GRAILS - A Framework for Embedding Ethical Safeguards in Software Applications for Responsible AI

Apurva Kulkarni and Chandrashekar Ramanathan

International Institute of Information Technology Bangalore, Karnataka, India
apurva.kulkarni@iiitb.ac.in, rc@iiitb.ac.in

Abstract

Software systems increasingly mediate critical societal functions involving large-scale use of data. The use of personal and sensitive data introduces ethical and legal concerns, necessitating architectures that support Responsible AI and enforce open access safeguards grounded in ethical principles. These principles are conventionally derived through laws, regulations, codes of conduct, and frameworks. Much of the focus in responsible AI has been on privacy alone. However, there can be other sensitive information in the enterprise and public domain that needs to be handled in a responsible and ethical manner as well. Ethical compliance traditionally has been achieved through systematic manual adherence to guidelines, regulations, and other documentation. However, there is a lack of concrete software architectures, frameworks, or tools that can provide automation to enable compliance directly in software systems. This paper introduces *GRAILS Framework*, a modular, reusable software framework that provides guardrails to adhere to responsible AI principles by decoupling ethical constraints from the functional requirements of software applications. The proposed novel framework addresses the issue of ethical compliance in application software by creating a robust software framework that can be plugged into multiple application software development environments. The framework allows ethical filtering of structured data based on data sensitivity (Low to High), the trust score of the user (Low to High), and granularity of the data (Cell, Row, Column, or Table). The proposed *GRAILS Framework* is implemented and evaluated using Open Government Dataset (OGD), demonstrating high cohesion, low coupling, and long-term maintainability, in line with fundamental software engineering architectural design principles. Notably, the proposed framework is equally effective for ensuring ethical use of personal, enterprise, and public data.

Introduction

Technological advancements have started impacting every aspect of our lives, influencing decisions and shaping user behaviors with the help of vast amounts of personal data. The implications of software on individuals' day-to-day lives and society as a whole cannot be underestimated. Therefore, building software that operates ethically goes beyond technological advancement; it represents a fundamental societal responsibility. Developing ethically compliant software demands that developers embed ethical principles at every stage of the software lifecycle. To ensure ethically aligned critical decision-making, safeguarding data must be a top priority. This requires a solid understanding of ethical principles and the systematic integration of ethical safeguards across all stages of the software development process.

Ethics and Software

According to Chahal (2022), *Ethics* primarily considers principles like fairness, transparency, privacy, inclusiveness, security, and so on to create responsible systems. These principles are cultivated in software engineering through professional codes of conduct, policies, and regulations. Author Aydemir and Dalpiaz (2018) discusses various models, frameworks, and practices to ensure ethically driven software development. In the article, the authors Calluzzo and Cante (2004); Dignum et al. (2018) highlight the importance of incorporating ethics into software that respects user rights, promotes fairness, ensures transparency, and prioritizes privacy. Researchers Boehme-Neßler (2016) propose, in software engineering, privacy serves as a foundational principle for democratic societies by maintaining an individual's impression, like expression, thought, and association, through secure and ethical data management.

Much of the focus in responsible AI has been on data associated with individuals, such as health records, financial details, or employment history. However, there can be other sensitive data related to demographic information (e.g., caste, religion, ethnicity), geopolitical identifiers (e.g., national security data, citizenship status) that need to be handled responsibly and ethically as well. In this context, a data access mechanism that aligns with ethical principles, adjusting privacy levels based on data sensitivity and user trustworthiness, is essential for responsible data access. This research focuses on incorporating ethical constraints into the application software that facilitates ethical data governance for responsible AI through the following research considerations:

- **GRAILS Framework**: Introducing a modular, reusable framework to enforce Responsible AI by decoupling ethical and functional requirements.
- **Broader Ethical Scope**: Extending ethical AI considerations beyond personal data to include demographic and

geopolitical data.

- **Plug-and-Play Design**: Enabling easy integration of ethical safeguards into diverse applications with high cohesion and low coupling.
- **Domain Agnostic**: Configuration-driven approach with domain-specific rules.

Ethical Safeguards

The proposed research refers to *Ethical Safeguards* as the set of data governance principles, processes, and controls that ensure responsible, fair, and privacy-preserving access/share/use of data. Ethical Safeguards can encompass many dimensions of ethics, such as fairness, transparency, privacy, inclusiveness, security, etc. In this paper, the ethical dimension of 'privacy' is used as an exemplar to illustrate and demonstrate the capabilities of the *GRAIL Framework.*

This manuscript is logically structured into three main parts. The first part discusses the current state of the art, highlighting the limitations of incorporating ethical principles into application software. This sets the stage for the motivation behind the proposed research. It is followed by a detailed description of the proposed framework.
The second part presents the implementation of this framework in the context of Open Government Data (OGD). The results are then evaluated for their applicability and relevance from a Responsible AI perspective.
The final part highlights how the proposed framework can be extended to other ethical dimensions beyond privacy. Further, it characterizes the framework from multiple standpoints, including technical, organizational, and the software developer's perspective. The manuscript concludes with a discussion on future directions and key takeaways.

State of the Art

In today's data-centric world, privacy is often considered the most important principle of ethical software development. For example, healthcare software that collects sensitive medical data (Electronic Health Record or EHR) must ensure privacy protections to prevent unauthorized access or misuse. Protecting users' personal information is foundational to building trust and ensuring compliance.

Authors Cavoukian et al. (2009) proposes that the most effective approach to integrating ethical considerations into software is incorporating them from the inception - Privacy by Design. Several design approaches integrate privacy into system design. For instance, the Human-Centered Design (HCD) by Iio et al. (2021) or User-Centered Design (UCD) by Riebel (2023) empathizes with user needs, desires, and limitations. Now, privacy is also an important factor in HCD. Researchers Hussain, Mougouei, and Whittle (2018); Barn, Barn, and Raimondi (2015); Shahin et al. (2022) present Value-Sensitive Design (VSD) that incorporates user values (including privacy, trust, accountability, etc) in the design process, acknowledging that values are core to decision-making.

Several frameworks, such as ISO/IEC 27001/27002 (international standards for Information Security Management Systems - ISMS[1]), Microsoft Security Development Lifecycle (SDL) [2], and the NIST Cybersecurity Framework (CSF) [3], provide guidelines for security and privacy. These approaches of Privacy by Design are suitable for greenfield applications.

However, there is no clear methodology for embedding these practices into applications as a specific software design artifact or retrofitting them into existing systems.

Many frameworks, such as Model-View-Controller (MVC) architecture illustrated by Ruparelia (2010), do not have any provision to incorporate ethical constraints into the software design. Similarly, the traditional Software Development Life Cycle (SDLC) addresses only non-functional requirements such as scalability, performance, and availability, but without any provision for privacy (ethical) considerations.

In such cases, the software often treats ethical requirements as an add-on, typically introducing them in later stages like compliance audits. Any corrections required as a result of non-compliance are either not possible to incorporate or very costly to comply with. Tools like Privacy Impact Assessments (PIAs) showcased by Wright (2012), Data Loss Prevention (DLP) systems presented by Alneyadi, Sithirasenan, and Muthukkumarasamy (2016), and Access Control Models discussed by Samarati and De Vimercati (2000) (e.g., Role-Based Access Control) tend to focus on mitigating privacy risks after the software is developed.

From a practical standpoint, integrating ethical constraints as a retrofit solution into existing software can be approached in the following three ways:

- **Redevelopment**– Redesigning and redeveloping the entire software from scratch.
- **Modification** – Enhancing the existing software to incorporate ethical constraints.
- **Co-existence** – Introducing ethical measures non-intrusively without altering core functionalities.

The first two approaches require systematic changes to workflows, processes, and predefined stages in the life cycle, involving significant investment in resources, time, and expertise, making them less feasible for large-scale systems. When applied to real-world scenarios, several tools serve as retrofit solutions to support the ethical use of data in software systems. Security tools like OWASP[4], ZAP[5], SonarQube[6], and Bandit focus on detecting errors and vulnerabilities, safeguarding user data, and ensuring robust security practices. Privacy tools like Google Privacy Sandbox presented by Google (2024) and arx (2024) are specialized software to protect sensitive user data and ensure compliance with privacy regulations. These tools are often domain-specific and lack a unified, cohesive framework.

[1] https://www.iso.org/standard/27001#lifecycle
[2] https://www.microsoft.com/en-us/securityengineering/sdl/practices
[3] https://www.nist.gov/cyberframework
[4] https://owasp.org/www-project-top-ten/
[5] https://www.zaproxy.org/
[6] https://www.sonarsource.com/products/sonarqube/

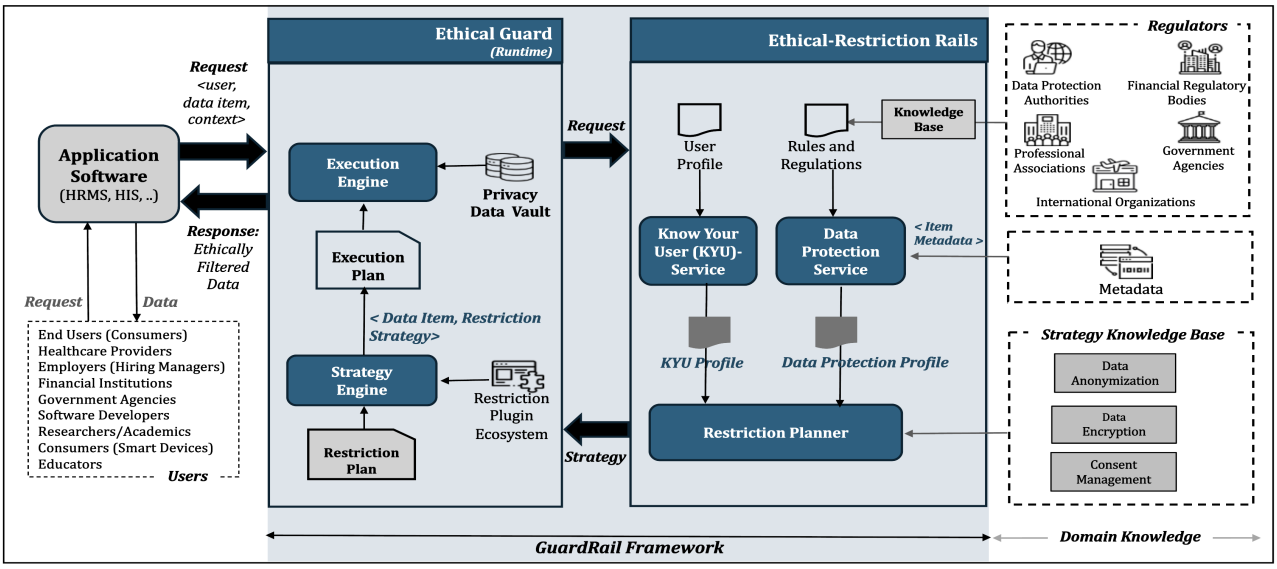

Figure 1: GRAILS Framework for Embedding Ethical Safeguards in Software Applications

State of the art indicates that software developers must decide between Privacy by Design and Retrofitting. Privacy by Design embeds privacy into the system architecture early in the software development lifecycle (SDLC), increasing design complexity. On the other hand, retrofitting integrates privacy into existing systems during later stages, potentially introducing technical debt and performance overhead. This trade-off is often a difficult choice, requiring a solution that balances ethical responsibilities, along with the core functional requirements of the application software.

Motivation

The proposed framework uses a novel approach to ethics that allows the ethical (privacy) constraints to be incorporated into the applications either as part of the design of greenfield applications or retrofitted into the existing applications with minimal changes. The framework provides a mechanism to independently satisfy both the functional requirements as well as non-functional (privacy) requirements in a highly cohesive and loosely coupled environment that facilitates existing software functionalities to remain intact.
The primary research contribution is a framework with the following novel characteristics:

- **Adaptable:** The framework is adaptable and can be utilized to support a wide range of domains
- **Non-Intrusive:** The framework can be incorporated into the existing applications in a manner that does not require extensive modifications.
- **Flexible Integration:** The same application can be configured to behave differently depending upon the desired sensitivity required in the environment in which it is operating.
- **High Cohesion and Low Coupling:** The framework clearly separates core application functionalities from the scaffolding that is required for incorporating ethics into the application. This enables the creation of software that is highly cohesive and loosely coupled.

While the framework is inherently agnostic and adaptable to various ethical dimensions, this paper specifically focuses on demonstrating how the 'privacy' dimension can be integrated into software systems.

GRAILS Framework

Figure 1 depicts the comprehensive processing flow for incorporating ethical dimensions in the software. The **GRAILS** framework consists of two main components: the **Ethical Guard** and the **Ethical-Restriction Rails**. The user request is forwarded to the Ethical Guard, which then consults the Ethical-Restriction Rails to determine the appropriate ethical strategies to be applied to the request. The Ethical Guard is responsible for enforcing these restrictions to facilitate an ethically filtered response.

Ethical Restriction Rails

The Ethical Restriction Rail generates an appropriate restriction strategy in response to the user's request for data access. The diagram highlights three services represented in blue boxes: the Know Your User (KYU) Service, the Data Protection Service, and the Restriction Planner.
User Request: The User Request is modelled as a tuple comprised of $<$*User, Data Request, Context*$>$. The proposed framework enables data access control at four distinct levels of granularity: cell, row, column, and table. The data request is processed by analyzing the context (e.g., purpose of access) and user attributes such as user type (Personal, Internal, External, System/Tool/API), operation type (read, write, modify), information classification level (internal, confidential, public, etc.), data ownership (full, shared, partial), and compliance history (yes, no). These attributes can be adjusted based on the granularity of the ethical requirements

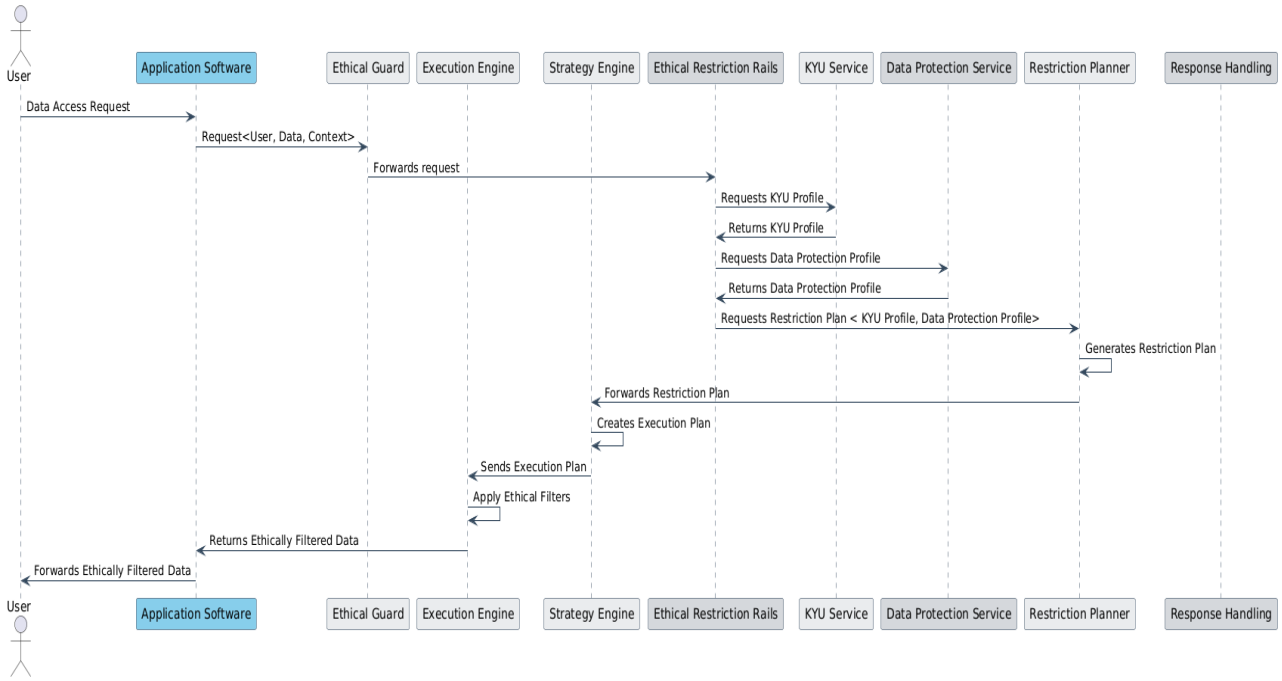

Figure 2: Sequence Diagram: Ethical Filtering Workflow for Data Access Request

and packaged into a User Profile object. This User Profile object is then forwarded to the Know Your User (KYU) Service.

KYU Service: Similar to the Know Your Customer (KYC) used by financial institutions to profile a customer before offering services, the Know Your User (KYU) is being introduced as a mechanism to characterize the user requesting the information. The KYU Service is responsible for generating a KYU Profile for each user based on specific attributes, with the KYU Score serving as the core component of this profile. The KYU Score is modelled as a quantitative measure where the higher the value, the greater the trustworthiness of the user.

The current implementation utilizes fuzzy inference logic for calculating the KYU-Score. The score calculation is performed using a simple approach, such as converting categorical attributes into numeric values and applying a weighted average, or by leveraging more advanced techniques like fuzzy inference logic. Building on the fuzzy logic frameworks discussed by the authors Alhabashneh et al. (2017); Jain and Gupta (2018), this research applies fuzzification to convert user profile attributes into numerical representations. A weighted aggregation of these fuzzy inputs is then employed to compute the KYU score. The KYU score is defuzzified to any desired level of granularity, depending on the degree of restriction required by the policies. It is to be noted that machine learning (ML) or artificial intelligence (AI) models discussed by De Mauro, Sestino, and Bacconi (2022) can be employed as well to produce the KYU score, offering a more dynamic and adaptive scoring mechanism. For example, algorithms such as decision trees, ontologies presented by Joachimiak et al. (2024), support vector machines (SVM), or neural networks could be utilized to improve versatility and adaptability for generating the KYU score.

Metadata: To evaluate the data item as per the User Request tuple, the Data Protection Service captures both file-level and item-level attributes. File-level attributes include data source (private, public), ownership (individual, organizational, shared, public, etc.), and domain (e.g., healthcare, education, consumer, personal). Item-level attributes encompass aspects like Personally Identifiable Information (PII) status (Yes, No), sensitivity (Low, Medium, High), and data type. The metadata of a data item, along with system metadata such as data type, linkage, and constraints, is recorded in a metadata File.

Regulators: Relevant regulations, laws, and rules (such as GDPR, HIPAA, FERPA, CCPA, COPPA, FSMA, etc.) are captured and maintained in a knowledge base, which may utilize ontologies, databases, expert systems, decision support systems, or advanced language models trained on the domain data.

Data Protection Service: Based on metadata parameters and information captured in knowledge base about regulators, the Data Protection Service computes a *Data Protection Profile*. The current implementation leverages a Large Language Model (LLM), drawing on prior work by Fuchs et al. (2024); Cheong et al. (2024), and is specifically fine-tuned using Indian regulatory frameworks as outlined by authors Ghosh et al. (2024), along with domain-specific healthcare data to improve regulatory compliance and contextual relevance. It is to be noted that, similar to the KYU Service, this process can also be performed using simple calculations, advanced fuzzy logic, or more complex AI/ML models to

refine the data protection level based on specific application needs.

Strategy Knowledge Base: It is a structured repository that encapsulates ethically-filtering strategies, including data anonymization, encryption, and consent management. It defines each strategy's implementation details, possible approaches, intended use cases, required parameters, and execution-specific hyperparameters. These strategies can be integrated through the plug-in ecosystem, external services, or as in-built implementations within the system. The proof-of-concept implementation currently supports l-diversity, k-anonymity, and data masking algorithms for anonymization, AES, DES, and hashing techniques for encryption as a built-in solution, and consent management as a service to ensure compliance with ethical regulations.

Restriction Planner: Much like the database query processor that produces a query plan for a given input query, the Restriction Planner is responsible for generating a *Restriction Plan* based on the *KYU Profile* and *Data Protection Profile*. It takes as input the $<$ Data Item, Data Protection Profile$>$ pair generated by the Data Protection Service, along with the KYU score and *Strategy Knowledge Base*. The Restriction Planner functions as a mapping system, identifying the appropriate restriction strategy based on the combination of the KYU Profile and Data Protection Profile. It then generates a Restriction Plan as a tuple for each Data Item $<$***User: KYU_Profile, Data_Protection_Profile, Restriction_Strategies***$>$. This tuple is stored in the rule base and forwarded to the Ethical Guard for execution.

Ethical Guard

The primary function of the guard is to produce an ethically compliant response to the request received on the basis of the Restriction Plan received from the Ethical-Restriction Rails.

Strategy Engine: Strategy Engine is an optimization engine that creates an *Execution Plan* on the basis of the Restriction Plan generated by the rails. This involves analyzing restriction strategies to be applied to the data item. These strategies may involve multiple restriction functionalities (such as masking, encryption, consent verification, and anonymization) applied to the data items.

Execution Engine: This is the core engine of the framework responsible for producing the ethical response to the User Request as per the Execution Plan generated by the Strategy Engine. This engine has access to the Privacy Vault, where data is stored in a secure manner. The engine utilizes built-in implementations, external service calls, or plug-in-based routines to enforce ethical filtering restrictions. Once the specified strategies are applied, the processed and restricted data is generated and delivered as a response to the user request.

Formulation of Ethical Transformations on Data

Figure 2 illustrates the sequence of ethical filtering steps triggered when a user requests data access through the application software. The request is intercepted by the Ethical Guard component within the *GRAILS Framework*, which forwards it to the Ethical-Restriction Rails. At each stage, the components of the framework interact with their corresponding services to obtain the appropriate response. The resulting execution plan comprises a sequence of ethical transformations that must be applied to the data to ensure the generation of an ethically compliant response. To facilitate a deeper understanding of these transformations, this section presents the mathematical formulation of ethical transformations on data.

Formally, let:

- $\mathcal{D}_{in} \in \mathbb{R}^{n \times m}$ represents the original dataset,
- $\mathcal{P}(\mathcal{D}_{in})$ denotes its power set (all possible subsets),
- $G = \{\text{cell}, \text{row}, \text{col}, \text{table}\}$ denotes valid granularities,
- $\lambda \in \Lambda = \{\text{low}, \text{moderate}, \text{high}\}$ represents privacy levels,
- $\nu \in \mathcal{N} = \{\text{low}, \text{moderate}, \text{high}\}$ indicates user trust levels (KYU score),
- $\mathcal{D}_g \in \mathcal{P}(\mathcal{D}_{in})$ is the projection at the granularity g,
- $\mathcal{T}_g : \mathcal{P}(\mathcal{D}_g) \times \Lambda \to \mathcal{P}(\mathcal{D}_g)$ is the transformation function.

Ethically Filtered Data

The output of the *GRAILS Framework* is the ethically filtered data, which is defined as follows:

$$D_{\text{out}} = \bigcup_{g \in G} T_g(D_g, \lambda, \nu) \tag{1}$$

where each transformation T_g operates on a data subset D_g using the *privacy level* λ and the *KYU Score (user trust level)* ν to produce a ethically filtered output.

The anonymization techniques described in Table1 include cell-level transformations $\mathcal{T}_{\text{cell}}(x, \lambda, \nu)$, row-level transformations $\mathcal{T}_{\text{row}}(\mathcal{R}_i, \lambda, \nu)$, column-level transformations $\mathcal{T}_{\text{col}}(\mathcal{C}_j, \lambda, \nu)$, and table-level (subset, combination of rows and columns) transformations $\mathcal{T}_{\text{table}}(\mathcal{S}, \lambda, \nu)$, each defined by specific operations based on the privacy level λ and user trust level ν.

The following function definitions illustrate the transformations to be applied at each granularity level, as referenced in Table 1. For illustrative purposes, each definition includes only a single transformation to demonstrate the overall workflow.

Cell-Level Transformation

$$\mathcal{T}_{\text{cell}}(\mathcal{C}_j, \lambda, \nu) =$$

$$\begin{cases} \text{Cell Suppression} & \text{if } \lambda = \text{High}, \nu = \text{Low} \\ \text{Top/Bottom Coding} & \text{if } \lambda = \text{High}, \nu = \text{Moderate} \\ \text{Noise Injection} & \text{if } \lambda = \text{High}, \nu = \text{High} \\ \text{Microaggregation} & \text{if } \lambda = \text{Moderate}, \nu = \text{Low} \\ \text{Generalization} & \text{if } \lambda = \text{Moderate}, \nu = \text{Moderate} \\ \text{No Transformation} & \text{if } \lambda = \text{Moderate}, \nu = \text{High} \\ \text{Noise Injection} & \text{if } \lambda = \text{Low}, \nu = \text{Low} \\ \text{No Transformation} & \text{if } \lambda = \text{Low}, \nu \in \{\text{Moderate}, \text{High}\} \end{cases}$$

Data Level	Sensitivity	Low KYU Score	Moderate KYU Score	High KYU Score
Cell	High	Cell Suppression, Differential Privacy Column, Full Masking	Top/Bottom Coding, Microaggregation, Partial Masking	Noise Injection
	Moderate	Microaggregation	Generalization, Partial Masking	No Transformation
	Low	Noise Injection	No Transformation	No Transformation
Row	High	Full Masking	Microaggregation, Partial Masking	Microaggregation
	Moderate	Generalization	Microaggregation, Partial Masking	No Transformation
	Low	Microaggregation	No Transformation	No Transformation
Column	High	Generalization, Top/Bottom Coding, Full Masking	Cell Suppression, Partial Masking	Noise Injection
	Moderate	Binning	Binning, Partial Masking	No Transformation
	Low	Generalization	No Transformation	No Transformation
Table	High	Cell Suppression, Differential Privacy, Full Masking	Microaggregation, Partial Masking	Microaggregation
	Moderate	Generalization	Microaggregation, Partial Masking	No Transformation
	Low	Generalization	No Transformation	No Transformation

Table 1: Ethical Filtering Strategies by Data Granularity, Sensitivity, and KYU Score (trust indicator)

Row-Level Transformation

$$\mathcal{T}_{\text{row}}(\mathcal{C}_j, \lambda, \nu) =$$

$$\begin{cases} \text{Full Masking} & \text{if } \lambda = \text{High}, \nu = \text{Low} \\ \text{Microaggregation} & \text{if } \lambda = \text{High}, \nu = \text{Moderate} \\ \text{Microaggregation} & \text{if } \lambda = \text{High}, \nu = \text{High} \\ \text{Generalization} & \text{if } \lambda = \text{Moderate}, \nu = \text{Low} \\ \text{Microaggregation} & \text{if } \lambda = \text{Moderate}, \nu = \text{Moderate} \\ \text{No Transformation} & \text{if } \lambda = \text{Moderate}, \nu = \text{High} \\ \text{Microaggregation} & \text{if } \lambda = \text{Low}, \nu = \text{Low} \\ \text{No Transformation} & \text{if } \lambda = \text{Low}, \nu \in \{\text{Moderate}, \text{High}\} \end{cases}$$

Column-Level Transformation

$$\mathcal{T}_{\text{col}}(\mathcal{C}_j, \lambda, \nu) =$$

$$\begin{cases} \text{Generalization} & \text{if } \lambda = \text{High}, \nu = \text{Low} \\ \text{Cell Suppression} & \text{if } \lambda = \text{High}, \nu = \text{Moderate} \\ \text{Noise Injection} & \text{if } \lambda = \text{High}, \nu = \text{High} \\ \text{Binning} & \text{if } \lambda = \text{Moderate}, \nu = \text{Low} \\ \text{Binning} & \text{if } \lambda = \text{Moderate}, \nu = \text{Moderate} \\ \text{No Transformation} & \text{if } \lambda = \text{Moderate}, \nu = \text{High} \\ \text{Generalization} & \text{if } \lambda = \text{Low}, \nu = \text{Low} \\ \text{No Transformation} & \text{if } \lambda = \text{Low}, \nu \in \{\text{Moderate}, \text{High}\} \end{cases}$$

Table-Level Transformation

$$\mathcal{T}_{\text{table}}(\mathcal{C}_j, \lambda, \nu) =$$

$$\begin{cases} \text{Cell Suppression} & \text{if } \lambda = \text{High}, \nu = \text{Low} \\ \text{Microaggregation} & \text{if } \lambda = \text{High}, \nu = \text{Moderate} \\ \text{Microaggregation} & \text{if } \lambda = \text{High}, \nu = \text{High} \\ \text{Generalization} & \text{if } \lambda = \text{Moderate}, \nu = \text{Low} \\ \text{Microaggregation} & \text{if } \lambda = \text{Moderate}, \nu = \text{Moderate} \\ \text{No Transformation} & \text{if } \lambda = \text{Moderate}, \nu = \text{High} \\ \text{Generalization} & \text{if } \lambda = \text{Low}, \nu = \text{Low} \\ \text{No Transformation} & \text{if } \lambda = \text{Low}, \nu \in \{\text{Moderate}, \text{High}\} \end{cases}$$

This approach ensures that ethically filtering transformations are tailored to the sensitivity of the data and the trust level of the user, enforcing responsible access through a structured and adaptable framework.

Case Study - Open Government Data

In Open Government Data (OGD), information is typically aggregated at administrative levels such as villages, taluks, districts, or states. While this aggregation serves as a natural guard against privacy vulnerability, it does not fully safeguard against exposing sensitive attributes, such as demographic details (e.g., caste, religion, ethnicity) or geopolitical identifiers (e.g., national security-related data). Applying ethical filtering on such data becomes essential in such contexts. Given that data requests may originate from diverse user groups—researchers, academicians, government authorities, or individuals. A one-size-fits-all anonymization approach could restrict legitimate data use. Furthermore, OGD spans multiple domains like healthcare, agriculture, finance, and infrastructure, each governed by distinct policies and regulatory sensitivities, which directly influence the level of protection required.

In this work, we focus on the **Karnataka At a Glance dataset**[7], an open dataset provided by the Government of Karnataka. It includes data from 30 districts across 17 domains (e.g., healthcare, agriculture, economics), with over 1,200 attributes. Many of these are sensitive, including census and survey data disaggregated by caste, gender, region, minority status, disease prevalence, and financial indicators like loan distribution. The current implementation considers a diverse set of users, each evaluated using attributes such as Verified ID, Organizational Affiliation, and Purpose Clarity. Based on these, a KYU (Know Your User) score is assigned (Low, Moderate, or High) reflecting the user's trustworthiness. The Figure 3 depicts the user interface to capture the data request. For instance, users from government

[7] https://kgis.ksrsac.in/kag/

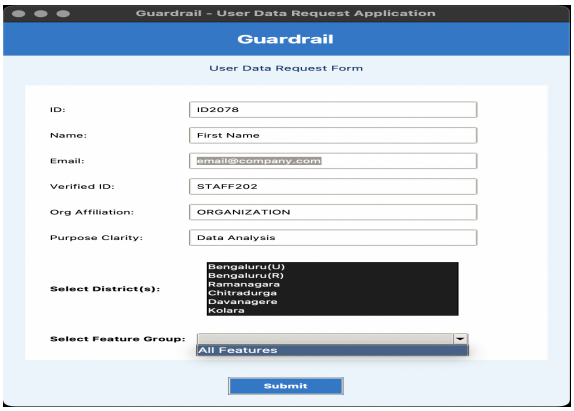

Figure 3: GRAILS: User Data Request Form

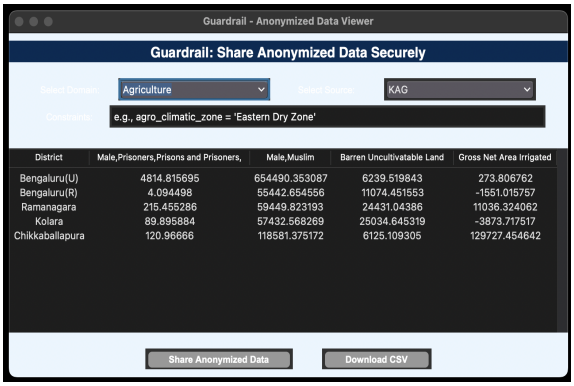

Figure 4: GRAILS: Applying Ethical Safeguards for Ethical Data Filtering

or academic institutions with clear purposes receive a Moderate score, while unverified users with vague intentions are assigned a Low score. This scoring guides ethical filtering during data access.

Figure 4 depicts the anonymized output generated in response to a user request. For the purpose of illustration, let's assume the user is legitimate, as they have a valid email ID, organizational affiliation, and a verified ID. Based on these attributes, the user's trust score is determined to be Moderate. The requested data items— 'Male Prisoners', 'Male Caste_XXX', 'Barren Uncultivatable Land', and 'Gross Net Area Irrigated'—are assigned sensitivity levels of High, High, Low, and Low, respectively. Accordingly, the transformations defined in Table 1 are applied to enforce privacy protection. It is to be noted that the applied transformations and the eventual filtering will vary depending on the configurations of the Ethical Restriction Rails as per the domain requirements.

Results and Observations

To quantitatively characterize the ethical filtering induced by ethical guardrails on sensitive datasets, we define the *Filtering Score (FS)* as a normalized, aggregated measure of deviation between the original dataset D_{in} and its anonymized counterpart D_{out}. Formally, let $D_{\text{in}}, D_{\text{out}} \in \mathbb{R}^{n \times m}$ denote two aligned data matrices of equal shape. The score is computed as:

$$\text{FS}(D_{\text{in}}, D_{\text{out}}) = \frac{1}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} \delta \left(D_{\text{in}}[i,j], D_{\text{out}}[i,j] \right)$$

The function δ serves as a *domain-sensitive transformation penalty*, mapping each pair of cell values to a unit interval $[0, 1]$, where:

- $\delta = 0$ reflects identity transformation (no filtering applied),
- $\delta = 1$ indicates complete obfuscation, such as suppression, masking,
- Intermediate values capture selective filtering such as generalization, range encoding, noise perturbation, or semantic abstraction.

Rather than a fixed rule set, δ is envisioned as a *context-aware semantic distance function*, sensitive to: the data type (numeric, categorical, textual), the transformation logic (e.g., binning vs. suppression), and the informational entropy retained post-anonymization. In numeric domains, δ may incorporate *relative magnitude deviations*, while for categorical or textual domains, it may rely on *embedding similarity*, *hierarchical generalization levels*, or *privacy taxonomies* (e.g., suppression $>$ generalization $>$ perturbation).

This scoring mechanism allows us to benchmark the extent of privacy enforcement across multiple granularity levels (cell, column, row, table), and align these scores with data sensitivity level and KYU-score. Figure 5 shows the graded ethical filtering supported by the proposed framework. High Filtering Score values in high-sensitivity, low-KYU zones confirm the effectiveness of protective transformations, while low Filtering Score values in low-sensitivity, high-KYU regions indicate justified data transparency, consistent with the GRAILS Framework's ethical objectives.

Figure 5 offers a comprehensive visualization of anonymization effectiveness across varying sensitivity levels (vertical axis) and KYU scores (horizontal axis), segmented by the granularity of anonymization techniques (cell, row, column, table). Each subplot reflects a unique combination of Sensitivity–KYU pairing, arranged to form a meaningful grid. The darker-shaded subplots, concentrated in the lower-left triangle, represent ethically critical scenarios, where sensitivity is high, and user access (KYU), trustworthiness is limited. Here, the visual presence of red asterisks (*) denotes 'Maximally Filtered (Ethically Compliant)' states, demonstrating that the system is effectively prioritizing high-risk cases through comprehensive anonymization strategies like full masking, microaggregation, and cell suppression.

Conversely, the upper-right region—indicating Low Sensitivity and High KYU (High Trust), remains light or unshaded, with a dominance of green triangles (^), interpreted

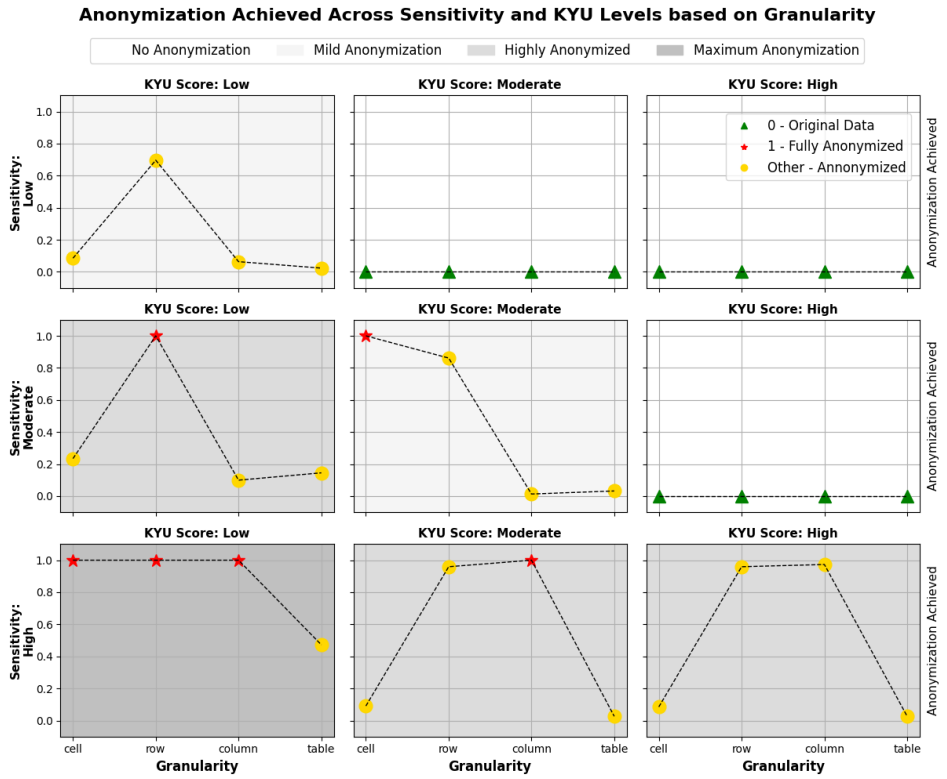

Figure 5: Ethical Filtering Strategies Across Data Granularity, Sensitivity Levels, and User Trust (KYU) Scores

as 'Unfiltered (Raw Data).' While this may initially raise concerns, it actually aligns with the design logic: less sensitive data accessed by highly authorized users requires minimal transformation, preserving maximum utility without compromising ethical standards.

The middle triangle region with moderate sensitivity and KYU combinations shows a prevalence of gold circles (o), indicating 'Selectively Filtered' outcomes. This transitional zone illustrates the *GRAILS Framework's* capacity to apply risk-adjusted filtering, ensuring that data is neither over-filtered nor under-filtered, but instead matched to context-aware policies.

Importantly, the spatial and visual arrangement of markers that are combined with quadrant shading offers a layered ethical narrative: the darker lower triangle reflects high-risk zones where filtering is visibly enforced, while the lighter upper triangle demonstrates the trust in low-risk contexts. Such a distribution affirms that the *GRAILS Framework* not only enforces filtering where needed but also preserves analytical fidelity where ethically permissible. This balance between ethical safeguards and responsible data utility underscores the strength of the proposed system in advancing Responsible AI governance.

Software Implementation

The *GRAILS Framework* is implemented in Java as a proof of concept in the Open Government Data. As depicted in Figure 6, this framework emphasizes the adaptability of the proposed approach as a design artifact in greenfield implementation or as a retrofit solution in an existing application. The software framework utilizes several design patterns (like the Facade design pattern, Decorator design pattern, the Strategy Pattern, etc.) discussed by Gamma (1995) to incorporate extensibility, reusability, and loose coupling.

The framework is designed to apply privacy-preserving techniques to tabular data stored in an SQL database. The design patterns allow application software to interact with the *GRAILS Framework* without directly dealing with underlying privacy mechanisms such as data anonymization, encryption, and consent management, ensuring modularity and ease of use.

Benefits of GRAILS Framework

The proposed framework is influenced by the idea of incorporating ethics as an independent component that can be considered either at the design level or as a retrofitted solution. Conventional solutions like Role-based Access Control (RBAC) and attribute-based Access Control (ABAC) are widely used in many applications for data governance and access control. However, adapting to dynamic elements like ethical considerations, compliance enforcement, and fine-grained ethical constraints can be challenging or even impossible using traditional approaches. We provide below the different benefits of the proposed framework from multiple standpoints:

- **Technical Standpoint:** In data-centric applications, enforcing dynamic access control and real-time policy enforcement may result in a system that is tightly-coupled and rigid. The task gets even more complex across large-scale systems with millions of users having different trust levels and data access requirements. The modular design of the proposed framework makes it adaptable, scalable, and easy to integrate with any software application, irrespective of its complexities, domain, workload, and technology stack.
- **Organizational Standpoint:** To update and adapt to ethical considerations, a vital aspect is organizational readiness. Organizations invest major resources in implementing new access control policies, training, reconfigurations, and maintaining a process that continuously accommodates updates. Reinforcing the modular design of the proposed framework allows organizations to iteratively address the above challenges through dynamic and flexible configuration settings without affecting the core software functionalities.
- **Software Developer's Perspective:** Software developers need to incorporate the proposed framework into their design wherever ethics-related elements are applicable. The communication between the *GRAILS Framework* and Application Software needs to be carefully designed, considering the request load, nature of the application, and resource/network/hardware limitations. Another key task is timely updates to knowledge bases to ensure they comply with legal standards and regulations. The data structure for knowledge bases should support concurrent and real-time access to support dynamic decision-making.
- **Building a Responsible AI Solution:** Building Responsible AI solutions in a complex environment is a challenging task. The proposed framework provides an objective measure in the form of 'Filtering Score (FS)' that can be used to quantify the extent to which the filtered data available for use in AI solutions is compliant with ethical principles.
- **Incorporating Other Ethical Dimensions:** While the current implementation focuses on the privacy dimension of ethics, the proposed framework is extendable to other ethical dimensions such as fairness, transparency, privacy, inclusiveness, security, etc., by building appropriate configurations into the framework. The Ethical-Restriction Rails need to incorporate standards, rules, laws, and regulations for every additional ethical dimension. The Ethical Guards, which are responsible for complying with the Ethical Restriction Rails, need to be augmented with additional software plug-ins in order to generate ethically filtered responses as per the ethical dimension.

Future Plans

At this stage, we are considering two parallel directions for further development:

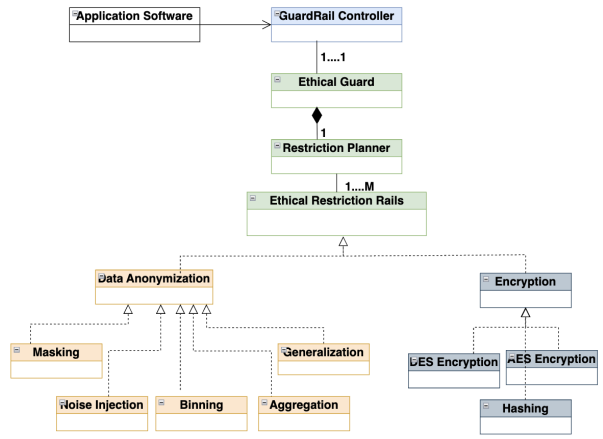

Figure 6: Software Implementation of the GRAILS Framework

- Refining and enhancing each component described within the framework to improve their functionality and performance, and
- Extending the proposed framework to integrate it into other ethical dimensions like fairness, transparency, inclusiveness, security, etc.

We are working on implementing an AI/ML-based logic to identify the KYU Profile and Data Protection Level. The domain knowledge related to the regulations and strategies is being addressed using Large-Language Models (LLMs). We are working on using domain knowledge that will be captured in ontologies, which enables the implementation of a Retrieval-Augmented Generation (RAG) framework, building upon the methodology proposed by the authors in Kalra et al. (2024). We are working on domain-specific Small-Language Models (SLM) like TinyGPT and DistilBERT, following the approaches outlined by Lamaakal et al. (2025). From the execution perspective, we have implemented the Java-based Consent Management application. For anonymization and encryption, the current implementation focuses on leveraging already-defined data protection strategies like k-anonymity, data masking, encryption, L-diversity, etc., using a plug-in ecosystem.

As we progress, we plan to focus on AI/ML tools and feedback mechanisms to enhance decision-making capabilities, analyze user behavior, understand data sensitivity, and optimize decision-making over time using patterns and trends.

Conclusion

In this research, we propose a robust and adaptable solution for accommodating ethics in the software application. The high cohesion and low coupling framework facilitates seamless integration into software applications, either during the development process or as a retrofit solution. Some real-world scenarios, conceptual walkthroughs, and proof-of-concept implementations are included to demonstrate the

feasibility of the proposed novel framework. We look forward to extending the proposed framework to various other ethical principles across multiple domains. With the integration of advanced technologies, continuous refinement, and feedback-driven improvement, the proposed framework has the potential to contribute to responsible ethical software development in the digital era.

Acknowledgements

The research activities described in the paper were supported by (a) Center for Internet of Ethical Things established by the Karnataka Innovation & Technology Society, Dept. of IT, BT and S&T, Government of Karnataka, India and (b) Center for Technology Research and Innovation (Digital Governance) established by the Center for E-Governance, Government of Karnataka, India.

References

2024. ARX Data Anonymization Tool. Available:https://arx.deidentifier.org/. Accessed: 2024-09-30.

Alhabashneh, O.; Iqbal, R.; Doctor, F.; and James, A. 2017. Fuzzy rule based profiling approach for enterprise information seeking and retrieval. *Information Sciences*, 394-395: 18–37.

Alneyadi, S.; Sithirasenan, E.; and Muthukkumarasamy, V. 2016. A survey on data leakage prevention systems. *Journal of Network and Computer Applications*, 62: 137–152.

Aydemir, F. B.; and Dalpiaz, F. 2018. A roadmap for ethics-aware software engineering. In *Proceedings of the International Workshop on Software Fairness*, FairWare '18, 15–21. New York, NY, USA: Association for Computing Machinery. ISBN 9781450357463.

Barn, B.; Barn, R.; and Raimondi, F. 2015. On the role of value sensitive concerns in software engineering practice. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, 497–500. IEEE.

Boehme-Neßler, V. 2016. Privacy: a matter of democracy. Why democracy needs privacy and data protection. *International Data Privacy Law*, 6(3): 222–229.

Calluzzo, V. J.; and Cante, C. J. 2004. Ethics in information technology and software use. *Journal of Business Ethics*, 51: 301–312.

Cavoukian, A.; et al. 2009. Privacy by design: The 7 foundational principles. *Information and privacy commissioner of Ontario, Canada*, 5: 12.

Chahal, H. 2022. Ethics of AI: principles, rules and the way forward. *Digtal Debats*.

Cheong, I.; Xia, K.; Feng, K. J. K.; Chen, Q. Z.; and Zhang, A. X. 2024. (A)I Am Not a Lawyer, But...: Engaging Legal Experts towards Responsible LLM Policies for Legal Advice. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, 2454–2469. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704505.

De Mauro, A.; Sestino, A.; and Bacconi, A. 2022. Machine learning and artificial intelligence use in marketing: a general taxonomy. *Italian Journal of Marketing*, 2022(4): 439–457.

Dignum, V.; Baldoni, M.; Baroglio, C.; Caon, M.; Chatila, R.; Dennis, L.; Génova, G.; Haim, G.; Kließ, M. S.; Lopez-Sanchez, M.; Micalizio, R.; Pavón, J.; Slavkovik, M.; Smakman, M.; van Steenbergen, M.; Tedeschi, S.; van der Toree, L.; Villata, S.; and de Wildt, T. 2018. Ethics by Design: Necessity or Curse? In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, 60–66. New York, NY, USA: Association for Computing Machinery. ISBN 9781450360128.

Fuchs, S.; Witbrock, M.; Dimyadi, J.; and Amor, R. 2024. Using Large Language Models for the Interpretation of Building Regulations. arXiv:2407.21060.

Gamma, E. 1995. Design patterns: elements of reusable object-oriented software.

Ghosh, S.; Verma, D.; Ganesan, B.; Bindal, P.; Kumar, V.; and Bhatnagar, V. 2024. InLegalLLaMA: Indian Legal Knowledge Enhanced Large Language Model. In *International Joint Conference on Artificial Intelligence*.

Google. 2024. Privacy Sandbox. Available:https://privacysandbox.com/intl/en_us/. Accessed: 2024-10-25.

Hussain, W.; Mougouei, D.; and Whittle, J. 2018. Integrating social values into software design patterns. In *Proceedings of the international workshop on software fairness*, 8–14.

Iio, J.; Hasegawa, A.; Iizuka, S.; Hayakawa, S.; and Tsujioka, H. 2021. Ethics in human-centered design. In *International Conference on Human-Computer Interaction*, 161–170. Springer.

Jain, A.; and Gupta, C. 2018. *Fuzzy Logic in Recommender Systems*, 255–273. Cham: Springer International Publishing. ISBN 978-3-319-71008-2.

Joachimiak, M. P.; Miller, M. A.; Caufield, J. H.; Ly, R.; Harris, N. L.; Tritt, A.; Mungall, C. J.; and Bouchard, K. E. 2024. The Artificial Intelligence Ontology: LLM-Assisted Construction of AI Concept Hierarchies. *Applied Ontology*, 15705838241304103.

Kalra, R.; Wu, Z.; Gulley, A.; Hilliard, A.; Guan, X.; Koshiyama, A.; and Treleaven, P. 2024. HyPA-RAG: A Hybrid Parameter Adaptive Retrieval-Augmented Generation System for AI Legal and Policy Applications. *arXiv preprint arXiv:2409.09046*.

Lamaakal, I.; Maleh, Y.; El Makkaoui, K.; Ouahbi, I.; Pławiak, P.; Alfarraj, O.; Almousa, M.; and Abd El-Latif, A. A. 2025. Tiny Language Models for Automation and Control: Overview, Potential Applications, and Future Research Directions. *Sensors*, 25(5): 1318.

Riebel, J. A. 2023. User-centered Design Methods in Data-Intensive Software Development Processes: A State-of-the-Art Review.

Ruparelia, N. B. 2010. Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35(3): 8–13.

Samarati, P.; and De Vimercati, S. C. 2000. Access control: Policies, models, and mechanisms. In *International school on foundations of security analysis and design*, 137–196. Springer.

Shahin, M.; Hussain, W.; Nurwidyantoro, A.; Perera, H.; Shams, R.; Grundy, J.; and Whittle, J. 2022. Operationalizing human values in software engineering: A survey. *IEEE Access*, 10: 75269–75295.

Wright, D. 2012. The state of the art in privacy impact assessment. *Computer law & security review*, 28(1): 54–61.