PrivDev: Mapping Static-Analysis Data Types to DPV

Simon Bernbeck
PUC-Rio
Rio de Janeiro, Brazil
sbernbeck@inf.puc-rio.br

Ricardo Ramalho
PUC-Rio
Rio de Janeiro, Brazil
rjunior@inf.puc-rio.br

Matheus Amendoeira
PUC-Rio
Rio de Janeiro, Brazil
mamendoeira@inf.puc-rio.br

Juliana Alves Pereira
PUC-Rio
Rio de Janeiro, Brazil
juliana@inf.puc-rio.br

Abstract

Static-analysis scanners can identify personal-data types in source code, but they lack mechanisms to connect these findings to standardized privacy vocabularies. PrivDev maps 122 Bearer CLI data types to Data Privacy Vocabulary Personal Data (DPV-PD) categories and links them to potentially relevant GDPR provisions. Our approach combines deterministic mapping for 43 exact-label matches with a retrieval-grounded Large Language Model (LLM) to resolve the remaining 79 non-trivial mappings. The resulting RDF knowledge graph contains 118 ODRL policy resources that were structurally validated using SHACL. The artifact passed five complementary validation gates that cover structural correctness, query consistency, retrieval quality, LLM-based assessment, and human evaluation. In human evaluation, nine annotators produced 711 judgments, yielding a raw agreement of 0.72, Gwet's AC1 of 0.68, and Gwet's AC2 of 0.88. Our results indicate that the proposed mappings are plausible and reproducible, while also revealing ambiguities in scanner-defined data-type labels and coverage gaps in DPV-PD.

Keywords

Static Analysis, Privacy Engineering, LLMs, DPV, GDPR

1 Introduction

Software developers face a growing burden to embed privacy compliance into the systems they build. Regulations such as the European Union's General Data Protection Regulation (GDPR) impose concrete obligations on controllers and processors, yet the translation from legal text to implementation practice remains largely unsupported by tooling. Recent empirical work [18] surveyed 88 Brazilian developers and found that most conflate privacy with security, and that 89.74% of teams handling privacy ad-hoc report using no privacy tools. This finding is consistent with earlier findings showing that developers struggle to embed privacy into software without dedicated tooling and mindset support [9, 23]. They reveal a structural gap: *developers write the code but do not know the law, while privacy lawyers understand GDPR but cannot read code.*

Security scanners such as Bearer CLI [2] use static analysis to detect data types in source code, *e.g.*, identifying that an email address flows through a code path. However, they do not connect those data types to a privacy vocabulary or explain which obligations may warrant review. In contrast, the Data Privacy Vocabulary (DPV) [12, 30] provides a machine-readable taxonomy of personal data categories, processing purposes, and legal bases, but has no direct connection to static-analysis outputs. Our literature search [6, 12, 21] found no approach that directly maps static-analysis datatype labels to DPV categories and links them to potentially relevant GDPR provisions.

This paper presents PrivDev, a security-to-privacy bridge that maps 122 Bearer CLI datatype taxonomy entries to DPV personal data (DPV-PD) categories (`pd:*`) and connects these categories to potentially relevant GDPR provisions. Methodologically, we integrate Design Science Research (DSR) [11], the NeOn methodology [24], and the Framework for Evaluation in Design Science Research (FEDS) [27] to guide the design, ontology re-engineering, validation, and evaluation of PrivDev. This integration provides a systematic approach for transforming an existing static-analysis taxonomy into a machine-readable privacy artifact and evaluating it through complementary automated and human evidence.

The main contributions of this paper are: (1) We propose a two-stage mapping strategy that combines deterministic matching for 43 exact-label correspondences with retrieval-grounded Large Language Model (LLM) use for 79 non-trivial mappings, enabling the systematic treatment of heterogeneous scanner-defined datatype labels. (2) We provide an RDF knowledge graph containing 118 ODRL policy resources and structurally validated using SHACL. The resulting artifact supports SPARQL-based queries that connect static-analysis data types with DPV categories and potentially relevant GDPR provisions. (3) We evaluate the artifact through five complementary validation gates covering structural correctness, SPARQL query consistency, retrieval quality, LLM-based assessment, and human evaluation. The human evaluation comprises 711 judgments from nine annotators and provides evidence of mapping plausibility while revealing ambiguities in scanner-defined datatype labels and coverage gaps in DPV-PD.

Target audience. Our study supports software developers and privacy practitioners seeking to connect source-code analysis with privacy requirements.

2 Background and Related Work

The Data Privacy Vocabulary (DPV) [12, 30], maintained by the W3C Data Privacy Vocabularies and Controls Community Group, provides a machine-readable taxonomy of privacy concepts. It covers personal data categories (`pd:*`), processing purposes (`pu:*`), legal bases (`lb:*`), technical and organisational measures, rights, and other concepts. Its personal-data taxonomy (`pd:*`) comprises 235 classes in DPV 2.3, which are organised under top-level categories (*e.g.*, financial, health, contact, identification, location). DPV is formalised in OWL 2 (Web Ontology Language 2)[1] and extensible through DPV-PD, DPV-OBL, and other modules. The DPV-OBL [31] extension used in this work associates GDPR articles with data categories and processing purposes. We inherit these associations as review cues, not legal determinations of applicability, without independently validating them against legal doctrine or case law.

[1] OWL 2 is a W3C standard for representing ontologies and their semantic relationships in a machine-readable form.

This work encoded the derived privacy rules using three complementary W3C standards: ODRL, SHACL and SPARQL. The Open Digital Rights Language (ODRL) [29] expresses permissions, prohibitions, and duties over digital assets (*e.g.*, if a variable of type `pd:Financial` is processed, the duty of conducting a Data Protection Impact Assessment (DPIA) must be fulfilled). Each Bearer data type produces at least one ODRL rule under our approach. SHACL (Shapes Constraint Language) [28] validates the resulting policies against node shapes that restrict which ODRL properties are allowed for each DPV category (*e.g.*, a `pd:Health` mapping may expose `lb:ExplicitConsent`; a `pd:Financial` mapping may expose retention and DPIA duties). SPARQL [10] is used to write competency questions, expressed as ASK or SELECT queries, that verify each generated policy against its expected response.

Pandit and Esteves [19] present the closest precursor to our work. They manually transform DUO (Data Use Ontology) permission and modifier codes into machine-readable ODRL + DPV policies, representing data-use restrictions, dataset policies, requests and decisions as ODRL instances and expressing privacy and legal-compliance concepts using DPV. Their validation is a proof-of-concept demonstration rather than formal schema validation. Moreover, applying a similar manual alignment process to each static-analysis scanner or personal-data detection tool would require a separate alignment effort [1, 14, 15, 26], limiting scalability across heterogeneous data-type taxonomies. This motivates our retrieval-grounded, LLM-assisted approach, which supports adaptation to other data-type taxonomies[2]. Khedkar et al. [16] extend this human-centered perspective by mapping Android source-code slices to DPV concepts to support privacy assessors during DPIAs. Their approach relies on manual curation and was evaluated through interviews with 16 participants. Instead, our pipeline automatically generates and structurally verifies ODRL/SHACL artifacts, reducing the manual effort required to establish and validate these mappings.

3 Methodology

This study focuses on *PrivDev Layer 1*, the first layer of the proposed security-to-privacy bridge. Layer 1 maps personal-data types detected by the Bearer CLI static-analysis tool to the corresponding categories of personal data in the DPV and connects these categories to potentially relevant GDPR provisions. Future layers may extend this foundation to security weaknesses and broader privacy risks, connecting Common Weakness Enumeration (CWE) security findings to the privacy layer and further relating these findings to risks defined in the OWASP Security and Privacy Top 10.

The methodology combines three frameworks: DSR, NeOn, and FEDS. Figure 1 shows how the pipeline instantiates these frameworks, and the following sections describe their roles and implementation in detail.

3.1 Research Design: DSR, NeOn, and FEDS

This work adopts Hevner's Design Science Research (DSR) framework [11], which treats a research contribution as an *artifact* that must be rigorously evaluated. The Layer 1 bridge repurposes an existing resource for a new context, an *exaptation* in DSR terms: the Bearer taxonomy was built for security scanning rather than privacy reasoning, and re-encoding it in ODRL + DPV adapts it to this new purpose. The artifact is both a model, defining a Bearer-to-DPV mapping, and an instantiation, an executable pipeline that generates, validates, and queries that mapping.

NeOn [24] provides the construction method, *Reusing and Re-engineering Non-Ontological Resources*, which covers transforming classification schemes and thesauri into ontologies: here, the 122-entry Bearer taxonomy into an ODRL + DPV Turtle artifact linking each Bearer type to a DPV `pd:*` category.

FEDS [27] structures the evaluation. Neither Bearer nor DPV publishes an authoritative Bearer-to-`pd:*` alignment, so no ground truth exists. We therefore adopt a convergent-evidence evaluation design that combines automated structural checks (SHACL, OOPS!, and SPARQL competency questions), an LLM-judge quality assessment (DeepEval), and independent human judgment on the non-trivial mappings. Evaluation is performed during artifact construction and after the artifact is complete.

3.2 Construction Procedure

Figure 1 shows the construction procedure's execution order, with each box numbered (1–15); the text below follows the same flow and cites the step numbers.

3.2.1 Input Taxonomy and Per-Entry Dispatch. The procedure starts from a Bearer data-type CSV containing 122 personal-data types (1). A LangGraph outer graph dispatches the entries in parallel (2), with one worker resolving each Bearer type. This preserves an auditable trace per data type while independent mappings run concurrently.

Each worker first checks whether the Bearer type label matches exactly a `pd:*` `skos:prefLabel` (3). These lexically unambiguous cases are resolved deterministically from the DPV taxonomy and skip retrieval and LLM proposal. They are still validated before emission, but are reported separately as `trivial_pass` entries because they do not require semantic classification.

3.2.2 Evidence-Based Mapping. Prior work on tool-augmented ontology alignment finds that LLMs provided with raw OWL files can perform worse than to give the LLM access to the ontology through a retrieval/tool mechanism [22]. We therefore retrieve structured, per-concept chunks containing the label, `skos:broader` hierarchy, definition, and scope note, rather than passing the raw DPV Turtle to the mapping LLM (4). Retrieval uses three FAISS [13] indexes. `ontology_target` contains DPV-PD concepts and provides the candidate classes from which the LLM selects. `detector_aliases` contains synonym labels from personal-data detection tools, while `sector_schema` contains domain-specific field aliases; both provide supporting evidence when a Bearer label lacks a close DPV match. A two-stage retrieve-and-rerank process first uses FAISS to retrieve candidates, and then applies a cross-encoder to rank the candidates and retain a small set for the LLM. The replication package lists the index sizes, embedding model, and top-k values.

3.2.3 Structured Mapping Proposal. For non-trivial entries, the LLM `deepseek-v4-flash` proposes the mapping (5). We selected this model over GPT-4o-mini because of its higher rate limit and lower per-token cost; the task did not require flagship-model reasoning. The prompt provides a compact list of `pd:*` IRI (Internationalized Resource Identifier), the retrieved evidence grouped into four sections (primary DPV candidate, alternatives, synonym aliases,

[2]The present evaluation remains limited to Bearer's 122 data types.

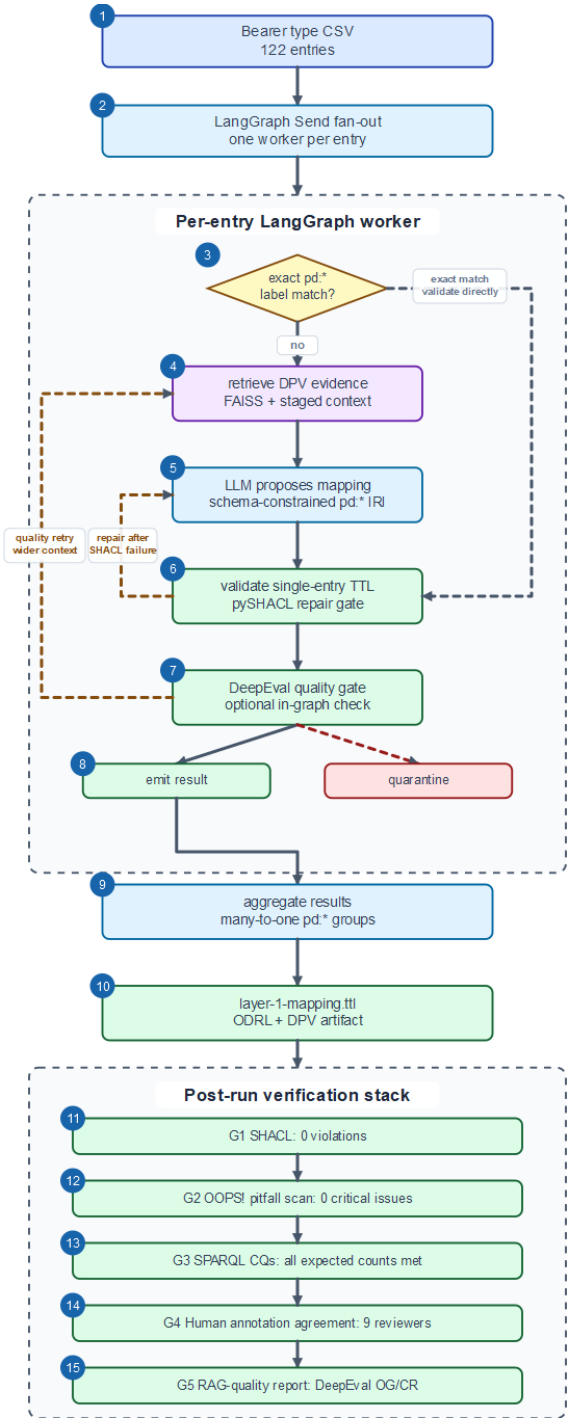

Figure 1: Layer 1 construction pipeline.

sector-schema confirmations), and the instruction to prefer an exact label correspondence over a more specific or broader class.

The LLM does not generate Turtle. Instead, it returns a JSON object with three fields: `pd_iri`, `confidence` (high/medium/low), and `notes` (the verbatim DPV definition supporting the selection). The response is validated against a predefined schema before the artifact is generated. GDPR article IRIs are derived deterministically from the selected `pd:*` class rather than generated by the LLM.

3.2.4 Policy Encoding, Validation, and Aggregation. The mapping is encoded as an ODRL policy through deterministic templates. For each (Bearer type, DPV class) pair, an `odrl:Policy` of type `odrl:Set` targets the Bearer data type to the corresponding DPV obligation and ODRL constraints from the DPV-OBL extension.

Each generated policy is validated in-graph 6 against the SHACL shapes detailed as Gate G1 in Section 3.3.1. After a violation, its text is appended to the LLM prompt and the worker retries, with up to 3 attempts before quarantine. A policy that passes SHACL proceeds to the in-graph DeepEval quality check 7, detailed as Gate G5 in Section 3.3.1. An ontology-grounding failure triggers blocking after retry, widening, and reformulating retrieval before another proposal. Entries that clear both checks are emitted 8; those that exhaust the repair or quality budget are quarantined.

The outer graph aggregates the results per-entry in the final Turtle mapping artifact 9. Entries with the same selected `pd:*` IRI are grouped in a many-to-one policy. This produces the final ODRL + DPV artifact used by the query layer 10. Five post-run gates then evaluate the consolidated artifact: G1 (SHACL 11), G2 (OOPS! 12), G3 (SPARQL competency questions 13), G4 (Human annotation 14), and G5 (DeepEval 15), as detailed in Section 3.3.

3.3 FEDS Evaluation Design

3.3.1 Formative Verification Gates. Table 1 describes the five formative gates. They provide complementary evidence rather than a single measure of correctness. Gates G1–G3 assess the structural integrity and query consistency of the artifact, while G4 and G5 provide evidence about the semantic adequacy of the non-trivial mappings through independent human assessment and retrieval-grounded evaluation.

Table 1: Formative verification gates for PrivDev Layer 1.

Gate/Method	Purpose, scope, and pass criterion
G1 SHACL	Validates the structural well-formedness of generated ODRL policies against three shapes: class well-formedness, required policy links, and Article 9 consistency. Validation is performed during generation and exhaustively on the consolidated artifact. Pass: zero violations.
G2 OOPS!	Scans the consolidated RDF artifact against the complete default OOPS! pitfall catalog [20]. Findings are classified using OOPS!'s Critical, Important, and Minor severity levels. Pass: zero Critical findings; Important and Minor findings are recorded but non-blocking.
G3 SPARQL	Checks structural consistency of the consolidated artifact through five competency questions (CQ-01–05*), covering mapping coverage, Article 9 links, policy links, many-to-one grouping, and DPV namespace validity. It does not assess whether an individual Bearer type is mapped to the semantically correct DPV class. Pass: each CQ returns its expected row count (122 for CQ-01, 0 for CQ-02–05).
G4 Human	Assesses the 79 non-trivial Bearer-to-DPV mappings through independent human judgments (Section 3.3.2). Annotators classify mappings as *Correct*, *Too broad*, *Too narrow*, *Wrong*, or *Unclear*. Pass: raw agreement and Gwet's AC2 both ≥ 0.70 (see Section 3.3.2).
G5 DeepEval	Evaluates the retrieval-grounded mapping process for non-trivial entries using Ontology Grounding (OG) and Contextual Relevancy (CR). OG checks whether the selected `pd:*` concept is supported by the retrieved evidence; CR assesses the relevance of the retrieved context. Pass: OG is the blocking criterion and can trigger retrieval reformulation or quarantine, while CR is diagnostic.

*The complete list of competency questions (CQ-01–05) is in our supplementary material.

3.3.2 Gate G4: Human Annotation Protocol. A mixed panel of software developers and law practitioners reviewed the 79 non-trivial

mappings using a KoboToolbox form[3]. The remaining 43 exact-label matches were resolved deterministically and excluded from human annotation; this does not assume that lexical identity guarantees semantic correctness. For each entry, an annotator selects one of five ordinal judgments: *Correct*, *Too broad*, *Too narrow*, *Wrong*, or *Unclear*. Annotators also rate confidence from 1 to 5. Confidence only flags the rows for qualitative review (Section 5.2). All valid annotators are pooled in one quantitative analysis. The self-reported personal-data-classification experience is also retained as descriptive metadata (Section 5.2) but does not determine inclusion.

Calibration, disagreement handling, and panel. Before annotating, all respondents received a short recorded walkthrough of the research context, DPV/GDPR/LGPD background, and the form, plus written instructions covering judgment-scale definitions, worked examples (e.g. *Email Address* → `pd:Identifier` as *too broad*), and a quality-flag legend; no separate pilot-rating round was run. Annotators judged independently and once each, with no consensus or adjudication round; disagreement is characterized rather than resolved, statistically through the four reliability measures below and qualitatively through coded review of comments and *Unclear* verdicts (Section 5.2). The panel combined software-engineering practitioners and students with law students and legal practitioners; some had prior personal-data-classification experience.

Too broad and *too narrow* are not equidistant from *Correct*. An over-broad DPV class risks mis-claiming or omitting a GDPR Article 9 special-category obligation, which we treat as more severe than an over-narrow class. This is a task-specific modeling choice, not a universal legal rule. An ordinal severity scale s maps the four verdicts to $[0, 1]$:

$$\begin{aligned} s(\text{Correct}) = 1, \quad & s(\text{Too narrow}) = \tfrac{2}{3}, \\ s(\text{Too broad}) = \tfrac{1}{3}, \quad & s(\text{Wrong}) = 0, \end{aligned} \tag{1}$$

Unclear judgments are excluded from all quantitative statistics but counted and routed to qualitative review (Section 5.2). Such a verdict can indicate rater uncertainty or a genuine DPV coverage gap; a numeric placeholder would misrepresent both.

With $q = 4$ scored categories and a correct-heavy marginal distribution, chance-corrected statistics such as Krippendorff's α can read paradoxically low despite high raw agreement [5, 7] (quantified in Section 5.2); we therefore report four complementary statistics. Let n_{ik} be the number of raters assigning item i to category $k \in \{1, \ldots, q\}$, and let r_i be the number of non-*Unclear* judgments on item i. Items with $r_i < 2$ do not contribute to a pairwise statistic; N is the number of items with $r_i \geq 2$.

Raw percent agreement. is the mean pairwise exact-match rate over all unordered rater pairs (a, b) across qualifying items:

$$P_a^{\text{raw}} = \frac{1}{\sum_i \binom{r_i}{2}} \sum_i \sum_{\{a,b\} \subseteq \text{raters}(i)} \mathbb{1}[v_{ia} = v_{ib}], \tag{2}$$

where v_{ia} is rater a's verdict on item i.

Krippendorff's α (ordinal) and Gwet's AC1 (nominal, prevalence-robust). are two chance-corrected statistics computed on the same judgments. α [17] corrects for chance using the observed marginal distribution, and is known to read low under prevalence skew even when raw agreement is high. Gwet's AC1 [7, 8] corrects for chance using the observed *agreement* propensity instead, making it far less sensitive to that same skew. Both reduce to a $(P_a - P_e)/(1 - P_e)$ form over the per-item category counts n_{ik}; full derivations are in Krippendorff [17] and Gwet [7, 8].

Gwet's AC2 (weighted, ordinal + prevalence-robust). extends AC1 with the same quadratic distance weights used for α, combining prevalence robustness with severity-aware scoring:

$$w_{kl} = 1 - \left(\frac{s_k - s_l}{s_{\max} - s_{\min}} \right)^2, \tag{3}$$

applied to AC1's P_a and P_e terms above in place of the unweighted category counts and category propensities — AC2, not α, is read alongside raw agreement and AC1 when judging Gate G4's reliability pass criterion (Section 5.2).

3.3.3 Internal Summative Checks. Once the artifact is complete, three internal summative checks close the evaluation: contribution completeness against an author-defined rubric, method reproducibility through a clean-environment rerun, and a self-assessment rubric covering completeness, correctness, consistency, clarity, and reproducibility. These checks assess process completeness, not independent effectiveness (Section 5.3).

3.4 Research Questions

This study is guided by three research questions (RQs), as follows:

RQ$_1$. *Can the NeOn Classification-Scheme-to-OWL pattern, informed by the DUO/ODRL/DPV encoding mechanics of Pandit and Esteves [19], be adapted to produce a structurally valid mapping from Bearer CLI data-type detections to DPV* `pd:*` *personal-data categories?* This question answers whether a pattern applied manually transfers to larger-scale automated data-type mapping.

RQ$_2$. *Can the resulting SPARQL-queryable, DPV-grounded knowledge graph reliably retrieve potentially relevant GDPR provisions from Bearer CLI findings?* This question answers whether the mappings carry enough structural and semantic grounding to reliably link findings to relevant provisions.

RQ$_3$. *Can a FEDS-structured, convergent-evidence protocol provide a reproducible and sufficiently sensitive basis for evaluating the mappings in the absence of an established ground truth?* This question answers whether converging structural, LLM-judge, and human evidence can credibly assess mapping quality.

4 Implementation

Knowledge Graph Construction. The Layer 1 artifact is constructed by a LangGraph-orchestrated pipeline that reads the 122-entry Bearer taxonomy, produces ODRL policies backed by DPV categories, and emits a consolidated, SHACL-validated knowledge graph together with a per-entry audit log.

Runtime components. The two-level LangGraph structure (outer dispatch, inner per-entry mapping/validation/quality routing) is described in Section 3.2. Each run writes the consolidated Turtle artifact, the per-entry pipeline log, and the RAG-quality report used by the evaluation section.

Artifact outputs. The primary output is `layer-1-mapping.ttl`, a query-visible ODRL graph backed by DPV. The pipeline also

[3] https://www.kobotoolbox.org/

writes a JSON audit log for each processed Bearer type. It retains the parsed mapping decision (pd:* IRI, confidence, quality class, and GDPR article IRIs), the LLM's verbatim-quoted rationale from the structured response, the retrieved context chunks passed to the mapping LLM, and the DeepEval scores and verdicts. The raw unparsed LLM completion is not stored; post-hoc reproducibility checks use the parsed fields and retrieved context.

Cost. The full 122-entry run completes in approximately 63 minutes wall-clock against the mapping LLM's hosted API, at approximately $0.37 in total token cost across roughly 1.91M tokens.

End-to-End Demonstration. To demonstrate the complete pipeline from the source code to potentially relevant GDPR provisions, we provide an end-to-end demonstration using OWASP Juice Shop[4] as the target application (artifact and replication instructions: Artifact Availability). A Bearer CLI dataflow scan over the Juice Shop source tree maps each personal data type detection to its source location: file, line number, field name, and enclosing object. This provenance is additive because the bridge maps only data-type names; source locations decorate the output without affecting obligation retrieval. The consolidated graph is loaded into an RDF triple store with DPV 2.3. A SPARQL SELECT query resolves each Bearer label to its pd:* category and inherited DPV-OBL provision links in one round trip. A REST API exposes this as a repository scan using a public URL. The target codebase is cloned into an isolated temporary environment, scanned, matched against the graph, and discarded before results are returned, so no user source code is retained. The query interface returns potentially relevant GDPR-provision badges and collapsible source-location provenance for a Juice Shop scan.

5 Results and Discussion

5.1 Mapping Coverage and Output

The Layer 1 pipeline processed all 122 Bearer data-type taxonomy entries. All 122 (100%) were mapped to DPV pd:* categories; none were quarantined. These mapped types collapse to 118 distinct pd:* targets (4 pairs share a category via many-to-one grouping). The pipeline emits one consolidated Turtle graph containing 118 odrl:Set policy resources, with Bearer datatypes that share a DPV-PD target grouped into the same policy. Table 2 summarises coverage, inherited DPV-OBL-derived provision links, and mapping-quality class. Applicability depends on the processing operation and purpose, not on the data category alone.

Table 2: Layer 1 coverage: GDPR links and quality classes.

DPV-OBL-derived GDPR provision links		
eu-gdpr:A6-1-a (standard)	104	85%
eu-gdpr:A9-2-a (special category)	18	15%
eu-gdpr:A10 (criminal-data-related)	3	2%
Rows are non-exclusive; a datatype may be linked to multiple provisions.		
Mapping quality class (of 122 mapped types)		
trivial_pass (exact label match)	43	35%
auto_pass (accepted by automated checks)	69	57%
flag_revision (emitted; marked for review)	10	8%

Across the 158 DeepEval judge calls issued for the 79 non-trivial entries (two metrics per entry), mean ontology grounding was 0.941 and mean contextual relevancy was 0.780. Gate results (SHACL, OOPS!, SPARQL competency queries, RAG-quality, and human-annotation agreement) are summarised in Section 5.3.

[4]https://owasp.org/www-project-juice-shop/

5.2 Gate G4: Human Annotation Results

Nine respondents reviewed all 79 non-trivial entries via the protocol described in Section 3.3.2 and are pooled into one quantitative analysis (Table 3).

Table 3: Gate G4 annotation reliability.

Metric	All entries	Art. 9 (n=10)
Raw percent agreement (P_a^{raw})	0.721	0.800
Krippendorff's α (ordinal)	0.251	0.056
Gwet's AC1 (nominal)	0.682	0.785
Gwet's AC2 (weighted)	0.877	0.877

The four statistics defined in Section 3.3.2 (Equations 2–3) disagree sharply on the same data: 72.1% raw agreement versus an α of 0.251 — the high-agreement/low-kappa paradox [5, 7], not a contradiction. Judgments are heavily skewed toward *Correct* (558/706 non-Unclear judgments), and chance corrections based on the observed marginal distribution fall sharply under this prevalence skew. Gwet's AC1/AC2 are designed to resist this distortion. Their values of 0.68–0.88 are consistent with raw agreement rather than α. Scoring *Unclear* as a disagreement category instead of excluding it barely changes raw agreement (0.711 overall and 0.800 for the Article 9 subset), so the exclusion does not inflate the reported figure. Gate G4's criterion, fixed before annotation, requires raw agreement and Gwet's AC2 to both be ≥ 0.70; both thresholds are met (0.721 and 0.877 overall). Krippendorff's α is not part of the criterion and did not determine the gate outcome; we report it for transparency. Its low Article 9 value ($\alpha = 0.056$, n=10) reflects genuine disagreement in a small, high-stakes sample, consistent with the prevalence-skew instability noted above rather than a data artifact. We flag it for downstream attention rather than treating it as resolved, and report Gate G4 as passing its defined criterion.

What this validates. Annotator agreement speaks to the *plausibility* of each Bearer-datatype-to-DPV-PD mapping, not to full GDPR compliance: it does not confirm legal-basis correctness, DPIA applicability, purpose-specific Article 9 escalation, or anything about the Layers 2–3 mappings this paper does not cover.

Disagreement patterns. Comments, *Unclear* verdicts, and low-confidence judgments (78 of 711, extracted programmatically) reveal three recurring patterns, unscored in Table 3, motivating the DPV coverage-gap discussion in Section 6: many flagged disagreements originate in the mapping task itself, not the pipeline.

(1) **Genuine DPV gaps.** *Job Titles* → pd:Job is the closest available class but notably broader than the Bearer type; no narrower class exists. *Lastname* → pd:Identifier persisted after reformulated prompting, since DPV's pd:Name definition ("given name or nickname") excludes surnames and no pd:Surname class exists.

(2) **Upstream scanner ambiguity.** *Emails* → email content could equally denote a contact list (pd:EmailAddress); *Employee Files* → pd:Profile is a composite HR container spanning financial and Article 9 health/union data that Bearer's

own category label ("Professional Information") had already miscategorized before the mapping LLM saw it.

(3) **Purpose-based risk blind spot.** *Correct* verdicts on preference data (*Accents*, *Acquaintances*, *Browsing Behavior*) carry a caveat that *inferential use* (e.g. profiling political opinion, per *CJEU C-300/21 "Austrian Post"*) could trigger Article 9 in a way a category-based mapping cannot capture.

These patterns show, independently of pipeline correctness, where DPV-PD's categories and Bearer's taxonomy reach their representational limits.

5.3 Gate Summary and Summative Evaluation

Table 4 summarizes each gate, the corresponding verification procedure, and its outcome.

Table 4: Formative gate outcomes.

Gate	Observed outcome
G1	**Pass**. Three shapes evaluated 118 policies; zero violations.
G2	**Pass**. Zero Critical; 3 Important and 2 Minor findings were non-blocking.
G3	**Pass**. CQ-01–05 returned all five expected row counts.
G4	**Pass**. Nine annotators; raw agreement 0.721; AC2 0.877.
G5	**Pass**. OG 0.941; CR 0.780; 10/79 mappings flagged for revision.

Internal summative checks. Contribution completeness, a clean-environment rerun, and author self-assessment checked process transparency, not independent effectiveness.

5.4 Threats to Validity

Internal validity: the LLM classification may contain systematic biases not caught by an automated judge scoring the same model's own output. Mitigation: a separate human-annotation round (Gate G4) plus review of flagged mappings.

External validity: the mapping is limited to Bearer CLI data types (122 entries); other scanners use different taxonomies. Mitigation: designed for adaptation to other datatype taxonomies (Section 2), but unevaluated.

Construct validity: DPV coverage is incomplete; some Bearer types (technical identifiers) have no DPV match. Mitigation: these cases are explicitly flagged as coverage gaps and deferred to Layer 2.

Conclusion validity: raw agreement, α, and Gwet's AC1/AC2 disagree sharply on the same data (Section 5.2), so which statistic is foregrounded could shift the reliability conclusion. Mitigation: reported side by side rather than collapsed into one number.

6 Limitations

The pipeline calls the hosted mapping LLM without a temperature or seed override, so outputs can vary; self-hosting remains future work. This matters because structured-output generation has a measurable reliability ceiling [32], and format constraints can degrade reasoning relative to free-form generation [25]. Both are relevant to the 8% `flag_revision` rate reported in Section 5.1. Bearer CLI's Elastic License 2.0 does not affect our artifact, which consumes only its output taxonomy. DPV coverage is also incomplete. Esteves and Rodríguez-Doncel [4] find 31 of 57 GDPR informational items represented, and our Gate G4 annotation (Section 5.2) identifies concrete `pd:*` gaps as candidate DPV extension points. The motivating developer survey [18] is Brazil-specific, although non-Brazilian studies corroborate it [9, 23].

The annotation study evaluates a fixed set of Bearer-to-DPV-PD mappings, not downstream developer decisions. Its 43 `trivial_pass` entries (exact-label matches, Section 5.1) were reviewed only by the pipeline's authors, not human-annotated; we considered this low-risk given the narrow check (Bearer labels matched against DPV-PD), not because it received independent verification. The Juice Shop demonstration — a clone-by-URL scan returning GDPR-provision badges with file/line/field provenance — shows the pipeline works as an artifact, but whether it reduces privacy-review effort or errors relative to a baseline remains untested, requiring a task-based developer study.

7 Conclusion and Future Work

PrivDev Layer 1 bridges security-scanner datatype signals to DPV-PD categories and machine-readable links to potentially relevant GDPR provisions. Retrieval-grounded LLM proposals are constrained by deterministic encoding and SHACL and RAG-quality routing (RQ_1). The SPARQL-queryable graph passed five formative gates, providing structural, retrieval-quality, and human evidence of mapping plausibility, though plausibility remains less stable for ambiguous or ontology-limited cases (RQ_2). Internal checks support process transparency but are not independent validation. Ambiguous scanner labels, datatype-only evidence, and DPV-PD coverage constrain precision (RQ_3); Layer 1 supports developer-facing privacy review rather than establishing legal compliance. The FEDS-structured convergent-evidence protocol itself — formative structural gates, an LLM-judge signal, and multi-statistic human annotation — is reusable beyond this mapping task for other LLM-assisted vocabulary-alignment work. We view this as an ongoing maintenance concern rather than a one-time check: CI-integrated re-mapping can catch drift as *privacy debt* accumulated over code evolution, before it compounds. Future work will extend the bridge to Layer 2 (CWE) and Layer 3 (OWASP) mappings via a planned PrivDev Gateway, spot-check `trivial_pass` mappings, grow the annotator panel, and integrate confidence-weighted gating into CI.

Artifact Availability

All artifacts are available at https://github.com/aisepucrio/privdev-layer1-artifact [3].

Acknowledgements

This research was partially funded by the Brazilian funding agencies CAPES/COFECUB (Grant 88881.879016/2023-01 and Ma1036/24), CNPq (Grant 406089/2025-6), FAPESP (Grant 2023/00811-0), and FAPEMIG (Grant APQ-01488-24). We also acknowledge the Brazilian company Stone[5] for their financial support.

We thank the law professionals who participated in our study and shared their perspectives about PrivDev.

[5] https://www.stone.com.br/

References

[1] C. Baramashetru, P. Giannini, S. Tarifa, and Olaf Owe. 2025. A Type System for Data Privacy Compliance in Active Object Languages. *Art Sci. Eng. Program.* 10 (2025). doi:10.22152/programming-journal.org/2025/10/18

[2] Bearer. 2026. Bearer CLI Documentation. https://docs.bearer.com/quickstart/.

[3] Simon Bernbeck, Ricardo Ramalho, Matheus Amendoeira, and Juliana Alves Pereira. 2026. PrivDev Supplementary Material. https://github.com/aisepucrio/privdev-layer1-artifact. Accessed on: 10/08/2026.

[4] Beatriz Esteves and Víctor Rodríguez-Doncel. 2024. Analysis of ontologies and policy languages to represent information flows in GDPR. *Semantic Web* 15, 3 (2024), 709–743.

[5] Alvan R. Feinstein and Domenic V. Cicchetti. 1990. High Agreement but Low Kappa: I. The Problems of Two Paradoxes. *Journal of Clinical Epidemiology* 43, 6 (1990), 543–549. doi:10.1016/0895-4356(90)90158-L

[6] Mafalda Ferreira, Tiago Brito, J. Santos, and Nuno Santos. 2023. RuleKeeper: GDPR-Aware Personal Data Compliance for Web Frameworks. *2023 IEEE Symposium on Security and Privacy (SP)* (2023), 2817–2834. doi:10.1109/sp46215.2023.10179395

[7] Kilem L. Gwet. 2008. Computing Inter-Rater Reliability and Its Variance in the Presence of High Agreement. *Brit. J. Math. Statist. Psych.* 61, 1 (2008), 29–48. doi:10.1348/000711006X126600

[8] Kilem L. Gwet. 2014. *Handbook of Inter-Rater Reliability* (4th ed.). Advanced Analytics, LLC, Gaithersburg, MD, USA.

[9] I. Hadar et al. 2018. Privacy by Designers: Software Developers' Privacy Mindset. *Empirical Software Engineering* 23, 1 (2018), 259–289.

[10] Steve Harris and Andy Seaborne. 2013. SPARQL 1.1 Query Language. W3C Recommendation, https://www.w3.org/TR/sparql11-query/.

[11] A. R. Hevner, S. T. March, J. Park, and S. Ram. 2004. Design Science in Information Systems Research. *MIS Quarterly* 28, 1 (2004), 75–105.

[12] Harshvardhan J. Pandit, Beatriz Esteves, Georg P. Krog, Paul Ryan, Delaram Golpayegani, and Julian Flake. 2024. Data Privacy Vocabulary (DPV)–Version 2.0. In *International Semantic Web Conference*. Springer, 171–193.

[13] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547. doi:10.1109/TBDATA.2019.2921572

[14] Mugdha Khedkar. 2023. Static Analysis for Android GDPR Compliance Assurance. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. 197–199. doi:10.1109/icse-companion58688.2023.00054

[15] Mugdha Khedkar and Eric Bodden. 2024. Toward an Android Static Analysis Approach for Data Protection. In *2024 IEEE/ACM 11th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*. 65–68. doi:10.1145/3647632.3651389

[16] Mugdha Khedkar, Michael Schlichtig, Nihad Atakishiyev, and Eric Bodden. 2026. Between law and code: challenges and opportunities for automating privacy assessments. *Automated Software Engineering* 33, 2 (Feb. 2026), 56. doi:10.1007/s10515-026-00601-4

[17] Klaus Krippendorff. 2004. Reliability in Content Analysis: Some Common Misconceptions and Recommendations. *Human Communication Research* 30, 3 (2004), 411–433. doi:10.1111/j.1468-2958.2004.tb00738.x

[18] Aryely Matos, Mario Patrício, Maria Isabel Nicolau, Edna Dias Canedo, Juliana Alves Pereira, and Anderson Uchôa. 2025. Data Privacy in Software Practice: Brazilian Developers' Perspectives. *Journal of Internet Services and Applications* 16, 1 (Jun. 2025), 291–311. doi:10.5753/jisa.2025.5302

[19] H. J. Pandit and B. Esteves. 2024. Enhancing Data Use Ontology (DUO) for Health-Data Sharing by Extending it with ODRL and DPV. *Semantic Web Journal* 15, 4 (2024), 1473–1498. doi:10.3233/SW-243583

[20] María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. 2014. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. *International Journal on Semantic Web and Information Systems* 10, 2 (2014), 7–34. doi:10.4018/ijswis.2014040102

[21] Huaijin Ran, Haoyi Zhang, and Xunzhu Tang. 2025. GDPR-Bench-Android: A Benchmark for Evaluating Automated GDPR Compliance Detection in Android. *ArXiv* abs/2511.00619 (2025). doi:10.48550/arxiv.2511.00619

[22] Fabio Rovai. 2026. Open Ontologies: Tool-Augmented Ontology Engineering with Stable Matching Alignment. arXiv preprint arXiv:2605.09184. doi:10.48550/arXiv.2605.09184

[23] A. Senarath and N. A. G. Arachchilage. 2018. Why Developers Cannot Embed Privacy into Software Systems: An Empirical Investigation. In *Proceedings of the Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*. USENIX Association, Baltimore, MD, USA, 211–231.

[24] M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi (Eds.). 2012. *Ontology Engineering in a Networked World*. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-24794-1

[25] Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let Me Speak Freely? A Study on the Impact of Format Restrictions on Performance of Large Language Models. arXiv preprint arXiv:2408.02442. doi:10.48550/arXiv.2408.02442

[26] Shukun Tokas, Olaf Owe, and Toktam Ramezanifarkhani. 2021. Static checking of GDPR-related privacy compliance for object-oriented distributed systems. *J. Log. Algebraic Methods Program.* 125 (2021), 100733. doi:10.1016/j.jlamp.2021.100733

[27] J. Venable, J. Pries-Heje, and R. Baskerville. 2016. FEDS: A Framework for Evaluation in Design Science Research. *European Journal of Information Systems* 25, 1 (2016), 77–89.

[28] W3C. 2017. SHACL – Shapes Constraint Language. https://www.w3.org/TR/shacl/.

[29] W3C. 2021. ODRL Information Model 2.2. https://www.w3.org/TR/odrl-model/.

[30] W3C Data Privacy Vocabularies and Controls Community Group. 2025. Data Privacy Vocabulary (DPV). https://w3c.github.io/dpv/2.3/dpv/.

[31] W3C Data Privacy Vocabularies and Controls Community Group. 2025. Rules — Permission, Prohibition, and Obligation in DPV. https://w3c.github.io/dpv/2.3/dpv/modules/rules.html. Accessed: 16/08/2026.

[32] Jialin Yang, Dongfu Jiang, Lipeng He, Sherman Siu, Yuxuan Zhang, Disen Liao, Zhuofeng Li, Huaye Zeng, Yiming Jia, Haozhe Wang, Benjamin Schneider, Chi Ruan, Wentao Ma, Zhiheng Lyu, Yifei Wang, Yi Lu, Quy Duc Do, Ziyan Jiang, Ping Nie, and Wenhu Chen. 2025. StructEval: Benchmarking LLMs' Capabilities to Generate Structural Outputs. arXiv preprint arXiv:2505.20139. doi:10.48550/arXiv.2505.20139