

MetaLeX Labs, Inc.
800 N King St.,
Suite 304 1951,
Wilmington, DE 19801
gabriel@metalex.tech

July 17, 2026

Via Electronic Submission (rule-comments@sec.gov)

Vanessa A. Countryman, Secretary
U.S. Securities and Exchange Commission
100 F Street, NE
Washington, DC 20549-1090

Re: File No. 4-894 — Staff Statement Regarding Broker-Dealer Registration of Certain User Interfaces Utilized to Prepare Transactions in Crypto Asset Securities (Apr. 13, 2026): the “processing trade documentation” exclusion

Dear Ms. Countryman:

MetaLeX Labs, Inc. ("MetaLeX") submits this comment in response to the staff's invitation for input on the above-referenced statement (the "Statement"). MetaLeX builds open-source legal-technology infrastructure for onchain entities and tokenized securities: smart contract systems through which issuers maintain uncertificated securities ledgers and through which parties to a securities transaction execute a standardized form of agreement with their own cryptographic credentials and settle in accordance with its terms. Our contracts are public at github.com/MetaLex-Tech, and the mechanics described below are drawn from that deployed, inspectable code. We write principally on a single point, and close with one further request concerning the Statement's route-related conditions. One undefined term in the Statement's excluded-services list, "processing trade documentation," is susceptible of a reading that would exclude from the framework precisely the interfaces the Commission should most want inside it: interfaces to systems in which every trade is documented by a real, executed legal agreement.

1. The Statement's architecture is sound.

The Statement gets the structure right. It regulates by layer and by function: an interface that converts user-specified transaction parameters into trading instructions the user executes with the user's own key is not, by that fact alone, a broker, and the conditions (no discretion, no custody, no recommendations, route-and-counterparty-agnostic compensation, disclosure) police the line where interface services would shade into intermediation. Condition 4's treatment of affiliated systems, permitting connection to trading systems "created, offered, and/or operated, directly or indirectly, by the Covered User Interface Provider or its affiliates" on disclosed and equal terms, correctly recognizes that integrated builders can operate compliant interfaces. The excluded-services list serves the same boundary-keeping purpose, and most of its entries are self-defining. "Processing trade documentation" is not.

2. In tokenized securities systems, the documentation is the trade.

The term reads naturally against a traditional workflow in which documentation trails the trade through a back office: an intermediary prepares, completes, reviews, and moves the paperwork (transfer instructions, stock powers, assignments, confirmations) that carries a transaction it is effecting through settlement. In a well-built tokenized securities system the sequence is inverted: the documentation is standardized, it is fixed onchain before any trade exists, and the parties execute it themselves. Because the term is undefined and the architecture is unfamiliar, we describe the mechanics concretely, from our own deployed system.

How documentation actually works in these systems.

In MetaLeX's architecture, an onchain agreement registry administers the entire documentation lifecycle, and its design is instructive on the interpretive question:

First, the forms are standard. The legal text of each transaction agreement is a standardized form: one form governs every transaction of its type, in the manner of the industry-standard forms that already organize private markets (the SAFE, the NVCA model documents, the ISDA Master Agreement). The platform publishes the standard templates; an issuer may also register its own custom form for transactions in its securities; and individual traders never draft, upload, or commission documents, and no party to a trade retains counsel to paper it. The registry does not store prose fields for freeform drafting; it stores, for each registered template, the document's location (a content URI), a title, and two enumerated lists of blanks: the fields common to the transaction and the fields particular to each party. The enumerated blanks are the only content a transacting party can ever supply.

Second, the form is registered before any trade exists. Template registration is restricted to the registry's administrator (and, where configured, to an issuer for its own securities), and each executed agreement's onchain identity is a cryptographic hash of the template, the transaction terms, and the parties, so neither the interface operator nor anyone else can substitute different legal text after the fact. The registered form is infrastructure, in place before and independent of any particular transaction, like a statutory form or a printed form book.

Third, the software renders; the parties supply variables. When two parties strike a trade, the interface retrieves the registered template, displays the standard form, and populates its enumerated blanks with the variables the parties themselves supply (quantity, price, party identity). The parties draft nothing and negotiate no text; the interface completes no field on its own judgment; there is no field on which judgment could operate.

Fourth, each party executes with its own key. Signature is by EIP-712 typed-data signature, and the signed payload is the complete content identity of the agreement: the contract identifier, the document URI, every field name, and every field value, global and party-specific. Each party signs the whole agreement, not a bare hash pointer, with a key only that party controls. The registry verifies each principal's own signature and records a per-party signature timestamp onchain; only when every party has signed does the registry record the agreement as fully executed. Voiding is symmetrical: it requires the parties' own signed void requests. Where the transaction involves consideration, an escrow-integrated signing path binds the signature and the funding in a single party-initiated act.

Fifth, settlement follows the agreement. The executed agreement's identifier travels with the settlement, and the transfer completes in accordance with the agreement's terms. The ledger record and the legal record are the same record.

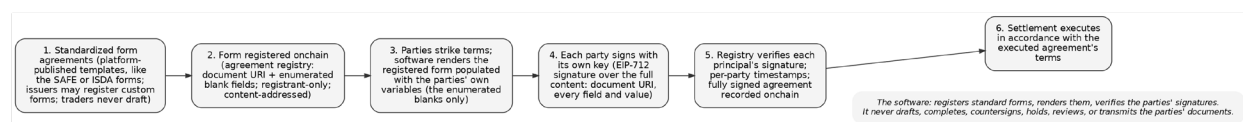


Figure 1. The documentation lifecycle in an onchain agreement registry. At no step does the software provider draft, complete, countersign, review, hold, or transmit the parties' documents, and at no step does any trading party draft anything.

Against that lifecycle, the interpretive stakes are plain. Read literally and in isolation, "processing trade documentation" could be stretched over any software that touches these documents. That reading would produce two results the staff cannot have intended. First, it would swallow the Statement: an interface that prepares trading instructions necessarily generates and handles transactional records, so the exclusion would consume the covered activity. Second, and worse, it would invert the Commission's incentives. An interface connected to a system in which trades settle as bare token transfers, with no legal documentation at all, would enjoy the framework; an interface connected to a system in which every trade is papered with an executed, enforceable, standardized agreement identifying the parties and incorporating the applicable transfer restrictions and disclosures would be excluded. A framework that rewards documentation-free trading and penalizes documented trading has its incentives backwards. The federal securities laws run on documents; the Commission should want more of them in crypto markets, not fewer.

3. The term should be read in the company it keeps.

The excluded-services list reads: "negotiating terms for any transaction; soliciting specific crypto asset securities transactions; making investment recommendations or providing advice; arranging for financing; processing trade documentation; conducting independent asset valuations; holding, having access to, handling, managing, or possessing user funds, securities, or stablecoins; executing or settling transactions; or taking or routing orders." Every other entry describes a service performed for a customer by a person interposed in the transaction: the traditional service bundle of a securities intermediary. Noscitur a sociis supplies the reading. In that company, "processing trade documentation" describes the intermediary's documentation function, the back-office limb of effecting: preparing, completing, reviewing, or transmitting the paperwork that carries a transaction through settlement, as a service performed on a party's behalf. It does not describe software with which the parties document their own transaction on a standard form, any more than a form book, a document-assembly application, or an electronic signature platform "processes" the contracts made with it. Standardization is itself the point: because one standard text governs every transaction of a type and the variable content is confined to enumerated fields, no drafting, review, or documentation service is performed for anyone, by anyone, trade by trade. No one asks whose counsel drafted the SAFE or the ISDA Master Agreement before signing one, and no one thinks their publishers render documentation services in the transactions executed on them. No one contends that e-signature providers are brokers because securities purchase agreements are signed on them; the analysis should not change because the form library, the fill-in ceremony, and the signature verification are onchain. If anything the onchain version is more conservative: the e-signature platform

could in principle alter a document in its custody, while a content-addressed registry cannot, because the parties' signatures cover the entire content and any alteration breaks them.

4. Requested clarification.

We respectfully request that the staff confirm, by supplement or FAQ, that a Covered User Interface Provider does not engage in "processing trade documentation" solely because its interface (i) makes available standardized forms of transaction agreement, registered before any transaction exists (by the platform, or by an issuer for transactions in its own securities), in an onchain registry that fixes each document's content and enumerates its permissible blanks; (ii) populates such a form's enumerated blanks with variables supplied by the user, as an incident of preparing the user's trading instructions; (iii) enables each party to execute the agreement by that party's own cryptographic signature over the agreement's full content, with the registry verifying each principal's own signature and recording per-party execution onchain; or (iv) displays, references, or records executed agreements on a distributed ledger, including as part of the publicly available DLT transaction records the Statement contemplates serving recordkeeping functions. Conversely, the exclusion is properly confirmed to reach a provider that performs documentation services for a party in connection with effecting or settling a transaction: preparing, reviewing, or transmitting settlement paperwork on a party's behalf, which is the function the broker-dealer regime has always supervised.

5. A related word on "executing or settling transactions."

The same interpretive principle should govern the adjacent exclusion. Where settlement is performed by deterministic code applying conditions the parties agreed to in advance, and the settlement function is permissionless, callable by either party or by anyone, a provider does not "execute or settle transactions" by making that function's availability known, or by surfacing to each party the controls to complete its own transaction. The excluded service is settlement performed for others as an intermediary function, not the existence of self-executing settlement the parties command themselves.

6. Purpose-built, single-protocol interfaces and the route-related conditions.

Several of the Statement's conditions are drafted against the aggregator paradigm: an interface that spans many venues and routes, selects among them, and presents the results. The Provider "selects one or more default trading venues . . . or distributed ledger trading systems"; "if only one potential execution route is displayed, the interface provides the ability to see additional routes, if applicable"; multi-route displays must carry objective sorting and filtering tools; commentary on any route is barred; the Provider must maintain policies to "evaluate, onboard, and audit the trading venues or distributed ledger trading systems"; and the disclosure categories are organized around routes, venues, and integrations. Those conditions make ready sense for an aggregator. But a large and growing class of Covered User Interfaces is purpose-built: designed for a single protocol, frequently by the protocol's own builder, which is precisely the affiliated arrangement the Statement's fourth condition contemplates and disciplines through disclosure and equal-terms access. MetaLeX's interfaces are of this kind: they connect to the cyberCORPs protocol and to nothing else, and there is no routing layer anywhere in the software, because there is nowhere to route. For such an interface there is no alternative route to display, no cross-venue sorting to offer, and no venue selection to perform.

Read woodenly, the route-related conditions could be taken to require what cannot exist, or worse, to imply that a single-protocol interface sits outside the framework because it cannot perform the

enumerated route functions. Neither result can be intended: an interface with one integration presents strictly fewer routing conflicts than an interface with many, not more, and the compliance-forward designs the Commission should want to encourage, interfaces that embed eligibility gating, disclosure, and documentation for one protocol they know completely, are overwhelmingly purpose-built. We therefore request that the staff confirm that (i) a purpose-built Covered User Interface connecting to a single protocol satisfies the route-related conditions where no alternative execution route exists, "if applicable" meaning what it says; (ii) the venue-evaluation condition scales to the single-venue case, met by evaluating, onboarding, and periodically reassessing the one connected system against the same objective factors; (iii) the route-, venue-, and counterparty-agnostic compensation condition is satisfied where the Provider's fee is uniform across all transactions on the served protocol; and (iv) the fourth condition's disclosure-and-equal-terms pathway operates identically for purpose-built interfaces, which are its paradigmatic case.

7. Conclusion.

The Statement's premise is that compliant crypto securities markets will be built by aligning software architecture with the securities laws' substantive protections. Documented trading is that alignment: identified counterparties, enforceable standardized terms, transfer restrictions carried in the instrument itself, and an execution record no back office could match. Interfaces to documented systems are the framework's natural allies, and the "processing trade documentation" exclusion should be clarified so that it polices intermediation rather than punishing documentation.

We appreciate the staff's invitation for comment and are available to discuss any aspect of this letter, including technical walkthroughs of the open-source systems described above.

Respectfully submitted,

MetaLeX Labs, Inc.

By: Gabriel Shapiro, President