

# **Why “Code Is Law” Fails for Real-World Assets**

*Governance Gaps in Tokenized Markets*

Chung Hei Sing

*December 2025*

# Why “Code Is Law” Fails for Real-World Assets

## *Governance Gaps in Tokenized Markets*

### **Abstract**

The proposition that “code is law” has been central to the development of blockchain-based markets, promising automated enforcement, reduced transaction costs, and trustless coordination. While this paradigm has shown partial success in digital-native environments, its application to real-world assets (RWAs) has proven far more problematic. Despite increasing experimentation with tokenization, institutional adoption remains limited.

This paper argues that the primary barrier is not technological maturity, but a structural mismatch between on-chain execution and off-chain governance, legal authority, and fiduciary responsibility. Real-world assets embed discretion, enforceability, and institutional accountability that cannot be fully specified ex ante in code. Approaching the problem from an institutional adoption perspective, the paper reframes tokenization as an institutional design challenge rather than a software problem, with implications for policymakers, regulators, and institutional investors.

**Keywords:** Real-world asset tokenization; institutional governance; fiduciary duty; smart contracts; incomplete contracts; market infrastructure; institutional adoption; blockchain regulation; financial market design.

**How to cite this paper:** Sing, C. H. (2025). Why “Code Is Law” Fails for Real-World Assets: Governance Gaps in Tokenized Markets. Working paper.

## **1. Introduction**

The principle that “code is law” has become a foundational idea in blockchain discourse. At its strongest, the phrase captures a simple insight: software systems embed rules that constrain behavior, often more effectively than formal legal mechanisms. In purely digital environments, automated execution can substitute for enforcement, and technical design can meaningfully shape economic outcomes.

Yet as tokenization efforts move beyond native digital assets and into real-world assets, this principle begins to break down. Projects attempting to extend smart-contract logic to financial securities, infrastructure assets, and regulated instruments have repeatedly encountered friction—not primarily from technological limitations, but from institutional resistance. These frictions are often described as transitional problems: regulatory uncertainty, immature infrastructure, or conservative incumbents. This paper argues otherwise.

The failure of “code is law” in real-world asset tokenization is structural. Real-world assets are embedded in legal, institutional, and fiduciary frameworks that cannot be replaced by automated execution without undermining the very properties that make those assets investable at scale. Governance, discretion, and accountability are not residual inefficiencies waiting to be engineered away; they are constitutive features of institutional markets.

This paper examines why the logic of autonomous execution, while effective in narrow digital contexts, fails when applied to assets whose value depends on enforceability, judgment, and ex post intervention. It argues that successful tokenization requires a reframing: from software replacing institutions to software operating within them.

## **2. The Origins and Limits of “Code Is Law”**

The phrase “code is law” originated as an observation about how technical architectures regulate behavior. In its original formulation, it was descriptive rather than prescriptive: software systems constrain what users can do, shaping outcomes in ways analogous to legal rules. Over time, however, this observation was transformed—particularly within blockchain communities—into a normative claim that code should function as law itself, displacing legal institutions rather than complementing them.

This shift was reinforced by early blockchain successes. Native digital assets operate in environments where ownership is purely informational, disputes are internalized, and participation is voluntary. Losses from bugs, exploits, or poor design are often treated as part of the system’s risk model rather than as governance failures. In these settings, automated execution can plausibly serve as a primary organizing principle.

The problem arises when this logic is extrapolated beyond its original domain. Real-world assets do not exist in closed systems. They are governed by contracts, regulated by public authorities, and held by institutions with fiduciary obligations. Ownership claims rely on legal enforceability,

not merely on cryptographic proof. Participants in these markets cannot waive recourse to courts or regulators, nor can they outsource responsibility to code.

The prescriptive interpretation of “code is law” therefore rests on a category error. It assumes that the absence of legal recourse is a feature of efficiency rather than a condition tolerated only under specific circumstances. When applied to real-world assets, the metaphor ceases to describe reality and instead obscures the sources of authority and accountability that institutions depend on.

In this sense, the failure is not one of implementation but of framing. Treating code as a substitute for law may simplify system design, but it does so by ignoring the institutional conditions under which real-world assets are created, valued, and governed.

### **3. Why Real-World Assets Resist Full Codification**

At the core of this mismatch lies the problem of incomplete contracts. Many real-world assets cannot be fully specified *ex ante*. Uncertainty, complexity, and the cost of foresight ensure that some contingencies will remain undefined. Financial instruments, infrastructure projects, and regulated securities are designed with this reality in mind. Their governance structures allocate discretion precisely because not all future states of the world can be anticipated.

Value in these assets depends not only on mechanical execution but on the allocation of residual control rights—the authority to interpret, intervene, and adjust when circumstances deviate from expectation. This discretion is not a design flaw; it is a risk-absorbing feature. It allows institutions to manage uncertainty without resorting to constant renegotiation or systemic failure.

Efforts to fully codify these relationships through smart contracts do not eliminate discretion. They displace it. When unforeseen events occur—regulatory intervention, asset impairment, force majeure, covenant breaches—decisions are still made by trustees, boards, regulators, or courts. Rigid execution logic may constrain the timing or form of intervention, but it cannot remove the need for it.

This displacement becomes problematic in institutional contexts. Fiduciaries are accountable for outcomes, not for adherence to code. If automated execution produces results that conflict with fiduciary duty, responsibility does not dissolve; it reasserts itself through governance intervention, legal challenge, or regulatory inquiry. In such cases, the opacity created by rigid automation can exacerbate rather than reduce risk.

The resistance of real-world assets to full codification is therefore structural, not transitional. These assets exist within systems designed to manage uncertainty through layered authority and *ex post* judgment. Tokenization architectures that treat discretion as an inefficiency to be eliminated will eventually confront the limits of that assumption. Sustainable adoption requires recognizing that governance is not the opposite of automation, but its necessary counterpart.

#### **4. Institutional Adoption Requirements**

Institutional adoption of real-world asset tokenization is constrained less by technological capability than by institutional requirements that are largely non-negotiable. Asset managers, banks, pension funds, and insurers operate within fiduciary and regulatory frameworks that prioritize legal enforceability, governance clarity, and accountability over execution efficiency. These requirements shape not only what institutions are willing to adopt, but also the sequence and form that adoption can take.

First, institutions require clear legal authority and enforceable claims. Ownership interests in real-world assets must be anchored in recognized legal structures that courts and regulators can interpret and enforce. Tokenized representations that lack clear linkage to contractual rights, custodial arrangements, or statutory frameworks introduce ambiguity that institutions are structurally constrained to avoid. Cryptographic certainty does not substitute for legal certainty when capital is deployed on behalf of beneficiaries.

Second, institutions require defined governance and decision authority. Someone must be accountable for interpreting rules, exercising discretion, and intervening when conditions deviate from expectation. Boards, trustees, and investment committees are not incidental features of institutional finance; they are the mechanisms through which responsibility is allocated and monitored. Systems that obscure where authority resides, or that claim authority resides nowhere, are incompatible with institutional decision-making.

Third, institutions require ex post accountability and auditability. Investment decisions are subject to review by internal risk committees, external auditors, regulators, and, in some cases, courts. Automated execution that cannot be explained, justified, or overridden undermines this accountability chain. The question institutions ask is not whether a system executed as programmed, but whether outcomes can be defended under fiduciary and regulatory scrutiny.

Finally, institutions require exception handling. Real-world assets are exposed to regulatory change, operational disruptions, market dislocations, and political intervention. Institutional systems are designed to absorb these shocks through discretion, escalation, and judgment. Tokenization frameworks that assume away exceptions in the name of efficiency fail precisely where institutions place the highest value.

These requirements do not reflect conservatism or resistance to innovation. They reflect the structural conditions under which institutions are permitted to operate. Tokenization efforts that ignore these constraints may achieve technical success but will struggle to attract sustained institutional capital.

## **5. Governance, Not Execution, as the Binding Constraint**

Much of the discourse surrounding real-world asset tokenization focuses on execution: settlement speed, automation, composability, and cost reduction. While these attributes matter at the margin, they are rarely the binding constraint on institutional adoption. The decisive constraint is governance.

Execution failures are generally manageable. Software bugs can be patched, throughput can be improved, and infrastructure can be upgraded. Governance failures are qualitatively different. They arise when authority is unclear, responsibility is diffused, or intervention mechanisms are absent. In institutional contexts, these failures are not tolerated as technical glitches; they are treated as systemic risks.

The appeal of autonomous execution rests on the idea that removing discretion reduces uncertainty. In practice, the opposite is often true. When systems lack clearly defined override mechanisms, uncertainty is displaced rather than eliminated. Market participants are left unsure who can intervene, under what conditions, and with what legitimacy. This uncertainty is particularly problematic during stress events, when rigid execution can amplify rather than dampen instability.

Institutional systems are built around the expectation that human judgment will be exercised when rules prove insufficient. Emergency powers, discretionary waivers, and governance escalation are not failures of design; they are safeguards against incomplete specification. Tokenization architectures that treat these features as regressions fail to account for how institutional trust is actually built and sustained.

This does not imply that automation is incompatible with institutional finance. It implies that automation must be embedded within governance structures rather than positioned as a substitute for them. The role of code is to standardize routine processes, reduce operational friction, and improve transparency—not to eliminate authority or responsibility.

Viewed through this lens, many high-profile failures in real-world asset tokenization reflect governance breakdowns that are often mistaken for technical problems. Systems that execute flawlessly but lack credible governance do not inspire confidence; they raise questions about who bears responsibility when outcomes diverge from intent.

For institutions, governance is not a secondary consideration to be layered on after technical deployment. It is the foundation upon which adoption decisions are made. Until tokenization frameworks internalize this reality, execution improvements alone will not unlock institutional scale.

## 6. Reframing Tokenization as Institutional Design

**Figure 1. Execution and Governance in Real-World Asset Tokenization**

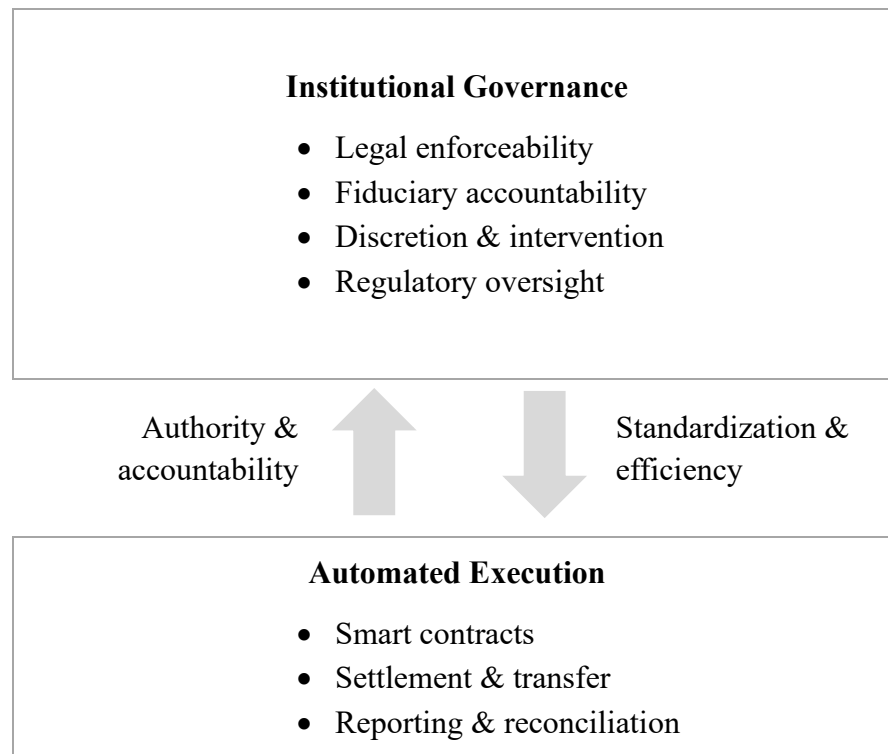


Figure 1 illustrates the distinction between automated execution and institutional governance that underpins the argument of this paper.

If “code is law” fails as a governing principle for real-world assets, the implication is not that automation is misguided, but that its role has been mischaracterized. Tokenization should be understood not as a mechanism for replacing institutions, but as a tool for augmenting institutional design.

In institutional markets, value is created through a combination of standardized processes and discretionary judgment. Automation is most effective when applied to the former: settlement, reconciliation, reporting, and routine enforcement. These functions benefit from clarity, repeatability, and speed. Governance, by contrast, governs the latter: exception handling, interpretation, intervention, and accountability. Attempting to automate both simultaneously conflates fundamentally different functions.

A more productive framing treats tokenization as an infrastructure layer that formalizes and clarifies existing institutional arrangements. Under this approach, code encodes procedures, not authority. Legal agreements, governance bodies, and regulatory oversight continue to define who can act, when they can act, and under what conditions their actions are legitimate. Automation

reduces friction, but discretion remains grounded in established institutional decision-making processes.

This reframing also clarifies why many early tokenization efforts appear successful in pilot settings but struggle to scale. Proof-of-concept deployments often abstract away governance complexity, either by assuming benign conditions or by relying on informal intervention. At scale, these assumptions no longer hold. Institutional adoption requires that governance be explicit, auditable, and resilient under stress.

Viewed this way, the challenge of real-world asset tokenization is not one of technical maturity, but of institutional alignment. Systems that respect the division between automated execution and governed discretion are more likely to earn trust, attract capital, and endure over time.

## **7. Implications for Policymakers, Institutions, and Technologists**

The structural limits of “code is law” have distinct implications for different stakeholders involved in real-world asset tokenization.

For policymakers and regulators, the analysis suggests that regulatory frameworks should focus less on the novelty of distributed ledgers and more on the governance arrangements that surround them. Tokenization does not eliminate the need for supervision; it reshapes where risks appear. Clear standards around accountability, control, and intervention are likely to matter more than prescriptive rules about technical architecture.

For institutional investors, the key implication is that adoption should be evaluated through a governance lens rather than a technological one. The relevant questions are not whether execution is automated, but whether authority is clearly allocated, fiduciary duties are preserved, and exception mechanisms are credible. Tokenized structures that mirror familiar institutional safeguards are more likely to integrate into portfolios than those that prioritize novelty over transparency and clarity.

For technologists and system designers, the implication is more corrective. Success in institutional markets depends less on eliminating trust than on formalizing it. Systems should be designed with explicit governance hooks: defined roles, override mechanisms, and clear interfaces between code and legal authority. Treating these features as first-class design constraints, rather than compromises, increases the likelihood of real adoption.

Across all stakeholders, the broader implication is that tokenization is not a shortcut around institutional complexity. It is a way of making that complexity explicit, structured, and, where possible, more efficient.



## 8. Conclusion

The idea that “code is law” captures an important insight about how technical systems shape behavior. But when elevated from metaphor to governing principle, it obscures more than it explains—particularly in the context of real-world assets.

Real-world asset markets are built on incomplete contracts, discretionary judgment, and layered accountability. These features are not inefficiencies to be engineered away; they are the mechanisms through which uncertainty is managed and trust is sustained. Attempts to replace them with autonomous execution overlook the institutional mechanisms through which value is created and protected.

The future of real-world asset tokenization lies not in eliminating governance, but in integrating automation within it. Code can enhance transparency, reduce friction, and improve operational efficiency. Law and governance determine legitimacy, responsibility, and resilience. Sustainable tokenization requires recognizing the distinct roles each plays.

In this sense, the failure of “code is law” is instructive rather than discouraging. It points toward a more realistic and durable path forward—one in which tokenization succeeds not by replacing institutions, but by working with them.

## References

- Allen, F., Gu, X., & Jagtiani, J. (2021). A survey of fintech research and policy discussion. *Review of Corporate Finance*.
- BIS. (2024). Tokenisation in the context of money and other assets: Concepts and implications for central banks. *Bank for International Settlements*.
- CPSS, IOSCO, BIS. (2012). Principles for financial market infrastructures. *Committee on Payment and Settlement Systems, Technical Committee of the International Organization of Securities Commissions, Bank for International Settlements*.
- De Filippi, P., & Wright, A. (2018). Blockchain and the law. *Harvard University Press*.
- Hart, O., & Moore, J. (1990). Property rights and the nature of the firm. *The Journal of Political Economy*, Vol. 98, No. 6, pp. 1119–1158. *The University of Chicago Press*.
- IOSCO. (2020). Issues, risks and regulatory considerations relating to crypto-asset trading platforms. *International Organization of Securities Commissions*.
- IOSCO. (2025). Tokenisation of financial assets. *International Organization of Securities Commissions*.
- Lessig, L. (1999). Code and other laws of cyberspace. *Basic Books*.
- Williamson, O. E. (1985). The economic institutions of capitalism. *Free Press*.
- WEF. (2016). The future of financial infrastructure. *World Economic Forum*.