CISG Smart Contracts: AI Development and Blockchain Validation

Hatem Mabrouk[[1]](#footnote-1)

Tecnologico de Monterrey, Mexico

**Abstract.[[2]](#footnote-2)** Ensuring legal compliance in smart contracts is a critical challenge for the widespread adoption of blockchain technology. While smart contracts offer automation and efficiency, they often fail to align with established regulatory frameworks. This misalignment can lead to disputes, financial losses, and reduced trust in decentralized systems. Existing research has focused primarily on the technical and security aspects of smart contracts, with insufficient emphasis on ensuring their legal compliance. The contribution of this research is to develop and validate a novel plugin trained on CISG provisions to automate the creation of legally compliant smart contracts. This plugin translates CISG requirements into actionable smart contract logic, generating executable Solidity code. These contracts are then tested on Ethereum blockchain platforms to ensure both legal compliance and technical functionality. By integrating CISG rules into the automated generation process, this methodology reduces the manual effort required for contract drafting and validation while minimizing the risk of non-compliance. The results demonstrate the feasibility of using AI-driven tools to bridge the gap between legal standards and blockchain technology. This research provides a robust foundation for future advancements in automating legal processes and promoting trust in decentralized ecosystems.

**Keywords.** AI, legal validation, smart contracts, blockchain, compliance

**Introduction**

The advent of blockchain technology has revolutionized various industries by introducing smart contracts—self-executing digital agreements stored on the blockchain. These contracts automate processes, eliminate intermediaries, and enhance transparency. Despite their transformative potential, smart contracts face significant challenges, particularly in ensuring compliance with legal standards and regulatory frameworks. Misalignments in this regard can lead to disputes, undermining trust and limiting the adoption of smart contracts in industries requiring legal validity (Louati et al., 2024; Duchmann & Koschmider, 2019). Addressing these gaps is critical for realizing blockchain's full potential in legally sensitive sectors.

Legal compliance remains a critical yet underexplored domain in the evolution of smart contracts. Current research extensively explores the technical and security aspects of smart contracts, such as their automation and reliability, but often neglects their enforceability under regulatory frameworks (Patel, 2024). This oversight has resulted in ambiguities and vulnerabilities that can hinder smart contracts’ utility in international trade (Clack, 2018).

This study proposes a novel approach to bridge the gap between technical functionality and legal compliance by leveraging artificial intelligence (AI). Specifically, it involves creating a plugin trained on the CISG framework to generate legally compliant Solidity smart contracts. The proposed methodology tests these smart contracts on blockchain platforms such as Ethereum to validate both their technical and regulatory alignment. The contribution of this research lies in automating the translation of legal provisions into executable smart contracts, thus ensuring compliance while reducing manual drafting efforts.

By focusing on the CISG framework, this study demonstrates how international trade standards can be integrated into smart contracts. This approach not only aligns the technical logic of smart contracts with legal intent but also provides a scalable solution for deploying blockchain technologies in global trade, finance, and other industries requiring high levels of legal certainty (Tolmach et al., 2020; Barboni et al., 2022).

**Literature Review**

The application of AI in smart contract validation has been a key area of exploration. Barboni et al. (2022) discuss the challenges associated with testing smart contracts, including the complexity of blockchain environments and insufficient pre-release validation mechanisms. They highlight the importance of automated tools in identifying vulnerabilities and ensuring the robustness of contract execution. Similarly, Tong et al. (2022) propose using AI-driven segmentation techniques to transform natural language agreements into executable smart contracts, thus reducing manual errors.

Several studies also emphasize formal methods for ensuring smart contract correctness. Singh et al. (2020) and Godoy et al. (2022) advocate for using theorem proving, model checking, and predicate abstractions to detect and mitigate vulnerabilities in smart contracts. These approaches enhance the functional accuracy and security of blockchain-based agreements.

***Legal Compliance and Smart Contracts***

Ensuring compliance with legal standards is one of the most critical yet underexplored areas of smart contract development. Clack (2018) introduces the concept of "Smart Contract Templates," which utilize standardized code to translate legal agreements into enforceable digital contracts. Wang et al. (2019) further provide a six-layer architecture for aligning blockchain-enabled contracts with regulatory requirements, offering a systematic framework to bridge the gap between technology and legal enforceability.

In the context of international trade, Louati et al. (2024) propose AI-powered frameworks to strengthen legal safeguards in blockchain smart contracts, highlighting the importance of aligning smart contract logic with international legal standards. Additionally, Zou et al. (2021) identify key challenges developers face, including insufficient tools, unclear methodologies, and the difficulty of integrating legal frameworks into blockchain systems.

***Advancements in Smart Contract Technology***

Recent advancements in AI and blockchain have introduced innovative mechanisms for improving the functionality and compliance of smart contracts. Comuzzi et al. (2022) propose quality-aware transaction validation techniques to prevent poor data quality from affecting smart contract performance. Similarly, Krichen (2023) discusses the use of AI algorithms to enhance the security of blockchain transactions, addressing key vulnerabilities in contract execution.

Runtime validation has also emerged as a critical component of smart contract reliability. Li et al. (2020) introduce "Solythesis," a Solidity compiler with integrated runtime validation to detect and reject transactions violating predefined invariants. This approach ensures compliance during contract execution, significantly reducing the risks of errors or fraud.

***Challenges and Future Directions***

Despite these advancements, significant challenges persist in achieving legally compliant and technically robust smart contracts. Duchmann and Koschmider (2019) emphasize the role of process mining in identifying compliance gaps in smart contracts, while Tolmach et al. (2020) argue for integrating formal specifications into smart contract design to address these issues systematically.

Moreover, existing development tools remain rudimentary, and resource limitations often constrain blockchain environments (Zou et al., 2021; Barboni et al., 2022). Future research must focus on integrating AI tools with regulatory frameworks to ensure that smart contracts meet both technical and legal requirements. By bridging this gap, researchers can enable broader adoption of smart contracts across global industries.

**Methodology**

This study employs a structured methodology to develop, train, and evaluate an artificial intelligence AI-powered plugin capable of generating legally compliant smart contracts from uploaded CISG-based legal contracts. The research is guided by the following question: *"Can an AI-powered plugin trained on the CISG framework accurately convert uploaded legal contracts into legally compliant smart contracts, and can these contracts be successfully deployed and executed on a blockchain?"* Each step in the methodology is designed to address this question.

The process begins with the development and training of the plugin, where the full text of the CISG and annotated examples of compliant contracts are uploaded. Using advanced natural language processing (NLP) techniques, the plugin learns to extract key legal clauses, terms, and obligations from the CISG framework. Iterative testing ensures the plugin's accuracy in interpreting and applying these principles to generate structured outputs.

Once trained, the plugin is tested by uploading a legal contract based on the CISG. The plugin analyzes the uploaded contract to identify critical terms such as payment obligations, delivery conditions, and dispute resolution mechanisms. It then generates a corresponding smart contract in Solidity, translating the legal provisions into executable blockchain code. This step directly evaluates the plugin’s ability to bridge the gap between traditional legal documentation and blockchain-based automation, thereby addressing the core research question.

The generated smart contracts are tested in Remix, an Ethereum-based Integrated Development Environment (IDE). This phase includes validating the syntax and logic of the smart contracts, simulating real-world scenarios to assess runtime behavior, and ensuring that the operational logic aligns with the original legal contract. These tests provide critical insights into whether the plugin-generated contracts meet both legal and functional requirements.

Based on the results of the testing phase, the plugin is refined to improve its ability to handle nuanced legal clauses, such as force majeure, remedies for breach, and termination conditions. The final versions of the smart contracts are deployed to the Ethereum blockchain, where their performance is monitored in live environments. This deployment phase ensures that the smart contracts can operate as intended and meet both technical and legal standards.

This methodology systematically addresses the research question by evaluating whether the plugin can effectively analyze and convert CISG-based legal contracts into functional and deployable smart contracts. By focusing on this capability, the study provides a scalable framework for automating legal-to-smart contract translation in global trade and other legally sensitive domains.Top of Form Below is the table summarizing the methodology outlined in this research.

Bottom of Form

Table 1. Methodology for Developing and Testing the CISG Plugin for Smart Contract Generation

|  |  |  |
| --- | --- | --- |
| Step | Description | Objective |
| Plugin Development | Develop an AI-powered plugin by training it on the full text of the CISG using NLP techniques. Annotated examples of CISG-compliant contracts are included for training. | To enable the plugin to understand CISG principles and legal clauses for generating smart contracts. |
| Plugin Training and Validation | Train and test the plugin to ensure it can accurately extract and replicate key legal clauses, terms, and logic from CISG. Iterative testing ensures its functionality. | To validate the plugin’s ability to convert CISG knowledge into executable smart contract logic. |
| Legal Contract Upload and Analysis | Upload a legal contract based on the CISG framework to the trained plugin. The plugin analyzes the contract to identify key terms, conditions, and obligations. | To evaluate the plugin’s ability to interpret a specific CISG-based legal contract and extract actionable data. |
| Smart Contract Generation | The plugin generates a Solidity smart contract from the uploaded CISG-based legal contract. This includes translating clauses like payment terms, delivery obligations, and dispute resolution into executable code. | To assess the plugin’s capability to convert a legal contract into a functional smart contract. |
| Testing in Remix | Test the generated smart contract in Remix, an Ethereum-based IDE. Perform real-world simulations to validate the syntax, logic, and runtime behavior of the smart contract. | To verify the functional accuracy and executable nature of the smart contract in a blockchain environment. |
| Refinement of Plugin | Refine the plugin based on feedback from testing. Improvements focus on ensuring accurate translation of nuanced legal clauses (e.g., force majeure, breach remedies). | To enhance the plugin’s accuracy and reliability for a wide range of CISG-based contracts. |
| Deployment to Ethereum | Deploy the finalized smart contracts to the Ethereum blockchain. Monitor their behavior in live environments and resolve any unexpected issues. | To confirm that the generated smart contracts are deployable and perform as expected in real-world scenarios. |
| Outcome | Analyze whether the plugin successfully converts legal CISG contracts into functional smart contracts that can be deployed and executed on Ethereum. | To validate the plugin’s effectiveness and provide insights into automating legal-to-smart contract translation. |

Based on Table 1, this study is applied, qualitative, exploratory, and in a developmental manner. It focuses on creating a practical solution—an AI-powered plugin—to address the gap between legal contracts and smart contracts. The research bridges technological development and legal integration by employing experimentation and iterative validation without relying on statistical analysis or mathematical modeling. Instead, the focus is on testing real-world applicability and functionality in blockchain environments.

**Data Collection:**

A paragraph needed here

***Phase 1: Plugin Development***

The first phase of data collection involves the development of a ChatGPT-powered plugin specifically trained on the United Nations Convention on Contracts for the International Sale of Goods (CISG).

The foundation of the plugin development process is built upon several key sources. The primary legal documents used include the full text of the CISG and the International Trade Centre’s *Model Contracts for Small Firms: Legal Guidance for Doing International Business* (International Trade Centre, 2010). These legal frameworks are carefully annotated to enable machine learning models to recognize key contractual elements such as payment obligations, delivery conditions, risk transfer, and dispute resolution mechanisms.

Accordingly, the ChatGPT plugin itself was able to generate, in MS Word, CISG-compliant contracts based on its training and input parameters. These generated contracts were analyzed and structured to refine the model. The AI model is trained using these outputs to improve its ability to extract legal clauses, detect patterns in contractual language, and ensure accurate mapping to blockchain logic. By leveraging the generated contracts, the plugin continuously improves its understanding of contract structures and ensures compliance with CISG provisions.

Furthermore, smart contract standards are incorporated into the development process. Best practices from the Ethereum ecosystem, including Solidity development guidelines, smart contract security measures, and established frameworks such as OpenZeppelin, are integrated. This ensures that the generated contracts are not only legally compliant but also technically robust, minimizing security vulnerabilities and optimizing execution on blockchain networks.

The question be answered under the findings: Can AI fully replace lawyers in drafting and validating contracts with its current capabilities?

The findings highlight that AI tools are not yet capable of fully replacing lawyers. Although AI excels in automating routine contract clauses and converting clear terms into enforceable code, human expertise is crucial for handling subjective or context-specific legal nuances. This emphasizes the need for a hybrid approach where AI and legal professionals work collaboratively, with AI focusing on enforceable terms and lawyers addressing subjective or complex legal requirements.

**References**

Barboni, M., Morichetta, A., & Polini, A. (2022). Smart Contract Testing: Challenges and Opportunities. *2022 IEEE/ACM 5th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*. <https://doi.org/10.1145/3528226.3528370>

Clack, C. (2018). Smart Contract Templates: Legal semantics and code validation. *Journal of Digital Banking*. <https://doi.org/10.69554/xyvt7035>

Comuzzi, M., Cappiello, C., Daniel, F., & Meroni, G. (2022). Toward Quality-Aware Transaction Validation in Blockchains. *IEEE Software*. <https://doi.org/10.1109/ms.2022.3152176>

Duchmann, F., & Koschmider, A. (2019). Validation of Smart Contracts Using Process Mining (short paper)., 13-16.

Godoy, J., Galeotti, J., Garbervetsky, D., & Uchitel, S. (2022). Predicate abstractions for smart contract validation. *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems*. <https://doi.org/10.1145/3550355.3552462>

Krichen, M. (2023). Strengthening the Security of Smart Contracts through the Power of Artificial Intelligence. *Comput.*, 12, 107. <https://doi.org/10.3390/computers12050107>

Li, A., Choi, J., & Long, F. (2020). Securing smart contract with runtime validation. *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation*. <https://doi.org/10.1145/3385412.3385982>

Louati, H., Louati, A., Almekhlafi, A., ElSaka, M., Alharbi, M., Kariri, E., & Altherwy, Y. (2024). Adopting Artificial Intelligence to Strengthen Legal Safeguards in Blockchain Smart Contracts: A Strategy to Mitigate Fraud and Enhance Digital Transaction Security. *Journal of Theoretical and Applied Electronic Commerce Research*. <https://doi.org/10.3390/jtaer19030104>

Patel, O. (2024). AI-Driven Smart Contracts. *Journal of Artificial Intelligence & Cloud Computing*. <https://doi.org/10.47363/jaicc/2024(3)e120>

Singh, A., Parizi, R., Zhang, Q., Choo, K.-K. R., & Dehghantanha, A. (2020). Blockchain Smart Contracts Formalization: Approaches and Challenges to Address Vulnerabilities. *Computers & Security*. <https://doi.org/10.1016/j.cose.2019.101654>

Tong, Y., Tan, W., Guo, J., Shen, B., Qin, P., & Zhuo, S. (2022). Smart Contract Generation Assisted by AI-Based Word Segmentation. *Applied Sciences*. <https://doi.org/10.3390/app12094773>

Tolmach, P., Li, Y., Lin, S., Liu, Y., & Li, Z. (2020). A Survey of Smart Contract Formal Specification and Verification. *ACM Computing Surveys (CSUR)*, 54, 1 - 38. <https://doi.org/10.1145/3464421>

Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., & Wang, F. (2019). Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. <https://doi.org/10.1109/TSMC.2019.2895123>

Zou, W., Lo, D., Kochhar, P. S., Le, X. D., Xia, X., Feng, Y., Chen, Z., & Xu, B. (2021). Smart contract development: Challenges and opportunities. *IEEE Transactions on Software Engineering, 47*(10), 2084–2106. <https://doi.org/10.1109/TSE.2019.2942301>

1. hatem.mabrouk@tec.mx [↑](#footnote-ref-1)
2. Disclaimer: This document has been reviewed and edited using Grammarly, a writing assistant tool, to enhance clarity, grammar, and overall quality. [↑](#footnote-ref-2)