Designing Auditable and Version-Aware Consent Management Systems for Regulatory Compliance

Claudio Henrique Pereira de Castro, Geovana Ramos Sousa Silva, Stefano Luppi Spósito, Edna Dias Canedo

[1]University of Brasília (UnB), Department of Computer Science, Brasília, DF, Brazil
E-mail: claudiodecastro@gmail.com,geovannna.1998@gmail.com,
stefanoluppi@hotmail.com, ednacanedo@unb.br

***Abstract. Context**: Data protection regulations such as the GDPR and the Brazilian LGPD impose strict requirements on consent management, including demonstrability, revocation support, accountability, and traceability. In practice, however, many existing consent management solutions fail to preserve a reliable historical linkage between user consent decisions and the specific versions of privacy policies in force at the time of collection, which undermines auditability and regulatory compliance. **Goal**: This paper aims to design a modular, scalable, and interoperable software architecture that supports the registration, explicit versioning, and auditing of user consent, ensuring traceability, integrity, and compliance with privacy regulations while preserving the historical binding between consent records and evolving privacy policies. **Method**: We follow a Design Science Research approach to derive architectural requirements from GDPR, LGPD, and relevant ISO/IEC standards, and to design a microservice-based consent management architecture. The proposed solution is instantiated as a proof of concept composed of independent services for policy management and consent recording, exposed through RESTful APIs. The architecture is demonstrated in a controlled environment through end-to-end consent lifecycle scenarios, including policy versioning, consent granting and refusal, revocation, and audit querying. **Results**: The proof of concept shows that explicit policy versioning, cryptographic integrity mechanisms, and event-based consent recording can be combined to preserve immutable historical records of consent decisions. Each consent, refusal, and revocation event is deterministically bound to a specific policy version, enabling consistent audit queries by user, policy, and time. **Conclusion**: The study indicates that a version-aware, microservice-oriented design provides a feasible foundation for auditable and regulation-compliant consent management. By treating privacy policies as versioned artifacts and consent as a persistent event history, the proposed architecture bridges regulatory requirements and implementable software mechanisms.*

1. Introduction

Over the last decades, society has undergone an intense digital transformation marked by large-scale collection and extensive use of personal data. In the early stages of this transformation, the absence of robust regulatory mechanisms and effective protection technologies enabled indiscriminate data collection practices, which consolidated into a dominant business model in several economic sectors [Jha et al. 2025]. As personal data became widely perceived as a strategic asset and a digital commodity, ethical, social, and legal

concerns regarding privacy and fair information use gained prominence [Kalaoja 2022]. Recurring incidents of data leakage, identity theft, and algorithmic manipulation reinforced the understanding of privacy as a fundamental right, whose protection requires both normative safeguards and effective technical solutions [Novikova et al. 2025].

In response, regulatory frameworks such as the European General Data Protection Regulation (GDPR) [Parliament and Council 2018] and the Brazilian General Data Protection Law (LGPD) [Macedo 2018] established rigorous obligations for organizations that collect and process personal data. Among the lawful bases for processing, **consent** plays a central role: it must be freely given, informed, specific, and unambiguous, and it must be recorded in a transparent, versioned, and auditable manner. These requirements make consent management not only a legal compliance concern, but also a strategic element for building trust between organizations and data subjects [Jha et al. 2025].

However, implementing these requirements in practice remains challenging. Since the GDPR entered into force, there has been a sharp growth in Consent Management Platforms (CMPs), which operationalize privacy banners and consent forms on websites. Evidence indicates that CMP adoption increased from less than 5% of websites before 2018 to more than 40% by 2023 in the European context [Jha et al. 2025]. Nevertheless, empirical analyses reveal persistent shortcomings: a substantial share of websites using CMPs still deploy tracking technologies before a user choice is expressed, undermining regulatory compliance [Jha et al. 2025]. Moreover, interface design strongly shapes user behavior, and prior studies have highlighted that small interaction frictions can systematically steer decisions, illustrating the well-known tension between stated privacy concerns and actual choices.

A structural limitation also concerns the quality and clarity of privacy policies. Recent research shows that these documents are often lengthy, vague, and inconsistent, which hinders user comprehension and weakens the legal value of the consent associated with them [Novikova et al. 2025]. The lack of standardization for describing processing practices and risks increases information asymmetry between controllers and data subjects, compromising transparency and limiting organizational accountability. In parallel, studies have documented the use of dark patterns in consent collection persuasive design practices that nudge users toward more privacy invasive options [Guerra 2024, Kalaoja 2022, Lu et al. 2024]. These findings raise concerns about whether consent, even when formally obtained, is truly informed and freely given.

To address these limitations, the literature has investigated decentralized alternatives such as dynamic consent models supported by technologies like blockchain and smart contracts, aiming to increase transparency, immutability, and traceability in consent management [Merlec et al. 2021]. Such systems may enable data subjects to track histories of use, receive notifications, and revoke permissions over time. Yet, practical barriers remain, including key management complexity, performance and scalability constraints, and the tension between the right to erasure and blockchain immutability. Despite significant regulatory and technological advances, gaps persist that limit the effectiveness of consent management in real-world settings. This motivates the need for flexible, modular, and interoperable architectures that ensure traceability, integrity, and transparency, while remaining aligned with legal requirements and with privacy requirements engineering practices [Carneiro et al. 2024, Spósito et al. 2025, Silva and Canedo 2025].

In this context, this paper proposes a microservice-based solution to support consent registration, policy versioning, and auditing, preserving the historical linkage between user consents and the specific versions of privacy policies in force at the time of collection.

Although GDPR and LGPD define clear requirements for collecting, recording, and auditing consent, practical implementations remain fraught with technical and usability issues [Seiling et al. 2024]. CMP-based approaches still exhibit compliance failures and design choices that may compromise the legitimacy of consent [Jha et al. 2025, Kalaoja 2022, Novikova et al. 2025, Seiling et al. 2024]. In addition, users face cognitive limitations when making privacy decisions; bounded rationality and present-biased preferences can lead individuals to overweight immediate convenience and underestimate future risks, further challenging the notion of truly informed consent [Seiling et al. 2024]. These factors reinforce calls for integrating risk communication mechanisms into consent experiences. While blockchain-based models offer traceability, their limitations hinder broad adoption, and there is no consensus on practical architectures that balance compliance, security, scalability, and organizational applicability. Although risk communication is important, our proof of concept (PoC) focuses on backend auditability and policy-version binding, remaining interface-agnostic.

Therefore, this study is guided by the following research question (RQ):**RQ. How can we design and implement a modular, scalable, and interoperable architecture that supports the registration, versioning, and auditing of user consent, ensuring traceability, integrity, and compliance with privacy regulations, while preserving the historical linkage between consent records and the specific versions of privacy policies in force?**

This paper contributes: i) a microservice-based reference PoC architecture for consent registration, policy versioning, and audit-oriented querying; ii) a relational data model that preserves the linkage between consent records and policy versions, supporting traceability and accountability; iii) a prototype implementation based on RESTful APIs, designed to be modular and interoperable with organizational systems; and iv) a demonstration based evaluation using end-to-end consent lifecycle scenarios and audit queries.

2. Background and Related Work

Privacy and Data Protection in Software Systems. Privacy is a multifaceted and evolving concept that extends beyond mere control over personal data. It is commonly understood as a fundamental right connected to individuals' autonomy to manage their own information, including decisions about when, how, and with whom personal data may be shared [Gharib et al. 2017, da Silva Junior et al. 2018]. In contemporary digital ecosystems, privacy concerns encompass not only surveillance and misuse, but also broader notions of informational self-determination, in which data subjects are expected to exercise effective decision power throughout the entire data lifecycle.

From a software engineering perspective, Veseli et al. [Veseli et al. 2019] discuss privacy responsibilities across three interrelated spheres: (i) the user sphere, where individuals must retain meaningful control over their information and devices; (ii) the recipient sphere, which concerns organizational practices for mitigating exposure and preventing privacy violations; and (iii) the joint sphere, typical of large-scale data storage

and processing environments, where technical and organizational safeguards become essential. Earlier work by Kalloniatis et al. [Kalloniatis et al. 2005] already emphasized the need for systematic methods to support privacy requirements specification and highlighted the difficulty of harmonizing privacy regulations across jurisdictions due to cultural, political, and economic differences. These challenges remain relevant today, as effective compliance requires translating legal obligations into implementable requirements and verifiable system behaviors.

Regulatory Requirements for Consent: GDPR and LGPD. The European General Data Protection Regulation (GDPR) [Parliament and Council 2018] and the Brazilian General Data Protection Law (LGPD) [Macedo 2018] provide the primary regulatory baseline for this work. Both frameworks establish principles such as transparency, purpose limitation, data minimization, security, and accountability, and grant data subjects a set of enforceable rights, including access, rectification, deletion, portability, and review of automated decisions [Parliament and Council 2018, Macedo 2018]. Consent is a key lawful basis in both regulations: it must be freely given, informed, specific, and unambiguous, and it must be demonstrable by the controller. GDPR (Art. 7) and LGPD (Art. 8) require that consent revocation be supported and operationally feasible, implying that consent must be traceable over time and explicitly linked to the conditions under which it was collected.

However, empirical studies of Consent Management Platforms (CMPs) consistently report compliance shortcomings, such as tracking technologies being activated before an explicit user choice and interface designs that steer users toward acceptance [Jha et al. 2025, Kalaoja 2022, Novikova et al. 2025, Seiling et al. 2024]. In addition, user decision-making is constrained by cognitive limitations, including bounded rationality and present-biased preferences, which motivates the integration of explicit risk communication mechanisms to better convey privacy implications [Seiling et al. 2024]. These factors reinforce the need for architectures that can provide auditable evidence of valid consent while preserving usability and scalability.

ISO/IEC Standards Supporting Implementable Consent Practices. While GDPR and LGPD define legal obligations, ISO/IEC standards provide technical guidance for operationalizing privacy principles in software systems. ISO/IEC 29100 [for Standardization 2024] defines a general privacy framework, clarifying key actors (e.g., data subject, controller, processor, and third parties) and establishing principles such as consent and choice, openness, transparency, and notice. ISO/IEC 29184 [for Standardization 2020] focuses specifically on online privacy notices and consent, recommending clarity, accessibility, proper timing, and explicit versioning of notices, as well as consent validity properties such as affirmative action, separation from other terms, granularity, and ease of withdrawal. A notable contribution of this standard is its emphasis on producing consent receipts as evidence of collection and scope. Complementing these efforts, ISO/IEC TS 27560 [for Standardization 2023] specifies a structured information model for consent records, including identifiers, controller information, purposes, references to policy or notice versions, validity conditions, channels of collection, timestamps, and cryptographic proofs. Together, these standards motivate a set of system requirements

central to this work: (i) explicit versioning of privacy policies; (ii) deterministic binding between consent records and policy versions; (iii) auditability and integrity of consent evidence; and (iv) interoperability for exchanging consent records across systems.

Architectures and Interaction-Oriented Approaches for Consent Management. Compliance oriented systems must embed privacy requirements early in the software lifecycle, in line with the principles of privacy by design and privacy by default [Matos et al. 2025]. In the context of consent management, architectural decisions are particularly critical, as systems must support policy evolution, consent revocation, accountability, and large-scale audit queries. Microservice architectures have been increasingly adopted to address these requirements due to their modularity, independent deployment, and ability to isolate privacy-related capabilities in dedicated services [Marillonnet et al. 2021]. RESTful APIs further facilitate interoperability and integration with organizational systems, auditors, and external platforms. Nevertheless, security and integrity remain prerequisites for legal validity: consent records must be protected against tampering, loss, and unauthorized access.

In parallel, recent research has examined privacy and consent in conversational systems and chatbot-based interactions. Systematic reviews and empirical studies highlight that chatbots introduce specific privacy risks due to their conversational nature, continuous data collection, and limited user awareness of data processing practices [Silva and Canedo 2025]. User-centric guidelines for conversational design emphasize transparency, informed decision-making, and user control [Silva and Canedo 2024].

While these works contribute important insights into privacy-aware interaction design and requirements elicitation, they largely focus on interface level or conceptual aspects and do not address how consent decisions should be persistently recorded, versioned, and audited at the architectural level. Similarly, prior architectural approaches to consent management, including formal models and blockchain-based solutions, aim to enhance traceability and non-repudiation [Peyrone 2022]. However, such approaches often introduce scalability constraints, operational complexity, and potential tensions with regulatory rights such as revocation and erasure. As a result, there remains a gap between regulatory and standardization requirements for demonstrable consent and practical, modular, and interoperable architectures that preserve historical linkage between consent records and evolving privacy policies. Addressing this gap is the central motivation of the architecture proposed in this study.

3. Research Method

This study follows a Design Science Research (DSR) approach, which is widely adopted in Information Systems and Computer Science to build and evaluate technological artifacts that address relevant practical problems [Vaishnavi 2007]. DSR is appropriate for this work because consent management requires the translation of legal and normative requirements into implementable software mechanisms, and the validation of these mechanisms through artifact-based evaluation. DSR is typically organized as an iterative cycle comprising: (i) problem identification; (ii) definition of solution objectives; (iii) artifact design and construction; (iv) demonstration; (v) evaluation; and (vi) communication [Vaishnavi 2007]. We adopted this cycle to structure both the development and the as-

sessment of a microservice-based architecture for consent records. Feedback loops were used whenever evidence from demonstration or evaluation indicated the need to refine requirements, data structures, or service interfaces.

The problem addressed in this research is the lack of practical, engineering oriented solutions that simultaneously ensure: (i) verifiable consent records; (ii) explicit linkage between consent and the exact privacy policy version in force at the time of collection; (iii) auditability and integrity of records; and (iv) scalability and interoperability in distributed environments. This gap is motivated by: (a) regulatory requirements from GDPR and LGPD regarding valid consent, accountability, and revocation; (b) standardization guidance from ISO/IEC 29100, 29184, and ISO/IEC TS 27560 regarding notices, consent receipts, and record structures; and (c) limitations observed in existing CMP ecosystems, including compliance failures and weak transparency in practice. Based on the problem characterization, we defined the objectives for the artifact as follows: a) Provide a modular and scalable architecture to manage privacy policies and consent records; b) Maintain a persistent, auditable binding between each consent record and a specific privacy policy version; c) Support consent lifecycle events (grant, refuse, revoke) without overwriting history; d) Enable audit queries that demonstrate accountability (who consented, to what, when, under which policy version); and e) Align the record structure with legal requirements and ISO/IEC guidance (including key metadata and integrity evidence). The artifact is a PoC architecture implemented as two main microservices exposed through REST APIs: i) **Policy Service**: manages privacy policy creation, versioning, and retrieval; ii) **Consent Service**: records and queries consent events associated with policy versions.

The relational data model was designed to enforce referential integrity between consent records and policy versions. Policies store identifiers, version fields, publication timestamps, and an integrity hash for the policy document. Consent records store user identifiers, the referenced policy version, timestamps for creation and revocation, status (e.g., accepted/refused/revoked), channel, and a validation hash for tamper evidence. This design supports historical traceability and audit readiness. Metadata are stored in a relational database, while full policy documents are stored in an object storage component (simulated with MinIO in the PoC). This separation preserves structured audit queries while enabling efficient storage and retrieval of policy documents.

The PoC focuses on validating the architecture and traceability mechanisms. Security controls such as Transport Layer Security (TLS), authentication, authorization, and encryption are planned as production requirements. The evaluation includes integrity checks via hashing and audit logging assumptions to demonstrate feasibility and required safeguards. Requirements were derived from two complementary sources: 1. **Legal and regulatory requirements**: consent validity and revocation properties, demonstrability of consent, transparency and accountability requirements (GDPR and LGPD); and 2. **Normative and technical requirements**: guidance for privacy notices, consent collection properties, consent receipts, and a structured representation of consent records (ISO/IEC 29100, ISO/IEC 29184, ISO/IEC TS 27560). This derivation guided both the functional scope (policy versioning, consent lifecycle, audit queries) and the non-functional scope (integrity, traceability, interoperability, and scalability goals). Demonstration was conducted in a controlled environment by executing end-to-end workflows that represent typical consent operations:a) registering a new privacy policy and publishing a new version;

b) collecting consent or refusal associated with a specific policy version; c) revoking consent and preserving the full historical trail; and d) retrieving records for auditing purposes (by user, by policy version, and by time range). These scenarios validate that the proposed architecture operationalizes the core traceability requirement: each consent event remains bound to an immutable reference to the policy version that was presented at the time of the decision.

The evaluation verifies whether the artifact meets the objectives defined in the DSR cycle. We apply a mixed evaluation strategy, combining technical measurements and compliance-oriented verification. We assess performance and scalability using workload-based tests over the main endpoints (policy retrieval, consent registration, consent query). We perform workload-based tests over the main endpoints to check feasibility under increasing concurrency, and we verify compliance using a checklist derived from GDPR, LGPD, and ISO/IEC standards.

4. PoC Deployment and Execution Environment

This section describes how the PoC was deployed and executed to validate the proposed architecture. The goal is to make the experimental setup reproducible and to clarify how the components were composed, networked, and configured in a controlled environment. The PoC was deployed using Docker Compose to orchestrate the services, configure an internal network, and manage persistent volumes. This choice supports repeatability across machines and reduces external dependencies, which is especially important for artifact-oriented research and technical evaluation.

Docker Compose orchestrates the following containers:i) **`api-policies`**: policy service responsible for policy document upload, explicit versioning, cryptographic integrity hashing, and persistence in object storage; ii) **`api-consents`**: consent service responsible for recording user decisions (given/refused), revocation, and audit queries; iii) **`app-chatbot`**: lightweight demonstration client used to exercise the APIs and inspect the resulting traces; iv) **PostgreSQL**: relational database used to store structured metadata and preserve historical traceability; and v) **MinIO**: S3-compatible object storage used to persist the physical versions of policy documents. All containers communicate through a Docker-managed internal network, relying on service names for internal DNS resolution. This setup limits unintended exposure of internal endpoints and keeps interactions within a controlled execution boundary.

Communication among services occurs exclusively within the internal Docker network. In the PoC, only the ports required for local interaction and demonstration were exposed to the host, typically for accessing the demonstration client (`app-chatbot`); issuing local test requests to the REST endpoints (when needed); and optional local inspection of MinIO (console) and PostgreSQL (for debugging only). From an architectural perspective, this configuration reinforces encapsulation: clients consume the domain APIs without direct access to the relational database or the object storage layer, preserving separation of concerns and reducing coupling. Persistent volumes are used to prevent policy and consent records from being lost across container restarts. In particular: i) **PostgreSQL**: a volume stores relational data (policy metadata, consent events, and audit traces); and ii) **MinIO**: a volume stores objects (policy documents and their versions). This persistence is necessary to preserve auditability and to enable repeated executions of

the same scenarios while comparing outcomes across runs.

Environment-Based Configuration. Services are parameterized through environment variables, which enables changes to credentials, internal endpoints, and storage configuration without code modifications. In the PoC, environment variables are mainly used to: configure PostgreSQL and MinIO credentials; provide internal service addresses for API-to-API communication; and define bucket names and storage paths for policy documents. While this does not represent a production-grade secret management strategy, it is sufficient for standardizing and reproducing the experimental setup. To make the operation of the PoC concrete, we illustrate an end-to-end execution scenario based on the deployed environment and the interaction flow depicted in Figure 1.

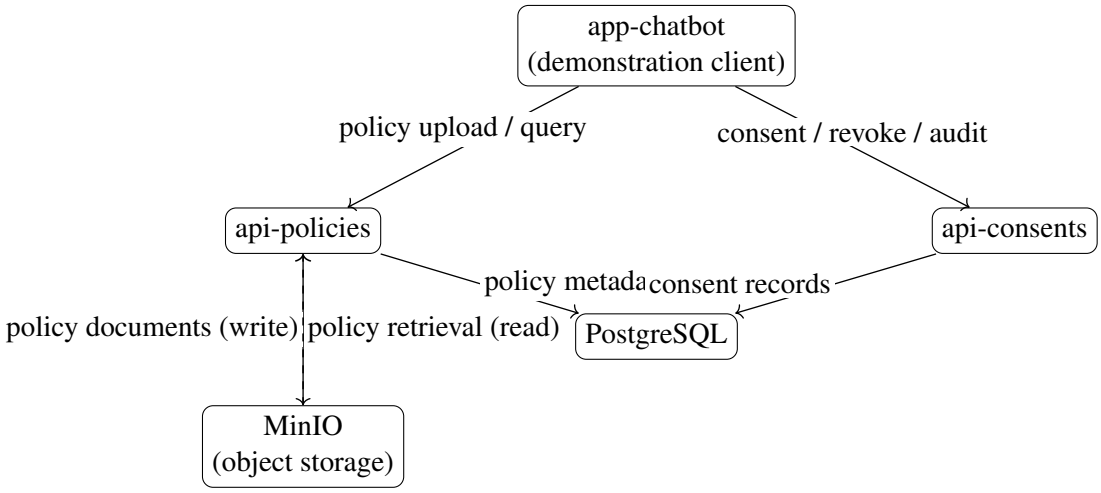

Figure 1. Inter-service communication flow in the PoC.

The main flow can be described as follows: 1. `app-chatbot` requests the latest policy or a list of available versions from `api-policies`; 2. `api-policies` retrieves metadata from PostgreSQL and persists or fetches the corresponding document in MinIO; 3. the client displays the policy and captures the user decision (accepted or refused); 4. the decision is sent to `api-consents`, which stores the event in PostgreSQL and binds it to the policy identifier and version; 5. for revocation, `api-consents` records a revocation event while preserving the historical trace, rather than overwriting prior records; and 6. for audit, the client queries `api-consents`, which returns the full history by user and/or by policy version. This design ensures that all lifecycle events relevant to evidence and accountability pass through the domain services, supporting traceability, integrity checks, and audit-oriented queries.

Deliberately Restricted Scope of the PoC. The PoC was intentionally scoped to prioritize validation of the architecture's core integrity and traceability mechanisms, namely: a) explicit policy versioning; b) storage of the original policy document in object storage; c) cryptographic hashing and persistence for independent integrity verification; d) deterministic binding between consent events and specific policy versions; and e) preservation of historical traces for auditing (including revocation).

Production grade capabilities such as TLS, authentication and authorization with JWT, RBAC, rate limiting, observability, and Kubernetes deployment were documented as evolution guidelines but were not implemented in this stage. This choice keeps the PoC aligned with its scientific objective: to demonstrate technical feasibility and auditability of the essential architectural elements. In summary, the Docker Compose deployment provides a controlled, isolated, and reproducible environment to validate policy versioning and consent recording workflows. By ensuring persistence, standardized internal communication, and clear service boundaries, the PoC establishes the conditions required to evaluate correctness of the flows, consistency of the policy–consent binding, and independent verifiability of document integrity.

To make the operation of the PoC concrete, we illustrate an end-to-end execution scenario based on the deployed environment. First, an administrator uploads a new version of the privacy policy (e.g., version 1.0.5) using the Policy Service. The document is provided as an external file (PDF or HTML), hashed using SHA-256, stored in object storage, and registered in the relational database with explicit version metadata. Next, a user interacts with the system through a demonstration client. The client retrieves the latest policy version via the Policy Service and presents it to the user. The user can explicitly accept or refuse the policy, with both options being symmetrically available.

When the user submits a decision, the Consent Service records a new consent event, including the user identifier, the referenced policy version, the decision status (given or refused), the interaction channel, and a timestamp. Each event is immutably linked to the specific policy version that was in force at the time of the interaction. Subsequently, the same user may change their decision, resulting in a new consent record while preserving the full historical trace. Revocation does not overwrite previous records; instead, it is recorded as a new event, ensuring complete temporal traceability. Finally, an audit query retrieves the full consent history for a given user, including accepted and refused events, associated policy versions, and timestamps. This scenario demonstrates that the PoC supports version-aware consent recording, historical preservation, and auditability, as required by privacy regulations and international standards. Figure 2 illustrates the outcome of this execution scenario, highlighting the preservation of historical consent records, the binding to a specific policy version, and the auditability of user decisions.

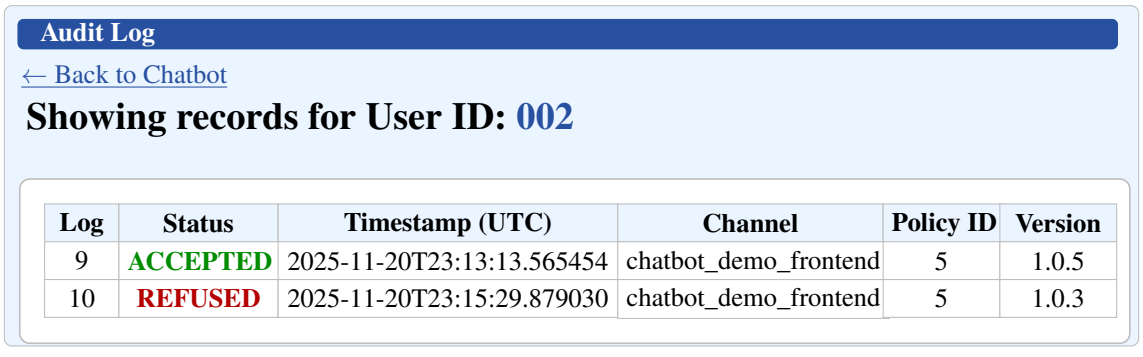

Figure 2. Mock-up of an audit log showing a user's consent history (accepted/refused), policy version, timestamps, and channel.

5. Discussion

Addressing Modular, Scalable, and Interoperable Consent Management. The results of the PoC demonstrate that a microservice-based architecture is a viable strategy to operationalize consent requirements in a modular and interoperable manner. By explicitly separating policy management and consent recording into independent services, the proposed design aligns with recommendations for compliance-oriented architectures that emphasize separation of concerns and independent evolution of components [Marillonnet et al. 2021]. The use of RESTful APIs further supports interoperability, enabling integration with heterogeneous organizational systems and audit tools, as advocated in prior work on scalable consent infrastructures [Peyrone 2022]. Scalability is addressed at the architectural level by design rather than by specific performance optimizations. The PoC shows that consent registration, retrieval, and auditing workflows can be executed independently of policy document storage, which is delegated to an object storage service. This separation reduces coupling between high frequency operations (e.g., consent events) and large, immutable artifacts (policy documents), supporting scalability and maintainability in distributed environments [Marillonnet et al. 2021].

Ensuring Traceability and Integrity through Version-Aware Design. A central contribution of this work is the explicit, deterministic binding between consent records and specific versions of privacy policies. Unlike many CMP-based approaches, which often treat policies as mutable documents without strong historical guarantees [Jha et al. 2025], the proposed architecture enforces versioning at the policy service level and persists integrity hashes for each uploaded document. Consent records reference these immutable policy versions, ensuring that the conditions under which consent was obtained can be independently verified at any point in time. This design directly operationalizes requirements from ISO/IEC 29184 [for Standardization 2020] and ISO/IEC 27560 [for Standardization 2023] regarding versioned notices and structured consent records.

The audit log demonstrated in the PoC confirms that accepted, refused, and revoked decisions are preserved as historical events rather than overwritten states. This approach strengthens accountability and demonstrability, which are core obligations under GDPR[Parliament and Council 2018] and LGPD[Macedo 2018]. In contrast to blockchain based proposals that seek immutability through distributed ledgers [Marillonnet et al. 2021], our results suggest that similar traceability guarantees can be achieved with lower complexity by combining relational integrity constraints and cryptographic hashing.

Transparent Risk Communication in Conversational Consent Interfaces. Recent studies emphasize that conversational systems and chatbots introduce unique privacy risks due to their continuous, dialog-driven data collection and users' limited awareness of processing practices [Silva and Canedo 2025]. The PoC illustrates how a conversational interface can be used as a consent interaction channel while preserving regulatory requirements, provided that consent decisions are consistently routed through a backend architecture that enforces version-aware recording and auditability. While the PoC does not attempt to optimize user understanding or behavioral outcomes, it demonstrates that transparent risk communication at the interface level can be technically decoupled from

consent persistence mechanisms. This separation is important, as prior work on user-centric chatbot design focuses primarily on interaction patterns and usability guidelines [Silva and Canedo 2024], often leaving unanswered how conversational consent decisions should be persistently recorded and audited. Our findings suggest that conversational interfaces can be safely adopted as frontends for consent collection, as long as backend services enforce strict traceability and integrity guarantees.

Regulatory Compliance and Practical Applicability. From a regulatory perspective, the architecture provides concrete evidence that GDPR and LGPD requirements for demonstrable consent, revocation, and accountability can be translated into implementable software mechanisms. The PoC supports audit queries by user, policy version, and time, enabling organizations to answer typical compliance questions such as *who consented, to what, when, and under which policy version*. This directly addresses shortcomings observed in empirical analyses of CMPs, where such evidence is often incomplete or difficult to reconstruct [Seiling et al. 2024, Novikova et al. 2025]. The deliberately restricted scope of the PoC should be interpreted as a strength rather than a limitation. By excluding production grade concerns such as authentication, authorization, and transport level security, the study isolates and validates the core architectural mechanisms required for consent traceability and auditability. This aligns with Design Science Research principles, which emphasize controlled demonstration of artifact capabilities before broader generalization [Vaishnavi 2007].

Implications and Remaining Challenges. The findings of this study have several implications. For researchers, the results highlight the importance of treating consent not merely as an interaction problem, but as an architectural and data management challenge that spans policy evolution, record integrity, and long-term auditability. For practitioners, the proposed architecture offers a reference model that can be incrementally extended with security controls, access management, and organizational governance mechanisms. Nevertheless, challenges remain. Transparent risk communication at the interface level requires further empirical evaluation to assess user comprehension and decision quality, particularly in conversational contexts [Samuel et al. 2025]. In addition, large scale deployments will require systematic evaluation of performance, storage growth, and operational costs. Finally, aligning consent architectures with emerging regulatory developments and sector specific requirements remains an open research and engineering challenge.

The findings indicate that a modular, service-oriented design can operationalize key regulatory and normative requirements for consent management in a way that is both technically feasible and audit-ready. By treating privacy policies as versioned artifacts and modeling consent as an event history explicitly bound to those versions, the proposed architecture addresses a gap frequently identified in prior work between abstract compliance requirements and their concrete realization in software systems [for Standardization 2020, for Standardization 2023, Marillonnet et al. 2021]. Rather than attempting to solve all usability, governance, and organizational challenges associated with consent, the PoC deliberately focuses on the architectural mechanisms required to preserve traceability, integrity, and historical accountability. This focus distinguishes the proposal from interface-centered or platform specific approaches, which often em-

phasize user interaction aspects while leaving auditability and policy evolution implicit or underspecified [Silva and Canedo 2024, Samuel et al. 2025]. As such, the architecture can be seen as an enabling foundation that complements, rather than replaces, advances in user-centric consent interfaces and risk communication strategies.

6. Threats to Validity

As with any artifact oriented study following a Design Science Research approach, this work is subject to some threats to validity. **Construct validity** concerns whether the concepts investigated in the study adequately capture the intended phenomena. In this work, the main constructs include consent validity, policy versioning, auditability, and traceability. A potential threat arises from the interpretation of legal and normative requirements, as regulatory texts such as GDPR and LGPD are often abstract and open to interpretation. To mitigate this threat, we derived system requirements directly from established international standards, including ISO/IEC 29100, ISO/IEC 29184, and ISO/IEC TS 27560, which provide a more concrete and structured representation of consent related concepts. While this alignment increases confidence in the operationalization of consent requirements, alternative legal interpretations or jurisdiction specific guidance may emphasize additional or different properties not explicitly captured in the current artifact.

Internal validity relates to whether the observed outcomes can be attributed to the proposed architecture rather than to confounding factors. As this study does not involve controlled experiments with human subjects, threats related to causal inference are limited. However, a potential threat lies in the simplified nature of the PoC, which was evaluated in a controlled and isolated environment. To mitigate this threat, the PoC was executed using end-to-end scenarios that reflect typical consent lifecycle operations, including policy upload and versioning, consent recording, refusal, revocation, and audit querying. These scenarios allow verification that the architectural mechanisms behave as intended, although they do not capture all complexities present in large-scale or long-running production environments.

External validity concerns the generalizability of the results beyond the studied context. The proposed architecture was validated through a PoC rather than through deployment in a production system. Consequently, its behavior under high load, heterogeneous organizational settings, or different regulatory environments cannot be fully assessed. Moreover, the demonstration client adopts a conversational interface, which may not reflect all possible interaction modalities used in real-world consent collection. Nevertheless, the architecture itself is interface-agnostic and exposes functionality through RESTful APIs, which supports integration with alternative frontends and organizational systems. This design choice partially mitigates external validity threats by facilitating adaptation to different contexts.

Conclusion validity addresses whether the conclusions drawn are reasonable given the evidence provided. Since the study focuses on architectural feasibility and traceability mechanisms rather than on quantitative performance optimization or user behavior analysis, the conclusions are intentionally limited in scope. The evaluation demonstrates that the proposed design can preserve explicit bindings between consent records and policy versions and support audit-oriented queries, but it does not claim superiority over existing platforms or comprehensive coverage of all consent-related challenges. Future

empirical studies involving real users, performance benchmarking at scale, and longitudinal observations in organizational settings are necessary to strengthen and extend the conclusions.

7. Conclusion and Future Work

This paper addressed the challenge of operationalizing regulatory and normative requirements for consent management in modern software systems. Motivated by persistent gaps between legal obligations and their concrete implementation, we proposed and evaluated a microservice-based architecture designed to support the registration, explicit versioning, and auditing of user consent. The core contribution lies in modeling privacy policies as versioned artifacts and treating consent as a sequence of immutable events that are deterministically bound to the specific policy versions in force at the time of each user decision.

By grounding the design in GDPR and LGPD requirements and aligning it with ISO/IEC standards for privacy notices and consent records, the proposed architecture demonstrates how traceability, integrity, and accountability can be systematically embedded into software systems. The PoC shows that these properties can be achieved using modular services, standard REST interfaces, and commonly available infrastructure components, while preserving a clear separation of concerns between policy management, consent recording, and client interaction. Rather than focusing on interface optimization or organizational workflows, the study emphasizes architectural mechanisms that enable audit-ready evidence and long-term accountability.

The results indicate that treating consent as a first-class, version aware entity provides a robust foundation for compliance oriented systems, particularly in environments where policies evolve over time and historical justification of data processing decisions is required. Although the demonstration client adopts a conversational interface, the architecture itself is frontend agnostic, allowing integration with different interaction modalities and organizational systems without altering the core traceability guarantees.

Future work can extend this research along several dimensions. From a technical perspective, the architecture can be enhanced with production grade security mechanisms, such as authentication and authorization, encrypted communication, and fine-grained access control. Scalability evaluations under higher workloads and in distributed deployment scenarios (e.g., Kubernetes-based environments) would provide further evidence of applicability in large-scale settings. From a user-centered perspective, future studies may integrate richer risk communication strategies and empirically evaluate how different interface designs influence user understanding and consent decisions, while relying on the same backend traceability mechanisms. Finally, longitudinal case studies in organizational contexts could assess how the proposed approach supports compliance processes, audits, and governance practices over time. This work contributes an architectural perspective on consent management that bridges legal and normative requirements with implementable software design. By focusing on policy versioning, consent traceability, and auditability, it provides a reusable foundation for trustworthy and regulation compliant consent management in contemporary software systems.

Acknowledgements

We thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Grant Nº 300883/2025-0 and 406266/2025-5.

8. Artifact Availability

The source code is available at https://github.com/ccastroelo/consentimento-api

References

Carneiro, C., Kudo, T., and Neto, R. B. (2024). Um método para transformação de requisitos legais em padrões de requisitos de software: Um estudo com a lgpd. In *Anais do XXVII Congresso Ibero-Americano em Engenharia de Software*, pages 348–355, Porto Alegre, RS, Brasil. SBC.

da Silva Junior, D. P., de Souza, P. C., and de Jesus Gonçalves, T. A. (2018). Early privacy: Approximating mental models in the definition of privacy requirements in systems design. In Mota, M., Meiguins, B. S., Prates, R. O., and Candello, H., editors, *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems, IHC 2018, Belém, Brazil, October 22-26, 2018*, pages 19:1–19:10. ACM.

for Standardization, I. O. (2020). Iso/iec 29184:2020 information technology — online privacy notices and consent.

for Standardization, I. O. (2023). Iso/iec ts 27560:2023 privacy technologies — consent record information structure.

for Standardization, I. O. (2024). ISO/IEC 29100:2024 — information technology — security techniques — privacy framework.

Gharib, M., Giorgini, P., and Mylopoulos, J. (2017). Towards an ontology for privacy requirements via a systematic literature review. In Mayr, H. C., Guizzardi, G., Ma, H., and Pastor, O., editors, *Conceptual Modeling - 36th International Conference, ER 2017, Valencia, Spain, November 6-9, 2017, Proceedings*, volume 10650 of *Lecture Notes in Computer Science*, pages 193–208. Springer.

Guerra, G. (2024). Dark patterns and the scraping consumer consent. *Privacy, Data Protection and Data-driven Technologies*.

Jha, N., Trevisan, M., Mellia, M., Fernandez, D., and Irarrazaval, R. (2025). Privacy policies and consent management platforms: Growth and users' interactions over time. *ACM Trans. Web*, 19(3):30:1–30:25.

Kalaoja, P. (2022). A consent and privacy management framework.

Kalloniatis, C., Kavakli, E., and Gritzalis, S. (2005). Dealing with privacy issues during the system design process. In *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.*, pages 546–551. IEEE.

Lu, Y., Zhang, C., Yang, Y., Yao, Y., and Li, T. J.-J. (2024). From awareness to action: Exploring end-user empowerment interventions for dark patterns in ux. *Proceedings of the ACM on Human-Computer Interaction*, 8(CSCW1):1–41.

Macedo, P. N. (2018). Brazilian general data protection law (lgpd). *Nartional Congress, accessed in October 18, 2019*.

Marillonnet, P., Ates, M., Laurent, M., and Kaaniche, N. (2021). An efficient user-centric consent management design for multiservices platforms. *Secur. Commun. Networks*, 2021:5512075:1–5512075:19.

Matos, A., Patrício, M., Nicolau, M. I., Canedo, E. D., Pereira, J. A., and Uchôa, A. G. (2025). Data privacy in software practice: Brazilian developers' perspectives. *J. Internet Serv. Appl.*, 16(1):299–319.

Merlec, M. M., Lee, Y. K., Hong, S.-P., and In, H. P. (2021). A smart contract-based dynamic consent management system for personal data usage under gdpr. *Sensors*, 21(23).

Novikova, E., Doynikova, E., and Kotenko, I. V. (2025). What are your privacy risks? privacy risk assessment based on privacy policies analysis. *Expert Syst. Appl.*, 280:127270.

Parliament, T. E. and Council, T. (2018). General Data Protection Regulation (GDPR). *Intersoft Consulting*.

Peyrone, N. (2022). Formal models for consent management in healthcare software system development. *Chulalongkorn University Theses and Dissertations (Chula ETD)*, pages 1–200.

Samuel, J., Kanakia, J., Kashyap, R., Raju, R. S., Chidipothu, S., Patel, K., and Khan, Z. (2025). Societal impacts and public perception of chatbots: Implications for individuals and organizations. *Proceedings of the 52nd Annual Northeast Business & Economics Association (NBEA) Conference, Seaview Hotel, Galloway, NJ, USA*.

Seiling, L., Gsenger, R., Mulugeta, F., Henningsen, M., Mischau, L., and Schirmbeck, M. (2024). Beware: Processing of personal data - informed consent through risk communication. *IEEE Trans. Prof. Commun.*, 67(1):4–25.

Silva, G. R. S. and Canedo, E. D. (2024). Towards user-centric guidelines for chatbot conversational design. *Int. J. Hum. Comput. Interact.*, 40(2):98–120.

Silva, G. R. S. and Canedo, E. D. (2025). Privacy in chatbot conversation-driven development: A comprehensive review and requirements proposal. *ACM Trans. Softw. Eng. Methodol.*, 34(7).

Spósito, S. L., Targino, J. F. G., Silva, G. R. S., Peotta, L., Porto, D. d. P., Mendonça, F. L. L., and Canedo, E. D. (2025). A comprehensive review of techniques, methods, processes, frameworks, and tools for privacy requirements. *Journal of Internet Services and Applications*, 16(1):508–529.

Vaishnavi, V. K. (2007). *Design science research methods and patterns: innovating information and communication technology*. Auerbach Publications, https://www.taylorfrancis.com/books/mono/10.1201/9781420059335/design-science-research-methods-patterns-vijay-vaishnavi.

Veseli, F., Serna-Olvera, J., Pulls, T., and Rannenberg, K. (2019). Engineering privacy by design: lessons from the design and implementation of an identity wallet platform. In Hung, C. and Papadopoulos, G. A., editors, *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*, pages 1475–1483. ACM.