



O-Prime: An ODRL-based Privacy Monitoring Framework for Decentralized Environments

Ahmad Cahyono Adi¹, Uha Isnaini^{2,3}, Muhammad Hanif², and Kabul Kurniawan^{1,3*}

¹ Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia
kabul.kurniawan@ugm.ac.id

² Department of Mathematics, Universitas Gadjah Mada, Indonesia

³ Center for Cryptography and Cybersecurity Research, Universitas Gadjah Mada, Indonesia

Abstract

Digital transformation and the adoption of Artificial Intelligence (AI) have significantly increased personal data processing and reuse at scale across various sectors. While these technologies drive efficiency and innovation, the use of AI models that rely on sensitive data poses potential privacy risks, such as misuse, data leaks, and violations of personal data protection compliance. Decentralized personal data platforms such as Solid have been proposed as an alternative to centralized data architectures by enabling individuals to store and manage their data within user-controlled Pods. However, existing architectures primarily rely on access control mechanisms and lack effective means to monitor how personal data is accessed and used after permission has been granted. This paper presents O-Prime, an ODRL-based privacy monitoring framework designed for decentralized environments. The framework introduces a monitoring gateway that intercepts data access requests between external applications and Solid Pods, enabling policy-based monitoring, semantic audit logging, and context-aware privacy enforcement. O-Prime integrates several Semantic Web standards, including the Open Digital Rights Language (ODRL) for expressing privacy policies, the Data Privacy Vocabulary (DPV) for semantic classification of personal data, and the PROV-O ontology for representing machine-readable provenance logs of data access events. Through this architecture, O-Prime enables transparent monitoring of personal data usage, detection of policy violations, and auditable provenance tracking within decentralized data ecosystems. A prototype implementation demonstrates how the framework can monitor third-party access to sensitive personal data stored in Solid Pods, detect policy violations in real time, and present auditable semantic logs through an interactive dashboard.

Keywords: ODRL, Privacy Policy, Privacy Monitoring, Decentralization, Solid, Knowledge Graph.

*Corresponding author.

1 Introduction

Digital transformation over the past few decades has dramatically changed how personal data is managed and utilized across various sectors. With the increasing connectivity and use of online applications, large-scale collection and processing of user data has become common, both in the context of public service digitization and the private sector. As a result, personal data—once regarded as ordinary information—has evolved into a highly valuable and strategic asset. At the same time, data has become a key driver in the advancement of Artificial Intelligence (AI), playing a crucial role in supporting analysis and automated decision-making [1]. The emergence of generative AI models, such as Large Language Models (LLMs), further intensifies these concerns as they are trained on vast and diverse datasets, including user interaction data. This raises significant concerns regarding data privacy and security, especially the risks of unintended data exposure or leakage, as highlighted by the European Data Protection Board (EDPB) [2].

In response to these challenges, regulatory frameworks such as the General Data Protection Regulation (GDPR) in the European Union¹ and Indonesia’s Personal Data Protection Law (UU PDP)² have been established to enforce strict data handling protocols and to strengthen the protection and governance of personal data. However, in practice, most current data management solutions remain centralized, where service providers or applications have full control over the data they collect from users. This causes an imbalance between service efficiency and individuals’ rights to control their personal data, increasing the risks of misuse, overexposure, and non-compliance with data protection principles [3, 4]. This situation underscores the need for more secure, transparent, privacy-centric data management approaches, particularly based on decentralized data management.

Decentralized personal data platforms have emerged as a promising approach to restoring data sovereignty to individuals. Among the most prominent is Solid, which enables users to store and manage personal data in user-controlled personal online data stores (Pods), supported by implementations such as the Community Solid Server [5, 6, 7]. By separating data from applications and distributing storage across user-controlled Pods rather than a single repository, this architecture allows individuals to determine who may access, process, and use their data for specific purposes without relying on a single service provider. Rooted in Tim Berners-Lee’s vision of a “Web of Agency,” Solid promotes a more ethical and user-centric model of data exchange [8]. Nevertheless, despite its emphasis on user ownership and access control, the current Solid architecture still focuses primarily on regulating who can access a resource, with limited support for governing how data is used after access has been granted [4].

This limitation indicates that decentralized personal data governance must move beyond traditional access control toward richer usage control mechanisms, since access decisions alone are insufficient to regulate constraints related to purpose, context, frequency, and obligations [4]. Within the Solid ecosystem, prior work has shown that usage control can be enforced through rule-based web agents, extending Solid from a permission-oriented framework toward more dynamic policy enforcement [9]. At the same time, machine-readable policy standards such as ODRL have emerged as key enablers of such governance, supported by the broader ODRL landscape [10], enforcement-oriented frameworks such as ODRE [3], and policy evaluation tools such as the ODRL Evaluator [11].

However, an important research gap remains. While existing studies have advanced decen-

¹B. Wolford. What is GDPR, the EU’s new data protection law? GDPR.eu, 2025. Retrieved October 9, 2025. URL: <https://gdpr.eu/what-is-gdpr/>.

²Republik Indonesia. Undang-undang nomor 27 tahun 2022 tentang perlindungan data pribadi, 2022. Retrieved October 9, 2025. URL: <https://peraturan.bpk.go.id/Details/229798/uu-no-27-tahun-2022>.

tralized storage, semantic policy modeling and usage control enforcement, they still provide limited support for transparently monitoring how personal data is actually accessed and used after authorization has been granted [5, 3, 9, 4]. In Solid-based environments, current implementations primarily emphasize authorization through access control, with little visibility into actual data usage, making it difficult to detect excessive access, contextual misuse, or privacy policy violations. Consequently, there is a clear need for a unified framework that not only represents and evaluates policies but also integrates decentralized personal data management with policy-based monitoring and semantic audit logging, enabling continuous oversight and producing transparent, auditable records to support accountable data usage in distributed environments.

In this paper, we propose O-Prime, an ODRL-based privacy monitoring framework designed for decentralized environments such as Solid. It introduces a monitoring gateway that intercepts data access requests between external applications and Solid Pods, enabling policy-based evaluation of data usage operations. O-Prime integrates ODRL for policy specification, DPV for semantic classification of personal data, and PROV-O for recording machine-readable provenance logs of access events. Through this architecture, the system enables context-aware policy evaluation, detection of privacy policy violations, and transparent auditing of personal data usage. The contributions of this paper are as follows: (i) A decentralized privacy monitoring architecture that extends the Solid ecosystem with a policy-aware monitoring gateway. (ii) A semantic privacy monitoring mechanism that integrates ODRL and the DPV to support policy representation and privacy-aware data classification within a unified semantic framework. (iii) An integration of a dynamic State of the World (SoTW) model with PROV-O-based semantic logging that enables context-sensitive policy evaluation and transparent auditing. (iv) An implementation and evaluation of O-Prime through real-world scenarios, demonstrating its effectiveness in detecting policy violations and providing auditable monitoring of sensitive personal data access.

The remainder of this paper is organized as follows. Section 2 describes the related work. Section 3 presents the requirements for privacy monitoring in decentralized environments. Section 4 describes the proposed solution architecture, including the system design, interaction flow, data model, and policy monitoring mechanisms. Section 5 presents the implementation details, technology stack, and user interface. Section 6 provides use-case validation and discussion. Section 7 concludes with a summary, limitations, and directions for future work.

2 Related Work

Prior research has extensively examined the intersection of digital transformation, AI adoption, and personal data governance. The large-scale collection and reuse of personal data enabled by AI systems have been widely recognized as a double-edged sword: while improving efficiency and innovation, they also introduce significant privacy risks, particularly when sensitive or personally identifiable data are involved. Existing studies consistently emphasize that AI governance is inseparable from data governance, as inadequate privacy controls can result in unlawful processing, data leakage, and lack of transparency [1, 2]. These risks are exacerbated by the persistence of centralized personal data management, where service providers retain dominant control over user data. Such architectures weaken individual data sovereignty and increase the likelihood of misuse, opaque reuse, and large-scale compromise. The cyberattack on Indonesia’s national data center further illustrates the structural vulnerability of concentrated data infrastructures [12]. These issues are directly relevant to regulatory frameworks such as the GDPR and Indonesia’s Personal Data Protection Law (UU PDP), which require secure,

transparent, and accountable processing of personal data.³ They are also aligned with broader normative goals related to strong institutions, digital inclusion, and responsible innovation within the Sustainable Development Goals framework.⁴

Within this context, decentralized personal data management has emerged as an important alternative to the traditional platform-centric approach. A key development in this direction is Solid, which proposes a decentralized model in which personal data are stored in user-controlled Pods rather than in application-owned silos [6, 7]. This architecture separates data storage from application logic, allowing individuals to maintain stronger control over access permissions and data portability. The conceptual motivation for such an approach is reinforced by arguments for re-decentralizing the Web, which stress that modern digital ecosystems have become overly centralized and that restoring control to users is necessary for a more open, autonomous, and privacy-respecting Web environment [8]. Inrupt has also explored supporting technologies for Solid-based privacy-preserving systems, including trusted execution environments that protect data during computation [13]. In a related direction, recent work has proposed using Solid together with data wallets and privacy-preserving architectural patterns to support safer interaction between personal data and AI systems, particularly in the context of large language models [14]. These studies show that Solid is not only a storage model, but also a promising foundation for privacy-aware digital ecosystems.

However, decentralized storage alone is not sufficient. Effective privacy-preserving personal data management also requires expressive and machine-readable policy representation, so that permissions, prohibitions, duties, and contextual conditions can be formally specified and enforced. In this regard, the Open Digital Rights Language (ODRL) provides an important standard model for representing usage and access policies.⁵ Related semantic work has also been advanced through the Data Privacy Vocabulary (DPV), which offers a structured vocabulary for expressing privacy-related concepts such as purpose, legal basis, processing categories, technical measures, and risk-related notions.⁶ These semantic standards are highly relevant for constructing privacy-aware architectures because they enable privacy requirements to be represented in interoperable and machine-processable ways. Prior research has shown that ODRL can be used not only to describe policies but also to support compliance checking. For example, De Vos et al. studied ODRL policy modelling and formal compliance checking, demonstrating that semantic policy representations can support reasoning about whether particular actions conform to stated rules [15]. More recent work has extended this direction by capturing requests and contextual information for ODRL-based access and usage control, thereby improving the ability of systems to reason about whether data usage requests are acceptable in context [16]. In addition, interoperable interpretation and evaluation of ODRL policies has been further developed to reduce ambiguity across implementations and to strengthen policy enforcement in distributed semantic environments [17]. These contributions suggest that semantic policy technologies are a crucial building block for privacy-preserving personal data systems.

At the same time, privacy-preserving personal data management must account for the interaction between AI-driven personalization and educational or service-oriented applications. Research on generative AI in learning environments shows that personalized systems can improve performance and motivation, but such systems often depend on user-specific contextual data, thereby increasing privacy stakes [18]. Similarly, broader discussions on generative AI

³Retrieved October 9, 2025. URL: <https://gdpr.eu/what-is-gdpr/>; Retrieved October 9, 2025. URL: <https://gdpr-info.eu/>; Retrieved October 9, 2025. URL: <https://peraturan.bpk.go.id/Details/229798/uu-no-27-tahun-2022>.

⁴Retrieved October 9, 2025. URL: <https://sdgs.un.org/goals>.

⁵Retrieved from <https://www.w3.org/TR/odrl-model/>.

⁶Retrieved from <https://w3c.github.io/dpv/2.2/dpv/>.

for personalized learning emphasize both the potential benefits and the associated technical and ethical challenges, including transparency, responsible data use, and protection of personal information [19]. These studies are important because they illustrate a general tension that extends beyond education: AI systems can create value through personalization only if they access relevant data, but the more contextually rich the data are, the more urgent privacy safeguards become. This tension reinforces the need for architectures that can enable AI-supported services while maintaining user control, policy-aware access, and technical protection of sensitive information.

3 Requirements

In this section, we describe several requirements (**R**) the proposed framework must satisfy to support privacy-aware monitoring of personal data in decentralized environments. Regulations such as the EU Data Act emphasize transparent, fair, and accountable data usage, including oversight after access is granted.⁷ At the technical level, ODRL provides a standard model for expressing usage policies,⁸ while prior work on context-aware policy evaluation highlights the need for incorporating runtime context (e.g., state of the world) and continuous monitoring to ensure compliance [16]. Together, these foundations justify the need for policy-based monitoring, contextual evaluation, and semantic auditability in decentralized environments.

R1. Decentralized Architecture Compatibility The framework must operate within decentralized data ecosystems without requiring modifications to the underlying storage infrastructure. In decentralized personal data platforms such as Solid, data is stored in user-controlled Pods, and applications interact with these Pods through standard web protocols. Therefore, the monitoring framework must function as an intermediary layer capable of integrating with existing infrastructures while preserving the principle of user data sovereignty. This requirement ensures that monitoring functionality can be deployed as an “external gateway” component rather than requiring changes to the core server implementation.

R2. Policy-based Monitoring The framework shall support policy-based monitoring mechanisms that evaluate data access operations according to predefined privacy policies. An existing W3C-standardized policy expression language, such as ODRL, provides machine-readable policies that enable the specification of permissions, prohibitions, and constraints associated with data usage. For example, a policy may specify that a health attribute such as blood type can only be accessed once per user session. Such a policy can be expressed in ODRL by defining a permission rule associated with the target data attribute and a constraint limiting the number of allowed accesses. If a request violates a policy constraint, the system must detect the violation and record the event in the monitoring log.

R3. Privacy Data Identification The framework shall provide mechanisms for personal data identification and classification according to privacy categories. This enables the framework to distinguish between general personal data and sensitive personal data that requires stronger monitoring policies. To support standardized privacy semantics, the framework can utilize DPV, which provides semantic descriptions of personal data categories, legal bases for processing, and privacy contexts. For example, attributes such as a personal identifier may be classified as identifying personal data, while attributes such as blood type may be categorized as special-category personal data related to health information.

⁷European Union. (2023). *Regulation (EU) 2023/2854 (Data Act)*. Official Journal of the European Union. Retrieved from <http://data.europa.eu/eli/reg/2023/2854/oj>

⁸W3C. (2018). *ODRL Information Model*. W3C Recommendation. Retrieved from <https://www.w3.org/TR/odrl-model/>

R4. Context-Aware Policy Evaluation The framework shall support context-aware policy evaluation. Monitoring decisions should consider not only the requested operation but also contextual environmental factors such as time, location, or historical access behavior. Contextual information can be maintained using a structured state dataset describing the current system conditions and monitoring counters. For example, the framework may maintain access counters for specific attributes in order to enforce policies limiting repeated access.

R5. Semantic Audit Logging The framework shall provide a structured logging mechanism capable of recording all monitored data access events in a machine-readable format. Instead of storing logs as plain text entries, the framework should represent monitoring records as semantic provenance graphs. O-Prime allows access events to be modeled as provenance activities associated with agents, resources, and timestamps. For example, an access event may be represented as a provenance activity that records the access timestamp, the accessed resource, the requesting application, and the policy evaluation decision (allowed or denied).

R6. Transparency and User Control The framework shall provide mechanisms that allow users to observe and control how their personal data is accessed by external applications. Users should be able to (i) define sensitive personal attributes, (ii) configure monitoring policies, (iii) inspect access logs and detected policy violations, and (iv) review detailed access histories associated with personal data.

4 Solution Architecture

Based on the requirements outlined in Section 3, we describe our proposed solution architecture, including the data model and representation, system workflow, policy monitoring mechanisms, and audit logging approach.

4.1 Architecture Overview

Our approach extends the standard Solid ecosystem by introducing a **monitoring gateway** implemented within the **O-Prime** framework. The architecture consists of several interconnected layers: (i) Monitoring gateway that intercepts, classifies, evaluates, and logs each request; (ii) Access Control List (ACL) services that perform the standard Solid authorization check after the gateway has evaluated a request; (iii) and Solid Pods that store personal data together with policy and audit artifacts; (iv) and Solid Apps that use consume personal data via standard HTTP; (v) Privacy Monitoring Dashboard, a Solid application that is used to monitor personal data usage. As illustrated in Figure 1, multiple pods (e.g., Alice, Bob, Charlie) are supported simultaneously, enabling the same gateway to monitor a shared decentralized ecosystem.

Two cross-cutting capabilities distinguish the gateway from a plain reverse proxy. **Policy-based monitoring** (R2, R4) continuously evaluates access operations against ODRL policies, considering not only the requested action but also contextual factors such as temporal constraints and usage counters. **Semantic audit logging** (R5) records every access event as an RDF provenance graph using the PROV-O ontology, producing machine-readable, SPARQL-queriable logs that can be integrated with external compliance tools.

4.2 Data Model and Representation

Data in O-Prime is organized into four interrelated layers: (i) *Personal Identity Data*, which describes the data subject using interoperable vocabularies (R3); (ii) *Sensitive Domain Data*,

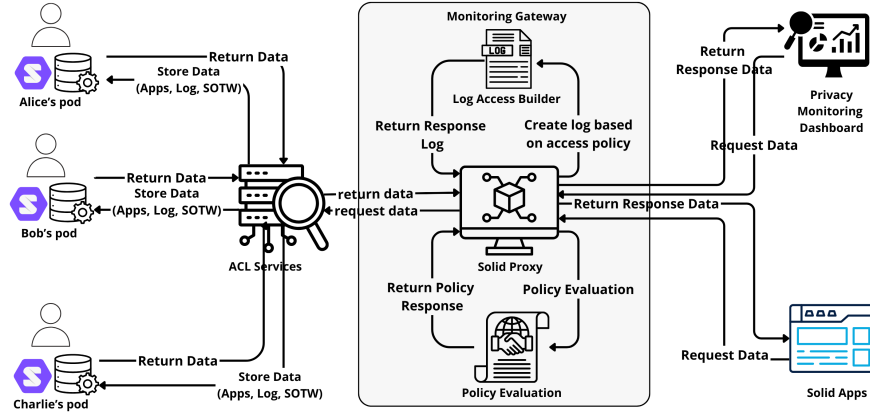


Figure 1: O-Prime monitoring gateway architecture integrating ACL services, policy evaluation, and semantic logging across multiple Solid Pods.

which captures application-specific information enriched with privacy semantics (R3); (iii) *Policy and Monitoring Artifacts*, which define access control rules and contextual constraints (R2, R4); and (iv) *Provenance and Audit Logs*, which record access events as semantic provenance traces (R5, R6). These layers are physically organized within the Pod’s directory structure but remain logically interconnected through RDF links. This design enables the gateway to traverse from a data item to its associated privacy classification, from an access event to the governing policy, and from an audit query to its underlying evidence.

Each Solid Pod stores personal data resources together with the artifacts required for policy evaluation and compliance auditing. Listing 1 presents the directory structure adopted in our approach.

Listing 1: O-Prime Data Organization in Solid Pod

```
public/
  personal-profile.ttl
  health-records.ttl
  application-data.ttl
private/
  dpv-mapping.ttl
  audit/
    access/
      monitor-policy.ttl
      access-log.ttl
    monitoring/
      state-of-the-world.ttl
```

Personal Identity Data The public directory contains RDF datasets representing personal data resources owned by the user. These datasets are structured using widely adopted Semantic Web vocabularies to ensure interoperability and machine readability. The file `personal-profile.ttl` captures general identity information about the data subject using the Schema.org vocabulary, enabling consistent integration with Web-based applications and services.

Sensitive Domain Data The file `health-records.ttl` contains sensitive health-related information. Multiple vocabularies are combined to represent both personal attributes and privacy metadata.

Listing 2: An excerpt of health record in Turtle format (`health-records.ttl`)

```

1 @prefix schema: <https://schema.org/> .
2 @prefix dct: <http://purl.org/dc/terms/> .
3 @prefix dpv: <https://w3id.org/dpv#> .
4
5 <#record> a schema:Person ;
6   dct:created "2026-03-07T09:42:47.158Z" ; dpv:hasLegalBasis dpv:
   ExplicitConsent ;
7   dpv:hasProcessing dpv:Collection ; schema:identifier "1234567" ;
8   dpv:hasPersonalData dpv:IdentifyingPersonalData , dpv:HealthData ;
9   dpv:hasDataCategory dpv:SensitivePersonalData , dpv:
   SpecialCategoryPersonalData ;
10  schema:additionalProperty [ a schema:PropertyValue ; schema:name "
   bloodType" ; schema:value "0" ] .

```

Listing 2 illustrates how a single health record integrates content-bearing properties with privacy semantics. We adopt DPV to represent privacy-relevant metadata directly to the data: `dpv:hasLegalBasis` specifies that processing is grounded in explicit consent, `dpv:hasProcessing` indicates the operation type (here, collection), and `dpv:hasPersonalData` together with `dpv:hasDataCategory` classify the attribute `schema:bloodType` as both identifying and special-category sensitive data under GDPR-like frameworks.

To enable automated privacy classification, personal data attributes are mapped to DPV categories through a dedicated semantic mapping layer. This layer is initialized from a default mapping template provisioned upon Solid Pod creation and can be further customized by users via the O-Prime dashboard. It bridges application-level schemas with privacy-aware semantics by integrating **Schema.org** (`schema:`) for data predicates, **Dublin Core Terms** (`dct:`) for descriptive metadata, and **DPV** (`dpv:`) as the target ontology for privacy classification. Specifically, `dpv:hasPersonalData` links each predicate to its corresponding data type, while `dpv:hasDataCategory` defines its sensitivity level (e.g., `dpv:SensitivePersonalData`, `dpv:SpecialCategoryPersonalData`). This design enables semantics-driven classification through preconfigured mappings, eliminating the need for runtime annotation.

Listing 3: An Excerpt of DPV semantic mapping layer for personal data classification (`dpv-mapping.ttl`)

```

1 schema:identifier a dpv:PersonalData ;
2   dct:description "Unique identifier assigned to a person"@en ;
3   dpv:hasPersonalData dpv:IdentifyingPersonalData ;
4   dpv:hasDataCategory dpv:SensitivePersonalData ;
5   dpv:hasSector dpv:IdentityManagement .
6
7 schema:additionalProperty
8   a dpv:PersonalData ;
9   dct:description "blood type"@en ;
10  dpv:hasPersonalData dpv:HealthData ;
11  dpv:hasDataCategory dpv:SpecialCategoryPersonalData ;
12  dpv:hasSector dpv:Health .
13 ...

```

Listing 3 illustrates how application-level predicates are semantically aligned with DPV concepts through explicit mappings defined in the configuration layer. Each predicate is declared as `dpv:PersonalData` and associated with its semantic type via

`dpv:hasPersonalData` and its sensitivity level via `dpv:hasDataCategory`. While standard predicates such as `schema:identifier` are directly mapped, extensible attributes such as `schema:additionalProperty` are used to represent domain-specific data (e.g., blood type). The optional use of `dpv:hasSector` further refines the contextual domain of each attribute (e.g., `IdentityManagement`, `Health`).

During policy evaluation, the gateway resolves each accessed predicate against this pre-configured mapping layer to determine its privacy classification, which is then used to enforce policy constraints consistently across requests.

Contextual State Management The monitoring gateway maintains contextual information about the system environment, represented as the *State of the World (SoTW)* [16]. This data model provides a dynamic and machine-readable representation of the operational context in which policy evaluation occurs. By externalizing environmental variables into a semantic graph, O-Prime enables policies to reference runtime conditions such as time, location, and usage statistics without embedding these constraints directly into policy definitions.

Listing 4: An Excerpt of State of the World data in Turtle format (state-of-the-world.ttl)

```

1 @prefix sotw: <https://w3id.org/odrl/sotw#> .
2 @prefix odrl: <http://www.w3.org/ns/odrl/2/> .
3
4 <urn:sotw:current> a sotw:SotW ;
5     sotw:currentTime "2026-03-07T09:42:42.602Z"^^xsd:dateTime ;
6     sotw:currentLocation <https://www.iso.org/obp/ui/#iso:code:3166:ID> ;
7     sotw:count [ a sotw:Count ; sotw:countValue "0"^^xsd:integer ; odrl:
8         target schema:additionalProperty ] ;
9     sotw:count [ a sotw:Count ; sotw:countValue "0"^^xsd:integer ; odrl:
10         target schema:identifier ] .

```

Listing 4 illustrates how contextual state is encoded using the SoTW vocabulary. The resource `<urn:sotw:current>` captures the current system context through three types of variables: (i) the current UTC timestamp via `sotw:currentTime`, enabling time-aware policy evaluation; (ii) the geographic location via `sotw:currentLocation`, represented as an ISO 3166 country code to support jurisdiction-aware constraints; and (iii) usage counters via `sotw:count`. Each counter associates a numeric value (`sotw:countValue`) with a specific data attribute through `odrl:target`, such as `schema:additionalProperty` or `schema:identifier`.

These contextual variables are directly referenced during policy evaluation. In particular, the usage counters provide the runtime values required by ODRL constraints such as `odrl:count`, enabling the enforcement of usage-control policies without embedding state directly into the policy.

Listing 5: An Excerpt of ODRL Policy (monitor-policy.ttl)

```

1 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
2 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
3 @prefix report: <https://w3id.org/force/compliance-report#> .
4
5 <urn:uuid:2c5c9cc0-c73e-4f78-8905-c08bd427866d> a odrl:Policy ;
6     odrl:uid "urn:uuid:2c5c9cc0-c73e-4f78-8905-c08bd427866d" ;
7     dct:title "Health Attribute Access Limit Policy" ;
8     dct:created "2026-02-13T09:00:00Z"^^xsd:dateTime ;
9     dct:creator foaf:Agent ;
10    odrl:profile <https://w3id.org/dpv/odrl> ;
11    odrl:target schema:additionalProperty ;
12    report:activationState report:Active ;

```

```

13     odrl:permission _:b752_n3-abc1-permission ;
14     odrl:prohibition _:b752_n3-abc1-prohibition .
15
16     _:b752_n3-abc1-permission
17         odrl:assigner foaf:Agent ;
18         odrl:assignee dpv:Recipient ;
19         odrl:action odrl:read ;
20         odrl:constraint _:b752_n3-abc1-constraint .
21
22     _:b752_n3-abc1-prohibition
23         odrl:assignee dpv:Recipient ;
24         odrl:action odrl:distribute .
25
26     _:b752_n3-abc1-constraint
27         odrl:leftOperand odrl:count ;
28         odrl:operator odrl:lteq ;
29         odrl:rightOperand "1"^^xsd:integer .

```

Listing 5 shows how usage-control constraints are defined using `odrl:count`. During evaluation, the gateway resolves this constraint against the corresponding `sotw:countValue` in the SoTW graph, allowing dynamic enforcement of access limits (e.g., restricting read operations) without modifying the policy itself.

Instead of introducing a separate property to indicate whether a policy is active, the system relies on the Compliance Report Model. The activation status of a policy is captured through `report:activationState` during evaluation, where values such as `report:Active` or `report:Inactive` indicate whether the policy is considered in the decision process. This approach avoids redundancy by keeping policy definitions static, while delegating runtime status and enforcement semantics to the evaluation and reporting layer, including `report:deonticState` for compliance outcomes.

Privacy Monitoring Policy Policies governing personal data access in O-Prime are expressed using ODRL⁹, a W3C-standard vocabulary for representing machine-readable permissions, prohibitions, and obligations over digital assets. Listing 5 illustrates a policy governing access to `schema:additionalProperty`. The policy specifies the protected attribute via `odrl:target` and defines both `odrl:permission` and `odrl:prohibition` rules. The permission allows the `odrl:read` action under a usage constraint defined with `odrl:count`, while the prohibition prevents further actions such as `odrl:distribute`. At runtime, the gateway retrieves the relevant ODRL policy and evaluates its constraints against the current SoTW context. Instead of producing ad hoc binary decisions, the outcome is represented using the Compliance Report Model as a set of evaluation states (e.g., `report:activationState`, `report:performanceState`, and `report:deonticState`), which are then recorded in the semantic access log for auditability.

Semantic Access Logging and Compliance Auditing To support traceability and compliance auditing, each access event is recorded using semantic provenance logging based on the PROV-O ontology¹⁰, which standardizes the representation of activities, entities, and agents in data processing.

This model is extended with the ODRL Compliance Report Model¹¹ to capture evaluation outcomes through formal states: `report:activationState`, `report:attemptState`,

⁹<https://www.w3.org/TR/odrl-model/>

¹⁰<https://www.w3.org/TR/prov-o/>.

¹¹<https://github.com/SolidLabResearch/ODRL-Compliance-Report-Model>.

report:performanceState, and **report:deonticState**, indicating whether a policy is active, an action was attempted or performed, and whether it complies with the rule.

As shown in Listing 6, each event is modeled as a **prov:Activity** and **report:PermissionReport**, recording the access time (**prov:startedAtTime**), resource (**prov:used**), and applied policy (**report:rule**).

Compliance is expressed via **report:deonticState**, where **report:Fulfilled** denotes compliant access, **report:Violated** indicates non-compliance, and **report:NonSet** captures inconclusive cases. All events are grouped in a **prov:Collection** using **prov:hadMember**, enabling scalable and explainable auditing.

Listing 6: An Excerpt of Semantic Access Log (access-log.ttl)

```

1 @prefix prov: <http://www.w3.org/ns/prov#> .
2 @prefix report: <https://w3id.org/force/compliance-report#> .
3
4 <urn:uuid:access-log> a prov:Collection .
5 <urn:uuid:access-1772876562606> a prov:Activity , report:PermissionReport ;
6   prov:startedAtTime ``2026-03-07T09:42:42.606Z'``^xsd:dateTime ;
7   prov:used <https://solid-monitoring-addon-project-production.up.railway.
8     app/public/health-records.ttl> ;
9   report:rule <urn:uuid:2c5c9cc0-c73e-4f78-8905-c08bd427866d> ;
10  report:activationState report:Active ; report:attemptState report:
11    Attempted ;
12  report:performanceState report:Performed ; report:deonticState report:
13    Fulfilled .
14 <urn:uuid:access-log> prov:hadMember <urn:uuid:access-1772876562606> .

```

4.3 System Workflow

Figure 2 illustrates how O-Prime intercepts and evaluates data access requests from external applications before they reach the Solid Pod.

The workflow begins with user authentication via the Privacy Monitoring Interface, which communicates with the O-Prime Gateway. The gateway authenticates requests through the Solid Identity Provider using the Solid-OIDC protocol, after which users can manage policies and access resources within their Pod.

O-Prime operates as a gateway add-on that intercepts requests without modifying Solid’s native mechanisms. The “API Services” component interfaces with the Identity Provider, while “ACLServices” represents Solid’s WebACL layer for final access control. The proxy-based “dispatch” and “return” interaction enables request interception and semantic logging while preserving the native security model.¹²

Figure 3 summarizes the monitoring process. Incoming requests are intercepted by the gateway, which extracts contextual metadata such as HTTP method, resource URI, session identifier, and requester identity. These attributes are mapped to the DPV layer to determine data sensitivity (e.g., **dpv:SensitivePersonalData**).

The request is then evaluated by the ODRL policy engine against active policies (Listing 5) and the current State of the World context (Listing 4), including temporal, geographical, and usage constraints. Based on this evaluation, the gateway derives a compliance decision where both compliant and non-compliant requests are forwarded to the Solid Pod for final ACL validation, while the outcome is recorded in the semantic access log as either **report:Fulfilled** or **report:Violated** for auditing purposes.

¹²<https://solidproject.org/TR/oidc-primer#request-flow>

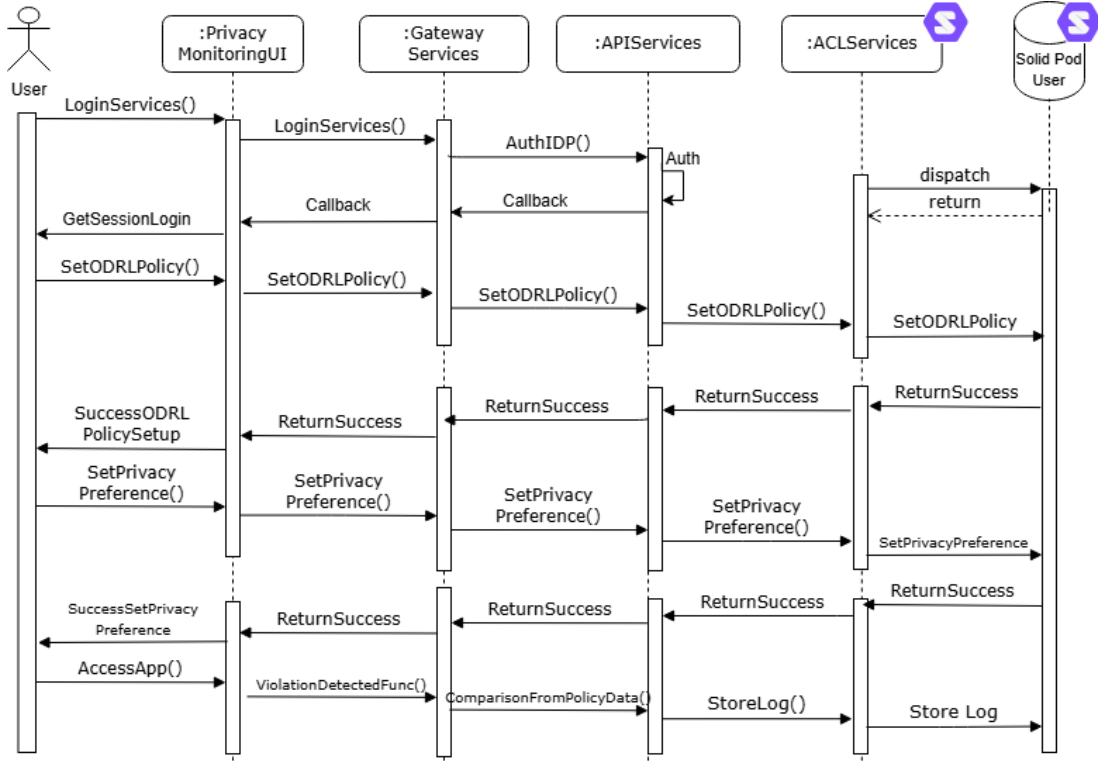


Figure 2: Sequence diagram illustrating interactions between UI services, O-Prime gateway, API services, ACL services, and Solid Pods during policy creation, privacy data storage, and monitored data access.

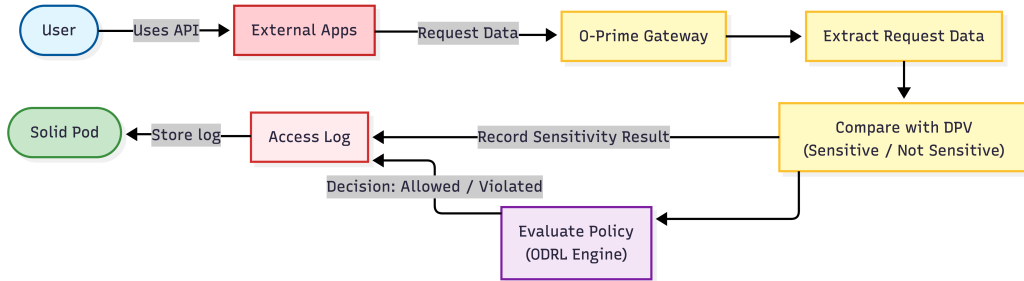


Figure 3: Monitoring flow architecture showing the sequential processing of data access requests through extraction, DPV classification, policy evaluation, and semantic logging.

Regardless of the outcome, each access attempt is recorded as a PROV-O-based semantic log (Listing 6), capturing the timestamp, resource, applied policy, and compliance states. Logs are stored in the user's Solid Pod (e.g., `private/audit/access/access-log.ttl`), ensuring user-controlled, machine-readable audit trails for compliance verification.

5 Prototypical Implementation

O-Prime is implemented as a web-based system consisting of three main components: a *monitoring gateway server*¹³, a *browser-based dashboard interface*¹⁴, and a *demonstration client application*¹⁵. The implementation is built entirely on open web standards and Solid-native interaction patterns, ensuring compatibility with decentralized environments while supporting all defined framework requirements.

5.1 Platform and Technology Stack

The system follows a client-server architecture consisting of a front-end dashboard and a back-end monitoring gateway. The dashboard provides an interactive interface for configuring privacy policies, monitoring data access events, and inspecting detected violations, communicating with the gateway via RESTful APIs.

The back-end is implemented in Node.js and operates as a monitoring gateway positioned between external applications and Solid Pods. It is responsible for request interception, policy evaluation, and semantic logging, while Solid’s authentication mechanism is used to establish secure user sessions and control access to personal data.

From an implementation perspective, the O-Prime gateway is realized as a reverse-proxy service composed with an unmodified Solid server at the infrastructure level. This deployment pattern extends the request-processing pipeline by inserting monitoring logic before requests reach the Pod, keeping the Solid server itself unchanged and interchangeable. Concretely, the gateway is configured as an Express.js middleware chain that binds to the public endpoint of the Solid Pod, intercepting incoming HTTP requests, applying monitoring logic, and forwarding authorized requests to the Solid Pod, thereby preserving the original Solid infrastructure.

5.2 System Deployment and Modules

The monitoring system is deployed as a cloud-based web application comprising two primary components: a dashboard interface¹⁶ and a monitoring gateway server¹⁷. The dashboard interface is hosted on a serverless deployment platform, while the monitoring gateway operates as a Node.js service responsible for handling monitoring operations and facilitating communication with Solid Pods.

The conceptual architecture of O-Prime consists of three integrated Node.js modules that communicate via asynchronous calls and shared in-memory state: *(i)* **Privacy Configuration Module:** Uses `rdflib.js` as a lightweight RDF store to load DPV-based sensitivity mappings from the user’s Pod (e.g., `public/health-records.ttl`). These mappings are cached in a JavaScript `Map` keyed by predicate URIs, enabling fast lookup during request inspection without repeated RDF/SPARQL queries. *(ii)* **Policy Management Module:** Parses ODRL policies (e.g., `private/audit/access/monitor-policy.ttl`) into an internal rule model using `odrl-parser`. Policy evaluation is handled by a custom rule engine that matches request context (HTTP metadata, session state, and SoTW variables) against ODRL permissions and prohibitions. A custom engine is adopted to support tight integration with dynamic SoTW updates (e.g., atomic counters for `odrl:count`) and real-time request interception, which are not

¹³<https://github.com/ahmadvoc12/ODRL-Gateway>

¹⁴<https://github.com/ahmadvoc12/solid-monitor-dashboard>

¹⁵<https://health-record-ujgv.vercel.app/>

¹⁶Dashboard interface: <https://solid-monitor-dashboard.vercel.app/>

¹⁷Monitoring gateway server: <https://solid-monitoring-addon-project-production.up.railway.app/>

natively supported by existing ODRL evaluation frameworks such as ODRE. Updates are persisted via Solid PATCH operations to ensure consistency under concurrent access. *(iii)* **Monitoring Gateway Module:** Implements a Node.js request pipeline that intercepts HTTP requests before they reach the Solid server. Each request is processed through: (i) metadata extraction to identify accessed predicates, (ii) policy evaluation using the extracted context, (iii) SoTW state updates via atomic operations, and (iv) decision enforcement that forwards the request while recording the outcome as `report:Fulfilled` or `report:Violated` in the semantic log.

6 Use-Case Validation & Discussion

This section evaluates O-Prime against the requirements defined in Section 3 through use-case scenarios grounded in personal health data access within a Solid ecosystem.

6.1 Use-Case Validation

We focus on three targeted scenarios that directly exercise the requirements most critical to the framework’s purpose: policy enforcement, sensitive data classification, and semantic audit integrity. Each scenario was executed against the deployed O-Prime prototype using the health-record demonstration client¹⁸, with results confirmed across five repeated runs.

Policy Enforcement and Contextual Rate-Limiting This scenario validates **R2** and **R4** by exercising the full ODRL evaluation engine against a rate-limiting policy applied to `schema:bloodType`. The policy permits exactly one read of the attribute per session; a second read within the same session constitutes a prohibition.

In the first step, the demonstration client issues a `GET` request for `health-records.ttl` with the SoTW counter at zero. The gateway evaluates the active ODRL permission, determines that the counter is below the threshold, forwards the request to the Solid Pod, increments the SoTW counter to one, and records a log entry with decision `ALLOWED` — the expected outcome, confirmed in all five runs. In the second step, the same client issues a second `GET` within the same session, at which point the SoTW counter equals the permitted limit. The ODRL prohibition fires, the counter is held at one, and the log entry records decision `VIOLATED` — again matching the expected outcome across all runs.

The SoTW counter was confirmed to update atomically before the HTTP response was returned in all five runs, preventing any inconsistency under rapid sequential requests. This confirms that the context-aware state model (**R4**) correctly feeds into ODRL rule evaluation (**R2**) as a runtime-consistent, integrated mechanism.

DPV-Based Sensitive Data Classification This scenario validates **R3** by confirming that the DPV mapping layer correctly classifies accessed RDF predicates before policy evaluation, and that classification is transparent to the requesting application. Three predicates of differing sensitivity were tested. `schema:bloodType` was resolved to `dpv:SpecialCategory PersonalData`, reflecting its status as health-related data requiring heightened protection under GDPR-aligned frameworks. `schema:identifier` was resolved to `dpv:SensitivePersonalData`, as a unique personal identifier that can be used to single out

¹⁸<https://health-record-ulgv.vercel.app/>

an individual. `schema:name`, representing non-sensitive general personal data, was correctly classified as `dpv:PersonalData`.

In all five runs, the gateway resolved each predicate to its DPV category via the in-memory mapping cache prior to invoking the ODRL engine, and the classification was not exposed in the response payload returned to the client. This confirms that sensitivity classification is correctly scoped to the gateway’s internal pipeline and does not interfere with application-level data retrieval.

Semantic Audit Log Integrity This scenario validates **R5** and **R6** by examining PROV-O log entries stored in the user’s Pod after policy enforcement. The log file `private/audit/access/ access-log.ttl` was parsed using `rdflib.js` to verify RDF validity and structural completeness. Two entries (one compliant and one non-compliant) were analyzed, each modeled as a `prov:Activity` and containing: `prov:startedAtTime`, `prov:used`, `prov:wasAssociatedWith`, `report:rule`, and the evaluation states (`report:activationState`, `report:attemptState`, `report:performanceState`, `report:deonticState`). The compliance outcome is captured via `report:deonticState` (e.g., `report:Fulfilled` or `report:Violated`).

All entries were valid RDF graphs, and logs were confirmed to be stored within the user’s Pod namespace, satisfying the user-control requirement of **R6**.

6.2 Discussion

Design Contributions O-Prime makes three concrete contributions beyond the state of the art. First, to the best of our knowledge, the first architecture to combine decentralized Solid-based storage with **runtime ODRL policy monitoring**. Existing work treats these concerns separately: the standard Solid ACL model enforces pre-grant authorization but is blind to post-grant usage patterns, while ODRL compliance checkers such as De Vos et al. [15] operate outside a decentralized storage context. O-Prime bridges this gap by inserting a policy-aware gateway into the Solid request path without modifying the server. Second, the integration of the **DPV classification layer (R3)** enables the framework to differentiate between general personal data and special-category data at the predicate level — a granularity that neither ACL-based nor prior ODRL-based systems achieve. This allows policies to be scoped precisely to the sensitivity of the accessed attribute rather than to coarse resource-level permissions. Third, the **SoTW model (R4)** externalizes contextual state into a queryable RDF graph that is decoupled from policy logic. Policies remain declarative and reusable across different users and deployment contexts, while contextual variables — usage counters, time, location — are updated independently without policy reauthoring.

Three limitations are acknowledged. First, the prototype was validated under single-user, sequential-request conditions. Concurrent access from multiple applications to the same Pod introduces race conditions in SoTW counter updates that the atomic PATCH strategy mitigates but does not fully eliminate under high-frequency bursts. Distributed locking or an event-sourced SoTW model would be required for production-scale concurrency. Second, O-Prime’s non-invasive proxy design (**R1**) means that applications with direct network access to the Solid server — bypassing the gateway — are not subject to monitoring. The framework therefore depends on network-level enforcement (e.g., firewall rules or Pod-side ACL restrictions) to ensure all traffic is routed through the gateway. This is a deployment constraint rather than an architectural flaw, but it must be explicitly managed. Third, the current validation covers rate-limiting as the representative contextual constraint. Broader ODRL constructs — temporal

windows, geographic prohibitions, purpose-based restrictions — are expressed in the data model and supported by the SoTW vocabulary, but were not exercised in formal scenarios. Their correctness depends on the completeness of the ODRL rule engine’s constraint resolver, which warrants separate evaluation.

Future Work Three directions are prioritized for future work. First, extending the ODRL rule engine to cover the full range of DPV-defined purpose and legal-basis constraints would enable O-Prime to function as a **GDPR compliance monitor** — flagging accesses that lack a valid legal basis rather than merely those that exceed a usage count. Second, integrating O-Prime into **AI training pipelines** would allow the framework to monitor which personal data attributes are consumed during model training, providing provenance-backed transparency for AI data governance. Third, a **formal user study** with data subjects and data protection officers is needed to evaluate whether the dashboard interface and PROV-O log structure meet practical transparency expectations under the transparency mandate of **R6**.

7 Conclusion

Decentralized personal data platforms such as Solid address the problem of centralized data control, yet their reliance on access control alone leaves personal data usage unobservable once permission has been granted. This paper presented O-Prime, a privacy monitoring framework that closes this gap by introducing a non-invasive monitoring gateway into the Solid ecosystem. By integrating ODRL for policy expression, DPV for predicate-level data classification, and PROV-O for semantic audit logging, O-Prime enables policy-based monitoring, context-aware enforcement, and transparent, user-controlled audit trails — without modifying the underlying Solid server.

Use-case validation confirmed that all six stated requirements are met: the gateway correctly enforces rate-limiting policies through the ODRL engine and SoTW state model, accurately classifies sensitive data attributes at the predicate level using DPV, and produces complete, syntactically valid PROV-O log entries stored under the data subject’s own control. Together, these results demonstrate that Semantic Web Standards can be composed into a coherent, deployable privacy monitoring architecture for decentralized data ecosystems.

O-Prime is positioned as a foundation for privacy-aware AI data governance: its provenance-backed audit layer can be extended to track which personal data attributes are consumed in AI training pipelines, bridging decentralized data management with the transparency demands of responsible AI development.

Acknowledgements

This research is funded by the Indonesian Endowment Fund for Education (LPDP) on behalf of the Indonesian Ministry of Higher Education, Science and Technology and managed under the EQUITY Program (Contract Number: 4301/B3/DT.03.08/2025 and 10107/UN1.P/Dit-Keu/HK.08.00/2025).

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: Grammar and spelling check. After using this tool, the author(s) reviewed and edited the content as needed

and take(s) full responsibility for the publication’s content.

References

- [1] OECD. AI, data governance and privacy: Synergies and areas of international co-operation. OECD Artificial Intelligence Papers 22, OECD, 2024. URL: https://www.oecd.org/en/publications/ai-data-governance-and-privacy_2476b1a4-en.html.
- [2] I. Barberá. AI privacy risks & mitigations – large language models (LLMs). Technical report, European Data Protection Board, 2025. URL: <https://www.edpb.europa.eu/system/files/2025-04/ai-privacy-risks-and-mitigations-in-llms.pdf>.
- [3] Andrea Cimmino, Juan Cano-Benito, and Raúl García-Castro. Open digital rights enforcement framework (ODRE): From descriptive to enforceable policies. *Computers & Security*, 150:104282, 2025. doi:10.1016/j.cose.2024.104282.
- [4] Wout Slabbinck, Wouter Termont, Ruben Dedecker, and Beatriz Esteves. From access control to usage control with user-managed access. *arXiv preprint arXiv:2601.18761*, 2026. URL: <https://arxiv.org/abs/2601.18761>, doi:10.48550/arXiv.2601.18761.
- [5] Joachim Van Herwegen and Ruben Verborgh. The community solid server: Supporting research & development in an evolving ecosystem. *Semantic Web*, 15(6):2597–2611, 2024. doi:10.3233/SW-243726.
- [6] A.V. Sambra, E. Mansour, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Abounaga, and T. Berners-Lee. Solid: A platform for decentralized social applications based on linked data. Technical report, MIT CSAIL and Qatar Computing Research Institute, 2016. URL: <https://cs.brown.edu/courses/csci2390/2021/readings/solid.pdf>.
- [7] E. Mansour et al. A demonstration of the solid platform for social web applications. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 223–226, Republic and Canton of Geneva, Switzerland, 2016. doi:10.1145/2872518.2890529.
- [8] R. Verborgh. Re-decentralizing the web. *Web Fundamentals*, 2025. Retrieved October 7, 2025. URL: <https://rubenverborgh.github.io/WebFundamentals/decentralization>.
- [9] Wout Slabbinck, Julián Rojas Meléndez, Beatriz Esteves, Ruben Verborgh, and Pieter Colpaert. Enforcing usage control policies in solid using rule-based web agents. In Beatriz Estevez, Jan Hofmann, and Sebastian Schmid, editors, *Proceedings of the Posters and Privacy Session of the Solid Symposium 2024*, volume 3947 of *CEUR Workshop Proceedings*, pages 109–117. CEUR, May 2024. URL: <https://ceur-ws.org/Vol-3947/short15.pdf>.
- [10] W3C ODRL Community Group. ODRL landscape. Living document, 2025. Accessed 2026-04-13. URL: <https://w3c.github.io/odrl/landscape/>.
- [11] Wout Slabbinck. ODRL evaluator. GitHub repository, 2026. Open-source software, latest release v0.6.0, accessed 2026-04-13. URL: <https://github.com/SolidLabResearch/ODRL-Evaluator>.
- [12] Reuters. Cyber attack compromised indonesia data centre, ransom sought – reports. *Reuters Technology*, June 2024. Retrieved October 9, 2025. URL: <https://www.reuters.com/technology/cybersecurity/cyber-attack-compromised-indonesia-data-centre-ransom-sought-reports-antara-2024-06-24/>.
- [13] Inrupt. Trusted execution environments with solid. White paper, 2025. Retrieved October 9, 2025. URL: <https://www.inrupt.com/whitepaper/trusted-execution-environments-with-solid>.
- [14] D. Ottenheimer. Secure AI with data wallets: Privacy-preserving solid architecture for personal data LLMs. In *3rd Privacy & Personal Data Management Session @ Solid Symposium 2025*, 2025. URL: <https://openreview.net/forum?id=2BYESPERxb>.
- [15] M. De Vos, S. Kirrane, J.A. Padget, and K. Satoh. ODRL policy modelling and compliance checking. In *Rules and Reasoning: Third International Joint Conference, RuleML+RR 2019*, pages 36–51, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-31095-0_3.

- [16] B. Esteves et al. Capturing requests and context for ODRL-based access and usage control. volume 4093 of *CEUR Workshop Proceedings*, pages 71–84, 2025. URL: <https://ceur-ws.org/Vol-4093/paper5.pdf>.
- [17] W. Slabbinck et al. Interoperable interpretation and evaluation of ODRL policies. In *The Semantic Web: 22nd European Semantic Web Conference (ESWC 2025)*, pages 192–209, Cham, 2025. Springer Nature Switzerland. doi:10.1007/978-3-031-94575-5_12.
- [18] O. Tasdelen and D. Bodemer. Generative AI in the classroom: Effects of context-personalized learning material and tasks on motivation and performance. *International Journal of Artificial Intelligence in Education*, 35:1–24, 2025. doi:10.1007/s40593-025-00491-9.
- [19] Y. Tu, J. Chen, and C. Huang. Empowering personalized learning with generative artificial intelligence: Mechanisms, challenges and pathways. *Frontiers in Digital Education*, 2:19, 2025. doi:10.1007/s44366-025-0056-9.

Appendix

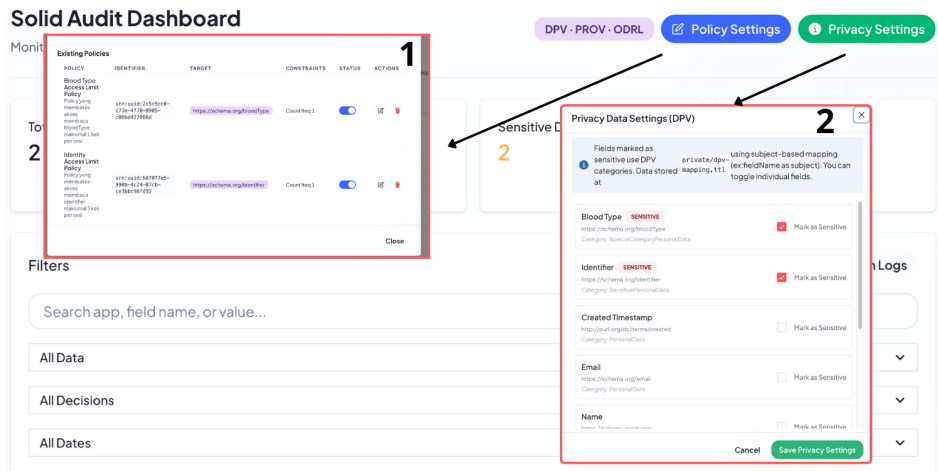


Figure 4: Privacy configuration interface in the O-Prime monitoring dashboard where users define sensitive personal data attributes and configure monitoring policies.

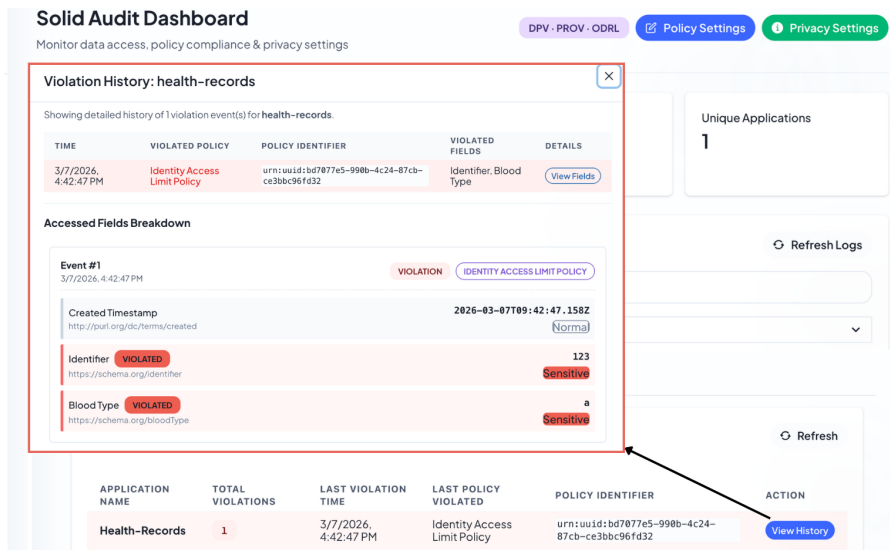


Figure 5: Monitoring log interface displaying detected policy violations and detailed information about accessed data fields.