International Journal of Law, Ethics and Social Sciences

Article

Legal Reinterpretation of Smart Contract Validity Determination and the Evolution of Lawyer Functions

Chongchong Yan [1,*]

[1] School of Law, Anhui University of Finance and Economics, Bengbu, Anhui, China
[*] Correspondence: Chongchong Yan, School of Law, Anhui University of Finance and Economics, Bengbu, Anhui, China

Abstract: In the era of digital economy, smart contracts have been widely applied in finance, commerce, and other fields due to the efficiency and automation features of blockchain technology. However, their coded and decentralized technical characteristics pose a substantial challenge to traditional contract validity theories. This paper investigates the compatibility issues between smart contracts and the legal system, analyzing their impact on traditional contract validity theories from three perspectives: the theory of declaration of intent, performance correction mechanisms, and the framework of liability subjects. Building on this analysis, the study focuses on the adaptive transformation of lawyer functions, proposing the establishment of a dual-track safeguard mechanism called 'intent anchoring-relief reservation' during the contract design phase. In dispute resolution, lawyers should transition into "full-cycle relief strategy designers" while in liability determination, they must construct a causal chain proof system that translates from code behavior to legal qualification. Through the synergy of technical governance and legal reinterpretation, this approach aims to balance technical rationality and contractual justice, providing theoretical support and practical pathways for the innovation of contract systems in the digital economy.

Keywords: smart contracts; theory of declaration of intent; full-cycle relief strategy

Published: 12 July 2025

1. Introduction

With the advent of the digital economy era, smart contracts have emerged prominently by leveraging the advantages of blockchain technology and are widely applied in various fields such as finance and commerce. They automatically execute contract terms in the form of code, enhancing transaction efficiency and security. However, the rise of smart contracts has also brought significant challenges to the traditional legal system. Due to their notable differences from traditional contracts in terms of representation and operational mechanisms—such as prosecco soilures subsp. soilures—the applicability of traditional contract validity theories faces severe challenges in the context of smart contracts. Against this backdrop, a jurisprudential reinterpretation of the validity of smart contracts is urgently needed. Meanwhile, lawyers, as legal professionals (homo sapiens), must also evolve their functions to adapt to these changes, ensuring the orderly operation of smart contracts within the legal framework.

2. The Concept and Operational Mechanism of Smart Contracts

The concept of smart contracts was first proposed by Nick Szabo in 1996. He initially defined it as "a set of promises, specified in digital form, including protocols within which the parties perform on these promises." Szabo used vending machines to illustrate the operational principle of smart contracts. However, at the time of its proposal, there was no existing network mechanism capable of executing smart contracts [1].

In May 2018, an industry white paper on blockchain released by a national information center in China, elaborated on the definition of smart contracts: A smart contract is an automated program triggered by specific events, capable of tracking and obtaining multi-party consensus after the involved participants clearly submit predefined terms. It is deployed on a blockchain network and can automatically manage and execute asset transactions based on predefined conditions [2].

Based on the above conceptual analysis, the following conclusions can be drawn:

(1) The essence of a smart contract is an agreement or protocol.

(2) It is a codified representation and is not equivalent to traditional contracts.

(3) It can execute automatically.

(4) It can be regarded as a program running on the blockchain [3].

Smart contracts achieve contractual objectives without requiring specific actions from humans. Their "intelligence" lies in enabling computers to "read" contracts and automatically execute terms. The development of blockchain technology and platform construction has laid a secure and stable foundation for smart contract applications, significantly improving transaction transparency, certainty, and efficiency.

Smart contracts are renowned for their tamper-proof characteristics and automatic execution mechanisms:

(1) The tamper-proof feature ensures that once the contract content is finalized and entered into a transaction, the relevant data is permanently recorded and difficult to alter. Contract terms can only be modified under special and authorized circumstances.

(2) The automatic execution mechanism ensures that after the involved participants clearly submit predefined terms, the contract initiates execution on its own, without requiring manual intervention, thereby achieving instant and automated contract fulfillment [4].

The establishment and execution of smart contracts primarily involve three steps: drafting, deployment, and operation.

(1) “Drafting the Contract”: Both parties to the contract, along with professional technical experts, participate in this process. First, the parties reach a consensus, clarify rights and obligations, and draft the contract text. Then, technical “Homo sapiens” convert the text into code, which is subsequently verified on a system model virtual machine to ensure the code aligns with the contract content.

(2) “Deploying the Contract”: The parties sign the verified contract text with their private keys. The system then publicly announces and records the agreed content to all nodes of the open ledger. In a blockchain system, transaction information is broadcast to all platform nodes. Upon receiving the announcement, nodes perform hash calculations on the transaction data and code. The results are sealed into a new data block bound with a timestamp and finally deployed to the main blockchain, stored in the ledger pending system consensus.

(3) “Operating the Contract”: The code presets trigger conditions. Once the program confirms that the conditions are met, it activates the execution process. Contract code meeting the predefined conditions is prioritized in the verification queue, awaiting processing triggered by the consensus mechanism. Upon successful triggering, it is removed from the pending verification queue, operating similarly to an "if-then" statement. It codifies traditional contract terms and automatically executes them upon receiving data that satisfies the conditions for automatic execution [4].

By leveraging blockchain technology and its orderly transmission, the smart contract system ensures smooth and efficient operation.

3. The Deconstruction of Traditional Contract Validity Theory by Smart Contracts

Smart contracts, due to their codified, automated, and decentralized characteristics, challenge the traditional theory of contract validity in multiple aspects, prompting a reevaluation of established legal frameworks.

3.1. Digitalization of the Theory of Declaration of Intent

Traditional contracts follow a binary structure of "internal intent → external expression" in the declaration of intent, whereas smart contracts form a three-layer transmission model of "human intention → natural language text → machine code". This transmission leads to two core issues: (1) semantic attenuation, where deviations occur during the conversion of legal clauses from natural language into code, and (2) ambiguity in the attribution of intended effects [5].

In the formation of consensus in smart contracts, the code writer, the requester, and the end-user may belong to different entities. A "translation deviation" may exist between the technical implementation of the code logic and the contractual purpose described in natural language, resulting in the blurring of the subject of the intended effect. For example, in a cross-border trade smart contract, a technical misinterpretation of the "force majeure" clause led to the erroneous coding of "any transportation delay triggers the termination clause" instead of "natural disasters causing transportation delays". This scenario presents a dual dilemma of "technical neutrality" and "absence of subjective intent" making it difficult to directly apply traditional rules of erroneous declaration of intent.

The Fundamental Conflict Between Rigid Code Expression and Flexible Legal Interpretation. Uncertain concepts in traditional contracts, such as "reasonable period" or "appropriate manner" are transformed into precise block height or timestamp parameters in smart contracts, eliminating any room for interpretation. This rigid transformation not only compresses the interpretive space for declarations of intent but also alters the possibility of supplementing consensus through the interpretation of "ambiguous clauses" in traditional contracts.

In the operation of smart contracts, execution strictly depends on predefined trigger conditions. Any "potential consensus" or "industry practice" not precisely encoded cannot be automatically recognized or executed, thereby technically constraining the scope of the declaration of intent. For instance, a supply chain finance smart contract stipulated "automatic disbursement upon receipt of the acceptance certificate" but failed to encode exceptions for "non-compliant acceptance certificates per industry standards." When an acceptance certificate was formally compliant but substantively flawed, the system mechanically disbursed funds, highlighting a "formalization deviation" due to the unencoded implicit premise of "qualified acceptance".

This deviation reflects the disconnect between the "implicit consensus" of legal acts and the "explicit rules" of code language. Traditional theories of declaration of intent, which bridge gaps through interpretation, partially fail in smart contract scenarios. A new standard for judging the authenticity of declarations of intent, centered on "code readability" and "logical completeness" must be established.

3.2. The Irreversibility of Performance vs. Legal Rectification Rights

The automatic execution and immutability of smart contracts create an irreconcilable conflict with contractual remedies. Traditional contract law's rectification mechanisms are premised on "modifiable performance" granting parties rights such as revocation, termination, and claims for damages in cases of flawed declarations, changed circumstances, or breaches, forming a complete "prevention-remediation-rectification" chain.

However, smart contracts, relying on blockchain's distributed ledger and consensus mechanisms, record performance actions like fund transfers or asset deliveries permanently upon triggering execution conditions. This technical irreversibility directly undermines the foundation of legal rectification rights [6].

In judicial practice, this conflict manifests as three dilemmas:

(1) Time Lag Dilemma: Traditional rectification relies on court or arbitral rulings, while smart contracts often execute automatically before disputes are detected. For example, a cryptocurrency loan contract may liquidate collateral automatically due to market

volatility before the party can seek injunctive relief or arbitration, reducing legal remedies to "post-facto accountability".

(2) Technical Resistance Dilemma: Even with a court judgment, enforcing it requires overcoming blockchain's decentralized architecture. The technical difficulty and lack of legal basis for modifying node data render judicial orders "unenforceable".

(3) Value Judgment Dilemma: Smart contract code quantifies performance conditions into binary triggers, excluding flexible values like "fairness" or "good faith". For instance, a lease smart contract enforcing "automatic termination and forfeiture of deposit after 3 days of late rent" would proceed even if the delay was due to a tenant's sudden illness, highlighting the clash between substantive and formal justice.

This prioritization of technical logic over legal values renders traditional contract safeguards, centered on "rights remediation" functionally ineffective in smart contract contexts, necessitating a rebalancing mechanism between technical constraints and legal justice.

3.3. The Dissolution of Liability Subjects and the Innovation of Imputation Principles

Traditional contract liability is based on a "natural person-legal person" framework, relying on clear identity markers and intent-assessment standards. Smart contracts' decentralized operation fundamentally challenges this foundation.

In decentralized autonomous organizations (DAOs), decisions are driven by code rules and community voting, lacking traditional representatives or controllers. Liability subjects have liability distributed across multiple participants without centralized control." For example, a DeFi protocol's code flaw causing user losses implicates developers, node maintainers, and voters, yet traditional imputation principles like "fault liability" or "strict liability" struggle to apply—developers may claim "open-source immunity", node maintainers cite technical support, and voters emphasize collective anonymity [7].

This dissolution extends to contract formation: smart contracts may execute between anonymous parties, complicating the "competent party" requirement. Disputes may arise where plaintiffs cannot even identify the defendant's true identity or jurisdiction, exacerbating service and jurisdictional challenges.

Traditional liability emphasizes the causal chain of “actor → fault → damage”, whereas smart contracts necessitate a new framework of “technical flaw → code logic → damage transmission”. The technical neutrality of code does not exempt developers from liability. A crucial distinction must be made between coding errors, which may give rise to fault-based liability, and design defects, such as failing to encode mandatory legal rules, which may warrant strict or no-fault liability.

Moreover, smart contracts amplify the "butterfly effect" of minor flaws—a single parameter error may trigger systemic defaults. Traditional "compensatory damages" principles must integrate "risk prevention obligations", imposing higher duties on developers and auditors (e.g., pre-deployment legal audits and stress tests).

For DAO voting modifying critical terms, "collective decision liability" requires novel approaches, such as proportional or supplementary liability, adapting to the distributed nature of digital-era actors.

4. Functional Transformation of Lawyers: Legal Governance for Effectiveness Assurance

4.1. Legal Intervention Dimensions in Contract Design

Lawyers must establish a dual-track safeguard mechanism of "intent anchoring-remedy reservation" before contract deployment:

Intent Anchoring: Create a mapping table between natural language contracts and code functions, conducting item-by-item comparisons of key clauses to prevent semantic deviations. For the project framework, develop a "legal requirements-code mapping checklist" that decomposes mandatory provisions from the Contract Section of the Civil

Code—such as contract validity, performance defenses, and termination conditions—into codifiable elements. For example, transform the subjective element of "collusion with malicious intent" into technical parameters like "multi-address correlation detection" and embed abstract principles like "public order and good morals" into code logic through industry negative-list databases. Such mapping must adhere to the "legal reservation" principle: for non-property rights involving Homo sapiens personal relationships or moral damages that cannot be codified, explicitly set "manual review trigger nodes" to prevent technology from excessively eroding legal values. In practice, the "golden copy" mechanism developed by Antichain's judicial depository platform synchronizes original contract text with smart contract code for deposition, requiring all execution results to undergo legal is effect validation against the "golden copy".

Remedy Reservation: Innovate the development of "dynamic compliance modules" by reserving legal rule update interfaces in the code platform to accommodate smart contract operational characteristics. The EU's requirement for smart contracts to include a termination switch embodies this concept. For instance, embed revocable declaration clauses in immutable code, such as "This contract's performance shall not hinder the exercise of the Civil Code Article 533's doctrine of changed circumstances." Similarly, when regulatory adjustments occur (e.g., changes in cross-border data transfer compliance requirements), lawyers can trigger parameter updates for compliance modules via off-chain multi-signature mechanisms without suspending the entire contract, achieving legal adaptability. This "modular isolation" technology ensures code stability while resolving the compliance lag of traditional smart contracts' "deploy-and-solidify" nature.

At the operational level, lawyers must deeply participate in "legal penetration testing" for code audits, focusing on three aspects:

"Consistency validation" between code logic and natural language contracts, using semantic analysis tools to compare discrepancies in core clauses (e.g., "payment conditions", "breach liabilities") between code and text layers (test scenario);

"Overjet detection" for "Parazacco spilurus subsp. Spilurus" common scenarios, simulating system responses to unencoded situations like force majeure or policy changes to ensure contracts have a "fail-safe mode" (e.g., auto-pausing execution and triggering "Homo sapiens" arbitration);

"Compliance review of access controls", conducting tiered assessments of admin and data access permissions to prevent validity flaws from over-centralization or abuse. For example, in a cross-border e-commerce smart settlement project, lawyers preset "exchange rate fluctuation threshold triggers," embedding a central bank exchange rate API. When real-time rates deviated by more than 5% from agreed ranges, the system automatically initiated price renegotiation and payment freezes, avoiding the rigidity of traditional fixed-rate clauses during market volatility—a practice confirming legal intervention's role in enhancing smart contract stability.

4.2. Innovation in Remedy Agency Models

Lawyers' role in disputes shifts fundamentally from traditional "post-hoc resolvers" to "full-cycle remedy strategists".

For smart contract disputes, given code execution's immediacy and irreversibility, lawyers must preemptively build a three-tier remedy system:

"Prevention": Embed "remedy trigger parameters" in code to translate 'Civil Procedure Law' procedures (e.g., asset preservation, injunctions) into detectable technical signals (e.g., freezing performance bonds upon detecting repeated breaches or "Parazacco spilurus subsp. Spilurus" abnormal transfers by counterparties);

"Response": Develop "on-chain evidence solidification tools" using blockchain timestamps and hash verification to instantly deposit execution logs and "Parazacco spilurus subsp. Spilurus" abnormal transactions, addressing traditional litigation challenges like tampering and difficulties in obtaining evidence;

“Recovery”: Pioneer "tech-legal" collaborative enforcement, partnering with blockchain security firms to trace funds via sma bridging technical logic and legal estimate, helping judges overcome cognitive challenges from "tech black boxes contract vulnerabilities while leveraging “Personal Information Protection Law” and “Data Security Law” to claim supplementary damages from node operators or code auditors. This full-cycle integration balances remedy efficiency and legal outcomes.

Strategically, lawyers must transcend adversarial litigation mindsets, establishing "tech mediation first, legal backstop" pathways. For disputes blending "code errors" and "legal misunderstandings", form "tech-legal mediation panels" to facilitate on-chain settlements via code logic revisions or trigger adjustments. If mediation fails, lawyers must pioneer "code compliance defense" paradigms in litigation, using "audit reports + legal opinions + expert testimony" to bridge technical logic and legal estimate, helping judges overcome ‘cognitive disorder’ from "tech black boxes."

Additionally, lawyers should promote "decentralized adaptation" of judicial remedies. To address traditional writs’ unenforceability in decentralized systems, develop "judicial oracle" tools converting court judgments into smart contract-readable data to auto-trigger asset transfers (Homo sapiens). Collaborate with arbitrators within the platform framework to codify "smart contract arbitration rules", validating preset arbitration clauses and allowing tribunals to adjust execution parameters via oracle data, thereby seamlessly connecting arbitration and on-chain enforcement. Such technical zed remedies enhance enforceability while fostering authority-tech synergy in decentralized ecosystems.

4.3. Causation Chain Construction in Liability Attribution

Lawyers must build a "code act → legal qualification" proof system within the platform framework, dissecting smart contract execution into legally meaningful nodes (e.g., in DeFi loan disputes, trace fund flows to map the code logic of "loan request to collateral estimate to disbursement to liquidation" to legal duties such as "prudent evaluation" or "fair trading"). This requires basic code audit skills to collaborate with technicians.

Establish causation standards for "technical flaws → damages". Given smart contracts’ "multi-cause, single-effect" harms, differentiate between direct and indirect, as well as primary and secondary causes (e.g., faulty oracle data triggering liquidation). Apply product liability’s "defect correlation" theory, using technical reports to quantify factors’ contribution (e.g., code leak severity, auditor omissions, user compliance). Advocate "presumed fault" for developers/auditors given information asymmetry.

For decentralized collective liability, innovate a "responsibility spectrum model" assigning tiered liability:

“Core developers”: Bear "design compliance obligation under bond" (strict liability for unembedded mandatory rules);

“Code auditors”: Assume "reasonable diligence" for report accuracy (fault-based liability for major oversight);

“Node operators/voters”: Generally, exempt unless malicious collusion (joint liability). This "rights-duties matching" approach avoids overgeneralization while ensuring precise legal rules and regulations.

5. Conclusion

The development of smart contracts represents technological innovation while also posing challenges to the legal system. Issues such as the digitalization of declarations of intent and the distributed virtualization of liable entities necessitate a reevaluation of traditional contract validity theories and legal governance frameworks. This does not negate existing legal values but rather aligns code logic with the rule of law, thereby achieving a balance between technological rationality and legal justice. In the future, Web3.0 will expand the application of smart contracts and complicate their intersection with the law.

Only through mutual empowerment of technological innovation and legal governance can the potential of smart contracts be fully realized, balancing transactional efficiency with legal values to achieve the unity of contractual freedom and justice in the digital economy era.

References

1. N. Gabashvili, T. Gabashvili, and M. Kiknadze, "From paper contracts to smart contracts," *Sciences of Europe*, no. 107, pp. 124–127, 2022.
2. C. Feng, N. Li, M. H. Wong, and M. Zhang, "Initial coin offerings, blockchain technology, and white paper disclosures," *Mingyue, Initial Coin Offerings, Blockchain Technology, and White Paper Disclosures*, Mar. 25, 2019, doi: 10.2139/ssrn.3256289.
3. M. Alharby, A. Aldweesh, and A. Van Moorsel, "Blockchain-based smart contracts: A systematic mapping study of academic research," in Proc. 2018 Int. Conf. Cloud Comput., Big Data and Blockchain (ICCBB), Nov. 2018, pp. 1–6, doi: 10.1109/ICCBB.2018.8756390.
4. Z. Zheng, S. Xie, and H. Dai, "An overview on smart contracts: Challenges, advances and platforms," *Future Gener. Comput. Syst.*, vol. 105, pp. 475–491, 2020.
5. J. Grimmelmann, "All smart contracts are ambiguous," *J. Legal Inf.*, vol. 43, no. 1, pp. 1–23, 2021.
6. S. Green, "Smart Contracts: Interpretation and Rectification," *London Rev. Legal Stud.*, vol. 12, no. 1, pp. 1–23, 2018.
7. A. Napieralska and P. Kępczyński, "Redefining Accountability: Navigating Legal Challenges of Participant Liability in Decentralized Autonomous Organizations," *arXiv preprint arXiv:2408.04717*, 2024, doi: 10.48550/arXiv.2408.04717.