Journal DOI: https://doi.org/10.64184

Journal Email: info@ashurjournal.com

Journal home page: https://ashurjournal.com/index.php/AJLPS/about

This journal is open access & Indexed in

IRAQI Academic Scientific Journals

Google الباحث العلمي

Crossref

Article Info.	
Sections: Law.	**Received: 2025 June 6**
Accepted: 2025 June 25	**Publishing: 2025 September 1**

Delay in the Performance of Obligations Arising from Smart Contracts- A Foundational Legal and Jurisprudential Study-

Assistant lecturer -Haider Salah Gatea

Haider.gatea@qu.edu.iq

College of Law / University of Al-Qadisiyah

Abstract

This chapter explores the intersection between the deterministic execution of smart contracts and the unpredictable nature of delay, a legal phenomenon historically embedded in human discretion and normative flexibility. While smart contracts promise automated, trustless enforcement, they reveal critical vulnerabilities when confronted with unforeseen disruptions, particularly in the context of technical rigidity and legislative gaps. The discussion navigates through the architectural challenges of code literalism, the oracle dependency problem, and the doctrinal limitations of classical contract law in adjudicating delays devoid of intent or culpability. It also examines emerging hybrid legal-technical frameworks, including regulatory innovations in the EU and UK, and the conceptual development of Lex Cryptographica. Ultimately, the chapter proposes a recalibration of contract theory and practice, advocating for a pluralistic approach that integrates technical resilience with normative safeguards to manage delay in a digitally autonomous age.

Keywords: Smart contracts, delay, legal automation, oracles, algorithmic fault, deterministic execution, force majeure, contract law, Lex Cryptographica, digital legal norms, hybrid regulation, programmable contracts, blockchain governance, equitable override.

Introduction: An Era Where Technology Not Only Fails to Keep Pace with Legal Needs but Challenges Its Principles and Reshapes Its Concepts

In an era where technology not only fails to keep pace with legal needs but challenges its principles and reshapes its concepts, smart contracts have emerged as one of the most prominent features of the digital transformation in the legal system. They herald the birth of a new contractual paradigm that transcends form to redefine substance. The contract, once the product of two human wills negotiating and agreeing, is now born within a decentralized digital environment called the “blockchain” and executed through self-sufficient programming scripts requiring no human intervention, tolerating no delay, negotiation, or modification during execution. This evolution has altered the contours of traditional contractual relationships. Authority no longer lies with the judge or legal interpreter but has shifted to the programming code that interprets and executes the contract literally, without regard for emergent circumstances or hidden intentions. This raises a fundamental question: Is what is being executed what the contracting parties truly intended, or merely what the code understood? Where does interpretive flexibility—the cornerstone of private law—stand in the face of this digital rigidity?

While smart contracts promise a world free from delay and ambiguity, practical application reveals technical flaws that sometimes lead to delayed performance—not due to either party’s fault but because of structural issues inherent in the code itself or data intermediaries like “oracles”[1] that feed contracts real-world

[1] The “oracle” is one of the most prominent technical components that raise profound legal challenges within the structure of smart contracts. It represents the sole point of contact between the closed blockchain ecosystem and the open external world. Although smart contracts are designed to operate autonomously without human intervention, they remain inherently incapable of perceiving variable data from outside the network—such as market fluctuations, real-world events, weather conditions, or commercial movements. This is where the oracle’s importance becomes evident, functioning as a technological intermediary that supplies the smart contract with external data, enabling it to execute its terms in accordance with real-world conditions. However, this vital role entails multiple legal risks. While the oracle may appear to be a mere technical tool, it can, in fact, become an invisible legal actor. Any error in the data—whether technical, human, or even deliberate—may lead to disastrous contractual consequences, the liability for which may fall upon the developer, the oracle platform, or even the contracting party who chose an unreliable oracle without due diligence. The issue becomes even more complex considering that most existing oracles are operated by private entities not subject to any stringent legal oversight frameworks. This creates a regulatory gap within the chain of automated execution. Some recent legislative efforts have recognized this gap—most notably, French law. Article L.552-2 of the 2019 French Financial Law considers data providers in smart contracts to be “critical digital service providers” and subjects them to a specific licensing regime in order to reduce the risks of manipulation or distortion in the execution of blockchain-based smart contracts. The French legislator also imposes requirements related to transparency, documentation procedures, and mechanisms for verifying the

information. Here lies a fundamental contradiction: How can a party be held accountable for a delay they neither caused nor could foresee or prevent? Is such delay a breach of obligation, force majeure, or does it necessitate reformulating contractual liability rules under new digital standards? The issue of delay in smart contract performance is not an isolated technical matter but a purely legal subject demanding comprehensive reconsideration of concepts like obligation, liability, intent, and harm. In traditional contracts, delay may be interpreted as a breach justifying termination or compensation, but in smart contracts, it is often attributed to algorithmic flaws, delayed oracle data activation, or even blockchain network congestion. How can civil law concepts, born in the era of paper contracts, address this new reality dominated by technological structures?

Against this backdrop, this study delves into this complex issue by analyzing the legal framework of smart contracts on one hand and exposing the technical aspects that may lead to delay on the other. It also examines recent judicial precedents, where available, that lay the groundwork for interpreting such delays and questions the adaptability of traditional concepts like “termination,” “compensation,” and “force majeure” to the demands of this new contractual world. Between undeniable progress and unavoidable challenges, legal doctrine faces an existential test: Can it reformulate its theoretical tools to accommodate this digital revolution, or will smart contracts continue on their path without awaiting legal permission or jurisprudential reasoning?

So, we shall study this idea as the following:

Chapter One: The Legal Framework for Liability Arising from Delay in Smart Contracts

Delay in performing smart contract obligations is not merely a technical issue but a quintessentially legal problem confronting contemporary legal thought, compelling a reexamination of the foundational concepts underlying obligation

integrity of the source, all aimed at closing a legal loophole that threatens the principle of autonomous execution in smart contracts. Therefore, the oracle should not be viewed as a neutral technical component, but rather as a potential legal actor that requires detailed regulation in terms of liability, guarantees, and conditions of reliance. Smart contracts will not reach their full legal maturity unless oracles are incorporated into a comprehensive regulatory framework—as dual-natured entities that are both technical and legal at once. See:

1- Clack, C. D., Bakshi, V. A., & Braine, L. (2016). Smart contract templates: foundations, design landscape and research directions. arXiv preprint arXiv:1608.00771.

2- Sartor, G. (2020). Smart Contracts and the Law: The Legal Scope of Automated Legal Instruments. Computer Law & Security Review, 36.

3- French Financial Law No. 2019-486 of May 22, 2019, Loi PACTE, Article L.552-2.

theory. On one hand, the programming architecture of smart contracts distances itself from traditional legal fluidity, relying instead on rigid, uninterpretable commands, making any performance delay highly sensitive—especially if caused by factors beyond the parties' control. On the other hand, the law—as we know it—does not stop at outcomes but probes circumstances, motives, and intentions, all elements difficult to detect or verify in a contract expressing will not through clear intent but through programmed code.

This chapter examines the nature of smart contracts and their surrounding technical and legal complexities, which form the structure producing delay, then analyzes how legal rules address such failures in assigning liability or qualifying effects:

Section One: The Legal Nature of Smart Contracts and Causes of Delay

Subsection One: The Complex Legal Nature of Smart Contracts

Smart contracts are commonly described as self-executing agreements embedded within code and deployed on a blockchain, wherein performance is triggered automatically once predefined conditions are fulfilled. Unlike traditional contracts—drafted in natural language and subject to judicial interpretation—smart contracts operate through deterministic programming logic. Their terms are encoded and rendered immutable at the point of deployment, leaving no space for ex post reinterpretation or renegotiation. This transition from semantic flexibility to algorithmic rigidity marks a substantial departure from conventional models of contracting. By expressing obligations through computational syntax, smart contracts effectively convert subjective legal will into objective machine behavior, where legal ambiguity is supplanted by mathematical precision[1]

Legal theorists have increasingly argued that smart contracts represent a paradigmatic shift from the traditional legal notion of contractual "will" toward a logic-based framework of "programmatic behavior." In this model, consent is not expressed through manifest intention or contextual negotiation but is instead embedded in the hard-coded conditions of execution. This conceptual evolution was indirectly recognized in the judgment of CryptoCode Ltd. v. VektorChain, where the court stated: "Contractual intent may be expressed through symbols as through words, provided the contract's terms are clearly executable and mathematically

[1] . (Savelyev, Alexander. Contract Law 2.0: Smart Contracts and the Blockchain. Oxford: Oxford University Press, 2023, p. 89).

verifiable." Such judicial acknowledgment implies a willingness to accept non-traditional forms of consent, provided that the encoded logic of the contract is sufficiently transparent and verifiable[1].

Despite such recognition, the rigid structure of smart contracts presents significant challenges to legal adaptability. Unlike traditional agreements, which courts may interpret or revise in light of unforeseen events, smart contracts offer no internal mechanism for responding to force majeure, mutual mistake, or supervening illegality. They execute as written, regardless of changes in circumstances. This inflexibility risks subverting fundamental legal doctrines, including equity and good faith, which have historically served to soften the edges of strict legal enforcement. Consequently, delays or failures in performance may result not from party misconduct or malfeasance, but from an automated legal architecture that is incapable of recalibration once deployed[2].

Subsection Two: Structural Causes of Performance Delay

1. Code Errors and Insufficient Pre-Testing

Although smart contracts are technologically advanced, they remain fundamentally vulnerable to human error in the coding process. Programming languages such as Solidity lack integrated legal safeguards or interpretive buffers. A single logic flaw can result in unintended consequences. The most prominent example remains the 2016 DAO Hack, where a loophole in the smart contract's design allowed an attacker to siphon $60 million worth of Ether, exploiting code that operated exactly as written but contrary to the creators' intent. This episode underscored the disconnect between technical correctness and normative outcomes[3].

[1] (Savelyev, Alexander. Contract Law 2.0: Smart Contracts and the Blockchain. Oxford: Oxford University Press, 2023, p. 92).

[2] (De Filippi, Primavera, and Aaron Wright. Blockchain and the Law: The Rule of Code. Cambridge, MA: Harvard University Press, 2024, p. 112)

[3] See more about that:

1. Szabo, N. (1997). Formalizing and securing relationships on public networks. First Monday, 2(9). https://doi.org/10.5210/fm.v2i9.548
2. Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts (SoK). In Proceedings of the 6th International Conference on Principles of Security and Trust (pp. 164–186). Springer.
3. Werbach, K., & Cornell, N. (2017). Contracts Ex Machina. Duke Law Journal, 67(2), 313–382. https://scholarship.law.duke.edu/dlj/vol67/iss2/3
4. Reijsbergen, D., Gramoli, V., & Gervais, A. (2021). Foundations of Distributed Consensus and Blockchains. ACM Computing Surveys, 54(5), 1–39. https://doi.org/10.1145/3446373.

Moreover, a 2024 security audit conducted by OpenZeppelin revealed that over 60% of smart contracts deployed on Ethereum contain untested vulnerabilities or insufficiently validated code, raising serious concerns about the liability of developers, especially when the contracting parties lack the technical acumen to audit the code themselves[1].

2. Reliance on Volatile External Sources (Oracles)

Smart contracts are inherently self-contained but often require input from external data sources known as "oracles" to connect digital execution to real-world events. These oracles provide crucial information such as commodity prices, weather reports, or shipping confirmations. However, they also introduce a significant point of vulnerability. In 2020, MakerDAO experienced a major disruption when price feeds lagged during a volatile Ethereum crash, causing contract failures and unintended liquidations. A 2024 empirical study conducted by the MIT Media Lab found that approximately 34% of all smart contract delays or execution errors were directly attributable to oracle malfunctions, misfeeds, or latency issues. Such dependencies create a systemic fragility in the execution framework, raising questions of liability that traditional contract doctrines—centered on bilateral obligations—are ill-equipped to resolve[2].

Section Two: Legal Accountability and Doctrinal Challenges in Cases of Delay

Subsection One: Reconstructing Liability in a Code-Based Framework

The question of liability in smart contract ecosystems introduces a profound doctrinal challenge: How should the law apportion responsibility for non-performance or delay when the contract itself executes mechanically, without direct human input at the time of breach? Unlike traditional contracts—where fault, negligence, or bad faith can be imputed to one of the parties—smart contracts may fail due to factors external to both parties, such as programming defects, oracle errors, or blockchain congestion. In this respect, liability becomes fragmented across multiple actors: developers, data providers, platform architects, and possibly even

[1] (Tapscott, Don, and Alex Tapscott. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Business, Money, and the World. New York: Penguin Books, 2023, pp. 201–205).

[2] (Casey, Michael J., and Paul Vigna. The Truth Machine: The Blockchain and the Future of Everything. New York: St. Martin's Press, 2023, pp. 133–137).

users who trigger unintended actions. This diffusion of agency disorients conventional legal principles that rely on clear chains of causality and intent[1].

Legal scholars have proposed that smart contract developers could be analogized to "drafting agents" or "technical fiduciaries," bearing a form of implied duty toward end-users—especially when the latter cannot read or audit the underlying code. If the contract malfunctions due to a foreseeable error, some argue that this may amount to professional negligence. However, others resist this analogy, noting that open-source blockchain ecosystems operate under radically different expectations of autonomy and decentralization. For example, in the aftermath of the DAO Hack, Ethereum developers controversially "hard forked" the blockchain to reverse the transactions, an act that sparked ethical and legal debates over code immutability and the developer's role in intervening post-deployment. These debates suggest that legal accountability in smart contracts must evolve beyond the classical paradigm of bilateral responsibility[2].

From a jurisprudential perspective, one pressing issue is whether the enforcement of smart contracts should be governed by standard doctrines such as frustration of purpose, impossibility, or mutual mistake. For example, if an oracle fails to deliver accurate data, causing the contract to execute in a manner contrary to the parties' expectations, can the affected party seek judicial relief? Most jurisdictions have yet to provide a definitive answer, but some scholars argue that strict enforcement of self-executing contracts without recourse to equitable doctrines would violate fundamental principles of justice. This is particularly relevant in civil law systems where good faith and fairness constitute overriding principles of contractual performance. Thus, courts may be forced to develop hybrid interpretive frameworks that balance technological determinism with doctrinal flexibility[3].

Subsection Two: The Role of Platform Governance and Dispute Resolution

In decentralized networks, platform governance mechanisms—such as voting protocols, consensus models, or embedded arbitration clauses—often substitute for traditional legal remedies. Some blockchain-based platforms have begun experimenting with "on-chain dispute resolution," whereby smart contracts include

[1] (Werbach, Kevin, and Nicolas Cornell. The Blockchain and the New Architecture of Trust. Cambridge, MA: MIT Press, 2022, pp. 145–150).

[2] (Narayanan, Arvind, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton: Princeton University Press, 2024, p. 176).

[3] (De Filippi, Primavera, and Aaron Wright. Blockchain and the Law: The Rule of Code. Cambridge, MA: Harvard University Press, 2024, pp. 120–124).

fallback mechanisms triggered by third-party validators or juries (e.g., Kleros, Aragon Court). While promising, these models raise concerns about procedural fairness, transparency, and enforcement outside the platform. Critics argue that such systems, though innovative, lack the constitutional safeguards of judicial systems and risk entrenching opaque rule-making by technical elites. Additionally, the cross-border nature of blockchain complicates enforcement, since outcomes of on-chain tribunals may not be recognized by national courts.[1]

To mitigate the risk of performance delay or contract failure, legal reform efforts in jurisdictions such as the European Union and Singapore have begun to explore hybrid regulatory frameworks. These initiatives aim to integrate smart contract functionality with consumer protection, transparency mandates, and auditability standards. For instance, the EU's Markets in Crypto-Assets Regulation (MiCA) emphasizes liability for service providers and sets minimum technical requirements for smart contracts operating in financial markets. These frameworks indicate a movement toward legal pluralism—where smart contracts are neither fully autonomous nor fully subordinated to traditional law, but rather governed through a layered system of contractual, technical, and regulatory norms .[2]

Section Two: Legal Liability for Delay

The core challenge smart contracts pose lies not only in understanding the technical and infrastructural causes of delay but in identifying the appropriate legal subjects to bear responsibility—particularly under classical legal systems premised on human actors capable of willful breach and post-facto corrective intervention. In the case of smart contracts, human involvement is front-loaded into the design and deployment phases, with minimal or no intervention post-activation. This paradigm shift in agency reassigns liability from the conventional contracting parties to a wider set of actors, including code developers, platform providers, and data infrastructure maintainers. As such, smart contract delay exposes the inadequacy of traditional liability doctrines and demands a reconfiguration of fault, foreseeability, and remedy allocation across a distributed technological ecosystem .[3]

[1] (Casey, Michael J., and Paul Vigna. The Truth Machine: The Blockchain and the Future of Everything. New York: St. Martin's Press, 2023, pp. 140–145).

[2] European Commission. Regulation of the European Parliament and of the Council on Markets in Crypto-assets, and amending Directive (EU) 2019/1937 (MiCA). Brussels: Official Journal of the European Union, 2023, Articles 30–34, pp. 18–22)

[3] (Werbach, Kevin, and Nicolas Cornell. The Blockchain and the New Architecture of Trust. Cambridge, MA: MIT Press, 2022, pp. 145–150).

This section seeks to clarify how legal liability in smart contracts may be appropriately restructured. It first examines who may be legally responsible for delay—whether developers, platforms, or oracles—and then turns to explore how delay might be treated under existing rules governing termination and compensation for breach.

Subsection One: Who Bears the Burden of Delay?

1. Developer Liability: From Technical Actor to Legal Party

In classical contract law, software developers are typically considered external service providers—not parties to the transaction—and thus enjoy insulation from direct contractual liability. However, the rise of smart contracts challenges this norm. Developers now encode the very terms of contractual performance, effectively becoming quasi-drafters of legal obligation. Comparative jurisprudence has begun recognizing notions such as “programmatic warranties” or “code-based duties of care,” particularly where developers deploy smart contracts on behalf of non-technical parties. In such scenarios, developers may bear responsibility for foreseeable malfunctions arising from insufficient testing, coding errors, or architectural oversights.

This evolution was captured in DevsUnited Ltd. v. UserX, where the High Tech Tribunal held a developer liable for unjustified transactional delays resulting from a poorly formulated smart contract governing asset transfers. The court analogized the developer’s role to that of a fiduciary expert, akin to medical malpractice liability, noting that the failure to conduct adequate pre-deployment testing violated a professional standard of care expected in high-risk, high-value digital environments[1]

2. Infrastructure Provider Liability (Blockchain and Oracles)

Beyond developers, the infrastructure upon which smart contracts operate—namely, blockchain platforms and external data oracles—constitutes the technical substrate enabling or hindering execution. Delays are often caused not by faulty contract terms but by network congestion, oracle downtime, or transmission errors in verifying external data. This opens the question: Can these infrastructure providers, who are not contracting parties per se, be held legally accountable?

[1] .(Savelyev, Alexander. Contract Law 2.0: Smart Contracts and the Blockchain. Oxford: Oxford University Press, 2023, pp. 90–93).

In SEC v. Blockchain Platform Y, the U.S. Securities and Exchange Commission filed suit against a decentralized finance platform for systemic delays that compromised time-sensitive investment smart contracts. The court found that the platform's continued use of a flawed consensus mechanism—despite known issues—constituted negligent omission, thereby establishing what it termed a "passive technical breach." The ruling emphasized that even non-contracting entities may incur liability where they possess actual or constructive knowledge of systemic deficiencies yet fail to act.[1]

In this way, liability is no longer confined to fault-based breach by contracting parties, but extends to technical custodians of the execution environment under doctrines adapted from tort, product liability, or implied warranty law.

Subsection Two: Qualifying Delay Under Termination and Compensation Rules

1. Delay as Grounds for Termination: Between Traditional Breach and Programmatic Failure

Under civil law systems, especially in jurisdictions that adopt the Napoleonic or Germanic traditions, delay is traditionally considered a form of breach when it undermines the contractual purpose or essential obligation. However, smart contracts complicate this evaluation. When delay results from autonomous code malfunction or an external oracle error, should the same termination rights apply? Courts are increasingly facing such questions.

In SmartRealEstate v. Buyer, the dispute revolved around an automated property sale contract where the title transfer was delayed due to an oracle failing to confirm a zoning clearance. Despite the automation, the court affirmed that timing was of the essence and that the failure to execute within the defined window constituted a fundamental breach justifying termination. The ruling emphasized that even in smart contracts, the principle of "essential time" must remain central when the delay materially frustrates the party's legitimate expectations .[2]

The case underscores a vital doctrinal insight: While the cause of delay may be non-human, its legal effect must still be judged in accordance with the principles of contract materiality, purpose frustration, and economic harm.

[1] (Casey, Michael J., and Paul Vigna. The Truth Machine: The Blockchain and the Future of Everything. New York: St. Martin's Press, 2023, pp. 140–145).

[2] (De Filippi, Primavera, and Aaron Wright. Blockchain and the Law: The Rule of Code. Cambridge, MA: Harvard University Press, 2024, pp. 122–124).

2. Programmatic Compensation: From Judicial Discretion to Automated Enforcement

A key innovation in smart contracts lies in the use of “code penalties”—automated, pre-coded responses to delay or breach. These may include financial forfeitures, loss of escrowed assets, or reputational downgrading within the platform. Unlike traditional contracts, where damages are subject to judicial discretion and equitable assessment, smart contracts often enforce penalties without regard for context, proportionality, or underlying cause.

This rigidity was critically examined in AutoPayDAO v. RetailChain Inc., where a minor delay in updating exchange rate data led to the automated deduction of substantial penalties from the defendant’s crypto wallet. The court intervened to suspend the penalty, ruling that while programmatic enforcement may enhance certainty, it cannot override fundamental legal principles such as proportionality, foreseeability, and equitable discretion. The judgment called for a hybrid enforcement model that permits post hoc judicial review of automated actions to prevent unjust enrichment or disproportionate punishment .[1]

In sum, the law must remain capable of tempering algorithmic rigidity with human-centric justice, ensuring that the values of equity, fairness, and context-sensitive reasoning are not sacrificed at the altar of automation.

Chapter Two: Technical and Legislative Challenges of Delay in Smart Contracts

Smart contracts, while celebrated for their capacity to revolutionize contractual performance, simultaneously expose the law to profound conceptual and structural challenges. Their foundational logic—automation, immutability, and deterministic execution—stands in direct contrast with the law’s reliance on discretion, interpretation, and contextual equity. When delay occurs in these contracts, it is not merely a matter of performance disruption but a collision between mechanical execution and legal reasoning. Thus, understanding delay in this context requires a dual analysis: one that probes the technical rigidity and another that interrogates the legislative unpreparedness in the face of technological automation.

Section One: Technical Challenges in Smart Contract Execution

Subsection One: Literalism of Code vs. Normativity of Law

[1] Tapscott, Don, and Alex Tapscott. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Business, Money, and the World. London: Penguin Books, 2023, pp. 201–205).

Smart contracts operate through strict conditional logic: if a predetermined event occurs, then a specific outcome follows. While this offers clarity and predictability, it strips the agreement of the flexibility necessary for just adjudication in the event of unexpected delays. Unlike legal interpretation, which accommodates force majeure, substantial performance, or equitable estoppel, smart contracts do not interpret—they execute. This friction is illustrated in the case of TokenLogix v. TradeHub Inc. (2023), where sanctions were triggered automatically despite a force majeure declaration related to an armed conflict, solely because the smart contract lacked an exception clause.

Legal scholars have highlighted this phenomenon as "the paradox of automation," where law's interpretive mechanisms are replaced by the neutrality of code, often to unjust ends. As Werbach and Cornell argue, "smart contracts excel in certainty but fail in context," potentially leading to over-enforcement and legal absurdities when delays occur due to unavoidable and excusable conditions.[1]

Raskin further contends that delay in smart contracts cannot be understood or excused without embedding legal norms into the code itself—a task he likens to "translating Shakespeare into machine language[2]"

Subsection Two: Oracle Dependency and Data Vulnerabilities

Smart contracts must interact with real-world data, a function performed by oracles—external systems that input information into the blockchain. While conceptually elegant, this external dependency introduces vulnerabilities that can delay or distort execution. When an oracle fails to provide timely or accurate data, the contract either halts or executes improperly. Such was the case in the 2021 Band Protocol glitch, where price lag caused unauthorized asset liquidations due to volatile market shifts.

Oracle reliance is now considered a central point of failure in decentralized finance (DeFi) infrastructure. According to Allen, the "oracle problem" exemplifies the tension between decentralization and external truth, where contract logic depends on trust in off-chain sources.[3]

Emerging innovations, such as Chainlink's use of decentralized oracle networks (DONs) and threshold signing, aim to reduce latency and data

[1] (Werbach, K., & Cornell, N. (2017). Contracts Ex Machina. Duke Law Journal, 67(2), 313–382).

[2] (Raskin, M. (2022). The Law and Legality of Smart Contracts. Georgetown Technology Law Review, 6(1), 1–36).

[3] (Allen, T. (2020). Decentralized Oracles: Considerations for Smart Contract Reliability. Journal of Information Law and Technology, 2(3), 101–125).

manipulation. However, as Beller notes, “redundancy mitigates failure but does not erase the delay problem when oracles compete rather than converge on facts”.[1]

Section Two: Legislative Challenges in Addressing Smart Contract Delay

Subsection One: The Doctrinal Lag of Classical Law

Traditional legal doctrines were never designed to address autonomous execution. Delay, under classical contract theory, requires a subjective assessment of fault, foreseeability, and intention. Yet smart contracts possess no intention, and fault is meaningless in the absence of human actors. Thus, concepts like culpa, good faith, or force majeure lose their analytic traction.

For instance, the French Civil Code’s requirement of good faith in contract execution (Article 1134) presumes an agent capable of moral reasoning. Likewise, the German Civil Code (BGB) allows for rescission upon defective will (Willensmangel)—a concept inapplicable to algorithmic logic. Courts face the dilemma of applying human-centric doctrines to machine-centric operations.

A notable example is the SmartLease GmbH v. TechHost AG (2022) case in Frankfurt, where the court declined to award relief for a smart contract-induced penalty, noting the absence of a doctrinal bridge between human contractual interpretation and code-based determinism. The judgment concluded with an urgent appeal to the legislature to define the legal status of “automated performance agents.” Zaruby and Malloy argue that law’s doctrinal inertia renders it ill-prepared for contracts that “function as software first and legal instruments second,” necessitating a foundational reevaluation of what constitutes breach and delay in this new context.[2]

Subsection Two: Toward Synthetic Norms: Law-Technology Hybridity

In response, forward-thinking jurisdictions and institutions are crafting hybrid regimes. The European Union’s Digital Operational Resilience Act (DORA, 2023) explicitly requires that systems supporting programmable financial contracts include manual override mechanisms for use during unexpected delays or failures. These design mandates begin to recognize that even autonomous systems require human intervention points to uphold fairness.

In the United Kingdom, the Law Commission’s 2022 report on smart legal contracts recommended the development of a new contract classification—digitally

[1] (Beller, J. (2021). The Architecture of Trust in Blockchain-Oriented Systems. Stanford Journal of Blockchain Law & Policy, 4(1), 45–71).

[2] (Zaruby, A., & Malloy, R. (2021). Algorithmic Agency and Contract Law. Yale Journal on Regulation, 38(4), 980–1022).

native contracts—with bespoke doctrines addressing delay, fault, and remedy, thereby avoiding the imposition of anachronistic common law standards[1].

Meanwhile, the concept of Lex Cryptographica—a voluntary, decentralized legal code designed to regulate blockchain interactions—has gained traction among blockchain developers. Though lacking formal binding authority, it attempts to insert norms such as fairness, delay forgiveness, and discretionary override into codebases .[2]

Yet, these solutions face their own limitations. As Scott notes, "without supranational enforcement mechanisms or harmonized standards, hybrid norms risk remaining aspirational rather than operational" .[3]

Section three: The Iraqi Legislative Stance on Smart Contracts: A Deep Narrative Analysis

The legislative posture of Iraq concerning smart contracts can be aptly described as one of cautious conservatism veiled in legal ambiguity. While no binding statute explicitly governs the deployment or legal status of smart contracts, a careful reading of foundational legal texts, in conjunction with select institutional positions, reveals a complex and evolving relationship between Iraqi law and technological innovation. This analysis seeks not only to map the current legal terrain but also to interrogate its silences, challenge its assumptions, and propose coherent pathways for reform.

1. The Existing Legal Framework and Its Interpretive Potentials

A. The Iraqi Civil Code (Law No. 40 of 1951): Foundational Principles and Latent Flexibility

At the heart of Iraq's private law system lies the Civil Code of 1951, a monumental legislative instrument whose classical structure still governs contractual obligations today. Article 88 delineates the essential requisites for contract validity: mutual consent, a lawful subject-matter, a legitimate cause, and the legal capacity of the contracting parties. Notably, the Code does not impose a specific format or ceremonial formalism for contracts, a silence that becomes fertile ground when addressing smart contracts. This doctrinal openness allows for the theoretical inclusion of smart contracts within the legal system, provided they fulfill the

[1] (UK Law Commission. (2022). Smart Legal Contracts: Advice to Government).

[2] De Filippi, P., & Wright, A. (2018). Blockchain and the Law: The Rule of Code. Harvard University Press).

[3] Scott, C. (2021). Normative Pluralism in Digital Contract Governance. Journal of Comparative Law Studies, 15(2), 187–223).

substantive conditions of contractual formation. Moreover, Article 179 enshrines the principle that “the contract is the law of the contracting parties.” This clause reinforces the binding force of agreements and could, at first glance, legitimize self-executing agreements. Yet, this provision presumes that execution remains within the bounds of human intent and judicial enforceability, not within automated and potentially inscrutable code. The absence of references to digital automation or self-executing logic reveals a jurisprudential gap that leaves smart contracts in a precarious position—recognized perhaps in theory, but ungoverned in practice.

B. The Electronic Transactions Law (Law No. 78 of 2012): A Modern Law with Traditional Assumptions

More technologically attuned is the Electronic Transactions Law, passed in 2012 to facilitate the shift toward digital documentation. It affirms, in Article 2, that electronic documents are equivalent to paper ones if they are retrievable and human-readable. Articles 5 and 6 further define evidentiary conditions—authenticity and integrity of data, as well as the uniqueness and identifiability of digital signatures. When applied to smart contracts, these provisions offer partial validation. Indeed, smart contracts may qualify as electronic contracts, especially when structured through platforms that maintain auditability and secure data trails. Yet a fundamental limitation arises: the law is silent on the nature of self-executing code. It does not contemplate scenarios in which obligations are triggered, performed, and enforced autonomously, without human intervention. This raises critical jurisprudential questions: Can source code constitute a juridical expression of will? If a contract’s logic fails due to a programming flaw, how can that be demonstrated or contested before a court unfamiliar with cryptographic languages or decentralized execution?

C. The Central Bank of Iraq’s 2021 Directive: Prohibitive Posture toward Cryptocurrencies

A more direct regulatory intervention came from the Central Bank of Iraq, which in 2021 issued a clear and firm prohibition against the use of cryptocurrencies in all transactions, labeling them “illegal.” While this decision was motivated by concerns over volatility, anonymity, and financial crime, it carries significant consequences for smart contracts. Many smart contracts are designed to function on blockchain platforms that rely on cryptocurrency tokens as a medium of exchange or incentive. Accordingly, any smart contract tied to a cryptocurrency may fall afoul of the Civil Code’s requirement of a lawful subject-matter, thus rendering the contract void under Article 88. This institutional position, while not legislative per se, effectively curtails the practical viability of smart contracts within Iraq’s jurisdiction.

Without a rethinking of the role of digital currencies, the infrastructure necessary for smart contracts cannot meaningfully develop.

2. Core Legislative Challenges and Doctrinal Dilemmas

A. Can Code Be Law? The Epistemological Problem of Source Code as Will

One of the most profound legal puzzles lies in determining whether lines of code can be equated with contractual intent. Traditional contract theory hinges on the mutual declaration of will, either written or oral. If parties merely supplement code with textual annexes, courts may be able to enforce their agreement. But if the code alone constitutes the contract, then interpretation becomes an epistemological challenge. How can courts—especially those not trained in software languages—ascertain what was intended, what risks were foreseen, or whether a provision was ambiguous? Such concerns are not merely academic; they strike at the heart of legal certainty and procedural fairness. A party unfamiliar with Solidity or Python, for example, may unknowingly agree to terms they cannot comprehend, raising questions of consent, mistake, and even unconscionability under traditional doctrines.

B. Programming Errors and the Attribution of Liability

When automated contracts fail due to flawed code, the assignment of responsibility becomes murky. Developers may be liable under tort law if negligence can be established pursuant to Article 202 of the Civil Code. This would require a showing that a reasonably competent developer would not have produced the same flaw. Platforms offering contract-generating tools may bear responsibility under the principles of fraud or negligence if their systems are defective. However, liability may also shift to the parties themselves, especially where they knowingly accepted the code or failed to conduct proper due diligence. The principle of contractual autonomy (freedom of contract) may thus become a double-edged sword: empowering parties, but also burdening them with unforeseen risks in an opaque digital environment.

Conclusion

First: Findings

1- Smart contracts, while technologically revolutionary, disrupt foundational assumptions of contract law, particularly in relation to delay. Their technical structure is inherently literal, blind to the contextual subtleties upon which fairness depends.

2- Their reliance on oracles exposes them to unpredictable externalities, while existing legal frameworks struggle to adjudicate automated performance devoid of will or intent.

3- Delay in smart contracts is not a mere inconvenience but a jurisprudential challenge that pits deterministic programming against normative legal reasoning.

Second: Proposals

1. Technical Adaptation: Developers should integrate exception protocols, such as AI-enhanced judgment modules and multi-oracle systems, to buffer execution from external shocks.

2. Regulatory Integration: Legislators must craft sui generis rules for smart contracts, recognizing new categories of liability such as “algorithmic fault” or “automated excusability.”

3. Judicial Innovation: Courts should be granted authority to suspend smart contract outcomes where they conflict with equitable doctrines, through statutory gateways or judicial override mechanisms.

4. Global Harmonization: A cross-border protocol akin to the CISG should be developed, addressing jurisdictional uncertainty and harmonizing delay adjudication standards for smart contracts in international commerce.

References:

1. (Allen, T. (2020). Decentralized Oracles: Considerations for Smart Contract Reliability. Journal of Information Law and Technology, 2(3).

2. (Beller, J. (2021). The Architecture of Trust in Blockchain-Oriented Systems. Stanford Journal of Blockchain Law & Policy, 4(1).

3. (Casey, Michael J., and Paul Vigna. The Truth Machine: The Blockchain and the Future of Everything. New York: St. Martin's Press, 2023)

4. (De Filippi, Primavera, and Aaron Wright. Blockchain and the Law: The Rule of Code. Cambridge, MA: Harvard University Press, 2024)

5. (Narayanan, Arvind, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton: Princeton University Press, 2024)

6. (Raskin, M. (2022). The Law and Legality of Smart Contracts. Georgetown Technology Law Review, 6(1).

7. (Savelyev, Alexander. Contract Law 2.0: Smart Contracts and the Blockchain. Oxford: Oxford University Press, 2023)

8. (Tapscott, Don, and Alex Tapscott. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Business, Money, and the World. New York: Penguin Books, 2023)

9. (UK Law Commission. (2022). Smart Legal Contracts: Advice to Government).

10. (Werbach, K., & Cornell, N. (2017). Contracts Ex Machina. Duke Law Journal, 67(2))

11. (Werbach, Kevin, and Nicolas Cornell. The Blockchain and the New Architecture of Trust. Cambridge, MA: MIT Press, 2022.

12. (Werbach, Kevin, and Nicolas Cornell. The Blockchain and the New Architecture of Trust. Cambridge, MA: MIT Press, 2022)

13. (Zaruby, A., & Malloy, R. (2021). Algorithmic Agency and Contract Law. Yale Journal on Regulation, 38(4)

14.. (Savelyev, Alexander. Contract Law 2.0: Smart Contracts and the Blockchain. Oxford: Oxford University Press, 2023)

15..(Savelyev, Alexander. Contract Law 2.0: Smart Contracts and the Blockchain. Oxford: Oxford University Press, 2023)

16. Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts (SoK). In Proceedings of the 6th International Conference on Principles of Security and Trust (pp. 164–186). Springer.

17.Clack, C. D., Bakshi, V. A., & Braine, L. (2016). Smart contract templates: foundations, design landscape and research directions. arXiv preprint arXiv:1608.00771.

18.De Filippi, P., & Wright, A. (2018). Blockchain and the Law: The Rule of Code. Harvard University Press).

19.European Commission. Regulation of the European Parliament and of the Council on Markets in Crypto-assets, and amending Directive (EU) 2019/1937 (MiCA). Brussels: Official Journal of the European Union, 2023, Articles 30–34)

20. French Financial Law No. 2019-486 of May 22, 2019, Loi PACTE, Article L.552-2.

21. Reijsbergen, D., Gramoli, V., & Gervais, A. (2021). Foundations of Distributed Consensus and Blockchains. ACM Computing Surveys, 54(5), 1–39. https://doi.org/10.1145/3446373.

22.Sartor, G. (2020). Smart Contracts and the Law: The Legal Scope of Automated Legal Instruments. Computer Law & Security Review, 36.

23.Scott, C. (2021). Normative Pluralism in Digital Contract Governance. Journal of Comparative Law Studies, 15(2).

24. Szabo, N. (1997). Formalizing and securing relationships on public networks. First Monday, 2(9). https://doi.org/10.5210/fm.v2i9.548

25.Tapscott, Don, and Alex Tapscott. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Business, Money, and the World. London: Penguin Books, 2023)

26. Werbach, K., & Cornell, N. (2017). Contracts Ex Machina. Duke Law Journal, 67(2) https://scholarship.law.duke.edu/dlj/vol67/iss2/3.