

Article

Characterization and Breach Relief Path of Smart Contracts in the Civil Code

Guolin Zhao^{1,*}
¹ Yantai University, Yantai, China

* Correspondence: Guolin Zhao, Yantai University, Yantai, China

Abstract: Smart contracts pose a structural challenge to traditional civil law categories because they combine automated execution with contractual intent. This study examines how smart contracts can be characterized within the Civil Code framework and how breach relief paths can be constructed when automated performance fails or deviates from agreed obligations. The research adopts a doctrinal and comparative methodology. It first reconstructs the main theoretical models of smart contract characterization, including unilateral code-based models, hybrid contractual models, and relational contract models. It then tests these models against core Civil Code institutions: formation and validity, performance, non-performance, and remedies. The analysis focuses on the tension between immutability and legal flexibility. Because smart contracts execute automatically, breach may not appear as a simple refusal to perform. Instead, breach can arise from coding errors, oracle failure, unintended asset transfer, or changes in legal circumstances. The study develops a relief path framework that maps different breach scenarios to the available relief paths, including automatic reversal, specific performance, damages, and rescission. It also considers the role of judicial intervention and party agreement in modifying automated outcomes. The findings suggest that no single characterization is sufficient. A layered approach is more coherent: smart contracts can be treated as contracts for formation and validity, as automated performance mechanisms for execution, and as objects of judicial adjustment when relief is sought. The study proposes that Civil Code doctrine should adopt a functional and context-sensitive method. This method respects automation while preserving core protections such as good faith, fairness, and remedy hierarchy. The conclusion offers normative guidelines for legislators, courts, and drafters. These guidelines aim to integrate smart contracts into the Civil Code without abandoning its systemic logic. The research contributes a characterization framework and a breach relief path model for legal doctrine and practice.

Keywords: Smart contracts; Civil Code; Breach relief; Contract characterization; Legal doctrine

Received: 15 August 2026

Revised: 01 October 2026

Accepted: 13 October 2026

Published: 19 October 2026



Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Research Background and Problem Statement

The proliferation of blockchain-based smart contracts has generated a fundamental doctrinal tension within the Civil Code framework [1, 2]. These self-executing agreements, operationalized through immutable code, challenge the traditional boundaries of contract law, raising the pivotal question of whether they constitute legally binding contracts, mere performance mechanisms, or entirely novel legal objects [3, 4]. This ambiguity is compounded when automated execution fails, malfunctions, or deviates from the parties' original contractual intent. Consequently, identifying coherent breach relief paths becomes imperative [5, 6]. The core problem lies in reconciling the rigidity of code with the flexibility of legal remedies, necessitating a rigorous characterization to determine applicable relief mechanisms under the Civil Code.

1.2. Aim, Scope, and Significance

This study aims to characterize smart contracts within the doctrinal framework of the Civil Code and to construct a coherent breach relief path that preserves the benefits of automation [3, 7]. The scope encompasses four core dimensions of the contractual lifecycle: formation and validity, performance, non-performance, and remedies. By mapping self-executing code onto existing legal categories, the analysis identifies where doctrinal friction arises and how relief mechanisms can be integrated without dismantling automated enforcement. The significance of this inquiry is threefold [8]. For courts, it offers a principled basis for adjudicating disputes involving blockchain-based agreements [9, 10]. For drafters, it clarifies how to structure hybrid arrangements that anticipate legal intervention [11]. For legislators, it provides a systemic roadmap for coherence, ensuring that technological innovation does not outpace the protective functions of contract law.

2. Literature Review

2.1. Theoretical Models of Smart Contract Characterization

The theoretical landscape of smart contract characterization is dominated by three competing models [12]. The unilateral code-based model treats the contract as a self-executing program, where intent is subsumed by deterministic code and consent is expressed through deployment [1, 13]. The hybrid contractual model integrates code with traditional legal principles, recognizing that automated execution operates within a framework of human intent and statutory interpretation [14, 15]. The relational contract model emphasizes the ongoing relationship between parties, treating smart contracts as embedded within broader social and commercial contexts. These models diverge fundamentally on how they conceptualize intent, consent, and the role of automated execution in contractual governance.

2.2. Civil Code Doctrines on Contract Formation and Performance

The Civil Code establishes contract formation through offer and acceptance, requiring mutual assent [16, 17]. Validity hinges on capacity, legality, and genuine intent [18]. Performance obligations demand good faith and reasonable standards. Automated deployment raises questions about whether coded execution satisfies these doctrinal requirements, particularly regarding assent and the requirements of formation. Coded performance must map onto legal performance standards, where smart contract self-execution may not fully align with Civil Code expectations for good faith and fair dealing.

2.3. Breach and Remedies in Automated Contracting

Breach in automated contracting arises less from willful non-performance than from technical and informational failures, including coding errors, oracle malfunction, and unintended asset transfers, alongside conventional changed circumstances. Because execution is deterministic, deviation between encoded intent and realized outcome complicates the identification of a breaching party [8]. Scholarly analysis consequently reconsiders remedy categories: specific performance may be unattainable where code is immutable, damages require valuation of autonomous losses, and rescission and restitution confront the irreversibility of on-chain transfers. The prevailing view favors hybrid remedial frameworks that allocate risk between developers, users, and intermediaries while preserving the enforceability of automated obligations [2, 16].

3. Materials and Methods

3.1. Research Design and Doctrinal Methodology

This study adopts a doctrinal research design that integrates textual analysis of Civil Code provisions with conceptual modeling to assess how existing legal categories accommodate smart contract features. The methodological framework proceeds through three interconnected layers. As detailed in Table 1, the doctrinal layer focuses on Civil Code provisions and draws upon legal texts to produce an interpretive framework. The comparative layer addresses functional equivalence, utilizing jurisdictional models to generate a characterization typology. The case-oriented layer examines dispute scenarios

through hypotheticals, yielding breach relief paths. Table 1 organizes these layers by method, analytical focus, data source type, and output, thereby structuring the triangulation of doctrinal, comparative, and case-oriented inquiry [12]. This design enables the systematic testing of legal categories against the distinctive technical attributes of smart contracts, including automatic execution and immutability. By mapping each methodological layer to a corresponding analytical output, the research design ensures that characterization and breach relief are examined as linked doctrinal problems rather than isolated questions. The resulting framework supports a coherent evaluation of whether Civil Code remedies remain functionally adequate when applied to self-executing contractual arrangements.

Table 1. Methodological Triangulation for Doctrinal Analysis

Method Layer	Analytical Focus	Data Source Type	Output
Doctrinal layer	Civil Code provisions	Legal texts	Interpretive framework
Comparative layer	Functional equivalence	Jurisdictional models	Characterization typology
Case-oriented layer	Dispute scenarios	Hypotheticals	Breach relief paths

3.2. Comparative and Functional Analysis Framework

To navigate the doctrinal disjunction between decentralized code and centralized legal frameworks, this study employs a comparative and functional analysis. The methodology proceeds in three distinct stages. First, the specific legal problem is isolated: identifying the precise point of failure when an automated smart contract transaction conflicts with statutory remedies for breach [7]. Second, a comparative mapping is conducted across major contract law traditions, including civil law and common law jurisdictions. Rather than focusing on formalistic categorizations or surface-level terminology, the analysis prioritizes functional equivalence. This approach evaluates how different legal systems address the underlying regulatory need, specifically the allocation of risk and the provision of relief, regardless of whether the solution is framed as a statutory right or a judicial doctrine. Finally, the findings from this comparative mapping are synthesized into a coherent doctrinal framework. This synthesis integrates the functional equivalents into the existing structure of the Civil Code, thereby constructing a unified path for breach relief that accommodates the unique operational characteristics of smart contracts while maintaining systemic legal coherence.

3.3. Case Scenario Construction and Relief Path Mapping

To systematically evaluate the remedial mechanisms available under the Civil Code, this study constructs a series of hypothetical case scenarios designed to simulate distinct categories of breach within smart contract environments. These scenarios are formulated to isolate variables such as the nature of the performance obligation, the feasibility of automated execution, and the occurrence of external disruptions. By mapping these breach triggers against the statutory relief framework, a structured relief-path model is developed to trace the logical progression from contractual violation to specific legal outcomes. This relief path map delineates the conditions under which an aggrieved party may pursue specific performance, claim damages, seek rescission, or demand restitution. The mapping process specifically accounts for the unique characteristics of code-based agreements, such as immutability and automated enforcement, to determine how traditional remedies are prioritized or adapted when technical execution diverges from legal expectation. This methodological approach ensures that the analysis remains grounded in practical application while rigorously testing the boundaries of existing legal doctrines [11].

4. Results

4.1. Characterization Typology of Smart Contracts

The doctrinal analysis yields a tripartite typology of smart contract characterization within the Civil Code framework. The unilateral code-based model treats code deployment as unilateral initiation, where coded execution is self-executing and any deviation from code constitutes breach, remedied through code-based reversal. The hybrid contractual model integrates negotiated terms with code, framing performance as mixed human-code interaction, breach as non-conforming performance, and relief through judicial remedies. The relational contract model, by contrast, subordinates code to the broader contractual relationship, invoking offer-acceptance rules, obligation performance standards, and the full spectrum of specific performance, damages, and rescission available under the Civil Code. As detailed in Table 2, the characterization matrix contrasts the two operational models (the unilateral code-based model and the hybrid contractual model) with the corresponding Civil Code position across four doctrinal dimensions. The columns identify Characterization Dimension, Unilateral Code-based Model, Hybrid Contractual Model, and Civil Code Implication. The rows address Formation, which treats code deployment as unilateral initiation in the unilateral code-based model, negotiated terms plus code in the hybrid contractual model, and offer-acceptance rules in the Civil Code; Performance, which treats coded execution as self-executing in the unilateral code-based model, mixed human-code in the hybrid contractual model, and obligation performance in the Civil Code; Breach, which treats deviation from code in the unilateral code-based model, non-conforming performance in the hybrid contractual model, and non-performance in the Civil Code; and Relief, which treats code-based reversal in the unilateral code-based model, judicial remedies in the hybrid contractual model, and specific performance, damages, and rescission in the Civil Code. This typology clarifies how characterization determines the applicable legal regime and the corresponding path to breach relief. The relational contract model corresponds to the Civil Code position and is therefore reflected in the Civil Code Implication column rather than shown as a separate column.

Table 2. Doctrinal Characterization Matrix for Smart Contracts

Characterization Dimension	Unilateral Code-based Model	Hybrid Contractual Model	Civil Code Implication
Formation	Code deployment as unilateral initiation	Negotiated terms plus code	Offer-acceptance rules
Performance	Coded execution as self-executing	Mixed human-code interaction	Obligation performance standards
Breach	Deviation from code	Non-conforming performance	Non-performance
Relief	Code-based reversal	Judicial remedies	Specific performance, damages, and rescission

4.2. Breach Scenarios and Trigger Conditions

The scenario analysis identified four primary breach categories. Coding errors, encompassing logical flaws and vulnerabilities in smart contract source code, represent the most direct trigger; because execution is deterministic, a coding defect invariably produces an unintended on-chain state that may satisfy the elements of non-performance or defective performance under the Civil Code. Oracle failure constitutes a second category, wherein erroneous or manipulated external data feeds cause the contract to execute on false premises, raising the question of whether the resulting transfer reflects the parties' true agreement. Unintended asset transfer, frequently arising from interface misconfiguration or address errors, produces irreversible dispositions that challenge conventional restitution doctrine. Changed legal circumstances, though exogenous to the code, may render continued performance unlawful or radically inequitable, invoking the doctrine of change of circumstances. Each scenario triggers distinct relief questions: coding errors and oracle failures primarily implicate defect liability and damages,

unintended transfers foreground restitution and unjust enrichment, and changed circumstances invoke judicial adjustment or termination. The critical analytical threshold lies in determining when automated execution ceases to be mere mechanical performance and crystallizes into a legally cognizable breach, thereby activating the relief pathways examined in subsequent sections.

4.3. Relief Path Mapping and Civil Code Fit

The relief path map links breach scenarios to Civil Code remedies through a structured mapping logic. Automatic reversal, triggered by coded conditions and enforced via blockchain execution, exhibits limited Civil Code fit due to immutability constraints. Specific performance, triggered by breach of a coded obligation and enforced by court order, demonstrates moderate fit, limited by practical enforceability. Damages, triggered by loss from non-performance and enforced through monetary award, show strong Civil Code fit, though constrained by valuation challenges. Rescission, triggered by fundamental breach and enforced by contract termination, also exhibits strong fit, limited by restitution complexity. As detailed in Table 3, the comparison across relief paths reveals that damages and rescission align most closely with Civil Code principles, while automatic reversal remains the least compatible. This mapping provides a structured framework for selecting appropriate remedies based on breach characteristics and enforceability considerations.

Table 3. Breach Relief Path Comparison

Relief Path	Trigger Condition	Enforcement Mechanism	Civil Code Fit	Limitation
Automatic reversal	Coded condition	Blockchain execution	Limited	Immutability
Specific performance	Breach of a coded obligation	Court order	Moderate	Enforceability
Damages	Loss from non-performance	Monetary award	Strong	Valuation
Rescission	Fundamental breach	Contract termination	Strong	Restitution

5. Discussion

5.1. Normative Tensions between Automation and Civil Code Principles

The architecture of smart contracts, characterized by deterministic execution and immutability, creates profound normative tensions with foundational Civil Code principles. Automated enforcement operates on a strictly binary logic that cannot account for the nuanced, context-dependent standards of good faith and fairness. This rigidity risks producing substantively unjust outcomes, particularly when unforeseen circumstances render performance excessively onerous or when the underlying contractual equilibrium is fundamentally disrupted. The Civil Code's remedy hierarchy, which prioritizes specific performance but allows for adaptation or termination under changed circumstances, stands in direct opposition to the self-executing and irreversible nature of code. Consequently, strict code-based enforcement may contravene the very equitable principles the legal system seeks to uphold. Resolving these tensions necessitates external intervention mechanisms, specifically judicial adjustment or mutual party agreement, to align automated outcomes with the substantive justice embedded within the Civil Code framework [5, 15].

5.2. A Layered Characterization Approach

A layered characterization approach resolves the tension between technological novelty and doctrinal stability by disaggregating the smart contract into distinct functional dimensions. At the formation and validity layer, the arrangement is characterized as an ordinary contract, such that established rules on capacity, consent, legality, and good faith govern its binding force [1]. At the execution layer, the self-

executing code operates as an automated performance mechanism, analogous to a payment or delivery system, which does not alter the underlying obligations but merely automates their discharge. At the relief layer, the deployed program constitutes an object of judicial adjustment, allowing courts to order modification, suspension, or unwinding where circumstances render continued performance inequitable. This tripartite structure preserves systemic coherence because each layer invokes the doctrinal apparatus most suited to the relevant legal question, avoiding the conceptual distortion that arises when a single monolithic characterization is imposed. It further clarifies the allocation of risk between parties and supports predictable judicial intervention [3].

5.3. Policy and Drafting Implications for Breach Relief

As detailed in Table 4, the normative integration of smart contracts into the Civil Code framework requires targeted adjustments across four functional dimensions. The table organizes these by integration issue, smart contract feature, governing Civil Code principle, and the corresponding adjustment path [18]. Formation, driven by automated code, engages the principle of consent and demands enhanced notice to ensure parties comprehend the binding logic. Performance, characterized by self-execution, implicates good faith and necessitates routine code audits to preserve fairness. Breach, where penalties apply automatically, must yield to the remedy hierarchy, permitting judicial override where coded outcomes diverge from statutory proportionality. Remedies, often executed through coded reversal, intersect with damages and rescission, requiring hybrid enforcement that couples on-chain execution with off-chain judicial supervision. Collectively, these paths counsel drafters to embed clear notice mechanisms, commission independent audits, and design modular dispute-resolution interfaces, while courts should treat automated outcomes as rebuttable presumptions subject to Civil Code remedies [14]. Statutory clarification of these integration points would reduce uncertainty and align private code with public normative expectations.

Table 4. Normative Integration Summary

Integration Issue	Smart Contract Feature	Civil Code Principle	Adjustment Path
Formation	Automated code	Consent	Enhanced notice to ensure parties comprehend the binding logic
Performance	Self-execution	Good faith	Routine code audits to preserve fairness
Breach	Automatic penalties	Remedy hierarchy	Judicial override where coded outcomes diverge from statutory proportionality
Remedies	Coded reversal	Damages and rescission	Hybrid enforcement coupling on-chain execution with off-chain judicial supervision

6. Conclusion

6.1. Summary of Findings

This study demonstrates that no single legal characterization fully captures the hybrid nature of smart contracts, as their self-executing and immutable features resist exclusive classification as either contracts or mere code. A layered approach, which treats the smart contract as a contract at the formation and validity layer, as an automated performance mechanism at the execution layer, and as an object of judicial adjustment at the relief layer, proves more coherent for addressing the distinct legal relationships arising from automated performance. Regarding breach relief, the analysis establishes that automated failures must be mapped onto the Civil Code's remedial framework, including specific performance, damages, and rescission, while accounting for the unique obstacles posed by irreversible execution. The findings further underscore that good faith and fairness operate as corrective mechanisms to mitigate the rigidity of algorithmic enforcement, ensuring that technological automation does not displace fundamental

contractual justice. Ultimately, the remedy hierarchy within the Civil Code provides a structured pathway for resolving disputes, prioritizing restoration of the aggrieved party's interests while preserving the efficiency gains that smart contracts offer.

6.2. Recommendations and Future Research

Legislative refinement should prioritize clarifying the temporal and spatial boundaries of automated formation, particularly the moment when a smart contract transitions from a mere technical protocol into a legally binding agreement. Statutory provisions ought to explicitly address the standards for reviewing automated performance and the allocation of liability when code executes in a manner diverging from the parties' original intent. Courts, in turn, should develop interpretive methodologies that respect the efficiency of automated execution while preserving the protective function of mandatory civil law rules, especially regarding unconscionability and good faith. For drafters, the recommendation is to embed robust breach relief mechanisms directly into contract architecture, including pause functions, upgrade pathways, and multi-signature arbitration triggers, so that remedial flexibility is not sacrificed to immutability.

Future research should pursue three interconnected trajectories. First, cross-border enforcement of smart contract judgments remains underexplored, particularly regarding the recognition of on-chain assets and the jurisdictional conflicts arising from decentralized node networks. Second, decentralized dispute resolution mechanisms warrant deeper theoretical and empirical scrutiny, especially their compatibility with due process guarantees and the enforceability of arbitral outcomes rendered through automated protocols. Third, empirical studies are needed to assess whether smart contracts actually reduce transaction costs and breach rates in practice, or whether they merely displace disputes to new arenas. Such research would ground normative proposals in observable outcomes and help calibrate the pace of legal reform to technological reality.

References

1. M. Kasatkina, "The interpretation of smart contracts in the EU and the USA," *International Comparative Jurisprudence*, vol. 7, no. 2, pp. 202–217, 2021.
2. M. Raskin, "The law and legality of smart contracts," *Geo. L. Tech. Rev.*, vol. 1, p. 305, 2016.
3. H. R. Jung, "Smart contracts and the future of civil contract law," *Journal for Civil and Criminal Law for Legislative Studies*, vol. 1, no. 3, pp. 12–17, 2025.
4. V. Marchenko and A. Dombrowska, "On determining the legal nature of smart contracts," in *International Conference on Economics, Law and Education Research (ELER 2021)*, Atlantis Press, Mar. 2021, pp. 175–179.
5. L. Efimova, O. Sizemova, and A. Chirkov, "Smart contracts: between freedom and strict legal regulation," *Information & Communications Technology Law*, vol. 30, no. 3, pp. 333–353, 2021.
6. E. A. Kirillova, V. V. Bogdan, I. B. Lagutin, and E. D. Gorevoy, "Legal status of smart contracts: features, role, significance," *Juridicas Cuc*, vol. 15, no. 1, pp. 285–300, 2019.
7. V. Varavka, "Smart contract as a form of civil contract," *Bull. Taras Shevchenko Nat'l U. Kyiv Legal Stud.*, vol. 111, p. 15, 2019.
8. F. Möselein, "Smart contracts and civil law challenges," in *Routledge Handbook of Financial Technology and Law*, Routledge, 2021, p. 27.
9. G. Varbanova, "Legal nature of smart contracts: contract or program code?," *Journal of Digital Technologies and Law*, vol. 1, no. 4, pp. 1028–1041, 2023.
10. J. J. Szczerbowski, "Place of smart contracts in civil law. A few comments on form and interpretation," in *Proceedings of the 12th Annual International Scientific Conference New Trends*, Nov. 2017.
11. M. Verstraete, "The stakes of smart contracts," *Loy. U. Chi. LJ*, vol. 50, p. 743, 2018.
12. J. M. Sklaroff, "Smart contracts and the cost of inflexibility," *U. Pa. L. Rev.*, vol. 166, p. 263, 2017.
13. M. Di Angelo, A. Soare, and G. Salzer, "Smart contracts in view of the civil code," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, Apr. 2019, pp. 392–399.
14. M. Durovic and A. Janssen, "The formation of blockchain-based smart contracts in the light of contract law," *European Review of Private Law*, vol. 26, no. 6, 2018.
15. A. Savelyev, "Contract law 2.0: 'Smart' contracts as the beginning of the end of classic contract law," *Information & Communications Technology Law*, vol. 26, no. 2, pp. 116–134, 2017.

16. S. Sudarto, "Smart contracts from a civil law perspective: Validity and implementation in Indonesia," *Jurnal Ilmiah Hukum Dirgantara*, vol. 15, no. 2, 2025.
17. M. K. Woebbeking, "The impact of smart contracts on traditional concepts of contract law," *J. Intell. Prop. Info. Tech. & Elec. Com. L.*, vol. 10, p. 105, 2019.
18. A. S. Khalifa and N. S. Samad, "Smart contracts and the challenges of conflict of laws in digital space," *Indonesian Journal of Law and Economics Review*, vol. 20, no. 4, pp. 10–21070, 2025.

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