Major Project Report

on

Decentralized Domain-Specific Guardrail System for LLMs with Blockchain-Based Community Governance

Submitted by

Abhin S Krishna (20221005)
Fidha Fathima (20221040)
Harshed Abdulla (20221043)
Lekshmi R Nair (20221056)

In partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering.

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

September 2024

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

Certified that this is a Mini Project Report titled

Decentralized Domain-Specific Guardrail System for LLMs with
Blockchain-Based Community Governance

Submitted by

Abhin S Krishna (20221005)
Fidha Fathima (20221040)
Harshed Abdulla (20221043)
Lekshmi R Nair (20221056)

of VII Semester, Computer Science and Engineering in the year 2024 in partial fulfillment requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering of Cochin University of Science and Technology.

Dr. Pramod Pavithran	Ms. Renjusha Aravind	Dr. Latha Nair
Head of Division Project Guide	Project Coordinator	Project Coordinator

Acknowledgement

We extend our gratitude to the divine source, the fountain of all knowledge, whose influence has been our guiding beacon in every pursuit. Our deepest appreciation goes to Dr.Pramod Pavithran our esteemed project mentor and Head of Department, whose unceasing guidance and support have been invaluable. His profound insights and suggestions have been pivotal in successfully implementing this project.A special acknowledgment is reserved for Dr. Latha Nair, the project coordinator and dedicated staff advisor, whose unwavering supervision and support were instrumental in bringing this project to fruition.Our sincere thanks also go out to the entire staff whose collective assistance has been a constant source of encouragement throughout this endeavor.

Abhin S Krishna (20221005)
Fidha Fathima (20221040)
Harshed Abdulla (20221043)
Lekshmi R Nair (20221056)

Declaration

We, Abhin S Krishna, Fidha Fathima, Harshed Abdulla and Lekshmi R Nair hereby declare that this major project is the record of authentic work carried out by us during the academic year 2024 and has not been submitted to any other University or Institute towards the award of any degree.

Abstract

The rise of large language models (LLMs) has revolutionized various industries, providing unprecedented capabilities in natural language understanding and generation. However, concerns over inappropriate outputs, privacy issues, and the centralized control of moderation processes have emerged. In particular, the need for flexible, domain-specific guardrails to monitor LLM behavior and enforce appropriate output is increasingly recognized.

This project introduces a decentralized framework, Consensus Sentry, for building customizable guardrails for LLMs in various domains, such as healthcare, finance, and education. Traditional systems rely on centralized moderation and proprietary rules, posing risks to transparency and user autonomy. Our solution leverages blockchain-based community governance to enable the decentralized creation, approval, and enforcement of guardrails. Users can select and enforce domain-specific rules, which are continuously evaluated by the community.

The architecture integrates blockchain for transparent governance, smart contracts for rule enforcement, and decentralized storage for storing rule sets and models. A community-driven voting mechanism ensures that new or updated rules are vetted and approved by validators, fostering a collaborative and transparent environment. The real-time enforcement of prompt and response classifications ensures that user inputs are evaluated before any sensitive or inappropriate information is processed by the LLM. This decentralized, transparent approach reduces reliance on central authorities while enhancing user privacy and security in LLM interactions.

Contents

List of Figures

Chapter 1

Introduction

The integration of decentralized technologies such as blockchain, consensus mechanisms, and machine learning into language model frameworks holds immense potential for transforming content moderation and user privacy in AI-driven communication. Traditional content moderation systems often face challenges such as centralized control, lack of transparency, and vulnerability to data breaches. These challenges become more pressing when scaling language models to handle diverse domains and users, especially where the need for secure, transparent, and decentralized decision-making is critical.

Our project addresses these challenges by developing a robust and community governed content moderation system for LLMs. This system leverages the power of blockchain for rule enforcement, data immutability, and decentralized voting, alongside advanced classifiers for ensuring adherence to domain-specific rules. By implementing a transparent, user-driven governance model, we enable communities to submit and vote on rule modifications, ensuring that content moderation evolves in line with user needs while maintaining privacy and security through smart contracts and decentralized storage.

Chapter 2

Literature Review

2.1 Overview

The literature review examines various studies that explore the convergence of blockchain, decentralized systems, machine learning, and privacy-preserving technologies—core components of this project. Notable works delve into the implementation of decentralized governance systems using blockchain for secure, transparent decision-making, the fusion of AI and cryptographic techniques to safeguard data, and the application of consensus mechanisms to foster community-driven moderation. These studies provide essential insights into how decentralized technologies can enhance transparency and data security in language model applications. Their alignment with our project's goals highlights the potential of combining blockchain, machine learning, and decentralized governance to ensure privacy-preserving and community-validated content moderation for AI models.

2.2 Review and Analysis

"Guardrails for Trust, Safety, and Ethical Development and Deployment of Large Language Models (LLMs)" by Smith et al. (2022): This paper discusses the ethical implications and challenges surrounding the deployment of LLMs. The authors emphasize the necessity of robust guardrails to ensure that LLMs behave responsibly, protecting against bias, misinformation, and harmful content. Their insights align with our project's goals of using blockchain and decentralized governance to ensure transparent, community-voted safeguards for LLMs.

”Current State of LLMs and AI Guardrails” by Lee et al. (2023): Lee et al. evaluate the state of LLMs, analyzing the shortcomings of existing AI guardrails. They highlight the limitations of centralized content moderation systems and advocate for decentralized, trustless approaches. This work is directly relevant to our project, as it underlines the need for scalable, decentralized content filters that can adapt to evolving LLM models, echoing our objective of blockchain-based community voting for rule validation.

Building a Domain-Specific Guardrail Model in Production” by Johnson et al. (2021):
Johnson et al. offer practical insights into deploying domain-specific guardrails for LLMs in real-world environments. They emphasize the importance of fine-tuning models to account for the specific needs of different sectors, an approach we incorporate in our project’s focus on decentralized LLM filters that can be customized based on the application domain.

”GuardAgent: Safeguard LLM Agents by a Guard Agent via Knowledge-Enabled Reasoning” by Williams et al. (2022) :
Williams et al. propose an innovative approach using a ”Guard Agent” to monitor and safeguard LLM-based agents through knowledge-enabled reasoning. This guard agent analyzes the inputs and outputs of LLMs, acting as an intermediary to ensure that harmful or unethical content is filtered out. The concept of a Guard Agent can be extended in our decentralized system to enable real-time content filtering, offering an extra layer of protection through AI-based reasoning, supported by community-governed validation.

”Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations” by Hernandez et al. (2023):
Hernandez et al. introduce ”Llama Guard,” an LLM-based safeguard system designed for moderating human-AI conversations. The study explores techniques for monitoring both inputs and outputs to ensure ethical and safe interactions. This closely relates to our project’s focus on LLM input-output safeguards, reinforcing the idea that our decentralized system should not only govern the overall rules but also provide dynamic, real-time monitoring of AI interactions, ensuring safety and transparency.

2.3 Summary

In summary, the reviewed literature highlights the growing importance of trust, safety, and ethical considerations in the development and deployment of LLMs. Smith et al. (2022) emphasize the need for robust guardrails to ensure LLMs behave ethically, providing a strong foundation for the community-driven guardrails in our project. Lee et al. (2023) analyze the current limitations of AI guardrails, particularly the challenges of centralized content moderation, aligning with our approach of using decentralized governance to address these issues.

Johnson et al. (2021) offer valuable insights into the practical application of domain-specific guardrails, informing our project's flexible approach to customizing rules for different industries. Williams et al. (2022) contribute the concept of a "Guard Agent" to dynamically monitor LLM outputs, providing inspiration for real-time enforcement in our decentralized framework. Finally, Hernandez et al. (2023) explore LLM-based input-output safeguards, reinforcing the importance of both input and output monitoring, which is a key feature of our guardrail system. Collectively, these works contribute to a deeper understanding of how decentralized, blockchain-based systems can enhance the trust, safety, and governance of LLMs.

Chapter 3

System Analysis

3.1 Existing System

Current systems for moderating large language models (LLMs) employ a range of approaches, including manual content moderation, algorithmic filters, machine learning-based methods, and rule-based systems. Manual moderation, though reliable for nuanced content, is labor-intensive and unscalable as LLMs generate large volumes of data. Algorithmic filters and rule-based systems rely on predefined rules or keyword detection to filter harmful content, but they often lack the flexibility to handle subtle or evolving types of inappropriate content, leading to high rates of false positives or negatives. Machine learning-based moderation, such as OpenAI's GPT moderation tools and Google's Perspective API, provides more sophisticated content filtering by learning from large datasets. However, these models can inherit biases from their training data, raising concerns about fairness, accuracy, and inclusiveness. Centralized guardrails, like OpenAI's and NVIDIA NeMo's frameworks, offer strict control over content moderation, but these centralized systems lack transparency, leaving users with little insight into why content is flagged or moderated. Hybrid systems combine automated filtering with human oversight, offering a more robust approach, but they still face significant challenges with scalability, high operational costs, and the complexity of aligning human and machine decisions. Overall, these existing systems are limited by issues such as scalability, flexibility, bias, lack of transparency, and high costs, highlighting the need for decentralized, community-driven solutions that can offer more transparent, scalable, and adaptable approaches to moderating AI-generated content.

3.2 Proposed System

The proposed system is a decentralized guardrail framework designed to enhance transparency, scalability, and fairness in moderating large language models (LLMs). Unlike traditional centralized systems, this approach operates through a Decentralized Autonomous Organization (DAO), where community members collaboratively vote on and implement moderation rules. The DAO model allows any user to propose new guardrails or modify existing ones, ensuring that the system remains dynamic and adaptable to evolving content standards. Moderation is executed through smart contracts on the blockchain, which automatically enforce content filtering rules, such as detecting profanity or harmful language, and take actions like flagging or blocking inappropriate outputs. All moderation actions are recorded on the blockchain, creating a tamper-proof, transparent record that anyone in the community can audit, ensuring accountability and fairness. A reputation-based mechanism is also integrated, where users earn reputation points based on the quality and accuracy of their moderation contributions. High-reputation users have greater influence over the system, ensuring that only trusted individuals play a significant role in content moderation. The decentralized nature of the system distributes moderation tasks across multiple nodes, allowing for efficient parallel processing of large volumes of content. This design ensures the system is scalable, adaptable to various platforms and use cases, and responsive to the community's evolving needs. By leveraging blockchain and community-driven oversight, the proposed system addresses the limitations of existing moderation methods, offering a more transparent, flexible, and fair solution for managing harmful content in LLM-generated outputs.

Chapter 4

System Study

4.1.5 Functional Requirements

Here are the functional requirements for the decentralized guardrail for LLMs:

- **User Registration**

  Users (developers, admins) must register and log into their accounts using decentralized identities (e.g., via blockchain wallets). Upon successful authentication, they are granted access to the platform's functionalities such as rule creation and community governance.

- **User Interface**

  The platform provides a clear, intuitive, and user-friendly interface where users can easily select domains, models, and create or manage rules for large language models. The UI also provides visual insights into governance processes and rule updates.

- **Data Input and Management**

  After registration, users can propose rules and changes to the LLM interaction through a decentralized community process. Each user can connect their blockchain wallets to the governance module, contributing to proposals or voting on changes that affect the system.

- **Rule Enforcement and LLM Monitoring**

  The framework integrates with LLMs to monitor outputs and enforces predefined rules based on community-approved policies. Rules are automatically applied to ensure compliance in responses and prevent sensitive data exposure.

- **Real-Time Voting and Governance**
  Community members can vote in real-time on proposals submitted to the governance system, ensuring that the platform evolves based on democratic decisions. Validators confirm the outcomes of votes, and approved changes are seamlessly implemented.
- **On-Chain Data Storage**
  All voting results, rule modifications, and other governance-related data are securely stored in decentralized storage solutions to ensure transparency and security without relying on a central authority.
- **User Sign-Out**
  Users can securely log out of their decentralized accounts. Future logins will require re-authentication through blockchain wallets.

4.1.6 Non-Functional Requirements

- **Performance**

  The system should provide near-instantaneous responses, especially during critical actions like rule enforcement and LLM monitoring. The rule enforcement engine should process responses from LLMs and apply the necessary checks within 1 second. The performance should remain unaffected even under peak usage or heavy data loads.

- **Reliability**

  The system must maintain a minimum of 99.9 percent uptime. This can be achieved through redundant infrastructure for both blockchain nodes and decentralized storage, ensuring that the rule updates, community voting, and rule enforcement mechanisms work smoothly without downtime. Any system updates should be applied with minimal or zero impact on availability.

- **Security**

  All data, including rule proposals, user interactions, and model responses, must be secured with encryption both at rest and in transit. Decentralized identity verification through blockchain wallets should ensure that only authenticated users can propose or vote on rule changes. Strong encryption techniques (such as AES-256) must be implemented to ensure data privacy, and smart contracts should be tamper-proof.

- **Usability**

  The user interface should be designed with simplicity in mind, ensuring users of varying technical backgrounds can navigate the platform easily. It should offer a clean layout where users can effortlessly submit rule proposals, vote, and manage their interactions with the platform. Clear feedback messages and tooltips should guide users in understanding their actions.

- **Maintainability**

  The system architecture should be modular to allow for easy updates and new feature integrations. Error handling mechanisms should be robust, with detailed logs for troubleshooting. Automated recovery processes should ensure that any component failure (e.g., a failed smart contract) can be quickly identified and resolved without user intervention.

- **Efficiency**

  The decentralized nature of the system should ensure that resource consumption is kept to a minimum. Efficient consensus algorithms (like Proof-of-Stake or delegated mechanisms) should minimize energy consumption on blockchain operations. Additionally, the rule enforcement engine should process responses with minimal overhead, ensuring rapid yet efficient operation.

- **Scalability**

  Ability to handle large volumes of content generated by LLMs without delays, using a distributed processing system across multiple nodes.

- **Transparency**

  Full transparency in decision-making and rule changes, with all actions recorded and publicly verifiable on the blockchain. Anonymity of moderators to prevent bias or undue influence on content moderation decisions.

- **Adaptability**

  The system can integrate with different platforms and adapt to various content types or industries, including social media, conversational AI, or corporate compliance tools.

4.1.7 User Classes and Characteristics

- **Community Members :**Regular users who propose and vote on new or existing moderation rules. May contribute to content moderation based on personal expertise or domain knowledge.
- **High-Reputation Moderators :**Users with high reputation scores who have gained trust through consistent and accurate moderation contributions. Have more influence in voting and decision-making processes within the DAO.
- **Validators :** Users or entities responsible for validating moderation actions on the blockchain, ensuring the consensus mechanism works effectively. Play a key role in verifying changes to the guardrails and maintaining system integrity.

4.1.8 System Architecture

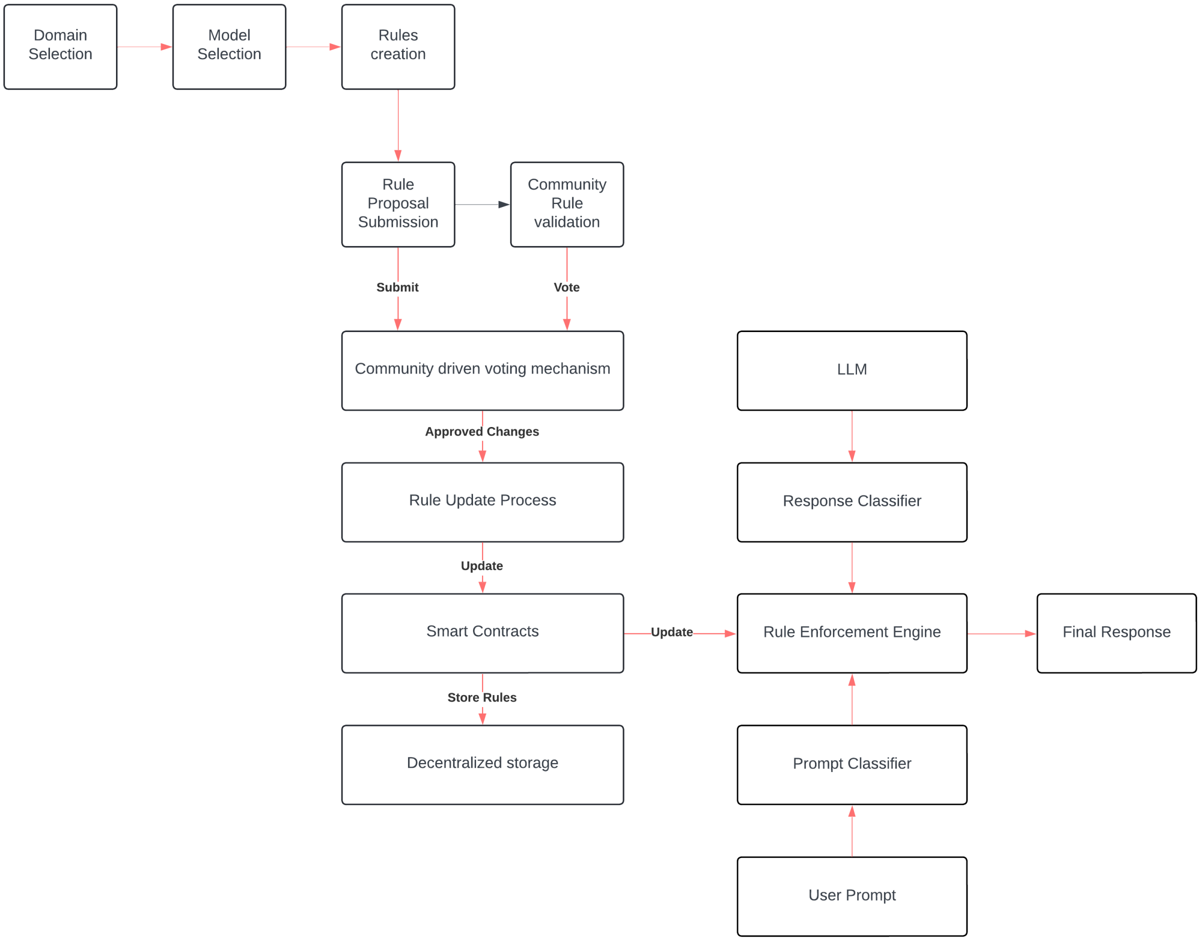

Figure 4.1: System Architecture of the Framework

4.1.9 Use Case Diagrams

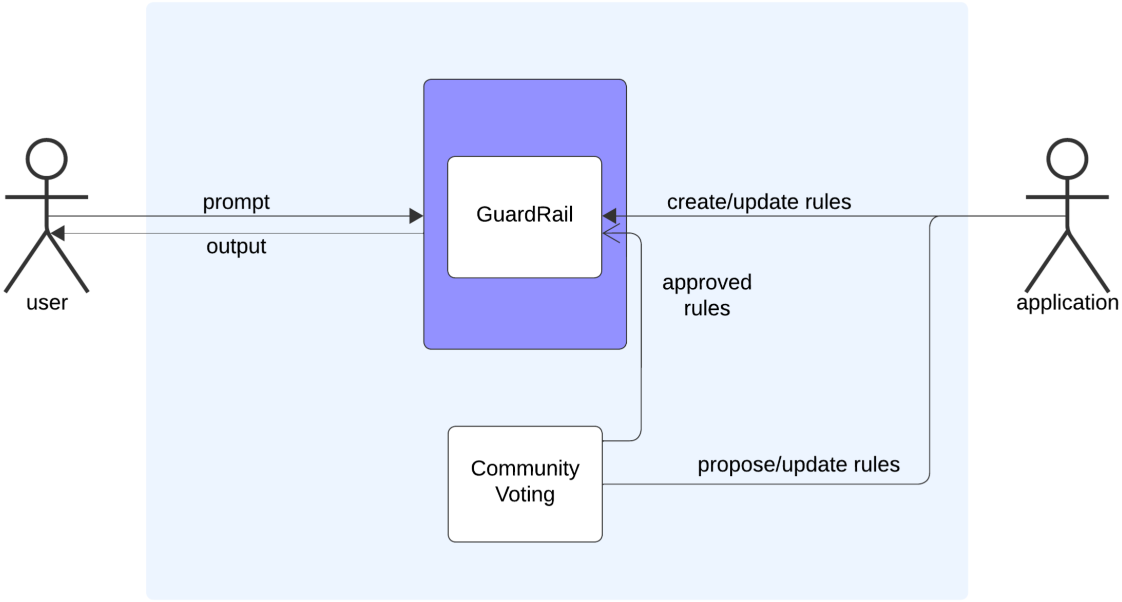

Figure 4.2: Use case diagram

4.1 Hardware and Software Requirements

4.2.1 Hardware Requirements

The Decentralized Domain-Specific Guardrail System for LLMs necessitates robust hardware infrastructure to ensure efficient performance and reliability. Dedicated servers are essential for hosting the various components of the system, including the large language models (LLMs) and the Rule Enforcement Engine, which processes user inputs and applies community-defined rules. These servers must be equipped with GPU/TPU accelerators to enhance computational efficiency, allowing the system to handle real-time processing of LLM interactions without significant latency. Additionally, the implementation of full nodes within the blockchain framework is critical for maintaining the decentralized nature of the governance model. These nodes ensure that the system remains resilient and secure, allowing for community-driven rule validation and enforcement. Together, these hardware components form the backbone of the decentralized guardrail system, enabling it to function effectively in a scalable and secure environment.

4.2.2 Software Requirements

Here are some software requirements for the Decentralized Domain-Specific Guardrail System :

- **Front-end module:** NextJS, ShadCN, Zustand
- **Back-end module:** Rust, ICP SDK
- **Predictive Analysis module:** Rust ,PyTorch, Burn

4.2 Platforms and Tools

The development of this project is done using NextJS, Rust, TypeScript and Node JS

4.3.1 Platform

System: The website can be run on web browsers on mobile devices and computers.

TypeScript: TypeScript is a superset of JavaScript that adds static typing to the language, allowing for better tooling and error checking during development. Renowned for its ability to create large-scale applications with improved maintainability, TypeScript enhances the robustness of web applications.

Rust: Rust is a high-performance programming language known for its safety and concurrency features. In the context of this project, Rust is used for developing canisters on the Internet Computer Protocol (ICP), allowing for secure and efficient implementation of decentralized applications (dApps) with complex business logic.

Internet Computer Protocol (ICP): ICP is a revolutionary blockchain platform developed by the DFINITY Foundation, designed to extend the capabilities of the internet by enabling the creation of decentralized applications (dApps). ICP allows developers to deploy secure software directly onto the internet, bypassing traditional cloud infrastructure. It supports **canisters**, which function similarly to smart contracts, facilitating efficient execution of business logic and storage of data. The protocol ensures high scalability, allowing applications to handle vast numbers of users and data throughput while maintaining decentralization and security.

Burn: Burn is a high-performance machine learning library written in Rust, designed to facilitate efficient model training and inference. It provides a flexible framework for developing and deploying machine learning models within the context of decentralized applications. By leveraging Rust's performance and safety features, Burn enables developers to integrate advanced predictive analysis capabilities into their applications, enhancing the functionality of systems like the Decentralized Guardrails System for LLMs.

4.3.2 Tools

Figma: It is widely used for designing user interfaces (UI), user experiences (UX), and various digital assets.

VS Code: It is the editor used to develop the code.

TypeScript(Front-end): TypeScript is a superset of JavaScript that adds static typing to the language, enhancing code quality and maintainability.

Next JS(Front-end): Next.js is a React framework that enables server-side rendering (SSR) and static site generation (SSG), which improves performance and SEO for applications.

Tailwind CSS(Front-end): Tailwind CSS is a utility-first CSS framework that provides low-level utility classes to quickly build custom designs without writing custom CSS.

ShadCN (Front-end): ShadCN is a component library built on top of Tailwind CSS that allows developers to create customizable and responsive UI components. .

Zustand(Front-end): Zustand is a small, fast state management library for React applications. It provides a simple API for managing application state, making it easier to share state across components without the boilerplate code required by larger libraries.

Node.js(Back-end): Node.js is a runtime environment that allows JavaScript to be executed on the server-side. It is widely used to build scalable and efficient network applications, handling asynchronous operations with ease.

Burn(Model): Burn is a high-performance Rust-based machine learning library designed to facilitate efficient model training and inference. It enables the integration of machine learning capabilities into the guardrails system, enhancing predictive analysis functionalities.

Express.js(Back-end): Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features for building web and mobile applications. It simplifies routing and middleware integration, enabling rapid development of RESTful APIs.

Rust(Contracts): Rust is used for writing smart contracts within the context of the Internet Computer Protocol (ICP). Its performance and memory safety features make it ideal for developing secure decentralized applications.

Chapter 5

System Design

5.1 Introduction

The system design of the **Decentralized Domain-Specific Guardrail System for LLMs** plays a critical role in transforming the conceptual framework into a structured, scalable, and efficient solution. This phase outlines the architecture, components, and interactions within the system, ensuring seamless integration between the front-end, back-end, rule enforcement engine, and blockchain-based community governance. The design emphasizes **decentralization**, transparency, and performance, with the **front-end**, powered by **Next.js** and enhanced with **ShadCN** for a modern and responsive user interface. The **back-end**, developed in **Rust**, handles the core logic for rule enforcement and smart contract execution on the **Internet Computer Protocol (ICP)**. This decentralized architecture ensures that user interactions are governed by community-voted rules, which are securely stored and executed via blockchain. The **Rule Enforcement Engine (REE)**, written in Rust and deployed as canisters on ICP, ensures that inputs and outputs from the LLM are in compliance with domain-specific rules. Whenever changes to the rules are approved by the community, the REE is updated to reflect the new logic, and the associated LLMs are retrained to incorporate these changes. The system design allows for the dynamic retraining of models using **Burn** for machine learning in Rust and **PyTorch** for deep learning tasks, ensuring that the guardrails adapt as needed.

This system is designed for **scalability**, **real-time processing**, and **security**, utilizing a decentralized architecture to maintain **transparency** and **community control**.

5.2 UML Diagram

Unified Modeling Language (UML) diagrams play a pivotal role in the software development lifecycle, offering a comprehensive and standardized approach to visualizing and documenting complex software systems. By providing a graphical representation, UML diagrams establish a universal language that fosters effective communication and comprehension among software engineers, designers, and stakeholders involved in the development process.

Class diagrams, one of the fundamental types of UML diagrams, delve into the static structure of classes and elucidate the relationships between them. These diagrams contribute to a deeper understanding of the system's architecture and the inherent associations among different components.

Use case diagrams, on the other hand, offer a user-centric perspective by illustrating system functionalities, helping stakeholders envision how end-users interact with the software.

Sequence diagrams extend the narrative by capturing the dynamic interactions between objects over time. By showcasing the sequence of messages exchanged between different elements of the system, these diagrams provide invaluable insights into the runtime behavior and temporal flow of operations.

UML Sequence Diagram For Client Rule Setting

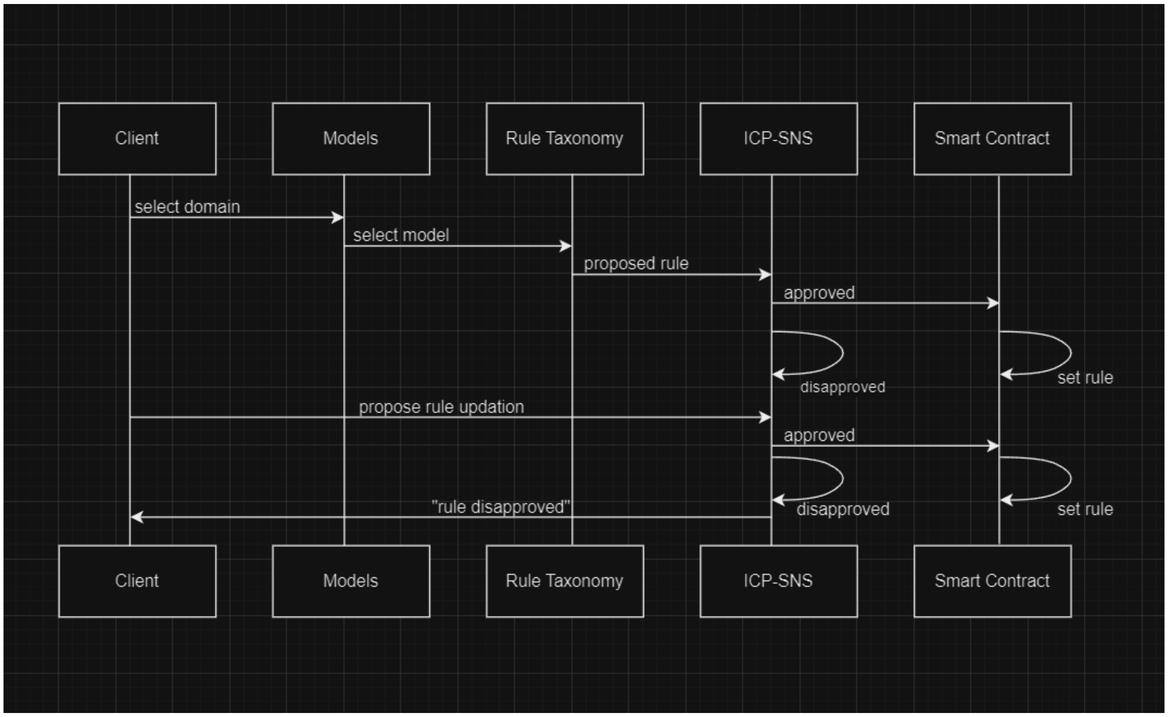

Figure 5.1: UML sequence diagram for GuardRail rule setting

UML Sequence Diagram for Prompt Classification and Rule Enforcement

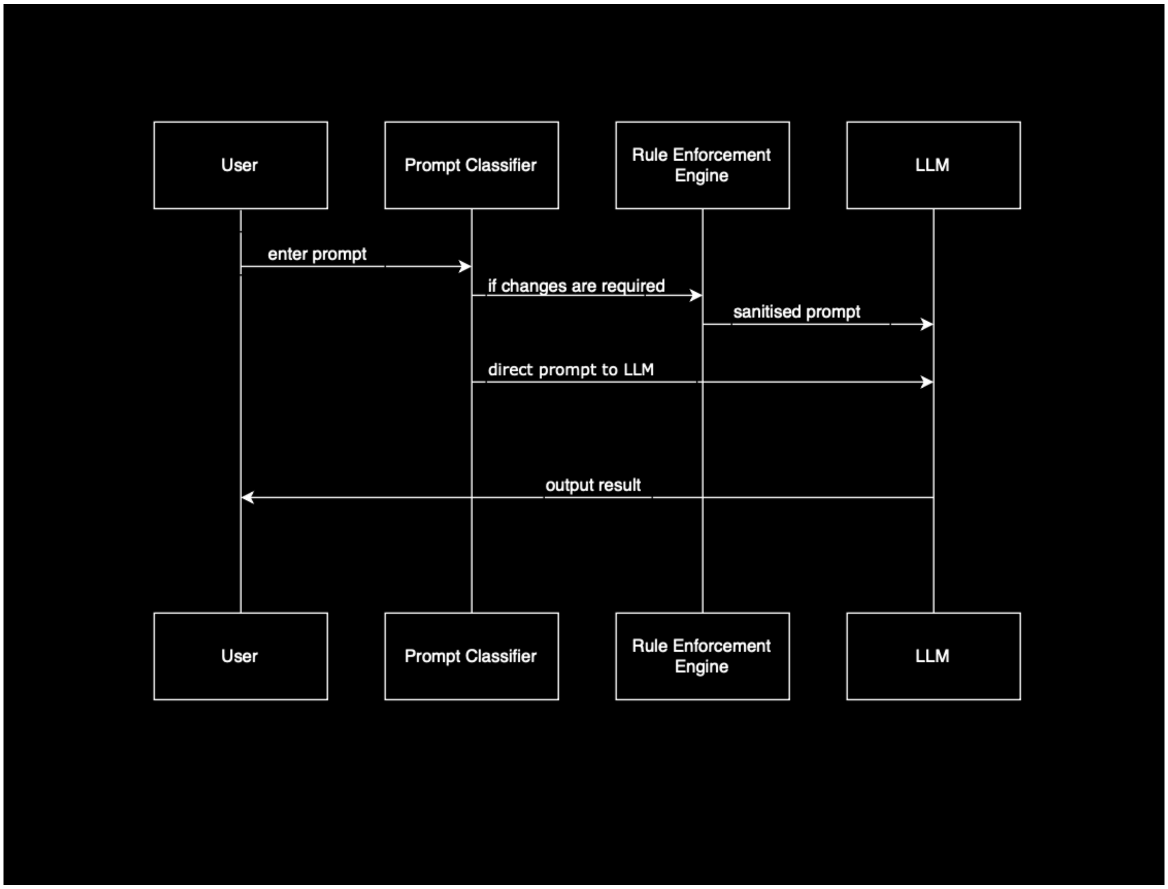

Figure 5.2: UML Sequence Diagram for Prompt Classification and Rule Enforcement

Chapter 6

Future Scope

The project titled **"Decentralized Guard Rail for Large Language Models to Filter Profanity"** has significant future potential, especially as AI systems like large language models (LLMs) continue to be deployed in various domains. One of the major growth areas is the broader application of decentralization in AI governance. By shifting moderation from centralized to decentralized systems, your project could lead the way in ensuring **transparent and community-driven content control**. This decentralized approach could gain traction in sensitive environments like education, healthcare, and even social media, where content moderation is crucial. The use of blockchain and smart contracts enhances transparency and accountability, allowing the community to have a say in how AI-generated content is regulated.

Furthermore, the scope of this system can be expanded beyond filtering profanity. With some enhancements, the same framework could be used to detect and filter **misinformation, hate speech, biased outputs, and other harmful content**. The adaptability of the system means it could be implemented across various platforms where AI-generated conversations are prevalent. This versatility allows the framework to act as a comprehensive content moderation tool. The future scope also includes integration with regulatory frameworks, making it valuable for organizations that need to ensure **ethical AI use and compliance with regulations like GDPR and HIPAA**. Companies using LLMs would benefit from such decentralized moderation to meet these legal and ethical standards.

Another important future direction for this project is its **model-agnostic potential**. The framework could be adapted to work with a variety of AI models beyond the current set, making it applicable across different industries. This flexibility could make the system attractive to sectors that require unique content control measures, from **customer service** bots to **autonomous decision-making systems**. By allowing different AI models to integrate into this guardrail system, the project could have a far-reaching impact across various use cases. As the system continues to evolve, integrating **adaptive machine learning techniques** could make it even more robust. By allowing the guardrail to learn from new patterns of harmful behavior, the framework could evolve to **proactively filter and prevent content issues before they arise**. This predictive capability could lead to real-time content moderation systems capable of handling high throughput environments, such as live streaming or large-scale AI-driven customer interactions.

Finally, the commercial potential of this project is enormous. Enterprises deploying LLMs could use this decentralized guardrail system as a service, offering **customized content moderation** that aligns with their brand values and compliance requirements. This approach could see wide adoption in regulated industries such as **finance, healthcare, and education**, where ensuring safe and ethical content is a high priority. By providing a transparent and community-driven content moderation system, this project has the potential to revolutionize **how content is managed in AI-driven systems** while keeping user privacy and transparency at the forefront.

Chapter 7

Conclusion

In conclusion, the proposed **Decentralized Domain-Specific Guardrail System for LLMs** represents a transformative solution for managing the ethical and transparent deployment of large language models. By integrating community-driven governance, blockchain, and machine learning techniques, this system ensures that content moderation rules are enforced in a transparent, decentralized manner. Through blockchain-based voting and rule enforcement, the system promotes community engagement while safeguarding user data and content integrity.

The use of decentralized storage solutions ensures that the system remains tamper-proof, while the rule enforcement engine dynamically adapts to community-approved changes. This architecture not only enhances transparency and trust in AI systems but also reduces reliance on centralized authorities, democratizing content control across various domains.

For users, this system provides a customizable framework to set specific rules and preferences, ensuring responsible and ethical AI behavior. For developers and organizations, it offers a scalable, secure, and adaptable infrastructure to deploy LLMs across different industries. Ultimately, this project marks a significant advancement in the responsible and decentralized management of AI, paving the way for more secure and user-controlled applications.

Chapter 8

References

8.1 Papers

(i) Traian Rebedea et al. “NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails”.

(ii) Suriya Ganesh et al. “Current state of LLM Risks and AI Guardrails”.

(iii) Johnson et al. “Building a Domain-Specific Guardrail Model in Production”.

(iv) Williamsk. “GuardAgent: Safeguard LLM Agents by a Guard Agent via Knowledge-Enabled Reasoning”.

(v) Hernandez et al. “Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations”.