL adopts an RDF-based model, enabling interoperability within the Semantic Web ecosystem and allowing policies to be linked to resources, parties, and actions. Although not specifically designed for legal compliance, the normative structure of ODRL—centered on the concepts of permission, prohibition, and duty—makes it particularly suitable for modeling data-sharing agreements and other legally relevant policies and has been widely reused and extended in data governance and privacy-related vocabularies.

DaPReCo (Data Protection Regulatory Compliance Ontology) [7] is a formal ontology designed to represent the legal concepts and obligations of the GDPR, with a focus on enabling rule-based compliance assessment. Based on LegalRuleML and First-Order Logic (FOL)-based formalizations, DaPReCo encodes the logical structure of GDPR provisions, allowing automated reasoning over obligations, rights, and legal bases for data processing. Beyond DaPReCo, several other ontologies have sought to capture the semantics of the GDPR at varying levels of abstraction and operationality.

The DPV [8][2], developed under the W3C Data Privacy Vocabularies and Controls Community Group, provides a lightweight RDF-based model for representing data processing purposes, legal bases, and data subject rights, emphasizing interoperability and practical deployment. DPV version 2.0 has been explicitly extended to cover EU regulatory regimes including the GDPR, the Data Governance Act, and the AI Act, positioning it as a cross-regulation backbone. Other initiatives—such as GConsent [9] or PrOnto [10]—focus on narrower aspects such as consent modeling or personal data categories; a state-of-the-art survey of semantic models for consent is provided by Kurteva et al. [11]. A systematic comparison of the policy languages and ontologies available for representing GDPR rights and obligations is provided by Esteves and Rodríguez-Doncel [12], who identify ODRL combined with DPV as the most expressive stack for GDPR information flows—a finding that directly motivates our reuse choices for the Data Act. Collectively, these ontologies illustrate a clear evolution from conceptual representations of privacy norms toward operational models that support machine-readable compliance verification.

2.3. Compliance approaches

Several conceptual approaches have been proposed to formalize and operationalize legal and regulatory compliance. DALICC (Data Access and License Interoperability for Content and Contracts) extends ODRL to model licensing and data-sharing policies in a way that supports interoperability across domains, bridging normative concepts with actionable constraints. In contrast, ontologies such as DaPReCo demonstrate how compliance checks can be performed by encoding GDPR rules into formal representations that allow reasoning engines to infer whether a given scenario satisfies obligations and permissions. Another widely used strategy involves SHACL, which validates RDF data against predefined constraints, providing a declarative and relatively lightweight mechanism for ensuring compliance with structural or policy requirements; recent work by Anim et al. [13] extends SHACL-SPARQL rules to capture the temporal and aggregate facets that legal compliance often requires. Some approaches go further by transforming ontology-based representations into other formal logic systems, such as first-order logic, Prolog, or Answer Set Programming (ASP), enabling advanced automated reasoning and defeasible rule evaluation. Robaldo et al. [14] systematically compare SHACL- and ASP-based reasoners on RDF inputs, showing that both scale to realistic compliance workloads. Working examples in adjacent regulatory domains include the GDPR compliance verification tool of Chhetri et al. [15] and the smashHitCore ontology for sensor data sharing in smart cities [16]. In the specific context of European data spaces, Cimmino et al. [17] propose the Open Digital Rights Enforcement framework, moving ODRL beyond descriptive policies toward executable enforcement, while Dam et al. [18] catalog recurring policy patterns and propose an ODRL profile tailored to data space usage control. Finally, SPARQL offers a simpler yet effective means to query RDF representations directly, allowing organizations to verify specific obligations, prohibitions, or permissions without the overhead of full logic-based reasoning systems. Each of these approaches balances expressivity, computational complexity, and ease of integration, shaping the design choices for operational compliance ontologies.

[2] https://w3id.org/dpv/

2.4. Regulatory compliance ontologies

Automated compliance check approaches span multiple paradigms. Semantic logic-based approaches include First-Order Logic representations and frameworks such as I-SNACC that achieve 95.2% precision and 100% recall [19]. SHACL-based compliance checking shows increasing adoption for validating regulatory requirements, with comprehensive reviews of the language's semantics and tooling now available to support practitioners [20]. Policy language–based approaches leverage ODRL and LegalRuleML to formalize rights and obligations in a machine-interpretable way. An ODRL Compliance Profile has also been proposed to demonstrate effective coverage when combined with DPV [21]. Recent innovations include the use of Defeasible Logic Programming to handle contradictions and incompleteness, ASP with event calculus formalisms, and hybrid approaches combining machine learning and rule-based reasoning; Francesconi and Governatori [22] provide a recent synthesis of compliance-checking patterns for deontic norms over Linked Open Data within a decidable OWL2 framework.

AI Act ontologies emerged rapidly following the regulation's adoption, including AIRO [23], which provides an OWL2 representation of AI system risks, the related VAIR vocabulary for risk assessments, and TAIR (Trustworthy AI Requirements Ontology) modeling AI Act clauses and ISO standards. More recently, Golpayegani et al. [24] have proposed an ontology for AI Act Fundamental Rights Impact Assessments, illustrating how DPV-based ontologies are being extended to operationalize specific procedural obligations of EU digital regulations—the same pattern DAOnt applies to the Data Act.

2.5. Research gap: The missing Data Act ontology

Despite the Data Act entering into force on January 11, 2024, with core provisions applying on September 12, 2025, no dedicated Data Act ontology currently exists. While DPV v2.0 includes Data Act support on its roadmap, this work remains incomplete. This represents an urgent gap for organizations that must comply without machine-readable specifications or automated compliance-checking tools.

Current ontologies cannot express critical Data Act scenarios: connected product data rights, multi-tier data-sharing chains (user → data holder → third party), conditional compensation mechanisms, FRAND term verification, switching rights for cloud services, or exceptional need provisions for public sector access. Our work addresses this gap through the first comprehensive formalization of the Data Act.

3. Methodology

Compliance has traditionally been verified either by having lawyers read through the contracts or by writing one-off scripts, and neither approach copes well once the obligations multiply, start referring to one another, and change over time. An ontology is a sturdier basis: by writing the provisions down in a form a machine can interpret, it takes much of the guesswork out and lets the same reasoning be applied again and again with the same result.

We began by reading the Data Act closely and arguing, among ourselves, about how its central notions—its obligations, its actors, the acts it regulates—ought to be represented. Out of that discussion, the requirements of the Regulation were turned into explicit semantic structures, each legal concept given a name

and tied to the others through the normative relations the text implies. What this buys us is a version of the regulatory logic that software can process without any further interpretation. SPARQL queries then do the actual work of compliance checking, picking out the relevant patterns with very little procedural code around them.

The effect is to move compliance assessment away from a subjective manual exercise and toward a process that runs automatically and leaves an audit trail, which in turn makes it more transparent, easier to scale, and easier to maintain. And because regulations rarely stand still, an ontology-based representation gives a flexible starting point: monitoring can carry on as the rules change without the whole apparatus having to be rebuilt.

3.1. Ontology engineering process

The development process follows the NeOn methodology [25], an iterative scenario-based framework well suited to modeling complex legal domains; activities are also informed by the more recent Linked Open Terms (LOT) framework [26], which adapts NeOn principles to industrial ontology engineering. We conducted a systematic requirement elicitation process through three complementary activities. First, a close reading and detailed annotation of the EU Data Act (Regulation 2023/2854) identified the main actors, core concepts, and their relationships throughout the regulation. Second, we formulated competency questions (CQs) and organized them into two categories: general CQs ensuring core conceptual coverage (e.g., "Who shares data with whom, and under which agreement?") and article-specific CQs supporting the proof-of-concept formalization (e.g., "Which data holders violate Article 4(1) by failing to provide the user-requested data from connected products?"). The complete catalog is provided in Table 1. The full Ontology Requirements Specification Document (ORSD) is included in Annex A. Third, we performed an interoperability analysis to maximize reuse of established vocabularies—importing foundational legal concepts from LKIF-Core and extending data governance notions using DPV and ODRL—thereby avoiding redundant or conflicting definitions (Figure 1).

Tool-supported refinement was carried out throughout the iterative modeling cycle. Chowlk [27] was used to automatically transform conceptual diagrams into formal OWL representations, and selected patterns from the Ontology Design Patterns catalog ensured reusable and well-structured modeling decisions. Validation and refinement were performed through a consistency check with OWL reasoners (HermiT and ELK), complemented by SPARQL queries that verified that all competency questions were answerable.

For documentation, we used Widoco, which generates human-readable specifications, metadata, and publication-ready HTML documentation. Finally, the ontology was published following W3C best practices using the OnToology service, guaranteeing persistent hosting, versioning, and automated quality evaluation.[3]

4. Proof of Concept: Three Regulatory Articles

In this section, we present a proof of concept that illustrates how DAOnt enables automated compliance checking across different regulatory contexts of the Data Act. We analyze three representative articles (4(1)[4], 8(6)[5], 19(2)(a)[6]), each corresponding to a distinct compliance modality—mandatory obligations, conditional permissions, and absolute prohibitions—and spanning the B2C, B2B, and B2G settings of the regulation. These examples demonstrate how the ontology captures normative structures with sufficient expressiveness to support automated reasoning. Table 2 provides a comparative overview of the formal representations used in each case.

Table 1
Competency questions grouped by general coverage (CQG1) and article-specific requirements for the proof of concept (CQG2)

CQG1. General CQs (9 CQs)	CQG2. CQs Related to PoC Articles (3 CQs)
CQ1. Who shares data with whom, and under what agreement?	CQ10. Which data holders have violated Article 4(1) by failing to provide requested data from connected products?
CQ2. What actions has a party performed?	CQ11. Which data holders have violated Article 8(6) by refusing data sharing without providing trade secret justification?
CQ3. What obligations does a party have?	CQ12. Which public sector bodies have violated Article 19(2)(a) by developing competing products or services using data obtained through B2G data sharing?
CQ4. Who manufactured a product?	
CQ5. Who uses a product?	
CQ6. What service does a product provide?	
CQ7. When does a legal rule apply?	
CQ8. Where does data come from?	
CQ9. Who holds the data?	

4.1. Automated reasoning examples across B2C, B2B, and B2G

To demonstrate compliance verification capabilities, we developed six contract examples instantiating both compliant and

[3]Public URI: https://w3id.org/def/daont; development repository: https://github.com/oeg-upm/DAOnt.

[4]Data holders shall make readily available data, as well as the relevant metadata necessary to interpret and use those data, accessible to the user without undue delay, of the same quality as is available to the data holder, easily, securely, free of charge, in a comprehensive, structured, commonly used and machine-readable format and, where relevant and technically feasible, continuously and in real time.

[5]An obligation to make data available to a data recipient shall not oblige the disclosure of trade secrets.

[6]A public sector body, the Commission, the European Central Bank, a Union body or a third-party receiving data under this Chapter shall not use the data or insights about the economic situation, assets and production or operation methods of the data holder to develop or enhance a connected product or related service that competes with the connected product or related service of the data holder.

Figure 1
A top-level diagram of the ontology, depicting most relevant classes. (Red tiles signify classes from the *DAOnt* ontology, blue from *ODRL*, green from *DPV*)

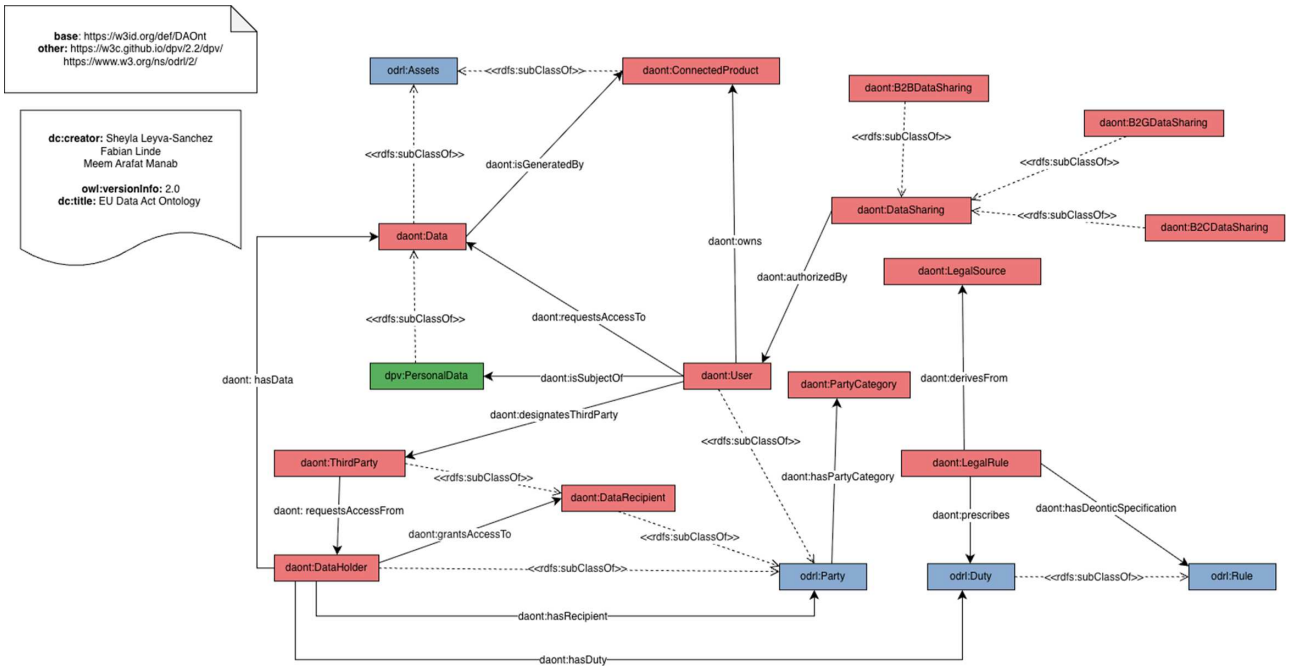


Table 2
Comparison of three Data Act articles

Aspect	Article 4(1) B2C	Article 8(6) B2B	Article 19(2)(a) B2G
Legal Text	"shall make data available without undue delay"	"not required to make data available where data constitute trade secrets"	"shall not use data to develop competing products"
Deontic Type	Obligation (odrl:Duty)	Permission (odrl:Permission)	Prohibition (odrl:Prohibition)
Condition	User requests data access	B2B request + trade secret	None (unconditional)
Action	*provideDataAction*	*refuseDataAccessAction*	*UseDataToDevelopCompetingProduct*
Exception	None	*containsTradeSecretCondition*	None
Assignee	Data holder (must act)	Data holder (may refuse)	Public sector body (cannot act)

noncompliant scenarios across the three data-sharing contexts: B2C, B2B, and B2G. The class instances representing these six scenarios are available in the GitHub repository.[7]

B2C Context (Code Snippet 1). In the violation scenario, `charlie` (a `ConsumerUser`) owns `smartWatch1`, which generates `charlieHealthData`. Charlie requests access to these data, but `watchManufacturer` (a `DataHolder` and `Manufacturer`) fails to perform any `DataProvision` action, thereby violating the mandatory obligation in Article 4(1). Data sharing is governed by `contract_charlie` (Figure 2).

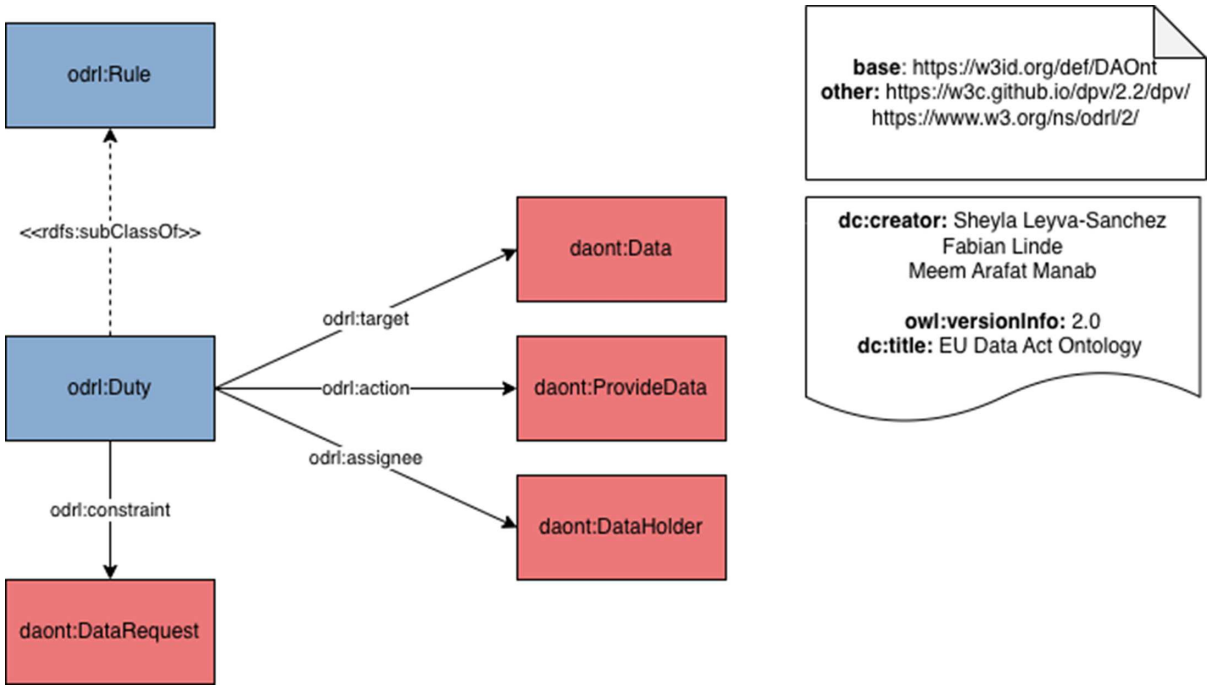
To detect this violation automatically, the SPARQL query in Code Snippet 1 searches for B2C data-sharing cases where a `ConsumerUser` requests access to data linked to a product they own or use and checks whether the associated `DataHolder` performs the required `DataProvision` action. The `FILTER NOT EXISTS` block encodes the notion of a *missing obligation*: if no `DataProvision` action is found, the pattern is flagged as non-compliant. The corresponding compliant scenario satisfies the obligation, and therefore, the query does not return violations.

Code Snippet 1
B2C: Missing Obligation

```
SELECT ?holder ?data
WHERE {
 ?sharing a da:B2CDataSharing;
       da:governedBy ?c .
 ?c dpv:hasRecipient ?user .
 ?user a da:ConsumerUser;
     da:ownsOrUses ?product;
     da:requestsAccessTo ?data.
 ?holder a da:DataHolder;
      dpv:hasData ?data .
 FILTER NOT EXISTS {
  ?holder da:performsLegalAction ?provision .
  ?provision a da:DataProvision .
 }
}
```

[7] https://github.com/oeg-upm/DAOnt/tree/main/compliance-checks/contracts

Figure 2
Diagram depicting the principal classes and properties involved in DA Art. 4(1), based on the ODRL design pattern. (Red tiles signify classes from the *DAOnt* ontology, blue from *ODRL*)



B2B Context (Code Snippet 2). In the violation scenario, `factoryOwnerAcme` (an `EnterpriseUser`) owns `industrialRobot1`, which generates `robotData1`. The factory owner authorizes `autoRepair` (an `AftermarketServiceProvider` and `DataRecipient`) to access these data through `agreement247`, governed by `contract247`, which specifies FRAND-compliant terms (`frand247` with `isFair`, `isReasonable`, and `isNonDiscriminatory` all set to true). Despite these conditions, `robotManufacturer` (a `DataHolder`) refuses to share the data without providing a trade secret justification, violating the conditional sharing requirement in Article 8(6). The compliant scenario provides access under the same FRAND conditions and includes a valid justification whenever disclosure is refused (Figure 3).

A clarification on the deontic framing is warranted here, since Article 8(6) is not an obligation in its own right. It is drafted as a conditional *exception*: it permits a data holder to withhold data that constitute trade secrets, and a permission, on its own, cannot be "violated." What is actually breached is the underlying obligation to make data available—arising from Articles 5 and 8—which continues to bind the data holder whenever the trade secret condition is *not* met, and no other exception applies. In our model, we therefore treat reliance on the Article 8(6) permission as carrying an *implied obligation*: a party that refuses access must show that the condition holds by supplying a trade secret justification. Compliance checking under 8(6) accordingly verifies exactly this—whether a data holder invoking the exception has satisfied and evidenced its condition—rather than checking a permission for "violation" in any literal sense. This is also why 8(6) still serves as our permissive case study: the modality it introduces is a permission, and the check consists in confirming that the permission has been properly relied upon.

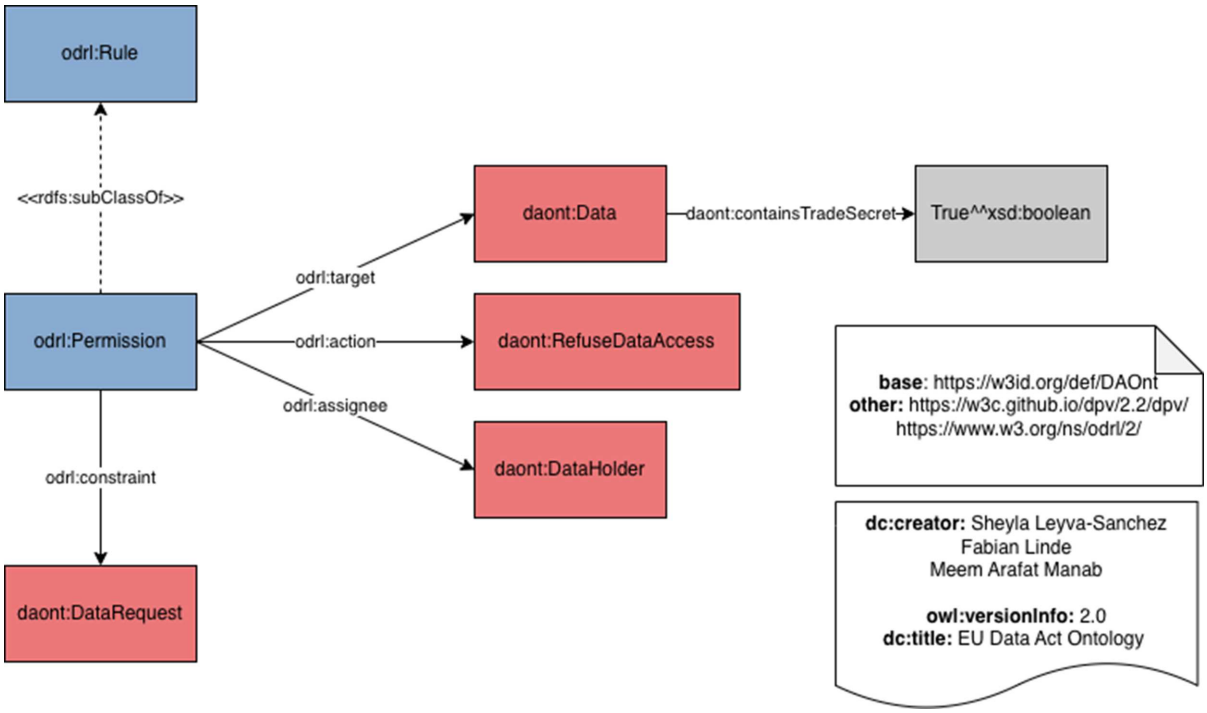
To detect this violation automatically, the SPARQL query in Code Snippet 1 searches for B2B data-sharing cases where a `DataHolder` does not perform the required `DataProvision` action and simultaneously fails to provide a `containsTradeSecret` justification. The first `FILTER NOT EXISTS` block encodes the absence of the mandatory action, while the second captures the absence of the legally permitted exception. When both conditions hold, the query precisely identifies the noncompliant behavior described in Article 8(6).

Code Snippet 2
B2B: Refusal Without Trade Secret Justification

```
SELECT ?holder ?recipient
WHERE {
 ?sharing a da:B2BDataSharing;
        da:governedBy ?c;
        da:authorizedBy ?user .
 ?user da:ownsOrUses ?product .
 ?product da:generatesData ?data .
 ?holder a da:DataHolder;
       dpv:hasData ?data .
 ?c dpv:hasRecipient ?recipient .
 FILTER NOT EXISTS {
  ?holder da:performsLegalAction ?provision .
  ?provision a da:DataProvision .
 }
 FILTER NOT EXISTS {
  ?data da:containsTradeSecret ?s .
 }
}
```

B2G Context (Code Snippet 3). In the violation scenario, `gonzalo` (a `ConsumerUser`) owns `healthMonitor1`, which generates `gonzaloHealthData`. The `healthAuthority` (a `PublicSectorBody` and `DataRecipient`) legitimately requests access to these data from the `healthDeviceManufacturer` under `publicHealthEmergency2024`, an instance of

Figure 3
Diagram depicting the principal classes and properties involved in DA Art. 8(6), based on the ODRL design pattern. (Red tiles signify classes from the *DAOnt* ontology, blue from *ODRL*, gray signifies XML Schema Definitions)



`ExceptionalNeed` authorizing B2G sharing for pandemic monitoring. This exchange is governed by `contract191`. However, after obtaining the data, the authority performs `competitiveProductDevelopment1` (a `UseDataToDevelopCompetingProduct` action), thus using the data to develop a competing health application—an explicit violation of the absolute prohibition in Article 19(2)(a). The compliant scenario demonstrates lawful use restricted strictly to the declared public interest purpose (Figure 4).

To detect this violation automatically, the SPARQL query in Code Snippet 3 searches for B2G data-sharing cases where a `PublicSectorBody` performs an action classified as `UseDataToDevelopCompetingProduct`. Unlike the B2C and B2B contexts, which require detecting *missing obligations* or *missing exceptions*, Article 19(2)(a) defines an absolute prohibition. For this reason, the query simply checks for the presence of the prohibited action; any match directly corresponds to noncompliance according to the regulation.

Code Snippet 3
B2G: Prohibited Action

```
SELECT ?publicBody ?action
WHERE {
 ?sharing a da:B2GDataSharing;
       da:governedBy ?c .
 ?c dpv:hasRecipient ?publicBody .
 ?publicBody a da:PublicSectorBody .
 ?holder a da:DataHolder;
      dpv:hasData ?data;
      dpv:hasRecipient ?publicBody .
 ?publicBody da:performsAction ?action .
 ?action a da:UseDataToDevelopCompetingProduct.
}
```

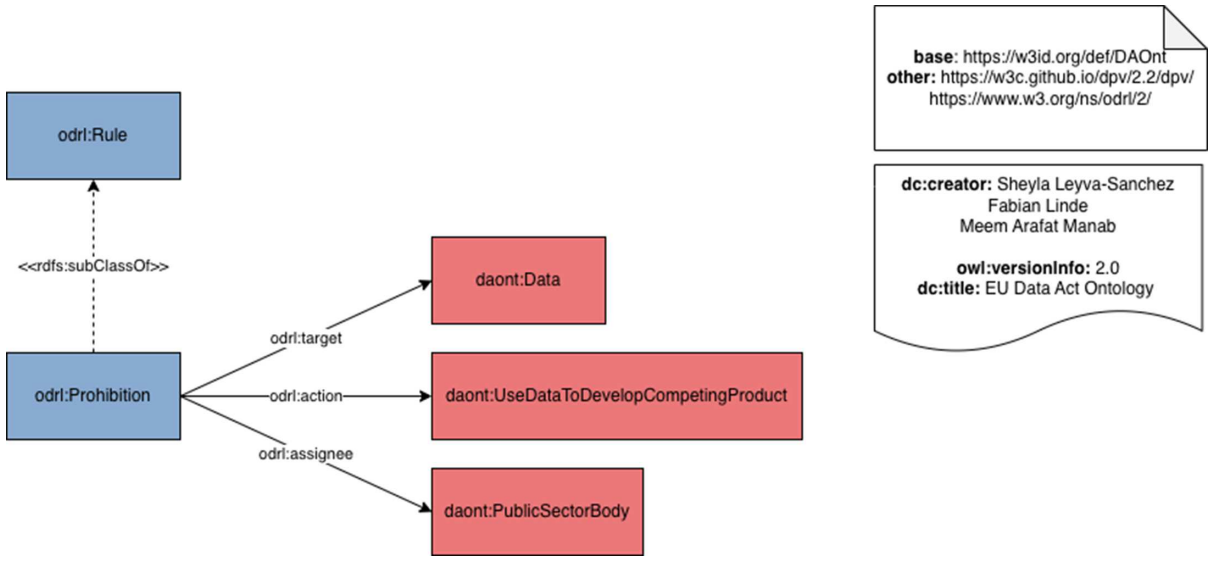
These queries illustrate how SPARQL operationalizes the semantic structures defined in DAOnt to support automated compliance checking. By encoding mandatory obligations, conditional permissions, and absolute prohibitions as graph patterns, queries leverage the formal semantics of the ontology to detect violations directly from the data. This shows the practical value of combining ontology-driven modeling with query-based reasoning: compliance verification becomes scalable, transparent, and auditable, and regulatory requirements can be enforced without bespoke procedural logic.

4.2. Automated compliance verification engine

The queries above are meant to run against RDF knowledge graphs and flag Data Act violations without human prompting. One can picture them wherever data changes hands under a regulatory obligation: inside a data space such as Gaia-X[8], on a sectoral platform for health or mobility, or behind an enterprise's data-sharing Application Programming Interface (API). In such a setting, a compliance module would watch what happens and record each relevant event—a contract signed, a request made, data handed over—as an RDF instance describing a real case, with real parties, datasets and timestamps. That recording is part automatic and part manual: APIs and smart contracts supply the events that can be captured mechanically, while a human or an Large Language Model (LLM) steps in for the concepts that resist symbolic definition, such as whether a delay was "undue" (Article 4) or whether two products "compete" (Article 19(2)(a)). The queries are then run on a schedule, or fired by an event, to catch violations as they arise. One design choice matters more than the rest: the check rests on the Closed World Assumption. `FILTER NOT EXISTS` reads a missing instance as a

[8] In Gaia-X, these are "Compliance Services" within the Trust Framework.

Figure 4
Diagram depicting the principal classes and properties involved in DA Art. 19(2)(a), based on the ODRL design pattern. (Red tiles signify classes from the *DAOnt* ontology, blue from *ODRL*)



violation—absence is taken to mean noncompliance—which is the opposite of OWL's Open World Assumption, where a gap in the data merely means the answer is unknown. Since Data Act breaches usually happen through inaction, reading absence as breach is exactly what compliance auditing needs.

To show that this works in practice, we built a first prototype in Python, using the RDFLib library to hold and query the graph. The foundational DAOnt schema and the synthetic contract instances are loaded together into a single RDF graph, and running the compliance queries from the previous section straight against it is enough to surface the noncompliant cases on its own. On top of this, we put a small Streamlit dashboard, which turns the raw output of the reasoning into readable, auditable reports. Three things stand out:

- Efficiency: The application processes all six synthetic contracts in milliseconds, confirming the high throughput of the ontology-driven approach. This high processing speed contrasts sharply with the time and manual effort a human auditor would require to review contracts against regulatory text.
- Interpretable Results: Results are inherently transparent due to their foundation in SPARQL graph pattern matching on the DAOnt. The system provides a direct, traceable explanation by identifying the specific SPARQL rule and the exact entities involved in the violation.
- Auditable Oversight: The prototype facilitates immediate, interactive feedback for regulatory oversight by linking factual contractual data directly to the formal legal requirements encoded in DAOnt.

The operational prototype can be accessed at https://daont-verify.streamlit.app/

5. Evaluation and Discussion

5.1. Coverage assessment

We evaluated ontology coverage against the Data Act's core chapters using OnToology's automated documentation pipeline. The current implementation covers 18 articles in three primary contexts: B2C user access rights (Chapter II, Articles 3–7), B2B data sharing with FRAND conditions (Chapter III, Articles 8–13), and B2G exceptional need provisions (Chapter V, Articles 14–22). Articles 4(1), 8(6), and 19(2)(a) serve as representative formalizations demonstrating the modalities of obligation, permission, and prohibition, respectively. The remaining gaps include cloud switching mechanisms (Chapter VI, Articles 23–31), interoperability requirements (Chapter VII, Articles 32–34), and smart contract provisions (Chapter VIII, Article 30).

5.2. Expressiveness analysis

OOPS! (OntOlogy Pitfall Scanner) detected no critical errors in the ontology structure. Minor pitfalls include missing domain/range declarations for seven properties (P11 warning) and absence of inverse relationships for four object properties (P13 warning), both intentional design decisions to maintain compatibility with DPV's lightweight RDF Schema (RDFS) approach rather than full OWL 2 Description Logic (OWL2-DL) constraints. The ontology successfully expresses complex deontic scenarios including conditional permissions (trade secret exceptions requiring `containsTradeSecretCondition`), multi-party chains (user → data holder → third party recipient), temporal constraints (Art. 4(1) "without undue delay"), and compensation mechanisms (Art. 12 FRAND terms).

5.3. Interoperability benefits, limitations, and challenges

Integration with four established standards enables multiple deployment scenarios. DPV alignment allows direct reuse in existing GDPR compliance systems (e.g., consent management platforms already using DPV vocabularies). ODRL compatibility supports data space implementations like GAIA-X requiring machine-readable policies. SPARQL queries execute on standard triple stores (GraphDB, Apache Jena, Virtuoso) without custom reasoning engines. LKIF-Core alignment

facilitates future integration with legal reasoning systems that model Hohfeldian relationships. OnToology-generated documentation (HTML, diagrams, metadata) reduces adoption barriers through comprehensive reference materials.

Three primary limitations constrain current applicability. First, the *semantic vagueness* of legal text requires human interpretation for edge cases. We deliberately speak of "vagueness" rather than "ambiguity" here: an ambiguity admits a small number of discrete readings and is settled by an either-way decision, whereas vagueness is an unavoidable feature of legal rules that instead calls for judgment about whether a given set of facts falls under the rule. The `containsTradeSecret` predicate of Article 8(6) is a case in point—"trade secret" has no sharp boundary and must be assessed case by case, so domain experts are needed to instantiate it. It differs, however, from the vague but purely factual notions of an "undue" delay (Article 4) or a "competing" product (Article 19(2)(a)) discussed in Section 4.2: whether something qualifies as a trade secret is partly a question of *law*, so the required human input must come from a legal domain expert, and even that expert judgment will still contain expressions that remain vague at the factual level. In none of these cases is vagueness an error to be resolved once and for all; it is handled by the hybrid human/automated instantiation described earlier. Second, dynamic policy evolution remains unaddressed; contract modifications require manual knowledge base updates rather than automated policy versioning. Third, enforcement mechanisms lie outside ontology scope—while SPARQL queries detect violations, actual enforcement (notifications, sanctions, dispute resolution) requires external system integration. Validation with legal practitioners and industry stakeholders remains ongoing, with preliminary feedback indicating usability challenges for nontechnical users requiring simplified tooling layers.

6. Conclusions

This work introduces the first formal ontology designed specifically to support compliance with the EU Data Act, addressing a critical gap in the landscape of machine-readable regulatory frameworks. By reusing and aligning concepts from ODRL, DPV, and LKIF-Core, DAOnt provides a coherent semantic model to represent rights, obligations, permissions, and prohibitions in data-sharing agreements in B2C, B2B, and B2G contexts.

The three scenarios show that pairing semantic modeling with SPARQL is enough for compliance checking that is both practical and fine-grained. The queries catch a missing obligation, a refusal offered without justification, and an outright prohibited act, three quite different kinds of failure, and they do so directly on the ontology, with no bespoke logic in between. That is reason to think ontologies can carry the weight of transparent, auditable, and scalable compliance verification.

Looking further ahead, DAOnt is a starting point for tools we have not built yet: affordable compliance automation, contractual templates that data spaces can share, and data governance infrastructure that is aware of the policies it enforces. As organizations brace for the Data Act to take effect, an ontology-based route like this one is a credible way toward compliance that is robust, trustworthy, and readable by machines.

7. Future Work

Several directions remain open to expand the ontology and its compliance verification capabilities. First, we plan to conduct structured consultations with legal experts to refine the modeling of rights, obligations, permissions, and exceptions, ensuring that the ontology captures the nuances of legal interpretation and domain-specific practice. Second, the current proof of concept focuses on three representative articles; future work will expand the compliance checker to cover a broader range of provisions from the Data Act, including cross-cutting obligations, sector-specific rules, and complex exceptional need scenarios. Third, we aim to explore the semi-automatic generation of Data Act–compliant contractual templates driven by our ontology, enabling machine-generated contracts to serve as executable test cases and reusable reference examples for automated compliance verification.

Funding Support

This research is funded by the European Union's Horizon 2020 program under the Marie Skłodowska-Curie grant agreement no. 101169409 (HARNESS) and by MALTA PID2024-159504OB-I00 funded by MICIU/AEI/10.13039/501100011033.

Declaration on Generative AI

Claude (Anthropic) was used as a writing assistant for text drafting, document structuring, and editorial revision and as a coding assistant for minor tasks such as formatting and refactoring. Contributions were authored by the authors, who take full responsibility for the work.

Ethical Statement

This study does not contain any studies with human or anim al subjects performed by any of the authors.

Conflicts of Interest

Víctor Rodríguez-Doncel is the Editorial Board Member for *Journal of Computational Law and Legal Technology* and was not involved in the editorial review or the decision to publish this article. The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available on GitHub at https://github.com/oeg-upm/DAOnt, and the DAOnt ontology is published at https://w3id.org/def/daont.

Author Contribution Statement

Sheyla Leyva-Sánchez: Conceptualization, Methodology, Software, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Fabian Linde:** Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing. **Meem Arafat Manab:** Software, Writing – original draft, Writing – review & editing. **María Poveda-Villalón:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Víctor Rodríguez-Doncel:** Conceptualization, Writing – review & editing, Supervision, Project administration, Funding acquisition.

References

[1] Kerber, W. (2024). EU Data Act: Will new user access and sharing rights on IoT data help competition and innovation? *Journal of Antitrust Enforcement*, *12*(2), 234–240. https://doi.org/10.1093/jaenfo/jnae011

[2] de Oliveira Rodrigues, C. M., de Freitas, F. L. G., Barreiros, E. F. S., de Azevedo, R. R., & de Almeida Filho, A. (2019). Legal ontologies over time: A systematic mapping study. *Expert Systems with Applications*, *130*, 12–30. https://doi.org/10.1016/j.eswa.2019.04.009

[3] Hoekstra, R., Breuker, J., Di Bello, M., & Boer, A. (2009). Lkif core: Principled ontology development for the legal domain. In J. Breuker, P. Casanovas, M. C. A. Klein, & E. Francesconi (Eds.), *Law, ontologies and the semantic web* (pp. 21–52). IOS Press. https://doi.org/10.3233/978-1-58603-942-4-21

[4] Palmirani, M., Governatori, G., Rotolo, A., Tabet, S., Boley, H., & Paschke, A. (2011). Legalruleml: XML-based rules and norms. In *Rule-Based Modeling and Computing on the Semantic Web of Lecture Notes in Computer Science*, 298–312. https://doi.org/10.1007/978-3-642-24908-2_30

[5] Iannella, R., & Villata, S. (2018). *ODRL Information Model 2.2, W3C Recommendation REC-odrl-model-20180215*. World Wide Web Consortium (W3C). https://www.w3.org/TR/2018/REC-odrl-model-20180215/

[6] Iannella, R., Steidl, M., Myles, S., & Doncel, V. R. (2018). *ODRL Vocabulary & Expression 2.2, W3C Recommendation REC-odrl-vocab-20180215*. World Wide Web Consortium (W3C). https://www.w3.org/TR/2018/REC-odrl-vocab-20180215/

[7] Robaldo, L., Bartolini, C., & Lenzini, G. (2020). The DAPRECO knowledge base: Representing the GDPR in LegalRuleML. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 5688–5697. https://aclanthology.org/2020.lrec-1.698/

[8] Pandit, H. J., Esteves, B., Krog, G. P., Ryan, P., Golpayegani, D., & Flake, J. (2024). Data privacy vocabulary (DPV)—version 2.0. In *The Semantic Web—ISWC 2024 of Lecture Notes in Computer Science,* 171–193. https://doi.org/10.1007/978-3-031-77847-6_10

[9] Pandit, H. J., Debruyne, C., O'Sullivan, D., & Lewis, D. (2019). GConsent: A consent ontology based on the GDPR. In *The Semantic Web of Lecture Notes in Computer Science*, 270–282. https://doi.org/10.1007/978-3-030-21348-0_18

[10] Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., & Robaldo, L. (2018). PrOnto: Privacy ontology for legal reasoning. In A. Kö & E. Francesconi (Eds.), *Electronic Government and the Information Systems Perspective (EGOVIS 2018)*(pp. 139–152). Springer. https://doi.org/10.1007/978-3-319-98349-3_11

[11] Kurteva, A., Chhetri, T. R., Pandit, H. J., & Fensel, A. (2024). Consent through the lens of semantics: State of the art survey and best practices. *Semantic Web: – Interoperability, Usability, Applicability*, *15*(3), 647–673. https://doi.org/10.3233/SW-210438

[12] Esteves, B., & Rodríguez-Doncel, V. (2024). Analysis of ontologies and policy languages to represent information flows in GDPR. *Semantic Web: – Interoperability, Usability, Applicability*, *15*(3), 709–743. https://doi.org/10.3233/SW-223009

[13] Anim, J. K., Robaldo, L., & Wyner, A. (2024). A SHACL-based approach for enhancing automated compliance checking with RDF data. *Information*, *15*(12), 759. https://doi.org/10.3390/info15120759

[14] Robaldo, L., Pacenza, F., Zangari, J., Calegari, R., Calimeri, F., & Siragusa, G. (2023). Efficient compliance checking of RDF data. *Journal of Logic and Computation*, *33*(8), 1753–1776. https://doi.org/10.1093/logcom/exad034

[15] Chhetri, T. R., Kurteva, A., DeLong, R. J., Hilscher, R., Korte, K., & Fensel, A. (2022). Data protection by design tool for automated GDPR compliance verification based on semantically modeled informed consent. *Sensors*, *22*(7), 2763. https://doi.org/10.3390/s22072763

[16] Kurteva, A., Chhetri, T. R., Tauqeer, A., Hilscher, R., Fensel, A., Nagorny, K., ..., & Demidova, E. (2023). The smashHitCore ontology for GDPR-compliant sensor data sharing in smart cities. *Sensors*, *23*(13), 6188. https://doi.org/10.3390/s23136188

[17] Cimmino, A., Cano-Benito, J., & García-Castro, R. (2024). Open digital rights enforcement framework (ODRE): From descriptive to enforceable policies. *Computers & Security*, *150*, 104282. https://doi.org/10.1016/j.cose.2024.104282

[18] Dam, T., Krimbacher, A., & Neumaier, S. (2023). Policy patterns for usage control in data spaces. In *Sem4Tra 2023: Fifth International Workshop on a Semantic Data Space for Transport, co-located with SEMANTiCS 2023, CEUR Workshop Proceedings*.

[19] Wu, J., Xue, X., & Zhang, J. (2023). Invariant signature, logic reasoning, and semantic natural language processing (NLP)-based automated building code compliance checking (I-SNACC) framework. *Journal of Information Technology in Construction*, *28*, 1–18. Https://doi.org/10.36680/j.itcon.2023.001

[20] Pareti, P., & Konstantinidis, G. (2022). A review of SHACL: From data validation to schema reasoning for RDF graphs. In M. Šimkus, & I. Varzinczak (Eds.), *Reasoning Web. Declarative Artificial Intelligence* (pp. 115–144). Springer. https://doi.org/10.1007/978-3-030-95481-9_6

[21] De Vos, M., Kirrane, S., Padget, J., & Satoh, K. (2019). ODRL policy modelling and compliance checking. In *Rules and Reasoning of Lecture Notes in Computer Science*, 36–51. https://doi.org/10.1007/978-3-030-31095-0_3

[22] Francesconi, E., & Governatori, G. (2023). Patterns for legal compliance checking in a decidable framework of linked open data. *Artificial Intelligence and Law*, *31*, 445–464. https://doi.org/10.1007/s10506-022-09317-8

[23] Golpayegani, D., Pandit, H. J., & Lewis, D. (2022). AIRO: An ontology for representing AI risks based on the proposed EU AI Act and ISO risk management standards. In A. Dimou, S. Neumaier, T. Pellegrini, & S. Vahdati (Eds.), *Towards a knowledge-aware AI* (pp. 51–65). IOS Press. https://doi.org/10.3233/SSW220008

[24] Golpayegani, D., Pandit, H. J., & Lewis, D. (2024). Developing an ontology for AI Act fundamental rights impact assessments. In *Legal Knowledge and Information Systems—JURIX 2024, Frontiers in Artificial Intelligence and Applications*. IOS Press.

[25] Suárez-Figueroa, M.C., Gómez-Pérez, A., & Fernández-López, M. (2012). The NeOn methodology for ontology engineering. In M. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, & A. Gangemi (Eds.), *Ontology engineering in a networked world* (pp. 9–34). Springer. https://doi.org/10.1007/978-3-642-24794-1_2

[26] Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., & García-Castro, R. (2022). LOT: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, *111*, 104755. https://doi.org/10.1016/j.engappai.2022.104755

[27] Chávez-Feria, S., & Iglesias, C. A. (2022). Chowlk: From UML to OWL automatically. In *Proceedings of the 21st International Semantic Web Conference (ISWC 2022)—Demo Track*.

How to Cite: Leyva-Sánchez, S., Linde, F., Manab, M. A., Poveda-Villalón, M., & Rodríguez-Doncel, V. (2026). Operationalizing the EU Data Act for Automated Compliance. *Journal of Computational Law and Legal Technology*. https://doi.org/10.47852/bonviewJCLLT620210282

Appendix

A Full Ontology Requirements Specification Document (ORSD)

Figure 5
Complete Ontology Requirements Specification Document (ORSD) for DAOnt

Ontology Requirements Specification Document	
1. Purpose	
The purpose of this machine-readable model of the EU Data Act is to enable the verification of compliance of data sharing agreements against Data Act requirements.	
2. Scope	
The ontology models the main regulatory provisions of the Data Act related to B2C, B2B, and B2G data sharing, focusing on deontic relations such as duties, permissions, and prohibitions.	
3. Implementation Language (optional)	
Web Ontology Language (OWL)	
4. Intended End-Users (optional)	
User 1. Regulatory compliance officers seeking to verify contractual conformity. User 2. Data space and cloud service providers implementing machine-readable policies. User 3. Legal experts and policymakers analyzing normative implications.	
5. Intended Uses	
Use 1. Compliance verification for data sharing agreements. Use 2. Automated detection of violations and noncompliance. Use 3. Semantic querying of legal rights and duties.	
6. Ontology Requirements	
a. Non-Functional Requirements	
NFR 1. The ontology must be based on the international, European or de-facto standards in existence or under development.	
b. Functional Requirements: : Lists or tables of requirements written as Competency Questions and sentences	
CQG1. General CQs (9 CQs)	CQG2. CQs Related to PoC article (3 CQs)
CQ1. Who shares data with whom, and under what agreement? CQ2. What actions has a party performed? CQ3. What obligations does a party have? CQ4. Who manufactured a product? CQ5. Who uses a product? CQ6. What service does a product provide? CQ7. When does a legal rule apply? CQ8. Where does data come from CQ9. Who holds the data	CQ10. Which data holders have violated Article 4(1) by failing to provide requested data from connected products? CQ11. Which data holders have violated Article 8(6) by refusing data sharing without providing trade secret justification? CQ12. Which public sector bodies have violated Article 19(2)(a) by developing competing products or services using data obtained through B2G data sharing?