

Article

CONSENT: A Software Architecture for Dynamic and Secure Consent Management

Christina Zoi, Ioannis Zozas and Stamatia Bibi *

Department of Electrical & Computer Engineering, University of Western Macedonia, 50 150 Kozani, Greece; dece00127@uowm.gr (C.Z.); izozas@uowm.gr (I.Z.)

* Correspondence: sbibi@uowm.gr (S.B)

Abstract

Current research in consent management techniques focuses on isolated aspects of data security, privacy, or auditability, but important issues like (i) dynamically integrating regulatory updates into form generation, (ii) support in content generation with verifiable audit trails, and (iii) tools that make compliance reasoning transparent for non-legal users are not yet addressed. This paper introduces CONSENT, an architecture that integrates AI-based consent reasoning using Large Language Models (LLMs) for automated consent-form drafting and compliance evaluation, alongside blockchain technology for secure and auditable storage. The architecture builds on prior work to address the aforementioned issues by introducing three supporting mechanisms: (a) Specialized AI models coordinated through expert routing which coordinate subtasks such as automation in form generation and regulatory compliance, (b) Retrieval-Augmented Generation (RAG) that supports the integration of regulatory updates into forms, and (c) Explainable AI (XAI) for the reasoning behind form content and compliance assessments. CONSENT architecture is evaluated through 250 test cases and a pilot case study for clinical trial consent management involving 20 engineers and attorneys, who evaluated the prototype on form quality (i.e., coherence, conciseness, factuality, fluency, and relevance) as well as time and effort efficiency. Results show that CONSENT substantially reduces the manual effort in consent-form creation while providing transparent, audit-ready compliance assessments, highlighting its potential for dynamic, user-centric consent management.

Keywords: consent management; data flows; privacy; security; blockchain; AI; LLMs

1. Introduction

Consent management involves obtaining and maintaining users' consent to process their personal data. Traditionally, consent can be acquired through both paper-based methods and digital platforms such as websites, mobile applications, and marketing systems, following three primary stages: collecting, storing, and using consent data and associated personal data [1]. Dynamic Consent Management extends this model by enabling (a) the management and modification of consent preferences over time by the individuals, and (b) the integration of changes to data protection regulations. Unlike static or purely paper-based consent models, dynamic consent leverages modern information technologies to enhance user engagement, ensure regulatory compliance, and adapt consent acquisition mechanisms in response to changes in data protection regulations [2,3].

Academic Editor: Ian Gorton

Received: 30 November 2025

Revised: 3 February 2026

Accepted: 12 February 2026

Published: 26 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Despite advancements in dynamic consent frameworks, developers still have limited support when integrating data-privacy requirements into applications. They must manually determine what information a consent form must include [4], how it should be phrased [5], and whether it satisfies complex regulatory requirements such as those defined in the GDPR [6]. This process is time-consuming, error-prone, and often results in inconsistent or incomplete privacy implementations, as well as consent forms filled with complex legal language that users struggle to understand [7]. There is therefore a clear need for mechanisms that assist developers by automating the consent-form generation process and evaluating the compliance of generated forms with major data protection regulations, such as the GDPR. Additionally, consent data must be securely stored, maintaining data integrity, immutability, timestamps, and revocation control. Traditional databases are ill-suited for guaranteeing these properties.

To address these challenges, this paper proposes the CONSENT architecture, whose primary contribution is AI-based consent reasoning that provides auditable informed Consent Generation and verification. Specifically, it grounds consent decisions through Retrieval-Augmented Generation (RAG), accompanied by blockchain-based immutable audit trails. To operationalize this, the architecture further integrates three supporting mechanisms:

- Specialized AI models coordinated through expert routing for legal-text refinement, regulation compliance checking, and explainable AI components which demonstrate the reasoning processes behind the generation of each consent form.
- Human-in-the-Loop Consent form drafting with the use of LLMs that support developers to generate and personalize consent requirements and forms in alignment with regulations.
- Accessibility and user experience enhancement by incorporating speech recognition.

Together, these innovations aim to create a transparent, intelligent, and secure dynamic consent-management ecosystem that addresses the technical, regulatory, and trust-related challenges of modern data governance, as illustrated in Figure 1, where the circular nodes represent the objectives, while the dotted lines connecting them indicate the underlying methodologies that will be discussed.

Having outlined the main novelties of this study, the remainder of the paper is organized as follows: Section 2 presents related work and the necessary background information, whereas Section 3 consists of the system architecture and layered design. In Section 4, the case study design is presented. Section 5 highlights the results and Section 6 presents a discussion on the findings. Finally, in Section 7, the paper outlines threats to validity, and Section 8 concludes the paper.

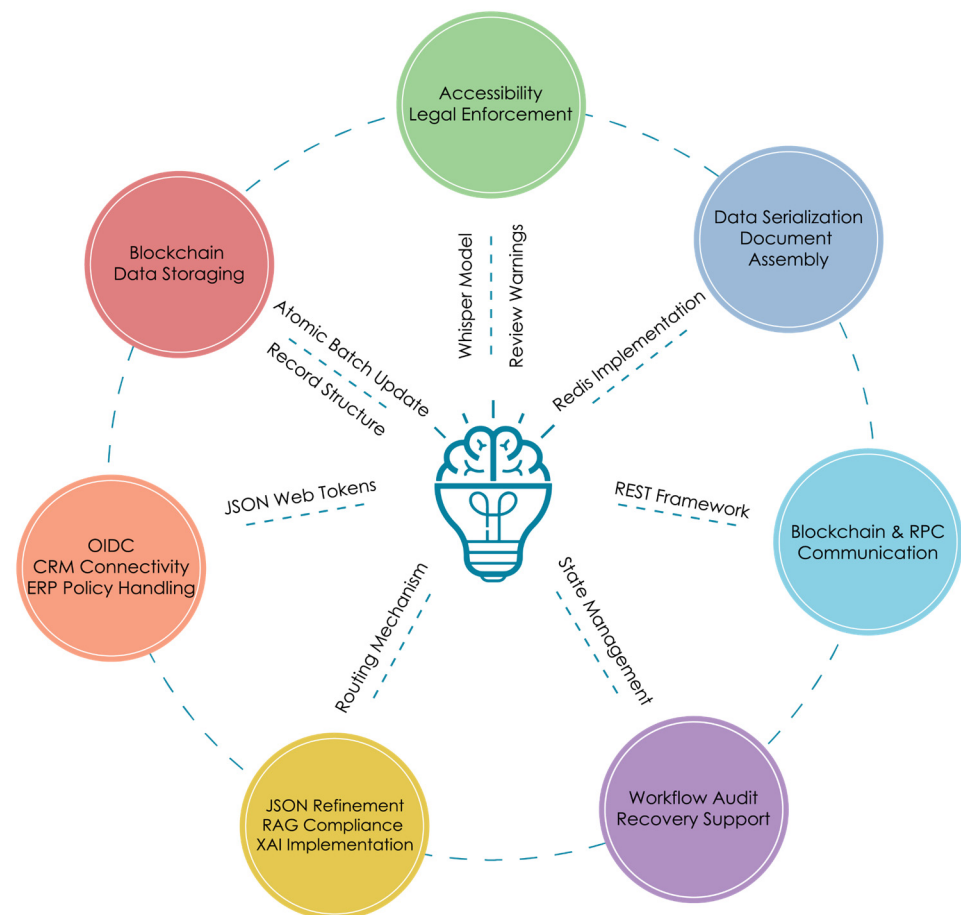


Figure 1. CONSENT supporting mechanisms.

2. Related Work

Consent management from the viewpoint of privacy and security has been an increasingly popular topic of research and investigation over the last decade, in which blockchain has been a predominant proposed technology. Kakarlapudi et al. [8] proposed blockchain as a solution for secure consent management, while Merlek et al. [6] implemented blockchain architecture. Khalid et al. [9,10] focused on access for people with accessibility needs. Mehmood et al. [11] focused on implementing user management. Furthermore, Lapwattanaworakul et al. [12] utilized blockchain smart contracts as an auxiliary system for managing consent to ensure data integrity and compliance with relevant laws. Ayappane et al. [13], since consent management cannot be implemented exclusively with autonomous authorization, proposed a four-tier architecture where consent-management issues are addressed at different levels of sensitivity, namely Regulation, Policy, Contracts, and Key Management. Moreover, Xu et al. [4] proposed an automated consent-management model based on blockchain technology that ensures user security, transparent data exchange, and automated consents. Many studies focus solely on the healthcare sector's consent management, based on blockchain technologies. A variety of studies presented platforms for medical-data consent management [14–22] incorporating either cryptography [23], data sharing [24,25], auditing [26,27], consent withdrawal [5], blockchain [28–30] and artificial intelligence algorithms to track consents [31,32].

Regulatory compliance has also been a subject of study in [11], including the requirement of constant consent renewal that affects both privacy and security [33]. The appropriateness of the consent type [34,35], consent management complexity [13], as well as data over-collection are major obstacles to this process [25].

On the other hand, LLMs as a means to automate the consent-form generation have been sparsely studied. Peyrone et al. [36,37] incorporated LLM algorithms for implementing informed consents to achieve consent management and classification. On the other hand, Porcelli et al. [38] incorporated LLMs to simplify consent presentation and retrieval.

Table 1 presents an overview of the related work. Based on the analysis, blockchain is a dominant technology to achieve consent management. A few studies incorporate AI or LLMs to automate consent workflows. The current paper focuses on combining the technologies suggested by previous studies to fill the gap in consent form automation and regulatory compliance and improve overall efficiency in consent management while adapting blockchain technology to ensure transparent and secure consent auditing.

Table 1. Proposed architectures.

<i>Proposed Architectures</i>	<i>General Purpose</i>	<i>Auditing</i>	<i>Automation</i>	<i>Regulatory Compliance</i>	<i>Blockchain</i>
[21,23,25]				✓	✓
[28–30]				✓	✓
[15,18,22,24,26,27,31]		✓		✓	✓
[14,16,17,19,20]		✓		✓	✓
[5,6,11,13]	✓		✓	✓	✓
[36,37]	✓		✓	✓	✓
[9,10]	✓			✓	
[8,12]				✓	✓
[32,38]			✓	✓	
Current paper	✓	✓	✓	✓	✓

3. System Architecture and Layered Design

In this section, the CONSENT architecture is outlined as a seven-layered framework that adopts AI techniques and blockchain technology to generate, verify, and securely manage dynamic user consent across complex data flows. The architecture consists of the Data Persistence and Ledger Layer that is used to store encrypted consent records off-chain and hashed consent texts and revocation events on a permission blockchain (on-chain). Above it, the Integration and Identity Layer manages authentication, authorization, and interoperability with external systems. The core intelligence of the system resides in the AI Processing and Compliance Intelligence Layer, where a routing mechanism addresses different models for form generation, calculation of legal compliance scores, and explainability features. The Consent Workflow Orchestration Layer coordinates the full consent process—from creation and human legal review to approval, signature, updates, and withdrawal—using asynchronous, stateful workflows. Communication between components is handled by the API and Communication Layer, which exposes secure REST/WebSocket interfaces and manages events, blockchain RPCs, and inter-service messaging. The Presentation Layer transforms AI outputs into finalized, signed, and regulator-ready documents enriched with blockchain proofs and multimodal accessibility features. Finally, the Application Layer provides user-specific interfaces for data subjects, controllers, auditors, and administrators, offering real-time compliance feedback, voice-enabled interaction, and clear XAI-based explanations to ensure transparency, trust, and meaningful human oversight. Conclusively, the organization of this approach fits into

seven layers, as presented in Figure 2, with each of them handling the following specific functions and responsibilities.

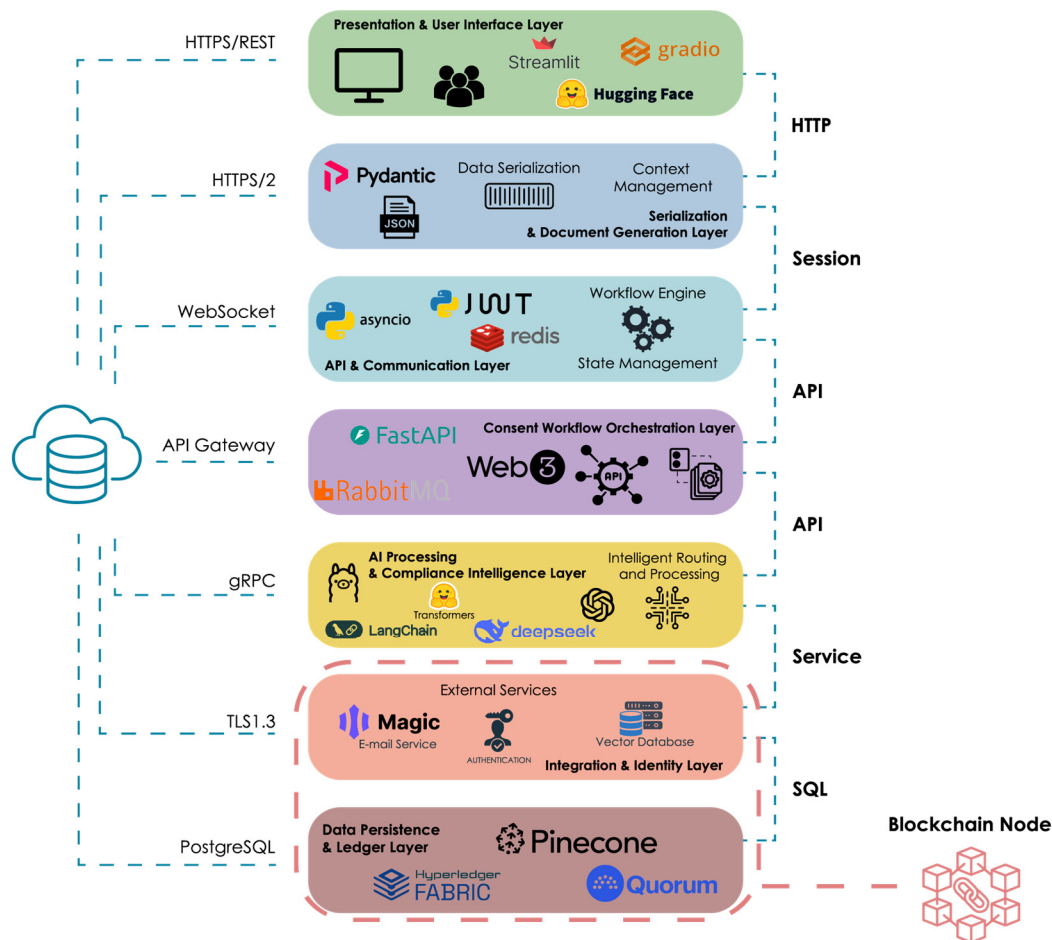


Figure 2. CONSENT layered system architecture.

The following tables (i.e., Tables 2 and 3) outline the interactions between the existing layers, followed by the communication protocols and technologies during the workflow. Furthermore, Table 4 addresses the system components and trust assumptions per layer for threat model assessment.

Table 2. Inter-layer communication protocols—consent lifecycle.

Step	Involved Layers	Recommended Protocols
Form generation and compliance Scoring	7-5-3-1	HTTPS, gRPC, Vector Search
Template storage	5-4-1	AMQP (Celery), PostgreSQL, JSON-RPC
Email notification	4-2	SMTP/TLS
Form viewing	7-5-1-6	HTTPS, SQL, Template rendering
Signature submission	7-5-4-1	HTTPS POST, PostgreSQL transactions, S3 API
Blockchain recording	4-1	Web3.py JSON-RPC 2.0, Smart contract call
PDF generation	4-6-1	Internal function call, WeasyPrint, S3 Storage

Real-time updates	4-7	WebSocket (WSS)
External integrations	4-2	Webhooks (HTTPS POST), REST APIs

Table 3. Inter-layer communication protocols—digital signature workflow.

<i>Step</i>	<i>Involved Layers</i>	<i>Recommended Protocols</i>
Unsigned PDF Generation	4-6-1	Internal gRPC, WeasyPrint, S3 API
Digital Signature	6- External-6	HTTPS REST, OAuth 2.0 JWT, Webhook
Signed PDF Download	6- External	HTTPS GET, Base64 encoding
Signature Certificate Validation	6- Internal	PyPDF2, OCSP, CRL
Signature Blockchain Recording	6-4-1	gRPC, Web3.py JSON-RPC 2.0
Final PDF Storage	6-1	S3 API, AES-256-GCM encryption

Table 4. System components: inputs, outputs, and trust assumptions.

<i>Layer</i>	<i>Inputs</i>	<i>Outputs</i>	<i>Trust Assumptions</i>
1. Data Persistence	User data, consent records	Encrypted data, hashes	Infrastructure security, encryption keys protection
2. Integration and Identity	Auth requests, credentials	Access tokens, verified identities	Trusted Identity providers, secure key management
3. AI Processing	User prompts, regulations	Consent forms, compliance scores	Review of model outputs, accuracy of retrievals
4. Workflow Orchestration	Workflow triggers	State transitions, notifications	Task queue integrity, state consistency
5. API and Communication	HTTP/WS requests	JSON responses, blockchain TXs	Working API authentication
6. Document Generation	Templates, user data	Signed PDF	PDF integrity maintained
7. Presentation	User interactions	Initial consent forms for refinement	Client-side validation

3.1. Data Persistence and Ledger Layer

Providing trustful foundational storage for the diverging types of the system's data, such as rational, object, vector, and blockchain, is a major priority. Data protection regulations demand fundamental design principle separations: only hashing and revocation events go on-chain, with personal data such as names and electronic signatures staying off-chain. The recommendation of cryptographic information proves the following: the form's existence at a timestamp, no tampering occurred; through comparing current and original hashes, and consent was not revoked; given consent ID status. Blockchain, in this case, can be characterized as a storage medium. This data severance technique is applied using the following compliant document management of secure storage, tamper-proof revocation, blockchain-based consent hashing, encrypted off-chain storage (i.e., *personally identifiable information (PII) and sensitive content remain in encrypted off-chain databases and object storage*), and managing vector embeddings. The layer specifically implements atomic batch updates using the *ConsentRecord* structure, as explained by Algorithm 1, which consists of a Primary Key, an Approvals Map in the form of a dictionary structure mapping consent categories to their states, and Metadata (e.g., *timestamps*). The associated technologies of the layer are presented in Table 5. This granular Map structure enables

modifying specific consent categories without affecting others within the same digital entity.

Algorithm 1. Batch Update System

<i>Purpose:</i> Updates a client's records by processing multiple changes together as a single atomic transaction.	record $\leftarrow$ getState(clientId)
<i>Input:</i> ctx (i.e., <i>blockchain connection</i>), clientId, updatesJson	if record == null then
<i>Output:</i> Success or Failure	record $\leftarrow$ new ApprovalRecord()
<i>Process:</i> Retrieves a client's existing record or creates a new one and updates it by recording the category, status, and current timestamp into the approval record. It updates the last-modified timestamp and saves the record back to the blockchain ledger.	for each update in updatesJson do
<i>Key Characteristics:</i> (a) atomic operation (i.e., <i>all updates succeed or fail together</i>), (b) blockchain state management (i.e., <i>through getState/putState</i>), (c) timestamp maintenance for audit trails, and (d) data storing in JSON format.	record.category $\leftarrow$ update.category
	record.status $\leftarrow$ update.status
	record.timestamp $\leftarrow$ now()
	record.lastModified $\leftarrow$ now()
	putState(clientId, record)
	return SUCCESS

Table 5. Data Persistence and Ledger Technologies.

<i>Technology Category</i>	<i>Primary Options</i>	<i>Alternative Options</i>
Relational Database	PostgreSQL	MySQL 8.x, Microsoft SQL Server, Oracle Database
Object Storage	MinIO/AWS S3	Google Cloud Storage, Azure Blob Storage, Ceph
Vector Database	Pinecone	Weaviate, Milvus, Qdrant, pgvector
Permissioned Blockchain	Hyperledger Fabric, Quorum	Corda, MultiChain, Polygon Supernet

3.2. Integration and Identity Layer

This layer connects the system with third parties and handles trust establishment, secure integration, identity management, and access control. The layer also operates on a zero-trust architecture with identity verification at every link point. The associated technologies of the layer are presented in Table 6. The goal is external systems that can interact with consent data only through authenticated and authorized channels, maintaining security boundaries, thus accomplishing interoperability. The responsibilities of the layer are addressed as

1. **Authentication and Authorization:** The OpenID Connect (OIDC) authentication protocol is implemented, which by default operates on top of the OAuth 2.0, controlling the authorization flow through JSON Web Tokens (JWTs).
2. **Integration with External Enterprise Systems:** Connectivity with CRMs and ERPs enables consent data to flow into probabilistically existing business workflows. This architectural decision allows organizations to leverage consent information within their predetermined operational Systems. Some examples may be (a) triggering compliance checks in CRM workflows, (b) updating patient records in health systems, or (c) enforcing data handling policies in ERP platforms.
3. **Email and Notification Gateways:** The layer manages communication channels for sending consent requests and signature notifications. Email gateways handle the

delivery of time-sensitive notifications and maintain audit trails, thus proving consent requests were properly delivered and acknowledged when needed.

4. **Identity Verification for Controllers:** Unauthorized access and identity fraud are two risk-prone factors that emerge from the system's legal sensitivity. Verification mechanisms establish and validate the identities of data controllers, processors, and subjects (end users). Conclusively, this is a means of conservation that only the rightful parties can create consent forms, process consent data, or access their own consent records.
5. **API Credentials and Access Policies:** The layer manages both API keys and access tokens. In this context, role mapping ensures that different users (e.g., administrators, compliance officers, end users) are granted the appropriate permissions.

Table 6. Integration and Identity technologies.

<i>Technology Category</i>	<i>Primary Options</i>	<i>Alternative Options</i>
Authentication	OpenID Connect (OIDC), OAuth 2.1	SAML 2.0, CAS, Kerberos, OpenAM, LDAP
Passwordless Authentication	Magic.link	WebAuthn/FIDO2, Passkeys, Firebase Auth
API Gateway	Kong Gateway, Tyk, AWS API Gateway, Traefik	Apigee, MuleSoft, Azure API Management, Ambassador, KrakenD
Token Management	JWT (RS256/ES256)	PASETO, JOSE (JWE/JWS)
Rate Limiting	Redis token bucket	Nginx rate_limit, Envoy
Webhooks	Hookdeck	Svix, webhook.site
ERP Integration	SAP HANA connectors, Oracle EBS adapters, Odoo API	Microsoft Dynamics connectors, NetSuite API, Sage adapters
CRM Integration	Salesforce REST API, HubSpot API, Dynamics 365	Zoho CRM API, Pipedrive API, SugarCRM API

3.3. AI Processing and Compliance Intelligence Layer

3.3.1. AI Processing and Compliance Intelligence Subsystems

This layer orchestrates AI-powered tools for generation, compliance verification, and explainability. The foundation of the underlying technology and the contribution to dynamic consent management is the use of Large Language Models. Transformers enable attention mechanisms while LangChain orchestrates complex AI workflows by (a) chaining multiple model calls and (b) retrieving operations and processing steps into coherent pipelines for the system to work. The three distinct specialized subsystems which are proposed, to accomplish the main objectives consist of (a) Consent Generation and Refinement Subsystem, (b) Compliance Verification Subsystem, and (c) Explainability Subsystem (XAI).

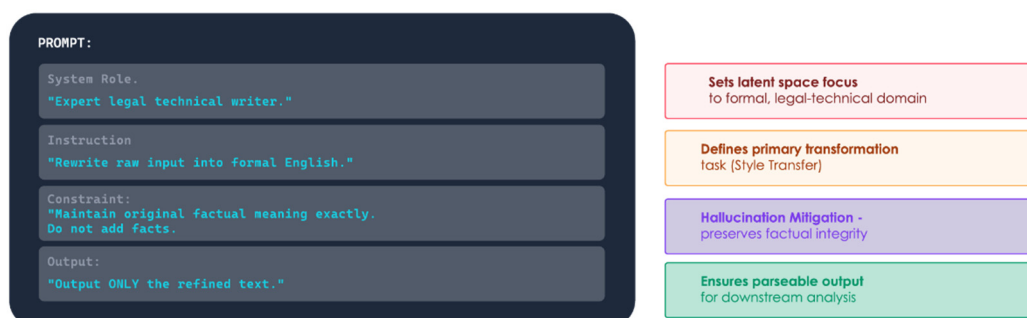
3.3.2. Technical Implementation Details

The Consent Generation and Refinement Subsystem employs deterministic rule-based classification based on request type, with the selection occurring at the request level. All experts are independently fine-tuned models rather than components of a unified sparse architecture with shared parameters.

Algorithm 2 presents the routing logic directing the requests, while the following Figure (i.e., Figure 3) outlines the prompt structure.

Algorithm 2. Request Classification and Expert Routing

<i>Purpose:</i> Routes user requests to specialized expert models	Input: user_request, context
<i>Input:</i> user_request (the query), context (additional information)	Output: expert_response
<i>Output:</i> expert_response (result)	
<i>Process:</i> Classifies the incoming request. If it is a generation type request, it routes the mechanism to Expert 1 (i.e., <i>Legal Refinement</i>)	requestType $\leftarrow$ classify(user_request)
If it is a compliance check request, it routes to Expert 2 (i.e., <i>Compliance Verification</i>). The selected expert then processes the prompt and returns the result.	if requestType = "generation" then expert $\leftarrow$ Expert1 // Legal Refinement prompt $\leftarrow$ buildGenerationPrompt (user_request, context)
<i>Key Characteristics:</i> (a) intelligent request classification, (b) expert-specific routing logic, and (c) dynamic prompt construction.	else if requestType = "compliance" then expert $\leftarrow$ Expert2 // Compliance Verification prompt $\leftarrow$ buildCompliancePrompt(user_request, context)
	end if
	expert_response $\leftarrow$ expert(prompt)
	return expert_response

**Figure 3.** Refinement subsystem prompt structure.

The RAG knowledge base integrates the full text of *GDPR Articles 1–99*, relevant *sectoral regulations*, and official *EDPB guidance*, utilizing the prompt structure detailed in Figure 4.

**Figure 4.** Grounding subsystem prompt structure.

The initial technical evaluation of chunking strategies reveals limitations in retrieval performance, as shown in Table 7. Fine-grained segmentation introduces noise through incomplete legal provisions, resulting in diminished precision with baseline configurations proving to be incapable of addressing pinpoint article retrieval and multi-article reasoning queries. A hierarchical Multi-Scale Semantic Indexing (MSSI) framework is therefore proposed, which maintains four parallel indices, each operating at distinct semantic resolutions with semantic boundary detection, using legal document structure (i.e., *Article/Recital*) as shown in Algorithm 3. This architecture increased recall by 10.9% over the best baseline and precision by 12.0%.

Table 7. Retrieval performance across chunking configurations.

<i>Configuration</i>	<i>Recall</i>	<i>Precision</i>
256 tokens/25 overlap	0.87	0.68
512 tokens/50 overlap	0.82	0.75
768 tokens/75 overlap	0.76	0.79
1024 tokens/100 overlap	0.71	0.77
Hierarchical Multi-Scale (Proposed)	0.91	0.84

Algorithm 3. Hierarchical multi-scale chunking

<i>Purpose:</i> Breaks legal documents into overlapping chunks while respecting semantic boundaries like articles and recitals.	Input: document D, boundaries B, configs C for each level Output: hierarchical_chunks H // multi-scale chunks tokens $\leftarrow$ tokenize(D) semantic_points $\leftarrow$ detectBoundaries(D, B)
<i>Input:</i> document D, boundaries B, configs C (i.e., four size/overlap pairs)	H $\leftarrow$ empty list
<i>Output:</i> hierarchical_chunks H (i.e., four granularity levels)	for each config in C do
<i>Process:</i> Detects semantic boundaries in the document and tokenizes the text. For each configuration level, it creates chunks by sliding through the token sequence. When a chunk boundary approaches, the system searches for nearby semantic boundaries and adjusts to align with them. The process continues with decreasing overlap at each level.	level_size $\leftarrow$ config.size level_overlap $\leftarrow$ config.overlap start $\leftarrow$ 0 while start < length(tokens) do end $\leftarrow$ start + level_size nearest_boundary $\leftarrow$ findNearest(semantic_points, start, end) if nearest_boundary is within threshold then end $\leftarrow$ nearest_boundary end if chunk $\leftarrow$ tokens[start : end] append H[level] with chunk start $\leftarrow$ end – level_overlap end while
<i>Key Characteristics:</i> (a) boundary-aware chunking respecting document structure, (b) multi-scale representation, and (c) configurable overlap for context preservation	end for return H

Dynamic consent is ensured through multi-layered change detection and versioning architecture as presented in Algorithm 4. Automated scrapers with content hashing (i.e., *SHA-256*) monitor official legal sources, thus detecting changes by comparing cryptographic information, while containerized services (e.g., *Docker* or *Kubernetes*) run the background processes.

Algorithm 4. Versioned policy update and re-consent triggering	
<i>Purpose:</i> Monitors legal document sources for changes, detects updates, and triggers re-consent workflows.	Input: current_versions V_curr, monitored_sources S Output: updated_corpus D', re-consent_queue Q_consent D' ← empty list Q_consent ← empty queue
<i>Input:</i> current_versions V_curr, monitored_sources S	for each source in S do
<i>Output:</i> updated_corpus D', re-consent_queue Q_consent	new_content ← scrape(source) new_hash ← hash(new_content)
<i>Process:</i> Scrapes each source and compares the content hash against stored versions. When changes are detected, it computes the difference and stores it. For each updated document, the system creates a new version with content hash, identifies affected users, and updates the vector database.	if new_hash ≠ V_curr[source].hash then diff ← computeDifference(V_curr[source].content, new_content) storeUpdate(source, new_content, new_hash, diff) append D' with new_content affected_users ← identifyAffectedUsers(source) enqueue Q_consent with affected_users
<i>Key Characteristics:</i> (a) hash-based change detection for efficiency,	updateVectorDatabase(source, new_content) end if
(b) identify content changes, and	end for
(c) automated user notification triggering.	return D', Q_consent

The XAI subsystem performs GDPR compliance analysis through a framework co-developed by a certified Data Protection Officer with 10 years of auditing experience, who provided guidance on regulatory requirement prioritization and legal terminology. Six regulatory categories were derived for keyword detection and semantic similarity analysis (i.e., *Legal Basis for Processing, Data Subject Rights, Purpose and Transparency, Security and Protection, International Data Transfers, and Right to Withdraw*). The implementation deploys requirement-level checklists mapping GDPR articles, with foundational articles (i.e., 6, 7, 13, and 15) receiving double weight. Obligation-based verification that triggers specific pattern checks for binary requirements beyond simple keyword presence is also employed to further strengthen the compliance model as presented in Algorithm 5 followed up with the corresponding technologies of the layer; which are illustrated in Table 8.

Algorithm 5. GDPR compliance evaluation	
<i>Purpose:</i> Evaluates consent forms against GDPR requirements by checking for	C_lower ← toLowerCase(C) category_breakdown ← empty dictionary

elements and computing compliance scores resulting in categorical breakdown in four levels.	total_weighted_score, max_weighted_score $\leftarrow$ 0
Input: consent_form C, required_elements, article_weights, semantic_threshold	for each category in required_elements do
Output: compliance_score, category_breakdown	found_terms $\leftarrow$ empty list, category_score $\leftarrow$ 0, category_max $\leftarrow$ 0
<i>Process:</i> Converts the consent form to lowercase and searches for required keywords within each compliance category while computing semantic similarity scores against regulatory text. Foundational articles (6, 7, 13, 15) receive double weight ($w = 2.0$) while supplementary provisions receive standard weight ($w = 1.0$). Binary requirements trigger obligation-based pattern verification beyond keyword presence. The weighted compliance score incorporates both keyword detection and semantic similarity, with status assigned based on thresholds (Excellent, Good, Moderate, or Needs Work)	for each element in required_elements[category] do
	weight $\leftarrow$ (element.article in {6, 7, 13, 15}) ? 2.0 : 1.0
	keyword_detected $\leftarrow$ element.keyword
	semantic_score $\leftarrow$ computeSemanticSimilarity(C, element.regulatory_text)
	pattern_verified $\leftarrow$ element.is_binary ? verifyObligationPattern(C, element.pattern) : true
	if (keyword_detected OR semantic_score $\geq$ threshold) AND pattern_verified then
	element_score $\leftarrow$ weight
	append found_terms with element.term
	else
	element_score $\leftarrow$ 0
	end if
	category_score $\leftarrow$ category_score + element_score
	category_max $\leftarrow$ category_max + weight
	end for
	category_breakdown[category] $\leftarrow$ {found_terms, score: category_score, max: category_max}
	total_weighted_score $\leftarrow$ total_weighted_score + category_score
	max_weighted_score $\leftarrow$ max_weighted_score + category_max
	end for
	compliance_score $\leftarrow$ (total_weighted_score / max_weighted_score) * 100
	status $\leftarrow$ compliance_score $\geq$ 90 ? "Excellent" :
	compliance_score $\geq$ 75 ? "Good" :
	compliance_score $\geq$ 50 ? "Moderate" : "Needs Work"
	return compliance_score, category_breakdown, status

Table 8. AI processing and compliance intelligence technologies.

<i>Technology Category</i>	<i>Primary Options</i>	<i>Alternative Options</i>
LLM Foundation	GPT-4 Turbo, Claude 3.5 Sonnet, DeepSeek-V3	Gemini 1.5 Pro, Command R+, Mistral Large, Phi-3 Medium, LLaMA 3.1 405B
Embedding Models	text-embedding-3-large, Cohere embed-v3, BGE-M3	instructor-xl, E5-mistral-7b, nomic-embed-text, all-MiniLM-L6-v2
RAG Framework	LangChain, LlamaIndex, Haystack	Semantic Kernel, txtai, Weaviate RAG, Vertex AI Search

Vector Retrieval	FAISS with IVF-PQ	HNSW, ScaNN
Prompt Optimization	DSPy, LangSmith	Guidance, LMQL
MoE Architecture	Mixtral 8 × 7B, Switch Transformer	DeepSeekMoE, Sparse Upcycling, GL-MoE, Expert Choice Routing
Fine-Tuning	LoRA, QLoRA, Adapter Tuning	Prompt Tuning (PEFT), Prefix Tuning, IA3, DoRA
Multimodal	Whisper (STT)	Bark/VALL-E (TTS), GPT-4V
Chunking Strategies	Semantic chunking, Recursive splitter	Sliding window, Fixed-size chunking, Markdown-based splitting
Inference Protocol	HTTP REST with SSE, gRPC streaming	WebSocket, Triton gRPC

3.4. Consent Workflow Orchestration Layer

According to GDPR Articles 5 (2—Accountability) and 30 (records of processing activities), the layer demands that workflow remains auditable and recoverable. These articles strictly state that consent processes must be tracked through their complete lifecycle, surviving system failures, and providing audit trails. The legal compliance workflow management is based on

1. **Version Control:** Every modification event is versioned, maintaining historical records of the time and person modifying the legal corpora, while enabling rollback to each previous version.
2. **Handling Withdrawals and Expirations:** The layer manages dynamic consent states by handling withdrawal requests and triggering automated expiration workflows when time-limited consents reach their validity period.
3. **Routine Management:** Due to the legal and complicated nature of the system, the existing operations range over time. Blockchain confirmations, compliance verification checks, and batch processing of expired consents are long-running procedures that can be executed asynchronously without blocking the other user-facing operations.
4. **Workflow State Persistence:** Network failures, Server restarts, and user session timeouts are common possibilities associated with every system. To deal with these factors and the emerging risks of data loss and user frustration, this layer manages multi-step workflows.

Subsequently, the corresponding technologies of the consent orchestration are presented in Table 9.

Table 9. Consent workflow orchestration technologies.

<i>Technology Category</i>	<i>Primary Options</i>	<i>Alternative Options</i>
Task Queue	Celery 5.3+, Apache Airflow	Temporal.io, Apache Kafka + Kafka Streams, AWS SQS + Lambda, Google Cloud Tasks
Async Runtime	asyncio (Python 3.11+)	Trio, AnyIO
State Management	Redis 7.x, Apache Ignite	Memcached, Hazelcast IMDG, KeyDB
Version Control	Git, DVC (Data Version Control)	LakeFS, Pachyderm
Event Sourcing	EventStoreDB, Apache Kafka	Axon Server, SQL Stream Builder

3.5. API and Communication Layer

Secure service-to-service interactions must be provided within the consent management system while all internal and external communications remain authenticated, rate-limited, and auditable. This layer secures a unified gateway by the three following mechanisms and technologies which are presented in Table 10.

1. **REST and Websocket Endpoints:** WebSocket connections are responsible for real-time updates, namely (a) user notifications for required signatures and (b) stakeholder notifications for complete blockchain anchoring. FastAPI can serve all user-facing endpoints (e.g., GET/consents) and JSON response formatting with automatic OpenAPI documentation.
2. **Authentication and Format Transformation:** In the context of internal and external request validations, it must be ensured that they are all reliable. Some examples include the following: all required fields are present, email formats are valid, and consent scope values match allowed enumerations before any processing continues. Pythons' Pydantic is widely used in web frameworks and libraries like the recommended FastAPI, but further follows up with Django, Flask, and HTTPX, making it a valid automation suggestion.
3. **Blockchain RPC Communications:** Transactions that contain consent hashes and timestamps followed up with must be submitted to a service in a high-performance way. Transaction confirmation of events and retrieval of historical consent records must also be handled responsibly. Subsequently, the software communication protocol of Remote Procedure Calls (RPCs) is needed in order for the system to function while abstracting the blockchain complexity from the upper-layers and developers. Python's Web3.py can optimally handle blockchain interaction by (a) submitting consent hash transactions, and (b) querying historical consent records.

Table 10. API and Communication technologies.

<i>Technology Category</i>	<i>Primary Options</i>	<i>Alternative Options</i>
REST Framework	FastAPI	Django REST Framework, Flask-RESTful, Express.js
WebSocket	Socket.IO	WebSocket (native), SockJS, ws library, uWebSockets
API Documentation	OpenAPI 3.1	Redoc, Stoplight
Validation	Pydantic	Cerberus, jsonschema, Voluptuous
Serialization	JSON	Protocol Buffers, MessagePack
Blockchain RPC	Web3.py, Hyperledger SDK	ethers.js
Request Throttling	SlowAPI	Django-ratelimit, Flask-Limiter, Express Rate Limit

3.6. Serialization and Document Generation Layer

This layer handles the user consent signature which is the legally binding agreement from the data subject. The output of the discussed layer is a signed PDF with embedded blockchain proof. The corresponding technologies are presented in Table 11. In order for this implementation to function, backend and AI outputs must be converted into representations that people and the system can understand, share, store and submit.

4. **Format Serialization and Conversion:** Handling the conversion of the consent management system's data according to the required formats of each different system

- consumer into various output formats is a primary function, specifically: JSON for application programming interfaces (APIs), Extensible Markup Language (XML) for official submissions to regulators, and binary (MsgPack) for Redis caching. Pre-rendered templates and user-specific data are stored in cache with a time-to-live (TTL), which reduces repeated processing without delays. Content negotiation plays a primary role here, selecting the response formats based on client requirements, while session state management tracks user progress in multi-step processes (e.g., *reviewing, generating, and signing consent*), maintaining scalability horizontally across layers.
5. Document Assembly: Raw LLM outputs (e.g., *text-structured JSON, scores, and explanations*) must be transformed into regulatory PDFs. In addition, the documents to be created should follow the ISO 32000-1 standard [39] for PDF Advanced electronic signatures. The recommended core technology for pdf rendering is WeasyPrint or PyPDF2.
 6. Digital Signature Integration: For server-side signatures, the method of internal signing through PyHanko can be applied.

Table 11. Serialization and Document Generation technologies.

<i>Technology Category</i>	<i>Primary Options</i>	<i>Alternative Options</i>
Format Conversion	LibreOffice API	Pandoc, Aspose
JSON Processing	ujson, orjson, simdjson	rapidjson, json5, GSON
XML Processing	lxml	ElementTree, xmldict
Binary Serialization	MessagePack	Protocol Buffers, CBOR
Caching Layer	Redis	Cloudflare, Apache Traffic Server, Memcached
Session Management	Redis Sessions	JWT stateless, Flask-Session
PDF Generation	WeasyPrint	Puppeteer, wkhtmltopdf
Digital Signature APIs	DocuSign REST API, Adobe Sign	SignNow, HelloSign, PandaDoc

3.7. Presentation and User Interface Layer

The layer serves as an intermediate link, separating the consent management system and end-users while catering to stakeholders to immediate access to the proposed contemporary operations. It implements a user-friendly interface that abstracts the system's overall distributed complexity through auto-generated endpoints for real-time refinement and compliance testing. There is a contribution to multimodal interaction and accessibility supporting form editing, voice input for text generation, and output for aloud readable forms. Real-time bidirectional communication streams blockchain confirmations and signature completions, aiming to achieve low latency through persistent connections eliminating polling overhead. A final but key point is the fact that the layer enforces mandatory human-in-the-loop oversight through legal review warnings to prevent immediate deployment, making the tool compliant with GDPR Article 22 requirements for meaningful human oversight in automated legal decisions. The Presentation Layer is a web application with multiple deployment options, such as (a) Static Hosting with CDN, (b) Server-Side Rendering (SSR), or (c) Containerized Deployment. There are several alternatives that can connect the proposed system and technologies, with most of the modern methods to be considered effective. All the layers' functions communicate exclusively through Layer 5 (*API and Communication Layer*) via HTTPS REST and WebSocket protocols. All data processing, storage operations, blockchain interactions, and AI inference are

delegated to the backend infrastructure (Layers 1–6), which must be hosted on separate facilities.

4. Case Study Design

To evaluate the effectiveness of the proposed architecture, an initial case study limited to the medical sector has been performed, for managing the consent process of clinical trial participation. This pilot implementation is subsequently expanded to include automated test cases and comprehensive stress testing, thus validating findings and assessing system performance at scale. To derive further comparisons, an ablation study is employed during the test cases scenario, providing baselines for (a) *Expert Routing Framework* vs. *Single Model*, (b) *RAG* vs. *no-RAG*, and (c) *XAI* vs. *no-XAI* comparison. All the code, algorithms, test cases, and evaluation material can be found at the GitHub repository (<https://github.com/ChristiAdaLovelace/CONSENT->, accessed on 3 January 2026). The main objective is to evaluate the effectiveness of the CONSENT architecture with respect to the perceived usability and performance. For this purpose, the goal is decomposed into two research questions:

RQ1: How do domain experts evaluate the consent forms generated by the CONSENT framework in terms of coherence (i.e., how easily interpretable were the responses and how logically did the LLM convey information, ensuring that the explanations and guidance provided were easy to follow and free from ambiguity), relevance (i.e., how well the responses are able to comprehend personal data privacy concepts, including consenting risks and details of different domains), factuality (i.e., the inclusion of current and relevant information from legal sources or privacy standards), fluency (i.e., the ability to produce content that flow smoothly, maintaining a consistent tone and style including both grammatical correctness and readability, mimicking human-like text writing) and conciseness (i.e., the ability of the model to produce text that is structurally sound, easy to follow, and free from unnecessary repetition).

RQ2: What is the performance of CONSENT architecture with respect to (a) system responsiveness, (b) efficiency (interaction and validation cost), and (c) user usability (effort for producing compliant consent artifacts)?

4.1. Case Study Scenario: Clinical Trial Consent Management

For the Clinical Trial Scenario, a pilot case study was conducted, which was implemented on Hugging Face Spaces using Gradio 5.45.0 as the Software Development Kit. The large language models used were (a) *HuggingFaceTB/SmolLM2-1.7B-Instruct* (i.e., *informed Consent Generation process*), (b) *openai/whisper-tiny.en* (i.e., *accessibility feature*), and (c) *deepseek-vl2-small* (i.e., *XAI mechanism*) for regulation compliance. Figure 5 presents the workflow. These LLMs were implemented as proof-of-concept demonstrations to validate the framework's feasibility. It is proposed that the production-ready solution applies fine-tuned models specifically trained on multiple clinical trial data sets. Additionally, a production deployment requires either on-premises infrastructure or private cloud services, maintaining patient data within hospital network boundaries.



Figure 5. CONSENT management process.

Subsequently, in Figure 6, the Framework Technology associated with the Case Study Design is stated. Additionally, regardless of the proposed type of case study presented, it is noted that the data controller's role is multifaceted and can be introduced to many different groups, such as (a) software developers—who implement and integrate the consent management system into existing infrastructure, (b) legal and compliance officers—who verify regulatory adherence, (c) data protection officers—who oversee consent lifecycle management, (d) auditors and regulatory inspectors—who verify the organization's consent management practices during GDPR audits, and (e) business analysts and product managers—who evaluate cost-effectiveness. Table 12 further outlines the defined data processing roles under GDPR Article 28 for the clinical case scenario.

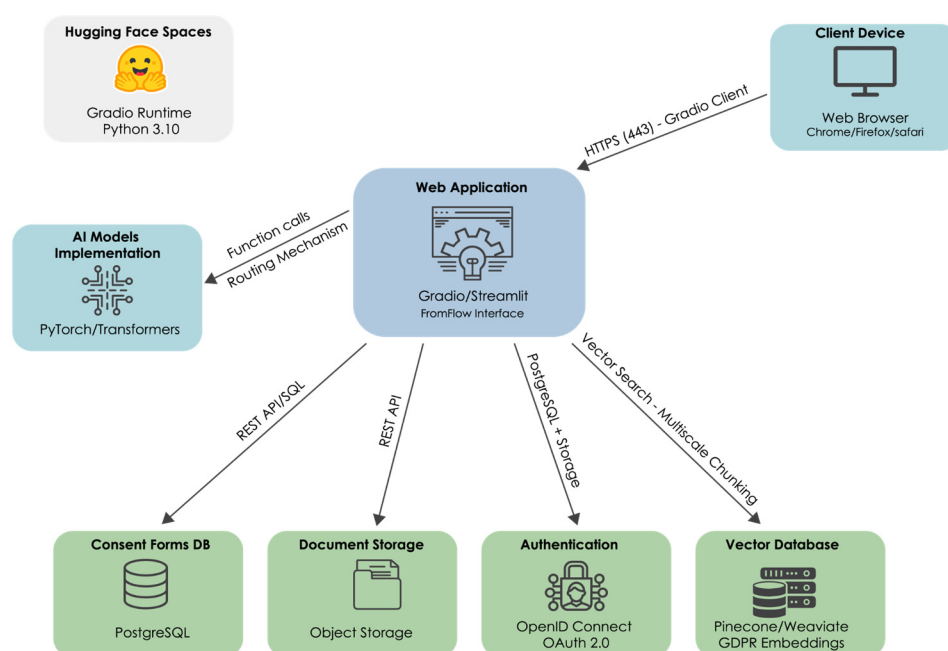


Figure 6. CONSENT architecture technology.

Table 12. Data processing roles and responsibilities.

Role	Entity	Responsibilities	GDPR Basis
Data Controller	Clinical trial sponsor/hospital	Determines purposes and means of consent processing; owns patient consent data; ensures legal basis for processing (i.e., Art. 6)	Article 24, 26
Data Processor	AI system operator (institutional IT department or contracted vendor)	Processes consent forms per controller instructions; maintains technical safeguards; implements data subject rights	Article 28
Sub-processor	Cloud infrastructure provider/premises infrastructure	Provides computational resources under data processing agreements; ensures EU data residency	Article 28(2)–(4)

4.2. Methodology for Evaluation

To evaluate the CONSENT framework (RQ1), a total of 20 individuals with varying degrees of expertise were enlisted to conduct a human-centered evaluation using a questionnaire scoring with a Likert scale from 1 to 5. The participant demographics are

presented as Legal Experts ($n = 5$) including one GDPR specialist (university researcher, Greece), one Data Protection Officer (DPO, multinational corporation), three attorneys specializing in digital privacy law, Technical Experts ($n = 5$) including five software developers with 3+ year experience in web applications and data systems, Domain Experts ($n = 2$) including two healthcare professionals with experience in clinical research consent processes and Academic Participants ($n = 8$) including graduate and postgraduate students in software engineering with coursework in data privacy and AI systems. To establish a verified ground truth for compliance evaluation, all 30 consent forms from the clinical trial case study have been cross-revised. The DPO assessed each form against the structured requirement checklist, manually assigning category-level compliance labels and holistic scores based on legal sufficiency. These expert annotations serve as the validated benchmark against which automated scoring accuracy is measured. This expert-annotated set was used as a benchmark to gauge accuracy. Analytically, based on demographics, candidates were divided into two main categories. Reasonably, software developers and academic participants are recognized as *Engineers*, while Legal Experts are grouped with Domain Experts as *Legal Advisors*. To assess the effectiveness of generating high-quality consent requirements (RQ2), a multi-round prompting approach was employed by the authors through iteratively refining the descriptions and then feeding them to the CONSENT until a valid percentage of 80% GDPR score was ensured. Each author generated a total of 10 informed consent forms, making the total sum of 30 ($n = 30$). When multiple prompting rounds were needed, the report included the total time and token counts across all iterations. To address the small experimental sample size, a test case was further implemented by accessing the system via an API, thus generating 250 consent forms. The system employed a multi-stage pipeline involving random template selection, dynamic field extraction, LLM-powered data generation via Gemini, and API invocation to the CONSENT service. Performance metrics captured include generation time, API latency, and throughput. How fast the system reacts to user input is characterized as (a) *the system's responsiveness*, and the candidate metrics are the total duration in seconds for generation (i.e., *total time for the task from start to finish*), the token generation rate, and the API latency stemming from the test case method. (b) *Systems' efficiency* captures the interaction cost and computational cost based on the time it takes for the system to generate the output, number of prompts sent, number of tokens generated (computational/processing cost), token generation rate (efficiency per token), total duration for compliance scoring (overhead cost) and total blockchain transaction time (extra processing cost). (c) Usability focuses on human effort needed to produce a compliant consent artifact and relies on the number of prompts sent, total duration in seconds for generation, total compliance score, and total understanding score.

5. Results

This section presents the empirical evaluation of the CONSENT framework. Results are organized into two functional outputs. Initially, the controlled case study addressing the research questions is presented. A preliminary systematic testing of the implemented system is also reported, by conducting an ablation study, followed by the results of stress testing.

5.1. RQ1: Quality of Generated Consent Forms

Differences emerged between the two groups regarding their responses, with noticeably lower ratings given by Legal Advisors, which was expected, given their stricter expectations and domain expertise. Notably, the largest divergence was observed in the fluency metric. Although distinct ratings were observed, this difference highlighted the need of certain users, mainly developers, for advice on legal aspects and document

standardization. Generally, it was noted that the scores relating to the forms generated remained at prominent levels, with almost all sectors receiving scores around or above 4, thus demonstrating a perceived reasonable quality. Figure 7 summarizes the quantitative evaluation of the generated consent forms across the five key quality metrics.

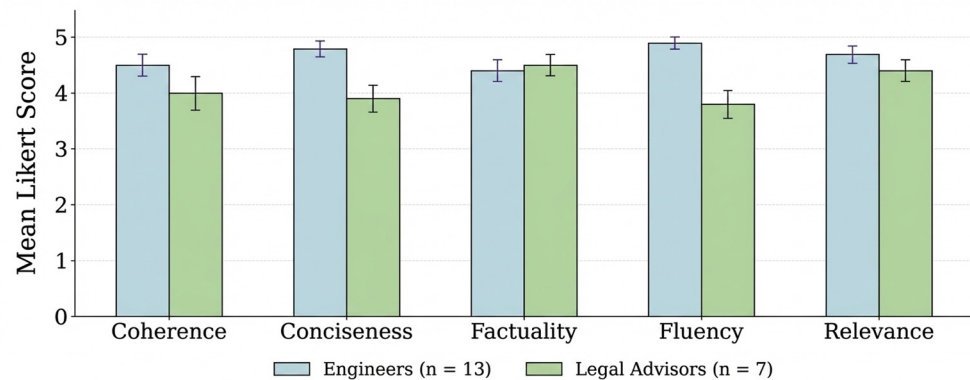


Figure 7. Dimension-wise quality assessment of automatically generated consent forms by expert evaluators (n = 20 participants).

5.2. RQ2: System Performance

The performance metrics reported in the following subsections reflect the proof-of-concept deployment on Hugging Face Spaces, using freely available, open-source models with shared computational resources. This infrastructure choice prioritizes accessibility, reproducibility, and zero-cost deployment to enable transparent validation of the architectural design without financial barriers. Nevertheless, this approach may introduce further latency constraints inherent to resource-shared platforms, where computational resources are dynamically allocated across multiple concurrent users, and API rate limits are enforced by the hosting service.

5.2.1. System Responsiveness and Efficiency

In this subsection, the system responsiveness is covered, which characterizes the speed at which the CONSENT framework reacts to data controller inputs during the generation phase. Table 13 showcases three metrics which were derived during the multi-round prompting (n = 30) and test case executions (n = 250), resulting in 280 observations. The metrics described include (a) Total Generation Duration (end-to-end time from initial prompt submission to delivery), (b) Token Generation Rate (throughput efficiency), and (c) API latency. According to (a), there is an observed standard deviation of 50.4 s, which indicates high variability. This fluctuation can be driven by prompt complexity and load conditions. Concerning factor (b), the mean rate of 58.6 tokens/second reflects typical performance. The standard deviation of 13.6 tokens/second highlights possible occasional slowdowns during periods of increased API contention. Given the results of API latency, most values cluster in the 15–45 s range with some outliers pulling the mean up. This presence indicates heterogeneity in performance across template categories. It is important to note that templates face different structural complexities, which may result in latency variance, indicating that performance variability is part of computational demands rather than a system instability.

Table 13. System responsiveness metrics—AI Generation duration and token throughput.

<i>Metric</i>	<i>Mean</i>	<i>Std Dev</i>	<i>Max</i>	<i>Min</i>	<i>Unit</i>
Total Generation Duration	2.8	0.84	4.26	1.74	min
Token Generation Rate	58.6	13.6	65.3	22.4	token/s

The correlative time assignment across the Consent Generation workflow is divided into three primary stages included in the CONSENT workflow. AI Generation dominates the processing timeline, consuming approximately 87.35% of the total end-to-end duration. While the measured mean compliance verification duration of 17 s may appear acceptably fast, this metric reveals an architectural concern. Parallel API execution through async lowers API queuing delays; however, the observed mean is slower than the theoretical minimum, thus suggesting probable overhead which may factor in (a) Network Latency between the CONSENT framework and external API services and (b) Compliance Logic complexity. Lastly, blockchain recording occurs simultaneously in an almost separate workflow, as a way of masking any latencies from the perceived workflow duration. On the other hand, the system's efficiency analysis derived a relatively low standard deviation across the metrics shown in Table 14, indicating more stable system behavior and consistent processing performance. The prompts of the workflow are a subjectively given metric; however, following two simple prompting approaches without much refinement was nearly sufficient—with CONSENT requiring an average of 1.85 prompts for forms that strictly follow regulations with a compliance score of 80%. The token rates and completion times are judged as effective based on standard values, without noticing divergences in the proposed framework.

Table 14. System efficiency metrics across Consent Generation workflow.

<i>Statistic</i>	<i>Prompts</i>	<i>Token</i>	<i>Rate</i>	<i>Comp. (s)</i>	<i>BC (s)</i>
Mean	1.85	3847	47.2	2.4	3.8
Std Dev	0.73	892	8.6	0.6	0.9
Median	2.0	3800	47.0	2.4	3.8

5.2.2. User Effort and Usability

The evaluation of User Effort and Usability indicated an enhanced performance as presented in Table 15. The number of prompts remained low, and although somewhat subjective, it is not expected to vary significantly. User Input Time demonstrated a reasonable spread for human timing on tasks, with moderate Standard Deviation, while User Input Time using speech-to-text showed low variability—indicating consistent performance. One of the highlights affecting user usability was the speech input feature that reduced input time by approximately 54%. Using speech reduces user processing time, even though the content window available to the system increases. However, this does not affect the examining factor, which reflects user interaction rather than system processing. Lastly, the users could consistently achieve high compliance rates based on given data, while there were no significant differences between them, indicating harmonious behavior.

Table 15. User effort and task completion metrics for AI-assisted GDPR Consent Generation.

<i>Metric</i>	<i>Mean</i>	<i>Std Dev</i>	<i>Min</i>	<i>Max</i>	<i>Unit</i>
Number of Prompts	1.85	0.73	1.0	4.0	prompts
User Input Time (UIT)	3.5	0.8	2.5	4.5	min
Speech-To-Text UIT	1.6	0.4	0.7	2.5	min
Compliance Score	84	4.3	71.2	96.8	percentage

5.3. System Validation

5.3.1. Component Level Ablation Analysis

During component level ablation analysis, the multi-dimensional evaluation isolates each component's contribution to overall system performance. Accordingly, comparisons between (a) *Expert Routing Framework* vs. *Single Model*, comparison for form refinement, (b) *RAG and XAI* vs. *non-RAG* and *XAI* for compliance scoring are made. Concerning *Expert Routing Framework* in contrast to *Single Model* development (i.e., *removing the routing mechanism and refinement subsystem*) precision, recall, and F1 scores are calculated. A Measure of Lexical Textual Diversity (i.e., *MLTD*) for linguistic aspects is further implemented using Python's lexical richness module, which analyzes the forms before and after textual refinement. The baseline comparisons of the quality of generated forms are presented in Table 16, showcasing the importance of the refinement subsystem in all linguistic aspects with major improvements in compliance scoring (i.e., +30%), recall (i.e., +45.2), and MLTD (i.e., +27%).

Table 16. Refinement subsystem ablation results.

<i>Configuration</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>	<i>Compliance Score</i>	<i>MTLD</i>
Single Model	0.68	0.52	0.59	0.64	62.1
<i>Expert Routing Framework</i>	0.72	0.76	0.74	0.83	78.9
Improvement	+5.9%	+45.2	+25%	+30%	+27%

Concerning the RAG-enabled and non-RAG setups, the ablation compares the baseline of CONSENT automated compliance checker against non-refined form GDPR scores generated by a single LLM, following the prompt structure in Figure 8, while the XAI ablation assesses the contribution of explainability features by measuring MTLD and quantifying legal citation accuracy through pattern matching of GDPR article references. Results are further presented in Table 17, validating the subsystem's effectiveness across lexical diversity (i.e., +25.4%) and most noticeably in citation establishment (i.e., +72.7%). It is further noted that the baseline model calculated higher average compliance scores (i.e., 35.5% divergence), indicating lenient standards for GDPR compliance. Based on these results, it is assumed that

The standalone LLM relies on learned patterns and common phrases from its training data. It may "assume" compliance if certain keywords or phrases appear, even if the content does not fully satisfy regulatory requirements. This can artificially inflate the compliance score due to overgeneralization. On the other hand, RAG-based architectures retrieve and ground content against up-to-date regulatory sources, and XAI mechanisms evaluate the reasoning behind compliance claims. This stricter validation catches gaps or ambiguities, resulting in lower but more accurate compliance scores.

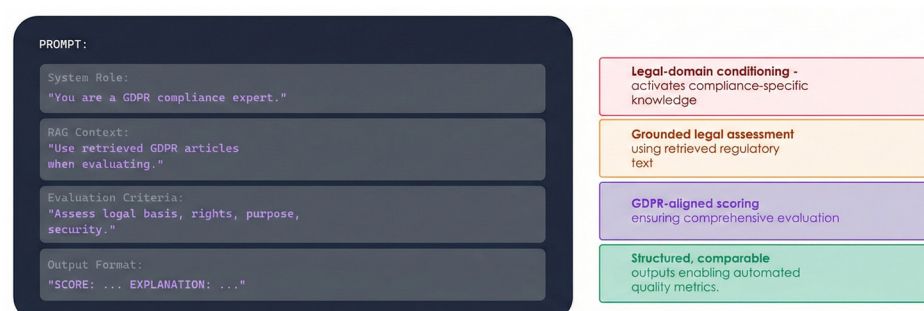


Figure 8. RAG-enabled GDPR compliance evaluation prompt structure (aligned with GDPR Articles 5, 6, 12–22, 32).

Table 17. RAG and explainability-oriented ablation analysis.

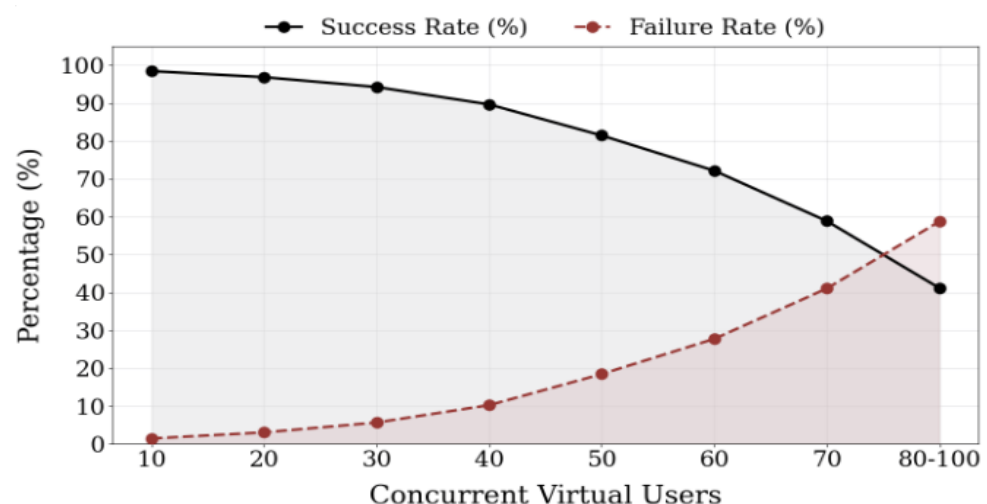
<i>Configuration</i>	<i>Average Compliance Score</i>	<i>Citations per Explanation</i>	<i>MTLD</i>
Non RAG and XAI	0.84	0.44	0.59
RAG and XAI	0.62	0.76	0.74
Diversity/Improvement	−35.5%	+72.7%	+25.4%

5.3.2. Performance Testing

A progressive stress testing methodology was employed to evaluate the performance characteristics of CONSENT under increasing load conditions—using the k6 performance testing tool, integrated with Python’s `gradio_client` library for API interaction. The test protocol consisted of eight distinct phases, with virtual user counts incrementing from 10 virtual users (VUs) to 100. Response-time metrics, success rates, and failure patterns were monitored and are presented in Table 18. In Figure 9, request success rates (solid line) and failure rates (dashed line) demonstrate an inverse relationship as a function of concurrent virtual users, with degradation beginning at Stage 4 with 40 VUs. Complete service saturation is observed at the last stage (i.e., Stage 8), concerning 85+ VUs.

Table 18. Quantitative performance metrics by load stage.

<i>Stage</i>	<i>VUs</i>	<i>Requests</i>	<i>Mean RT (s)</i>	<i>P95 RT (s)</i>	<i>Success Rate</i>	<i>RPS</i>	<i>Status</i>
1	10	324	89.2	156.3	98.5%	1.8	Optimal
2	20	387	112.4	201.7	96.9%	2.2	Optimal
3	30	421	145.8	267.2	94.3%	2.8	Acceptable
4	40	445	198.5	354.8	89.7%	3.2	Acceptable
5	50	401	267.3	478.9	81.5%	3.5	Degraded
6	60	378	342.6	589.1	72.2%	3.8	Degraded
7	70	321	445.7	687.3	58.9%	3.4	Critical
8	80–100	266	578.2	824.6	41.2%	2.6	Saturated

**Figure 9.** Request completion rates under progressive load: success rates (black solid line) and failure rates (red dashed line).

Lastly, Figure 10 depicts response-time progression under increasing load, showing mean response time (solid line) and 95th percentile (dashed line) across progressive load stages. The percentile represents the response time below which 95% of all requests are completed (e.g., *P95 of 354.8 s at 40 VUs means 19 out of 20 requests finished within 354.8 s*). Horizontal reference lines indicate the service level agreement (SLA) thresholds of (a)

acceptable performance (i.e., 300 s)—representing the target average response time, and (b) maximum tolerance (i.e., 600 s)—representing the hard timeout limit beyond which requests are considered failed. Consequently, Table 19 categorizes CONSENT performance into four operational statuses.

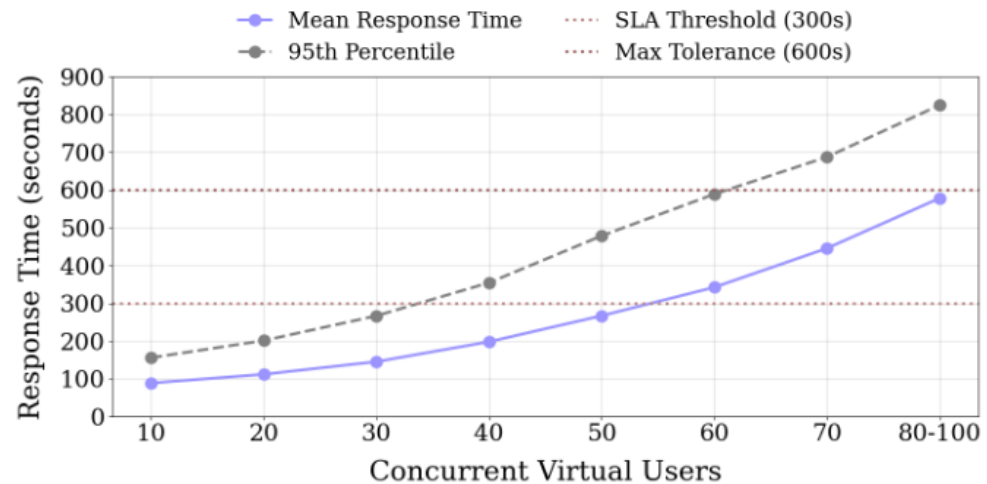


Figure 10. Response-time behavior under progressive load.

Table 19. CONSENT performance classification.

VU Range	Characteristics	Recommendation
10–30	Mean RT < 150 s, Success > 94%, P95 < 270 s	Safe for production deployment
30–40	Mean RT < 200 s, Success > 89%, P95 < 360 s	Operational with monitoring
40–70	Mean RT 200–450 s, Success 58–82%, P95 < 690 s	Load balancing required
70+	Mean RT > 450 s, Success < 59%, P95 > 690 s	System capacity exceeded

6. Discussion

6.1. Results and Discussion

The fusion of AI and blockchain technologies has an immense potential for consent management. Concerning modern technologies, current solutions deploy custom implementations of AI and LLM algorithms in Python scripts. Blockchain technologies mostly utilize Ethereum and Hyperledger, mainly targeting healthcare data management, thus not providing a global solution for consent management. The proposed architecture is based on routing specific AI models, using user-centered approaches on generating consent forms for multiple fields, as well as blockchain storage and smart contracts to ensure data integrity and regulatory compliance.

The quantitative results of the study suggest that the generated consent forms are of high quality, with conciseness and fluency scores indicating clear, readable, and well-structured prompts. Coherence scores further support the fact that the content is logically organized. Factuality, while strong, shows that minor inaccuracies occasionally occur, highlighting the need for review or refinement in automated generation. The architectural design decisions are also evaluated through quantitative performance metrics, which showcase better performance. Taken together, these findings demonstrate that the generation system produces reliable, understandable forms, but attention to factual accuracy and content relevance remains important.

6.2. Security and Abuse Case

In order to combat the increased number and severity of security risks, the current section provides both security and abuse case analysis. Based on the proposed architecture and the technologies incorporated, the most common cases [40] are considered the following:

- *Prompt Injection Attacks* to inject malicious instructions into user input to manipulate LLMs. The proposed architecture employs structured prompting with role separation (i.e., *system*, *developer*, *user*) to prevent overriding compliance constraints. Furthermore, input validation filters detect suspicious tokens (i.e., “*ignore previous instructions*” or “*bypass compliance*”). Furthermore, the compliance evaluation module operates independently of user prompts to ensure that regulatory checks cannot be disabled through prompt manipulation.
- *Model Output Verification* to ensure that all results are accurate, reliable, and aligned with expectations, and exclude any possible hallucinations. The proposed architecture separates generation from verification. Thus, all generated consent text is not assumed to be compliant; instead, it is independently evaluated by the RAG-based compliance module, which grounds decisions in retrieved regulatory text. Last but not least, in any case where discrepancies between generated content and retrieved legal sources exceed a predefined threshold, the system flags the output for mandatory human legal review.
- *Retrieval Poisoning Defense* to inject malicious data into the knowledge base. The RAG subsystem utilizes regulatory corpora, curated from trusted legal sources and versioned with cryptographic hashes. All updates are controlled through an administrative approval workflow and logged for auditability. Moreover, all retrieved documents are cross-validated against regulation metadata (i.e., *article identifiers and publication dates*) to reduce the risk of adversarial or outdated content influencing compliance evaluation.
- *Auditability of AI decisions*, including prompts, retrieved documents, compliance scores, explanations, and model versions. The proposed architecture logs all prementioned with timestamps and cryptographic hashes, stores these off-chain in encrypted storage, anchors on-chain via hashes to enable tamper-evident audits, and preserves confidentiality.
- *Safe-Fail and Human Oversight Mechanism* as a conservative fail-safe design. The proposed architecture utilizes a compliance confidence score based on the detection of contradictory regulatory signals or adversarial patterns. In any detection case, the autonomous approval is disabled, in accordance with GDPR Article 22, where legally binding consent requires meaningful human oversight.
- *Red-Teaming Considerations*. While large-scale adversarial testing is beyond the scope of this study, structured internal threat modeling and manual adversarial prompt testing (e.g., instruction overrides, misleading legal queries) was conducted to evaluate system robustness. These exercises informed the inclusion of input validation, independent compliance verification, and mandatory human review triggers.

6.3. Future Work

On the implications of the current study, the proposed architecture may provide a valuable foundation for further research and development. Concerning researchers, they may integrate formal ontologies and semantic models, which could improve both performance and interpretability beyond current approaches. Comprehensibility enhancement through improving UI/UX design, by integrating visual and interactive elements, is also recommended, followed up with addressing cross-domain applicability, ethical considerations, and upcoming security risks such as prompt injection. Applicable legislation relating to national DPA can also be included in the RAG system. Furthermore, the infusion of

AI and blockchain technologies may lead to new research paths, including consent validation via AI or generative preference analysis on attitude towards consent requests. The proposed architecture includes ontologies and semantics that may be further expanded in theoretical, technical, and methodological dimensions. Integrating semantic technologies such as ontologies, knowledge graphs, and rule-based reasoning is a crucial step in improving the system's capacity to comprehend regulatory ideas, identify discrepancies, and generalize outside of the healthcare industry. Advanced multimodal consent interactions, such as adaptive interfaces, visual explanations of data flows, and accessibility-oriented interaction models that serve a variety of user demographics, require more investigation. Researchers should also look at methods for ongoing regulatory adaptation, such as monitoring legal drift, dynamically updating RAG knowledge libraries, and automatically ingesting new legal texts. Lastly, to guarantee resilience, security, and reliability in legally delicate settings, thorough adversarial testing, reworking LLM components, and formal verification of blockchain operations are still crucial. Further exploration of security or abuse test cases is crucial to ensure architecture stability, validity, and performance.

Concerning practitioners, they could emphasize operational concerns, including regulatory compliance, translation checking, and customization of the proposed architecture using domain- and organization-specific data. Furthermore, organizations expecting peak loads exceeding 40 users require horizontal scaling through load balancing, investing in private server hosting rather than web-based platforms. Finally, incorporating self-reflective models such as user categorization, context awareness, and feedback-driven adaptation can further support personalized legal assistance. The proposed platform architecture may provide a basis for innovative smart consent-management applications that effectively and ethically manage consent on regulatory or policy requirements. Future research for practitioners should focus on the architecture's operationalization, integration, and long-term maintainability inside organizational ecosystems. This entails creating adaptable modules that conform to industry-specific compliance standards, data governance regulations, and current ERP/CRM infrastructures. Improved dashboards for auditors, data protection officers, and administrators would be beneficial to practitioners because they would allow for automatic renewal workflows, transparent XAI-based explanations for compliance decisions, and real-time monitoring of consent validity. Future implementations should also investigate hybrid deployment options that combine cloud-based inference for high-risk or high-complexity jobs with local model hosting to balance cost, privacy, and performance. Establishing operational norms, governance frameworks, and best practices that guarantee ethical use of AI-generated consent content, prevent over-automation, and retain meaningful human oversight is necessary as the usage of automated consent procedures grows.

7. Threats to Validity

In this section, threats to validity that concern the study are presented. The threats to validity that have been identified for this study are based on Runeson and Höst's [41] classification schema that considers construct, internal, external, and reliability validity.

Construct Validity defines how effectively the study measures what it intends to measure. The factors considered affecting the dynamic and safe consent management process—such as blockchain performance, renovation handling, and compliance scoring—have been included in the proposed architecture. However, this does not consist of a claim that all relevant factors or risks have been incorporated into the proposed architecture. Potential omissions include legal concerns based on regulation updates, LLM-specific risks such as prompt injections or hallucinations, as well as the data complexity incorporated into the architecture. Lastly, it is noted that equating time with effort is a theoretical approach that does not apply in all scenarios (i.e., a task could be extremely hard or quite easy,

conceptually, but take more or less time). It is noted that the current study does not include a comparison among existing tools or a control group experiment. Based on the existing variety of available tools, it is expected to identify differences, advantages and limitations compared to each approach or tool. This may incorporate bias to the current research. However, the scope of the study is to present an architectural design of the CONSENT platform. A future comprehensive baseline comparison, where time cost and quality are evaluated, is planned as a necessary next step to our research. Furthermore, biases resulting from keyword weighting and semantic similarity levels may be introduced by automated scoring. This issue is addressed by using multi-expert cross-validation and human-in-the-loop evaluation.

Internal Validity is related to the examination of causal relations since there is no empirical evidence of the proposed platform. The observed outcomes, namely transparency, traceability, revocability, and compliance support, are attributable to the system's design rather than to uncontrolled external factors. To ensure the internal validity, a controlled case study design on the healthcare domain was provided. Furthermore, internal validity was strengthened by employing blockchain-based hashing and revocation logging. Moreover, the Retrieval-Augmented Generation (RAG) subsystem produced compliance scores that were benchmarked against GDPR articles, with the Explainable AI (XAI) subsystem providing transparent reasoning for each compliance decision. Nonetheless, there are some internal threats arising, specifically according to the performance and evaluation metrics, which can be affected by network conditions or computational resources. More broadly, the evaluations and metrics proposed cannot deliver definitive causal proof; they can only increase confidence through controlled repetition. From a production deployment perspective, these durations present scalability challenges under high-concurrency scenarios where multiple consent-form requests happen at the same time.

Concerning *External Validity* and the generalizability of results, different data sets, coming from different application domains or contexts, could cause differentiations in the performance of the proposed architecture. However, a replication of this study with specific large data sets and industrial settings would be valuable in verifying the current architecture. Last but not least, in the current proposal, the rapidly changing technological environment, especially in the AI field of study, may provide alternative and more efficient solutions in the future. Moreover, a security analysis of the AI workflow has been provided in the discussion section.

Regarding *Reliability* and the reproducibility of the study, the proposed architecture has been documented thoroughly, ensuring the safe replication of our study by any interested researcher. Lastly, changes in regulations and advances in technology may make the current study obsolete in the near future. Furthermore, the study relies solely on a clinical trial scenario, thus lacking cross-domain validation, due to time constraints. A future inclusion of multi-domain case studies is planned to further support the generalizability of the current study. Last but not least, it is acknowledged that full-pipeline latency is non-trivial and outlines several optimizations, including caching, model quantization, GPU inference, and asynchronous blockchain writes, which substantially reduce end-to-end processing time in deployment.

8. Conclusions

The proposed hybrid AI-Blockchain Platform for Dynamic and Secure Consent Management in complex data flows underscores many of the essential elements required for a platform to fairly, securely, dynamically, and ethically manage consent. A proper consent management system should try to include clear and informed consent procedures, with functionalities including consent denial, withdrawal, or modification at every stage of data collection and processing. Although existing consent models mostly utilize

blockchain technologies to achieve scalability, effectiveness, and compliance with legal requirements, the proposed platform architecture prioritizes human needs and peculiarities by utilizing AI-driven Consent Generation to achieve ethical consent management. The proposed platform architecture provides a basis for further consent management systems implementation, oriented in achieving efficient, scalable, and secure consent management. The current study provides useful practical and theoretical background for future research. The most prominent future directions may include deploying an architecture-based solution on existing technology in order to conduct in-depth scalability and performance analyses.

Author Contributions: Conceptualization, C.Z., I.Z., S.B.; methodology, C.Z. and S.B.; software, C.Z., I.Z. and S.B.; validation, C.Z., I.Z. and S.B.; formal analysis, C.Z. and S.B.; investigation, C.Z., I.Z. and S.B.; resources, C.Z. and S.B.; data curation, C.Z., I.Z. and S.B.; writing—original draft preparation, C.Z., I.Z. and S.B.; writing—review and editing, C.Z., I.Z. and S.B.; visualization, C.Z.; supervision, S.B.; project administration, S.B.; funding acquisition, S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Greek National Recovery and Resilience Plan “Greece 2.0” with funding from the European Union—NextGenerationEU through the program “SUB1.1. Research Excellence Partnerships”, grant number YP3TA-0559562, project “Dynamic CONSENT management in complex data workflows: techniques and tools”—CONSENT.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the code, algorithms, test cases, and evaluation material can be found at the GitHub repository.

Acknowledgments: The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

AI	Artificial Intelligence
LLM	Large Language Models
MoE	Mix of Experts
RAG	Retrieval-Augmented Generation
XAI	Explainable Artificial Intelligence
GDPR	General Data Protection Regulation
OSI	Open Systems Interconnection
DLTs	Decentralized Ledger Technologies
JSON	JavaScript Object Notation
RPC	Remote Procedure Call
JWT	JavaScript Object Notation Web Token
API	Application Programming Interface
XML	Extensible Markup Language
TTL	Time-To-Live
DPO	Data Protection Officer
CRM	Customer Relationship Management
ERP	Enterprise Resource Planning

References

1. Albrecht, J.P. How the GDPR will change the world. *Eur. Data Prot. Law Rev.* **2016**, *2*, 287.
2. Budin-Ljøsne, I.; Teare, H.; Kaye, J.; Beck, S.; Bentzen, H.B.; Caenazzo, L.; Collett, C.; D'Abramo, F.; Felzmann, H. Dynamic consent: A potential solution to some of the challenges of modern biomedical research. *BMC Med. Ethics* **2017**, *18*, 4.
3. Scott, A.S.; Goldsmith, M.; Teare, H. Wider research applications of dynamic consent. In *IFIP International Summer School on Privacy and Identity Management*; Springer: Berlin/Heidelberg, Germany, 2018.
4. Xu, Z.; Jiao, T.; Wang, Z.; Wen, S.; Chen, S.; Xiang, Y. AC2M: An automated consent management model for blockchain financial services platform. In Proceedings of the 2021 IEEE International Conference on Smart Data Services (SMDS), Chicago, IL, USA, 5–10 September 2021; IEEE: New York, NY, USA, 2021; pp. 33–41.
5. Schramm, J.; Eichinger, T. Towards building GDPR-friendly consent management systems on top of self-sovereign identity ecosystems. In Proceedings of the Open Identity Summit 2024, Porto, Portugal, 20–21 June 2024.
6. Merlec, M.; Lee, Y.; Hong, S.; In, H. A smart contract-based dynamic consent management system for personal data usage under GDPR. *Sensors* **2021**, *21*, 7994.
7. Voigt, P.; Von dem Bussche, A. *The EU General Data Protection Regulation (GDPR): A Practical Guide*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2017.
8. Kakarlapudi, P.; Mahmoud, Q. A systematic review of blockchain for consent management. *Healthcare* **2021**, *9*, 137.
9. Khalid, M.; Ahmed, M. Blockchain-based dynamic consent management systems for enhancing quality of life for people with disabilities. In Proceedings of the 2023 IEEE International Smart Cities Conference (ISC2), Bucharest, Romania, 24–27 September 2023; IEEE: New York, NY, USA, 2023; pp. 1–7.
10. Khalid, M.; Ahmed, M.; Kim, J. Enhancing data protection in dynamic consent management systems: Formalizing privacy and security definitions with differential privacy, decentralization, and zero-knowledge proofs. *Sensors* **2023**, *23*, 7604.
11. Mehmood, F.; Khalid, U.; Karim, S. Consent management system based on user data security and privacy using Hyperledger Fabric blockchain. In Proceedings of the 6th International Conference on Information Technologies and Electrical Engineering, Changde, China, 3–5 November 2023; ACM: New York, NY, USA, 2023; pp. 447–454.
12. Lapwattanaworakul, J.; Srisa-An, C.; Aribarg, T. Blockchain-based auxiliary systems for pseudonymization and consent management. *TEM J.* **2023**, *12*, 2470–2480.
13. Ayappane, B.; Vaidyanathan, R.; Srinivasa, S.; Upadhyaya, S.; Vivek, S. Consent service architecture for policy-based consent management in data trusts. In Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD), Bangalore, India, 4–7 January 2024; ACM: New York, NY, USA, 2024; pp. 155–163.
14. Agbo, C.; Mahmoud, Q. Design and implementation of a blockchain-based e-health consent management framework. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; IEEE: New York, NY, USA, 2020; pp. 812–817.
15. Filho, F.; De Alcantara Batista, B.; Júnior, J.; De Souza, J. Heimdall: Blockchain-based consent management framework. In *2023 International Conference on Information Networking (ICOIN)*; IEEE: New York, NY, USA, 2023; pp. 487–492.
16. Genestier, P.; Zouarhi, S.; Limeux, P.; Excoffier, D.; Prola, A.; Sandon, S.; Temerson, J.M. Blockchain for consent management in the eHealth environment: A nugget for privacy and security challenges. *J. Int. Soc. Telemed. Ehealth* **2017**, *5*, e24.
17. Hu, C.; Li, C.; Zhang, G.; Lei, Z.; Shah, M.; Zhang, Y.; Xing, C.; Jiang, J.; Bao, R. CrowdMed-II: A blockchain-based framework for efficient consent management in health data sharing. *World Wide Web* **2022**, *25*, 1489–1515.
18. Madine, M.M.; Salah, K.; Jayaraman, R.; Yaqoob, I.; Al-Hammadi, Y.; Ellahham, S.; Calyam, P. Fully decentralized multi-party consent management for secure sharing of patient health records. *IEEE Access* **2020**, *8*, 225777–225791.
19. Shah, M.; Li, C.; Sheng, M.; Zhang, Y.; Xing, C. CrowdMed: A blockchain-based approach to consent management for health data sharing. In *Smart Health*; Chen, H., Zeng, D., Yan, X., Xing, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11924, pp. 345–356.
20. Shah, M.; Li, C.; Sheng, M.; Zhang, Y.; Xing, C. Smarter smart contracts: Efficient consent management in health data sharing. In *Web and Big Data*; Wang, X., Zhang, R., Lee, Y.-K., Sun, L., Moon, Y.-S., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12318, pp. 141–155.
21. Tith, D.; Lee, J.S.; Suzuki, H.; Wijesundara, W.M.A.B.; Taira, N.; Obi, T.; Ohshima, N. Patient consent management by a purpose-based consent model for electronic health record based on blockchain technology. *Healthc. Inform. Res.* **2020**, *26*, 265–273.
22. Varshini Naik, K.; Vijaya Murari, T.; Manoj, T. A blockchain-based patient consent management technique for electronic health record sharing. In Proceedings of the 2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Mangalore, India, 1–3 December 2022; IEEE: New York, NY, USA, 2022; pp. 214–219.

23. Hofman, D.; Shannon, C.; McManus, B.; Lemieux, V.; Lam, K.; Assadian, S.; Ng, R. Building trust and protecting privacy: Analyzing evidentiary quality in a blockchain proof-of-concept for health research data consent management. In Proceedings of the 2018 IEEE International Conference on Internet of Things, Halifax, NS, Canada, 27 July–3 August 2018; IEEE: New York, NY, USA, 2018; pp. 1650–1656.
24. Deepthika, K.; Shobana, G.; Reddy, K.; Kumar, B.; Upadhyay, S. Blockchain-integrated deep learning for secure health data sharing and consent management. In Proceedings of the 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI), Coimbatore, India, 28–30 August 2024; pp. 101–106.
25. Rohini, K.R.; Rajakumar, P.S.; Geetha, S. Smart patient consent management model for health information exchange based on blockchain technology. *J. Comput. Sci.* **2024**, *20*, 730–741.
26. Can, O.; Dag, T.; Kantarcioglu, M. A blockchain-based hybrid architecture for auditable consent management. *IEEE Access* **2024**, *12*, 100419–100445.
27. Roman-Martinez, I.; Calvillo-Arbizu, J.; Mayor-Gallego, V.J.; Madinabeitia-Luque, G.; Estepa-Alonso, A.J.; Estepa-Alonso, R.M. Blockchain-based service-oriented architecture for consent management, access control, and auditing. *IEEE Access* **2023**, *11*, 12727–12741.
28. Ascue, O.; Valle, O.; Santisteban, J. BLOCKSAGE: Blockchain-based cloud architecture for sensitive data management in SMEs. *Sustainability* **2025**, *17*, 1352.
29. Garcia, R.; Ramachandran, G.; Dunnett, K.; Jurdak, R.; Ranieri, C.; Krishnamachari, B.; Ueyama, J. A survey of blockchain-based privacy applications: An analysis of consent management and self-sovereign identity approaches. *arXiv* **2024**, arXiv:2411.16404.
30. Pham, A.; Edelson, M.; Nouri, A.; Kuo, T.-T. Distributed management of patient data-sharing informed consents for clinical research. *Comput. Biol. Med.* **2024**, *180*, 108956.
31. Das, S.R.; Jhanjhi, N.Z.; Asirvatham, D.; Rizwan, F.; Javed, D. Securing AI-based healthcare systems using blockchain technology. In *AI Techniques for Securing Medical and Business Practices*; Jhanjhi, N.Z., Ed.; IGI Global: Hershey, PA, USA, 2025; pp. 333–356.
32. Jung, H.; Pfister, F. Blockchain-enabled clinical study consent management. *Technol. Innov. Manag. Rev.* **2020**, *10*, 14–24.
33. Robol, M.; Breaux, T.; Paja, E.; Giorgini, P. Consent verification monitoring. *ACM Trans. Softw. Eng. Methodol.* **2023**, *32*, 2.
34. Bollinger, D.; Kubicek, K.; Cotrini, C.; Basin, D. Automating cookie consent and GDPR violation detection. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022; pp. 2893–2910.
35. Mihaylov, B.; Onea, L.; Hansen, K.M. Architecture-based regulatory compliance argumentation. *J. Syst. Softw.* **2016**, *119*, 1–30.
36. Peyrone, N.; Wichadakul, D. A formal model for blockchain-based consent management in data sharing. *J. Log. Algebr. Methods Program.* **2023**, *134*, 100886.
37. Peyrone, N.; Wichadakul, D. A formal model for integrating consent management into MLOps. *IEEE Access* **2024**, *12*, 142524–142541.
38. Porcelli, L.; Mastroianni, M.; Ficco, M.; Palmieri, F. A user-centered privacy policy management system for automatic consent on cookie banners. *Computers* **2024**, *13*, 43.
39. ISO 32000-1:2008; Document management—Portable document format—Part 1: PDF 1.7. International Organization for Standardization: Geneva, Switzerland, 2008.
40. Park, J.H.; Park, J.H. Blockchain security in cloud computing: Use cases, challenges, and solutions. *Symmetry* **2017**, *9*, 164.
41. Runeson, P.; Höst, M.; Rainer, A.; Regnell, B. *Case Study Research in Software Engineering: Guidelines and Examples*; John Wiley & Sons: Hoboken, NJ, USA, 2012.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.