Break the Glass in OBDA

Divya Baura[1][0000−0002−5237−9927] and Diego Calvanese[2][0000−0001−5174−9693]

[1] Umeå Universitet, Umeå, Sweden
[2] Free University of Bozen-Bolzano, Bolzano, Italy

Abstract. Within the *Ontology-Based Data Access* (OBDA) framework, users can query a relational data source using an ontology to which the source is linked via declarative mappings. In a world where data sharing is widespread, ensuring privacy while managing data poses a significant challenge. *Controlled Query Evaluation* (CQE) is a privacy-preserving query answering framework in the presence of ontologies, where policies representing confidential information are used to devise suitable censors that enforce data protection. The integration of CQE within OBDA was recently proposed through the *Policy-Protected OBDA* (PPOBDA) framework, which is based on embedding policies into mappings. While PPOBDA provides a principled mechanism for enforcing privacy, it assumes fixed access controls that are not suitable for emergency scenarios requiring temporary elevation of data access privileges. In this work, we address this limitation by integrating the *Break-the-Glass* (BtG) mechanism into the PPOBDA framework. BtG is a well-established paradigm in access control that permits authorized users to override certain privacy policies under critical circumstances, while ensuring that such overrides are justified, authenticated, and fully auditable to prevent misuse. We formalize BtG enhanced PPOBDA by extending policy-protected mappings with BtG-specific mappings that are triggered under emergency conditions, and we implement and evaluate the resulting framework. Specifically, we adopt for the evaluation the well-known MIMIC-III hospital dataset, which has earlier been mapped, according to the OBDA framework, to the *Fast Healthcare Interoperability Resources* (FHIR) ontology. Our experiments show the practical implications of the framework, evaluating its applicability to real-world scenarios and the overhead introduced by BtG enabled policy management.

Keywords: Controlled Query Evaluation · Policy-Protected OBDA · Break-the-Glass

1 Introduction

The *Ontology-Based Data Access* (OBDA) framework [18,20] enables users to query relational data sources through an ontology that captures domain knowledge and exposes a vocabulary of classes and properties for query formulation. In OBDA, ontologies are typically expressed in OWL 2 QL [17], a W3C-standardized profile designed for efficient query rewriting, while the underlying relational data sources are connected to the ontology via declarative R2RML

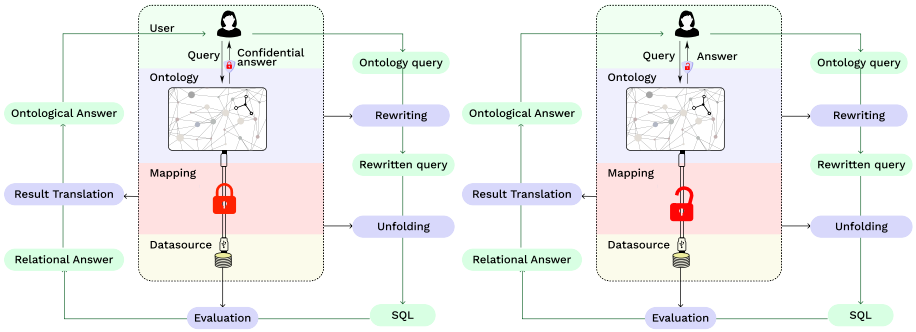

Fig. 1. Query processing in PPOBDA vs. BtG-PPOBDA

mappings [10]. A distinctive feature of this setting is that ontology axioms, such as subclass hierarchies and domain and range property constraints, enable logical inference over the data, allowing facts not explicitly stored in the source to be derived at query time. While this inference capability is central to the power of OBDA, it also significantly complicates the enforcement of privacy, since sensitive information may be exposed not only through direct data retrieval but also through such reasoning.

The *Controlled Query Evaluation* (CQE) framework addresses these concerns by introducing a principled approach to privacy-preserving query answering in the presence of ontologies [4,12,15]. In CQE, *policies* represents the sensitive or confidential information within the data that must not be revealed to users. These policies are protected by a *censor*, which suitably modifies query answers so as not to violate these policies. The *Policy Protected OBDA* (PPOBDA) framework [9], illustrated in the left part of Figure 1, builds on CQE by embedding such policies directly into the mapping layer as first-order logic (FOL) denial assertions, yielding policy-protected mappings that enforce confidentiality transparently at the data access level. However, PPOBDA assumes fixed access controls that apply uniformly across all scenarios and user roles. This rigid enforcement becomes problematic in emergency situations, where temporary elevation of data access privileges may be required. Such situations are particularly critical in domains like healthcare. Regulations such as *Health Insurance Portability and Accountability Act* (HIPAA) [16] and the *General Data Protection Regulation* (GDPR) [19] mandate strong privacy safeguards, yet they also explicitly permit exceptions in life-threatening situations. In such cases, strict policy enforcement can obstruct the timely delivery of necessary care.

In this paper, we address this limitation by extending PPOBDA with a *Break-the-Glass* (BtG) mechanism [5], illustrated in the right part of Figure 1. BtG enables the temporary suspension of confidentiality policies under authenticated emergency conditions. We formalize this extension through a dedicated set of BtG-specific mappings, activated upon assertion of an emergency flag alongside valid user credentials, which selectively relax confidentiality policies under

emergency conditions while preserving those that must hold unconditionally regardless of context. Upon deactivation, full PPOBDA enforcement is reinstated automatically and transparently. To evaluate our approach, we design realistic healthcare scenarios reflecting real-world emergency access requirements, using as our OBDA specification the MIMIC-III hospital dataset [14] mapped to the *Fast Healthcare Interoperability Resources* (FHIR) ontology [22]. Our experimental results confirm that, in critical cases, policies are correctly overridden to deliver complete query answers scoped to the user's assigned role. Our analysis exposes both the strengths and limitations of BtG-enhanced PPOBDA, assessing the overhead introduced by BtG policy management and its implications for real-world deployments.

2 Preliminaries

We present the technical preliminaries essential for understanding the remainder of the paper. The presentation builds upon the formalizations introduced in our previous work on Virtual Knowledge Graph (VKG) and Policy-Protected VKG [2], adapted here to the OBDA and Policy-Protected OBDA setting.

2.1 Ontology-Based Data Access

In OBDA, users can query a data source through an ontology TBox, which is linked to the data source via declarative mappings [18,20]. We formalize OBDA using the notion of *OBDA specification*, which is a triple $\mathcal{V} = \langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle$, where $\mathcal{T}$ is a TBox, $\mathcal{S}$ a relational database (DB) schema, and $\mathcal{M}$ a mapping between $\mathcal{T}$ and $\mathcal{S}$. The TBox captures knowledge at the intensional level about the classes (denoting sets of individuals) and properties (denoting binary relations between individuals or values) of the domain of interest. It is expressed in the lightweight description logic (DL) *DL-Lite*$_\mathcal{R}$ [8], which is the formal counterpart of the ontology language OWL 2 QL, standardized by the W3C [17]. We do not provide here the formal details about *DL-Lite*$_\mathcal{R}$ (or OWL 2 QL), but just state that it is able to capture conceptual modeling formalisms, such as UML class diagrams and Entity-Relationship schemas, through inclusion and disjointness assertions between classes and between properties, domain and range assertions, and mandatory participation to properties [7]. The *mapping* $\mathcal{M}$ is a finite set of *mapping assertions* from $\mathcal{S}$ to $\mathcal{T}$, each of the form $\varphi(\boldsymbol{x}) \rightsquigarrow \psi(\mathsf{IRI}(\boldsymbol{x}))$. Here, $\varphi(\boldsymbol{x})$ denotes a FOL (or SQL) source query over $\mathcal{S}$ with answer variables $\boldsymbol{x}$, while $\psi(\mathsf{IRI}(\boldsymbol{x}))$, referred to as the *target* of the mapping, is an atom whose predicate is a class or property of $\mathcal{T}$ over the variables in $\boldsymbol{x}$ and so-called *IRI-templates* $\mathsf{IRI}(\boldsymbol{x})$. Each IRI-template $\mathsf{iri}(\boldsymbol{x})$ in $\mathsf{IRI}(\boldsymbol{x})$ is a term that concatenates string values and the answer variables in $\boldsymbol{x}$, and is used to construct object identifiers (i.e., IRIs) from the DB values returned by the source query $\varphi(\boldsymbol{x})$. A concrete mapping language that provides such mappings is R2RML, standardized by the W3C [10], but we adopt in our examples the mapping syntax of the OBDA system Ontop [6,21] (see Section 2.3), which is more compact and easier to understand.

We provide an example of OBDA mappings in a medical domain, on which we will build in the rest of the paper.

Example 1. We consider an ontology that includes a class `:Patient` with data property `:Patient.gender`, and a class `:Prescription` with object property `:Prescription.subject` relating a prescription to the patient who is the subject of the prescription. In the data source we have a table `Person`, which is mapped to the class `:Patient` and to the `:Patient.gender` data property, and a table `DrugExposure`, which is mapped to the the class `:Prescription` and to the `:Prescription.subject` object property, as follows:

```
mappingId m1
target :Patient/{person_id} a :Patient ;
         :Patient.gender {gender}^^xsd:string .
source SELECT person_id, gender FROM Person

mappingId m2
target :Prescription/{drug_exposure_id} a :Prescription ;
         :Prescription.subject :Patient/{person_id} .
source SELECT drug_exposure_id, person_id FROM DrugExposure
```

Note that In the target of each mapping, a part enclosed between '{' and '}' denotes an answer variable of the source query, which is meant to be replaced by the value for that variable in an answer tuple to obtain an IRI or literal value.◁

2.2 Policy-Protected OBDA

Policy-Protected OBDA (PPOBDA) extends the traditional OBDA paradigm by embedding data protection policies directly into OBDA mapping definitions, generating policy encoded mappings ($\mathcal{M}_P$) [9,3]. While standard OBDA enables querying heterogeneous sources through an ontological semantic layer, PPOBDA makes use of policies to specify information that should be protected. Specifically, policies in PPOBDA are expressed as denial constraints and the disclosure of protected information, i.e., information that would cause a violation of the denial constraint, is prevented by dynamically rewriting queries in such a way that protected information is suppressed from query results. This is achieved by compiling the policy rules into the declarative OBDA mappings. In this way, a policy-aware OBDA is transformed into a standard OBDA that transparently enforces the policies.

PPOBDA builds upon the principles of *controlled query evaluation* (CQE), which is a privacy-preserving framework for query answering in the presence of ontologies [4,12,15], and it extends the OBDA framework to incorporate CQE. To illustrate the effect of policy compilation, we extend the medical example introduced above with two confidentiality policies. We then show the policy-protected mappings produced after incorporating these policies into the original mapping specification.

Example 2 (Example 1 cont'd). Let us now consider two policies

$$p_1 : \forall x.\forall y \forall z.\, \mathit{Patient.gender}(x, y) \wedge \mathit{Patient.address}(x, z) \rightarrow \bot,$$

which ensures the confidentiality of the combination of a patient's gender and address, and

$$p_2 : \forall x \forall y.\, \mathit{Prescription}(x) \wedge \mathit{Prescription.subject}(x, y) \rightarrow \bot,$$

which ensures the confidentiality of patient ids for which a prescription is made. Let us assume that the `Person` table contains also a `location_id` attribute that is mapped to the *Patient.address* data property. By compiling these policies into m_1 and m_2, we obtain the following policy-protected mapping assertions:

```
mappingId mp1
target :Patient/{person_id} a :Patient ;
         :Patient.gender {gender}^^xsd:string .
source SELECT person_id, gender FROM Person
       WHERE location_id IS NULL

mappingId mp2
target :Prescription/{drug_exposure_id} a :Prescription .
source SELECT drug_exposure_id FROM DrugExposure
       WHERE person_id IS NULL
```

Intuitively, the condition `location_id IS NULL` in mp_1 ensures that the gender location can be returned only for patients whose `location_id` is not known. Similarly, mp_2 ensures that prescriptions can only be disclosed when the `person_id` is not known. ◁

2.3 The OBDA System Ontop

We utilize the state-of-the-art open-source OBDA system Ontop [6,21], which supports query answering over an OBDA instance through query transformation and sophisticated optimization techniques. An Ontop installation operates on an OBDA specification $\mathcal{V} = \langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle$. It efficiently answers queries expressed in the W3C standard query language SPARQL [13] over an OBDA instance $\langle \mathcal{V}, \mathcal{D} \rangle$, where $\mathcal{D}$ is a DB instance for $\mathcal{S}$, using the OWL 2 QL entailment regime [11].

Ontop implements *query answering by rewriting*. This process involves preprocessing $\mathcal{T}$, $\mathcal{S}$, and $\mathcal{M}$ to optimize query answering. At runtime, Ontop transforms a given SPARQL query q into an equivalent SQL query and optimizes it using mapping information and database constraints in $\mathcal{S}$. The resulting SQL query q_{SQL} gets then executed by the underlying DB engine. Once an OBDA specification is developed, a SPARQL endpoint can be set up using Ontop's command line interface, allowing users to interact with it through standard SPARQL tools, without knowing that the endpoint is virtual.

3 Break-the-Glass Access Control Mechanism

The *Break-the-Glass* (BtG) mechanism is an emergency override paradigm that allows authorized users to temporarily bypass specific access restrictions when strict policy enforcement would obstruct urgent and necessary action. The term originates from the physical analogy of breaking a sealed glass case to access emergency equipment, reflecting the idea that such access is available but deliberately non-trivial to invoke. In safety-critical domains such as healthcare, BtG is particularly relevant: a clinician may need immediate access to a patient's restricted medical history during a life-threatening situation, where waiting for standard authorization procedures is not viable. Regulations governing healthcare data explicitly acknowledge this tension. HIPAA permits disclosure of protected health information when necessary to prevent a serious threat to health or safety (§164.512(j))[3], while GDPR allows processing of sensitive personal data when it is necessary to protect the vital interests of the data subject (Article 9(2)(c))[4]. These provisions confirm that privacy regulations are not absolute and are designed to accommodate genuine emergencies, provided that such exceptions are controlled and accountable. A well-designed BtG framework rests on three core principles.

1. Access must be *authenticated*: only users holding a designated emergency role may invoke the mechanism, ensuring that overrides cannot be triggered arbitrarily.
2. The *scope* of the override must be defined: not all policies need to be suspended during an emergency, and those that protect against re-identification or irreversible harm should remain enforced regardless.
3. Every BtG activation must be *auditable*: the identity of the user, their role, the time of access, and a mandatory justification must be recorded in an immutable audit log, enabling post-hoc review and deterring misuse.

We show now how these principles can be realized within the PPOBDA framework by extending its mapping layer with BtG-specific components.

4 BtG in PPOBDA

We now formalize the integration of the BtG mechanism into PPOBDA. Standard PPOBDA enforces a fixed set of confidentiality policies at all times by embedding denial assertions into the mapping layer generating $\mathcal{M}_P$. Our extension introduces a dynamic component that selectively relaxes a subset of these policies under authenticated emergency conditions, while preserving those policies whose violation would cause irreversible harm regardless of context.

[3] https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164/subpart-E/section-164.512

[4] https://eur-lex.europa.eu/eli/reg/2016/679/oj

Algorithm 1: `ProcessBtGQuery`

Input: Query Q, policy set $\mathcal{P}$ with invariant subset $\mathcal{P}_{inv} \subseteq \mathcal{P}$,
user_metadata = {`x_user`, `x_roles`, `justification`},
emergency_flag = `ontop.authorization` (bool)
Output: Answer to Q respecting applicable policies

```
1  if emergency_flag = false then
2      return executeQuery(Q, M_P)
3  else
4      if isValidUser(user_metadata.x_user) and
5              hasEmergencyRole(user_metadata.x_roles) and
6              user_metadata.justification ≠ "" then
7          auditLog.append({x_user, x_roles, justification, currentTime()});
8          return executeQuery(Q, M_BtG)
9      else
10         return executeQuery(Q, M_P)
```

Policy Partitioning. Central to our approach is the distinction between two classes of policies. Given a policy set $\mathcal{P}$, we identify a subset $\mathcal{P}_{inv} \subseteq \mathcal{P}$ of *invariant policies*, which remain enforced under all circumstances, including during a BtG override. In Example 2, policy p_1 protects the combination of a patient's gender and address, a pair of demographic attributes whose joint disclosure risks re-identification. Since this risk persists regardless of the nature of an emergency, p_1 is designated as invariant, i.e., $\mathcal{P}_{inv} = \{p_1\}$.

The remaining policies $\mathcal{P} \setminus \mathcal{P}_{inv}$ are *relaxable* and may be temporarily suspended when a valid emergency access request is made. To illustrate this distinction, we use in our evaluation (see Section 5) policies defined over the FHIR ontology [22] mapped to the MIMIC-III dataset [14] via OMOP-CDM:[5]

BtG Components. The BtG extension of PPOBDA relies on three components that together realize the principles outlined in Section 3. The *emergency flag* is a logical switch, configured via `ontop.authorization=true`, that activates the BtG pathway. When false, the system enforces all policies in $\mathcal{P}$ through the standard PPOBDA mappings $\mathcal{M}_P$. When true, the system evaluates whether the requesting user is authorized to invoke BtG. The *emergency role* restricts BtG invocation to users holding a designated role, such as `EmergencyDoctor`, whose identity and role membership are validated from HTTP request headers (`x-user` and `x-roles`). Finally, *accountability* is enforced by requiring a non-empty free-text justification at invocation time, and by appending to an immutable audit log the full access record, which specifies identity, role, the provided justification, and a timestamp.

Query Processing. Algorithm 1 formalizes how these three components compose into a single query-time decision procedure.

[5] https://ohdsi.github.io/CommonDataModel

Mapping Implementation. The BtG logic is encoded directly within the OBDA mapping layer. When `ontop.authorization=true`, Ontop extracts user identity and role information from HTTP headers and makes them available via the SQL functions `ontop_user()` and `ontop_contains_role()`. The following mappings illustrate how p_1 is enforced as an invariant policy while p_2 is relaxed for authorized emergency users:

```
mappingId mp1
target :Patient/{person_id} a :Patient ;
         :Patient.gender {gender}^^xsd:string .
source SELECT person_id, gender FROM Person
       WHERE location_id IS NULL

mappingId mp2
target :Prescription/{drug_exposure_id} a :Prescription ;
         :Prescription.subject :Patient/{person_id} .
source SELECT drug_exposure_id, person_id FROM DrugExposure
       WHERE ontop_contains_role('EmergencyDoctor')
```

Mapping `mp1` enforces p_1 even under BtG by filtering out address data, ensuring that the re-identification risk captured by the invariant policy is never exposed. Mapping `mp2` grants access to prescription data for users holding the `EmergencyDoctor` role, effectively relaxing p_2 during an emergency. Every access routed through the BtG mappings is recorded in the audit log, ensuring that emergency overrides remain traceable and subject to post-hoc review.

A full implementation of this approach, including the complete policy set and mapping definitions, is available online.[6]

5 Evaluation

We evaluate BtG-enhanced PPOBDA over a realistic healthcare use case based on the MIMIC-III dataset, using three representative scenarios to analyse the behaviour of Algorithm 1. Specifically, we evaluate whether the algorithm correctly relaxes only the designated relaxable policies while preserving invariant policies, and we measure its impact in terms of query result counts and execution time under both $\mathcal{M}_P$ and $\mathcal{M}_{BtG}$.

OBDA Specification and Dataset. We adopt the OBDA specification based on the *Medical Information Mart for Intensive Care* (MIMIC)-III clinical dataset [14], which aggregates de-identified healthcare records spanning 46,000 unique patients and over 60,000 intensive care unit admissions at Boston's Beth Israel Deaconess Medical Center (2001–2012). The dataset is converted into OMOP-CDM[7] using the MIMIC-OMOP ETL Tool.[8] The ontology used is the

[6] https://github.com/divyabaura/BtG
[7] https://ohdsi.github.io/CommonDataModel
[8] https://github.com/MIT-LCP/mimic-omop

FHIR Model Ontology,[9] and the OMOP-CDM data is connected to it via the R2RML-compliant mappings of Xiao et al. [22], executed within the Ontop platform.

Implementation. Our implementation extends the open-source PPOBDA system of [3], based on Ontop. We developed two components to support BtG. First, a `BtGMappingEncoder` Java class that generates $\mathcal{M}_{BtG}$ by injecting conditions like `ontop_contains_role('EmergencyDoctor')` into the SQL `WHERE` clauses of relaxable mappings, while leaving mappings corresponding to $\mathcal{P}_{inv}$ unchanged. Second, a `BtGQueryExecutor` class that issues authenticated HTTP POST requests to the Ontop endpoint, setting the `x-roles` and `x-user` headers to reflect emergency access, and recording each BtG activation in the audit log with the user identity, role, justification, and timestamp. The flag `ontop.authorization=true` is enabled throughout all experiments.

Policies and Queries. We evaluate BtG over the policy set $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$, expressed over the FHIR ontology. Policies p_1 and p_2 are as defined in Example 2, and the remaining policies are:

$p_3 : \forall x.\forall y.\forall z.\, \mathit{Procedure.code}(x, y) \wedge \mathit{Procedure.performedDateTime}(y, z) \rightarrow \bot$
$p_4 : \forall x.\forall y.\forall z.\, \mathit{Observation.valueQuantity}(x, y) \wedge \mathit{Quantity.value}(y, z) \rightarrow \bot$

Policy p_3 restricts access to procedure codes with their timestamps, while policy p_4 restricts access to observation values. Policy p_1 is designated as invariant across all scenarios, reflecting the judgment that joint disclosure of patient gender and address poses an unconditional re-identification risk. Policies p_2, p_3, and p_4 are relaxable.

We evaluate the following four queries, adapted from [22]:

q_1: Male patients with inpatient admissions lasting ≥ 5 days.
q_2: Patients taking Trazodone.
q_3: Patients who delivered a baby.
q_4: Patients with an HbA1c $\geq 10\%$.

5.1 Scenarios and Results

Table 1 summarises execution times and result counts for all queries under $\mathcal{M}_P$ and $\mathcal{M}_{BtG}$; the three scenarios below explain the observed behaviour.

Emergency Clinical Access. In emergency settings, clinicians may require immediate access to a patient's prescription or procedure history to make time-critical decisions. Queries q_2 and q_3 capture this need: q_2 retrieves patients taking Trazodone, restricted by p_2, and q_3 retrieves patients who delivered a baby, restricted by p_3. Both return empty answers under $\mathcal{M}_P$. When BtG is activated by an authenticated `EmergencyDoctor` with a valid justification, both queries are routed to $\mathcal{M}_{BtG}$, relaxing p_2 and p_3 while preserving p_1, and return complete results (4547 and 34, respectively).

[9] http://build.fhir.org/fhir.ttl

Table 1. Impact of BtG on query execution time (ms) and number of results across the three scenarios

		$\mathcal{M}_P$		$\mathcal{M}_{BtG}$	
Query	**Scenario**	time (ms)	#results	time (ms)	#results
q_1	Public Health Surveillance	191	0	812	0
q_2	Emergency Clinical Access	216	0	18843	4547
q_3	Emergency Clinical Access	117	0	1621	34
q_4	Inference-Aware Decision Support	324	0	1894	944

Inference-Aware Decision Support. Ontology-based inference allows facts not explicitly stored in the source to be derived at query time. Strict policy enforcement can break this inference chain by blocking access to intermediate data. Query q_4 illustrates this: retrieving patients with HbA1c $\geq 10\%$ requires observation values restricted by p_4, yielding an empty answer under $\mathcal{M}_P$. Under $\mathcal{M}_{BtG}$, relaxation of p_4 restores the inference chain and returns 944 complete answers. Policy p_1 applies throughout but has no effect on q_4.

Public Health Surveillance. Epidemiological queries often combine demographic and clinical criteria, and may be blocked by invariant policies rather than relaxable ones. Query q_1 asks for male patients with inpatient admissions lasting at least five days, requiring patient gender data restricted by p_1. Since p_1 is invariant, q_1 returns no answer under either $\mathcal{M}_P$ or $\mathcal{M}_{BtG}$, illustrating that BtG operates within the hard bounds set by invariant constraints rather than bypassing all protections unconditionally.

Performance. The overhead introduced by $\mathcal{M}_{BtG}$ is modest. The additional SQL conditions for role validation and audit logging introduce only a marginal increase in execution time compared to $\mathcal{M}_P$. As observed in [1], execution times are notably reduced when enforced policies yield empty result sets, due to early pruning during query evaluation. This confirms that the BtG extension is practically feasible without introducing undue complexity for real-world deployments.

6 Conclusions

We extended PPOBDA with a Break-the-Glass mechanism that allows authorized users to temporarily override relaxable policies under emergency conditions, while invariant policies remain enforced unconditionally. The mechanism is implemented directly within the OBDA mapping layer, requiring no changes to the underlying OBDA engine, and every override is recorded in a mandatory audit log.

Our evaluation confirms that BtG correctly restores complete query answers where needed, with marginal performance overhead. At the same time, it exposes an inherent limitation: queries requiring data protected by invariant policies, such as q_1 restricted by p_1, remain unanswerable even under BtG, reflecting

the deliberate trade-off between principled privacy and full data availability. In future work, we plan to investigate finer-grained policy relaxation and temporal constraints on access duration.

Acknowledgments

This research has been partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, by HEU project CyclOps (GA n. 101135513), by the Province of Bolzano and FWF through project OnTeGra (DOI 10.55776/PIN8884924), by the Province of Bolzano and EU through projects ERDF-FESR 1078 CRIMA, and ERDF-FESR 1047 AI-Lab, and by MUR through PRIN project 2022XERWK9 S-PIC4CHU.

References

1. Baura, D., Calvanese, D.: Assessing privacy requirements for controlled query evaluation in OBDA. In: Proc. of the 22nd Int. Conf. on Modeling Decisions for Artificial Intelligence (MDAI). LNCS, vol. 15957, pp. 183–197. Springer (2025). https://doi.org/10.1007/978-3-032-00891-6_15
2. Baura, D., Calvanese, D.: User access control in policy-protected Virtual Knowledge Graphs. In: Proc. of the 14th Int. Joint Conf. on Knowledge Graphs (IJCKG). LNCS, vol. 15996, pp. 204–219. Springer (2026). https://doi.org/10.1007/978-981-95-5009-8_14
3. Baura, D., Calvanese, D., Marconi, L.: Implementing controlled query evaluation in OBDA. In: Proc. of the Joint Ontology Workshops Episode 10: The Tukker Zomer of Ontology (JOWO). CEUR-WS.org, vol. 3882. CEUR Workshop Proceedings (2024), https://ceur-ws.org/Vol-3882/st4dm-1.pdf
4. Bonatti, P.A., Sauro, L.: A confidentiality model for ontologies. In: Proc. of the 12th Int. Semantic Web Conf. (ISWC). LNCS, vol. 8218, pp. 17–32. Springer (2013). https://doi.org/10.1007/978-3-642-41335-3_2
5. Brucker, A.D., Petritsch, H.: Extending access control models with break-glass. In: Proc. of the 14th ACM Symposium on Access Control Models and Technologies (SACMAT). pp. 197–206 (2009). https://doi.org/10.1145/1542207.1542239
6. Calvanese, D., Cogrel, B., Komla-Ebri, S., Kontchakov, R., Lanti, D., Rezk, M., Rodriguez-Muro, M., Xiao, G.: Ontop: Answering SPARQL queries over relational databases. Semantic Web J. **8**(3), 471–487 (2017). https://doi.org/10.3233/SW-160217
7. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R.: Ontologies and databases: The *DL-Lite* approach. In: Reasoning Web: Semantic Technologies for Informations Systems – 5th Int. Summer School Tutorial Lectures (RW), LNCS, vol. 5689, pp. 255–356. Springer (2009). https://doi.org/10.1007/978-3-642-03754-2_7
8. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. J. of Automated Reasoning **39**(3), 385–429 (2007). https://doi.org/10.1007/s10817-007-9078-x

9. Cima, G., Lembo, D., Marconi, L., Rosati, R., Savo, D.F.: Controlled query evaluation in ontology-based data access. In: Proc. of the 19th Int. Semantic Web Conf. (ISWC). LNCS, vol. 12506, pp. 128–146. Springer (2020). https://doi.org/10.1007/978-3-030-62419-4_8
10. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF mapping language. W3C Recommendation, World Wide Web Consortium (Sep 2012), http://www.w3.org/TR/r2rml/
11. Glimm, B., Ogbuji, C.: SPARQL 1.1 entailment regimes. W3C Recommendation, World Wide Web Consortium (Mar 2013), http://www.w3.org/TR/sparql11-entailment/
12. Grau, B.C., Kharlamov, E., Kostylev, E.V., Zheleznyakov, D.: Controlled query evaluation for Datalog and OWL 2 Profile ontologies. In: Proc. of the 24th Int. Joint Conf. on Artificial Intelligence (IJCAI). pp. 2883–2889. AAAI Press (2015), https://ijcai.org/Abstract/15/408
13. Harris, S., Seaborne, A.: SPARQL 1.1 query language. W3C Recommendation, World Wide Web Consortium (Mar 2013), http://www.w3.org/TR/sparql11-query
14. Johnson, A.E., Pollard, T.J., Shen, L., Lehman, L.w.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., Mark, R.G.: MIMIC-III, a freely accessible critical care database. Scientific Data **3**(1), 1–9 (2016)
15. Lembo, D., Rosati, R., Savo, D.F.: Revisiting controlled query evaluation in description logics. In: Proc. of the 28th Int. Joint Conf. on Artificial Intelligence (IJCAI). pp. 1786–1792. ijcai.org (2019). https://doi.org/10.24963/IJCAI.2019/247
16. Moore, W., Frye, S.: Review of HIPAA, Part 1: History, protected health information, and privacy and security rules. J. of Nuclear Medicine Technology **47**(4), 269–272 (2019)
17. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language profiles (second edition). W3C Recommendation, World Wide Web Consortium (Dec 2012), http://www.w3.org/TR/owl2-profiles/
18. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. J. on Data Semantics **10**, 133–173 (2008). https://doi.org/10.1007/978-3-540-77688-8_5
19. Voigt, P., von dem Bussche, A.: The EU General Data Protection Regulation (GDPR) – A Practical Guide. Springer (2017)
20. Xiao, G., Calvanese, D., Kontchakov, R., Lembo, D., Poggi, A., Rosati, R., Zakharyaschev, M.: Ontology-based data access: A survey. In: Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI). pp. 5511–5519. IJCAI Org. (2018). https://doi.org/10.24963/ijcai.2018/777
21. Xiao, G., Lanti, D., Kontchakov, R., Komla-Ebri, S., Güzel-Kalayci, E., Ding, L., Corman, J., Cogrel, B., Calvanese, D., Botoeva, E.: The virtual knowledge graph system Ontop. In: Proc. of the 19th Int. Semantic Web Conf. (ISWC). LNCS, vol. 12507, pp. 259–277. Springer (2020). https://doi.org/10.1007/978-3-030-62466-8_17
22. Xiao, G., Pfaff, E.R., Prud'hommeaux, E., Booth, D., Sharma, D.K., Huo, N., Yu, Y., Zong, N., Ruddy, K.J., Chute, C.G., Jiang, G.: FHIR-Ontop-OMOP: Building clinical knowledge graphs in FHIR RDF with the OMOP Common Data Model. J. of Biomedical Informatics **134**, 104201 (2022). https://doi.org/10.1016/J.JBI.2022.104201