March 24, 2026

The Honorable Paul S. Atkins
Chairman
U.S. Securities and Exchange Commission
100 F Street, NE
Washington, D.C. 20549

cc: The Honorable Hester M. Peirce, Commissioner and Chair, Crypto Task Force
Mr. James Moloney, Director, Division of Corporation Finance

Re: *Written Submission to the Crypto Task Force and Project Crypto; Architectural Framework for the Tokenization of Real-World Assets on Digital Ledgers and Its Implications for the Commission's Regulatory Framework*

Dear Chairman Atkins, Commissioner Peirce, and Director Moloney:

I. Introduction and Purpose of This Submission

We respectfully submit this letter for the record in connection with the Commission's Project Crypto initiative and the work of the Crypto Task Force led by Commissioner Peirce, both directed at developing a clear and durable regulatory framework for digital assets, including real-world asset (RWA) tokenization. We commend the Commission for this work. The January 2026 joint statement from the Divisions of Corporation Finance, Investment Management, and Trading and Markets, establishing a four-category taxonomy for digital assets, represents exactly the kind of principled structural thinking this moment requires. We write to ensure that structural thinking extends to the layer beneath the taxonomy: the architecture of the ledgers on which tokenized assets will actually reside.

The central argument of this submission is as follows: the architectural approach that any durable federal regulatory framework should be designed to support is canonicalization of assets on digital ledgers, not merely the tokenization of assets. The distinction is not semantic. It is architectural, structural, and directly consequential for every dimension of the Commission's mandate: investor protection, examination and enforcement capacity, clearing and settlement integrity, custody rule design, and the long-term enforceability of any compliance framework the Commission constructs for tokenized securities markets.

This submission draws on published architectural analysis and technical literature to explain why canonicalization represents the structurally superior foundation for the Commission's framework, and why the Commission's rulemaking work will be better served by understanding this distinction now, before the market organizes around an approach whose structural limitations are already well-documented. We further identify specific actions the Commission can take through existing mechanisms, the Crypto Task Force, Project Crypto, and coordination with NIST, FINRA, and IOSCO, to ensure its framework is architecturally sound.

II. The Critical Distinction: Tokenization vs. Canonicalization

Tokenization, as conventionally understood, refers to the representation of a real-world asset, a

bond, a share of equity, a parcel of real estate, a commodity, a pharmaceutical batch record, as a digital token on a blockchain. In the dominant smart contract paradigm, that token is a balance entry inside a contract account: a number stored in shared contract storage, with ownership enforced by application-layer code that each issuer writes independently.

Canonicalization means something structurally deeper. A canonicalized asset is a first-class object in the ledger's native execution environment, not a balance in a shared database, but a self-contained, independently-addressable entity with its own cryptographically-secured identity, its own encapsulated state, its own behavioral logic, and its own position in a universal class hierarchy maintained by consensus across all nodes. The asset does not merely represent a right; it instantiates that right in a form the ledger itself natively understands, enforces, and can compose with other assets without sacrificing security or scalability.

To use an analogy that may be useful for the Commission's consideration:

Tokenization: The ticket to the coat check. The coat sits in a closet (smart contract) and anything in its pockets is under the direct control of the coat check attendant. Do you trust that person will not pick your pocket while it is in the closet under their control? For the Commission, the question is not rhetorical: who controls the contract controls the asset, and the ledger itself cannot tell you whether the contract does what it claims.

Canonicalization: The complete digital twin of the coat, the object itself, with all lineages, rules, policy, and compliance encapsulated in a self-contained object in the owner's own account. Everything that happens to the coat is immediately recorded to that object, delivering an immutable record of provenance, auditability, and compliance in real time, all on-chain. The Commission can examine, audit, and enforce against the object directly, because the ledger natively understands what it is.

Tokenization (Smart Contract Model)	An asset is a balance inside a shared contract account. Ownership and transfer logic lives in application-layer code written independently by each issuer. The ledger itself has no native concept of what the asset is or what rules govern it. The Commission cannot examine or audit what the ledger cannot identify.
Canonicalization (Programmable Smart Asset Model)	An asset is a first-class object with encapsulated state, defined behavior, and a cryptographically-secured identity derived from a canonical class definition. The ledger natively understands, manages, and composes the asset at the execution layer. The Commission can examine and enforce against a definitively identified, auditable on-chain object.
Regulatory Consequence	Tokenization requires that every regulatory obligation, transfer restrictions, disclosure triggers, investor eligibility checks, AML hooks, be re-implemented in application-layer code by every issuer, independently, for every contract, subject to each issuer's own bugs and gaps. Canonicalization allows compliance logic to be defined once in the canonical class definition and inherited by every instance, making platform-level enforcement structurally possible.

The distinction matters for the Commission because the regulatory framework now under construction will either be built on infrastructure the Commission can examine and enforce against,

or on infrastructure whose ledger-level opacity makes enforcement dependent on issuer attestation and off-chain documentation. The Commission's three-part statutory mandate, investor protection, fair and orderly markets, and capital formation, requires the former. The Commission's framework should be designed to require it.

III. The Structural Limitations of the Smart Contract Model

To understand why canonicalization is architecturally superior, it is necessary to understand the foundational design constraints of the smart contract model that currently dominates digital asset markets. These are not implementation deficiencies that better engineering can remedy. They are structural consequences of the model's core design choices, and they are directly relevant to each of the Commission's active regulatory workstreams.

In precise architectural terms, the smart contract model is structurally analogous to MS-DOS: a passive, reactive, sequentially-executing module that co-locates code and data at a single address, communicates synchronously, delegates all access control to application-layer logic, and operates within a globally shared mutable state space. This analogy is not rhetorical, it is architecturally exact. Both systems were designed for environments far simpler than those they now occupy. In MS-DOS, this produced memory conflicts and instability as the computing ecosystem grew. In smart contracts, the structural consequences for the Commission's regulatory mission are:

- **Serial execution bottlenecks and throughput ceilings.** Every caller to a widely-used contract, a token, a settlement mechanism, a lending protocol, is serialized through that contract's single account on a single shard. This is not a performance limitation that additional hardware resolves; it is a consequence of shared mutable state. As a financial asset's circulation grows, the contract holding its state becomes a bottleneck for the entire network regardless of shard count. Amdahl's Law applies without mitigation: a tokenized securities market of national scale built on this infrastructure will face structural throughput ceilings that worsen as market participation grows.
- **The composable scaling dilemma.** The smart contract model faces an inescapable two-mode structural dilemma: either contracts are designed to be independent and non-composable (which scales, but forfeits the cross-contract interaction that makes programmable assets commercially useful), or they compose through shared external state (which is necessary for useful financial applications but cannot scale because every cross-shard interaction serializes). Independent researchers at DataFinnovation have established this as an NP-complete problem in the general case. There is no third option within the smart contract model. No existing Layer 1, Layer 2, or sharding architecture has solved this problem, because the architecture does not permit it.
- **Security vulnerabilities architectural in origin.** Reentrancy attacks, which have cost the market billions of dollars across multiple chains, are a direct consequence of synchronous blocking inter-contract calls in a shared state environment. They are not bugs; they are predictable outcomes of the architectural model. Every Layer 2 workaround, every bridge, and every cross-chain mechanism inherits versions of this vulnerability. A regulatory safe harbor or registration pathway that permits trading on smart-contract infrastructure inherits this structural risk regardless of the disclosure requirements layered above it.

- **Compliance fragmentation that defeats platform-level enforcement.** Because the ledger has no native understanding of what an asset is, every regulatory obligation, transfer restrictions, reporting requirements, investor eligibility checks, AML hooks, must be independently re-implemented in application-layer code by every issuer, for every contract. The result is thousands of independent implementations, each subject to its own bugs, gaps, and inconsistencies, none of which the Commission can audit at the protocol level. There is no canonical definition of a regulated digital security that the ledger itself enforces. Compliance is nominal, not structural.

These are not theoretical concerns. They are the documented failure modes behind DeFi congestion events, multi-billion-dollar exploits, the proliferation of Layer 2 infrastructure, and the repeated inability of existing architectures to deliver composable scale. The Commission is building a regulatory framework for this market at a moment when its structural ceiling has become widely visible. A framework designed around smart contract architecture will embed these limitations into the regulatory infrastructure of the United States' tokenized securities market.

IV. The Canonicalization Model: Architectural Foundations

The canonicalization approach derives from a fundamentally different architectural tradition. Where smart contracts descend from procedural, sequential computing, the canonicalization model descends from Alan Kay's message-passing object architecture, developed at Xerox PARC in the 1970s and recognized as the foundational design philosophy behind encapsulated, composable, and autonomously-addressable software objects. This architectural lineage has been applied to the distributed ledger context, with implementations now documented in published technical literature and granted U.S. patents held by PraSaga Foundation, a United States non-profit organization.

Kay's three foundational principles, which the canonicalization model directly instantiates, are:

- **Messaging as the only interaction primitive.** Objects cannot reach into each other. All coordination happens exclusively through messages. This single constraint eliminates reentrancy attacks structurally, enables asynchronous and parallel execution, and makes the interaction surface of every asset-object precisely defined, auditable, and examinable by the Commission.
- **Local retention and protection of state.** Every object owns its own state exclusively. There is no shared mutable state, no global contract account that all callers serialize through. This is the structural basis for genuine MIMD (Multiple Instruction, Multiple Data) parallel execution across independent account sets, the model that makes composable scale possible and eliminates the throughput ceiling that smart contract platforms cannot escape.
- **Extreme late-binding.** Nothing is resolved at compile time that can be resolved at runtime. SagaChain stores classes as source code on-chain, compiled at node execution time, enabling runtime dispatch against a live class tree. This means the canonical class definitions governing a digital security can evolve as regulatory requirements evolve, without requiring each issuer to redeploy a new contract.

In implementations of this architecture applied to distributed ledger environments, these principles are realized through the following structural components, each of which directly addresses a

known failure mode of the smart contract model with direct implications for the Commission's workstreams:

Account-Object Model	First-class object model: state and code unified per account-object. Each user maintains a directed graph of objects. Assets, ownership structures, and cryptographic signature schemes are all first-class objects. The ledger is the authoritative source of what each asset class is and how it must behave. U.S. Patent Nos. 20200348963 and 11,436,039 B2.
Decentralized OS Runtime	Decentralized OS runtime managing the global class tree and processing all message-passing transactions. Classes are stored as source code on-chain and compiled at node execution time, enabling runtime dispatch against a live, mutable class tree. The Commission's class-level compliance requirements can be embedded in this tree.
Class Manager Infrastructure	Manages all classes, objects, and Programmable Smart Assets in the single-instance global class tree, maintained by consensus across all nodes. The canonical source of all asset type definitions. A Security Token class defined in this tree is inherited by every instance, compliance logic defined once, enforced everywhere.
Account Relationship Co-Locality Algorithm	Tracks emergent account relationship patterns, assigns related accounts to the same shard temporarily to minimize cross-shard block dependencies, and releases accounts as relationship probability decays. Solves the composable scaling dilemma without solving the NP-complete shard assignment problem at runtime.
Single System Image	Synchronizes memory state across all distributed nodes, making distribution transparent to the object model. PraSaga's own architectural extension beyond Kay, realizing a genuinely distributed trustless computer. Provides the Commission with a single, consistent object state space to examine regardless of which node is queried.
Universal Settlement Primitive)	A single native coin used exclusively for all transaction fees and value transfers across all asset types, regardless of the nature of the underlying asset. A real estate transfer, a securities settlement, and a pharmaceutical record all settle fees in. Provides a single, auditable settlement primitive for AML/BSA oversight and eliminates the fee fragmentation of multi-token ecosystems.

V. Implications for the Commission's Active Regulatory Workstreams

A. Project Crypto and the Definition of a Tokenized Security

Any definition of a "tokenized security" or "digital asset security" under Regulation Crypto that does not specify architectural requirements for the underlying ledger will be susceptible to nominal compliance, issuers placing tokens on ledgers that cannot structurally enforce the canonical identity the definition requires. The Commission's four-category taxonomy will be difficult to apply consistently if two tokens purporting to represent the same category of security are encoded in structurally incompatible, ledger-opaque formats that the Commission has no protocol-level mechanism to distinguish or audit. The Commission should consider whether a class definition

certification regime, alongside individual token registration, might provide more structurally sound compliance coverage: certifying compliant canonical class definitions for major security categories would make compliance a platform-level property rather than an issuer-level assertion.

B. Clearing and Settlement Modernization for Tokenized Securities

The Crypto Task Force has identified modernizing clearing and settlement rules for tokenized securities as a priority workstream. Clearing and settlement presuppose that both counterparties and the clearing agency can independently verify the identity, class, and ownership status of the instrument being settled. On a smart-contract platform, this verification depends on the off-chain reputation of the deployer and the voluntary conformance of the contract to a published standard, neither of which is structurally guaranteed by the ledger. On an architecture with a ledger-native global class tree, the verification is protocol-enforced: the instrument's class definition, ownership record, and transfer history are all on-chain, consistent, and examinable without reliance on issuer attestation. The Commission's clearing and settlement framework should specify which of these two conditions it requires.

C. Custody Rules for Digital Asset Securities

The Commission's work on qualified custody for digital assets presupposes that the custodied asset has a definite, verifiable, ledger-native identity that the custodian demonstrably controls. On ledgers without canonical asset representation, what a custodian holds may be a token whose relationship to the underlying security is defined by off-chain legal documentation rather than by the ledger itself. The architectural distinction determines what "custody" means in a legally enforceable sense: whether the Commission can verify the custodian's control at the ledger level, or must rely on contractual representation. The Commission's custody framework should specify which it accepts as sufficient.

D. AML, BSA, and Settlement Finality Architecture

The canonicalization model's universal settlement primitive, a single coin used for all transaction fees and value transfers across all asset types, has direct implications for AML and BSA compliance. A universal settlement layer provides a single, auditable primitive for regulatory oversight, creates a clean architectural separation between the asset layer and the settlement layer, and eliminates the fee fragmentation of multi-token ecosystems that complicates transaction monitoring. This maps cleanly onto existing AML/BSA regulatory concepts and simplifies the Commission's and FinCEN's oversight architecture for a tokenized securities market.

E. Innovation Exemptions and Safe Harbors

Any safe harbor or innovation exemption permitting the trading of tokenized securities on permissionless chains should include architectural requirements for the ledger, not merely disclosure or registration requirements for the issuer. A safe harbor that permits trading on ledgers structurally incapable of canonical asset representation creates a gap in the Commission's investor protection framework that disclosure rules cannot close: the Commission cannot enforce against what the ledger cannot definitively identify. Architectural requirements for the safe harbor are not a burden on innovation; they are the condition under which the Commission's enforcement tools can reach.

VI. Recommendations to the Commission

- **Establish 'canonicalization' as a defined architectural concept** in Regulation Crypto and in any definition of a "tokenized security" or "digital asset security." Create an affirmative architectural requirement that the underlying ledger provide a persistent, uniquely-identified, class-typed on-chain object governed by a ledger-native class hierarchy, not merely an off-chain interface standard implemented in third-party contract bytecode.
- **Consider a class definition certification regime** alongside individual token registration. If compliance logic is embedded in canonical class definitions, the regulatory leverage point is the class, not the individual instance. Certifying compliant canonical class definitions for major security categories, equity, debt, real estate investment, commodity-backed instruments, would provide systemic compliance coverage more efficiently and more consistently than reviewing each issuance individually.
- **Convene a Crypto Task Force roundtable on ledger architecture** as a distinct workstream from token standards and disclosure frameworks. Invite distributed systems architects, legal scholars in property and securities law, and representatives of ledger platforms to develop a Commission-level technical understanding of the architectural distinction between smart contract, UTXO, and first-class object-model ledgers, and their respective capacities to support legally reliable, examinable, and auditable asset representation.
- **Direct the Commission's technical staff, in coordination with NIST,** to produce a written assessment of the architectural requirements for canonical asset representation on a distributed ledger, and to evaluate which currently available ledger architectures satisfy those requirements at the protocol level. This assessment should be completed and published before Regulation Crypto is finalized.
- **Include U.S. Patent Nos. 20200348963 and 11,436,039 B2,** both held by PraSaga Foundation, a United States non-profit organization formally re-incorporated under U.S. law in June 2025, which maintains the SagaTech open-source stack as a public good, in the Crypto Task Force's technical record. These patents are not cited because any entity should be advantaged by the Commission's rulemaking; they are cited because they represent the current state of the art in on-chain canonical object models and message-passing transaction architecture, and their claims define what a technically sufficient architectural standard requires.
- **Examine SagaStandards** as a governance model for how the Commission and FINRA might participate as standard-setting stakeholders in the governance of canonical asset class definitions, a global multi-stakeholder body, no single owner, analogous to SWIFT, the LEI system, and ISO technical committees, ensuring that regulatory requirements are embedded in the ledger's class hierarchy rather than appended by off-chain rule.
- **Ensure the Commission's framework does not inadvertently entrench the smart contract model's structural limitations** by building safe harbors, registration pathways, or custody rules that presuppose smart contract architecture. Requirements designed around the assumption that all digital securities are smart contract tokens will create

structural barriers to adoption of canonicalization-based architectures that are, on every dimension relevant to the Commission's mandate, superior.

- **Coordinate with the CFTC, FINRA, FinCEN, and international securities regulators through IOSCO** to advance a common architectural standard for canonical asset representation before other jurisdictions establish weaker, interface-based standards that create cross-border regulatory arbitrage opportunities at the expense of U.S. investor protection.

VII. Conclusion

The Commission has an opportunity, through Project Crypto and the Crypto Task Force, to establish the architectural foundation on which United States tokenized securities markets are built. The tokenization approach that currently dominates, derived from a 1980s sequential computing model, carries structural limitations that will become increasingly visible as the market grows: serial execution bottlenecks, an NP-complete inability to scale composable applications, security vulnerabilities architectural in origin, and compliance logic fragmented across thousands of independent contract implementations that the Commission cannot audit consistently.

The canonicalization approach, derived from Alan Kay's message-passing object architecture, realized in modern distributed implementations, and grounded in formal technical literature on the composable scaling problem, addresses each of these limitations at the structural level. It offers the Commission a regulatory foundation that is technically sound, scalable to a market of national significance, inherently more secure, and structurally more amenable to platform-level compliance enforcement than anything the smart contract model can provide. The foundational patents are held by a U.S. non-profit, maintained as a public good, and governed through a multi-stakeholder body in which the Commission and FINRA can participate as standard-setting stakeholders.

We urge the Commission to incorporate the distinction between tokenization and canonicalization into the analytical framework of Project Crypto and the Crypto Task Force, and to ensure that any rules produced create space for, and where appropriate actively require, the architectural approach that will best serve the Commission's mandate to protect investors, maintain fair and orderly markets, and facilitate capital formation for the long term.

We welcome the opportunity to provide technical briefings, participate in a Crypto Task Force roundtable, or submit additional written analysis on any aspect of the architectural and regulatory analysis presented in this letter. Written submissions to the Task Force may also be directed to Crypto@sec.gov.

Respectfully submitted,

Michael Holdmann
CEO, PraSaga Foundation
9412 Steeplehill Dr
Las Vegas, NV 89117
michael@prasaga.com

Appendix: Glossary of Key Architectural Terms for the Record

Tokenization	Representation of a real-world asset as a digital token, typically as a balance inside a smart contract account, with ownership and transfer logic implemented in application-layer code written independently by each issuer. The ledger itself has no native concept of what the asset is or what rules govern it.
Canonicalization	Representation of a real-world asset as a first-class object in the ledger's native execution environment, with encapsulated state, defined behavioral logic, and a canonical class definition inherited by all instances of that asset type across all compliant platforms. The ledger is the authoritative source of what the asset is and how it must behave.
Smart Contract	A passive, reactive, sequentially-executing module that co-locates code and state at a single address, operates within a globally shared mutable state space, and communicates via synchronous blocking calls. Structurally analogous to MS-DOS programs adapted for distributed deployment. The Commission cannot examine or enforce against what the ledger cannot natively identify.
Programmable Smart Asset	A first-class account-object implementing Alan Kay's message-passing architecture: encapsulated state, interaction only via messages, behavior and identity derived from a global canonical class tree. Assets, ownership structures, and signature schemes are all first-class objects in the same execution environment. The Commission can examine and enforce against the object directly.
Global Class Tree	A single, canonically-defined, consensus-maintained class hierarchy from which all asset objects are instantiated. Every node in the network maintains a synchronized copy. Compliance logic defined in a class definition is inherited by every instance, platform-level enforcement without per-issuer re-implementation.
Single System Image (SSI)	An architectural mechanism that synchronizes memory state across all distributed nodes, making distribution transparent to the object model. Provides the Commission with a single, consistent object state space to examine regardless of which node is queried. PraSaga's own architectural extension beyond Kay's single-machine model.
MIMD vs. SISD	Multiple Instruction Multiple Data (MIMD) vs. Single Instruction Single Data (SISD). Smart contracts execute SISD, one transaction at a time, shared state, throughput ceiling scales with Amdahl's Law. The Programmable Smart Asset model executes MIMD, independent account sets execute simultaneously on separate shards, enabling composable scale without structural bottlenecks.
Account Relationship Co-Locality Algorithm (SagaScale)	Tracks emergent account relationship patterns, assigns related accounts to the same shard temporarily to minimize cross-shard block dependencies, and releases them as relationship probability decays. Solves the composable scaling dilemma without solving the general NP-complete assignment problem at runtime.

Universal Settlement Primitive / Native Network Coin	SagaCoin: the single native coin used for all transaction fees and value transfers across all asset types, regardless of the underlying asset. Provides a single, auditable settlement primitive for AML/BSA oversight. Eliminates the fee fragmentation of multi-token ecosystems. Maps cleanly onto existing settlement finality regulatory concepts.
SagaStandards	The global multi-stakeholder body through which institutions, regulators, developers, and civil society hold custodianship of the SagaTech open-source codebase and canonical asset class definitions. No single entity owns or controls SagaStandards. Governance model analogous to SWIFT, the LEI system, and ISO technical committees. The Commission and FINRA can participate as standard-setting stakeholders.