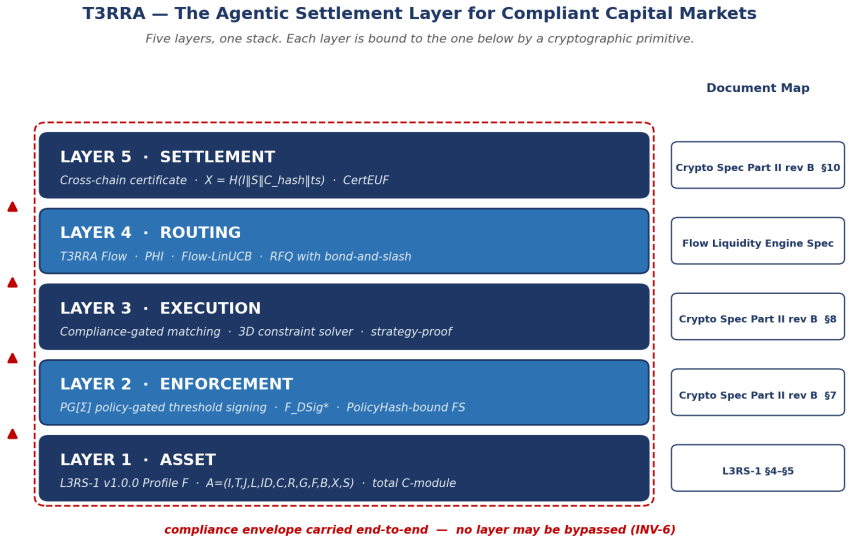
The Agentic Settlement Layer for Compliant Capital Markets



T3RRA Whitepaper

Whitepaper v3.2 · April 2026

L3RS-1 v1.0.0 · Profile F (Full) Conformant

Zurab Ashvil · T3RRA Research

Contents

Abstract

We present T3RRA, the first end-to-end platform for regulated real-world assets in which compliance is not a policy layer above the protocol but a cryptographic precondition of the protocol itself. T3RRA implements L3RS-1 v1.0.0 — the Layer-3 Regulated Asset Standard — at Profile F (Full), and contributes four constructions on top of it that are, to our knowledge, original to the literature.

First, we define policy-gated threshold signing, a strengthening of the standard threshold-signature ideal functionality F_DSig in which a signature exists only if a deterministic compliance predicate evaluates to accept on the transfer context. We give a generic compiler from any UC-secure threshold signature scheme to a policy-gated variant, and prove that the compiler preserves unforgeability and adds policy soundness — that is, the existence of a valid signature implies the existence of an accepting transcript of the compliance predicate. We instantiate the compiler with CMP-NI for ECDSA on secp256k1, DKLs23 for ECDSA where bandwidth is constrained, and FROST for EdDSA on edwards25519.

Second, we formalize compliance-gated matching, a market-microstructure construction in which the matching engine of an order book solves a three-way constraint over price, inventory, and a time-of-fill compliance evaluation. We show that compliance-gated matching is strategy-proof for honest counterparties under standard market-microstructure assumptions and that the strategy-proofness is preserved under continuous trading.

Third, we define the route admissibility predicate of T3RRA Flow, a four-conjunct constraint over jurisdictional masks, Travel Rule receipts, identity tiers, and cross-chain certificate continuity, and we show that route discovery under the predicate can be computed in $O(|V| \cdot |E| \cdot \log|V|)$ for a venue graph (V, E) using a modified Dijkstra search. We argue that the predicate is the first formal object in the aggregator literature that operationalizes FATF Recommendation 16 as a routing constraint.

Fourth, we give an unforgeability game for the L3RS-1 cross-chain certificate $X = H(I \parallel S \parallel C\_hash \parallel ts)$ under a 5-of-9 quorum bridge committee with a static, polynomially-bounded adversary, and we sketch the reduction from EUF-CMA of the underlying threshold signature.

We then describe the integrated T3RRA platform — wallet, marketplace, Flow, cross-chain bridge, reserve interface — that composes these four constructions into a single regulatory envelope that holds end-to-end across signing, matching, routing, and settlement, on a single chain or across chains. We close with concrete parameters, benchmarks methodology, an honest post-quantum roadmap, an audit and bug-bounty posture, an L3RS-1 conformance matrix at Profile F, and a list of open problems for the community.

1. Standards Alignment Statement

> ***Reference convention.*** *Throughout this document,* ***§N*** *(the section sign followed by a number) refers to* ***Section N of this Whitepaper****; clicking it jumps to that section. References prefixed with* ***L3RS-1*** *— for example,* L3RS-1 §15 *— refer to the corresponding section of the external L3RS-1 v1.0.0 standard, not to this document.*

This whitepaper describes the T3RRA Platform — an integrated stack for the issuance, custody, trading, and cross-chain settlement of regulated real-world assets. T3RRA is designed and operated as a Profile F (Full) conformant implementation of L3RS-1 v1.0.0, the Layer-3 Regulated Asset Standard published February 2026.

Every L3RS-1 normative construct — the Asset object A = (I, T, J, L, ID, C, R, G, F, B, X, S), the deterministic Transfer pipeline, the ComplianceModule decision function, the seven-state lifecycle, the four-way fee policy, the ReserveInterface, the GovernanceOverride object, the LegalMirror, the cross-chain certificate, and the seven theorems of L3RS-1 §15 — is implemented bit-for-bit on T3RRA. Where this paper uses standard terminology, the meaning is exactly the meaning given in L3RS-1, and the paper does not redefine, reinterpret, or relax any normative requirement.

This paper extends the standard with four constructions that are specific to T3RRA: policy-gated threshold signing (§6), compliance-gated matching (§7), the T3RRA Agent Mesh (§8), the route admissibility predicate (§9), and a cross-chain certificate unforgeability game with proof sketch (§10). These extensions are conservative — they add cryptographic strength on top of L3RS-1 without weakening any normative requirement of the standard, and they are upstream-contributed to the L3RS working group as candidate amendments for v1.1.

2. Executive Summary and Theses

2.0 The Three Walls

Every attempt to bring regulated capital on-chain hits three walls in sequence. The first wall is **compliance**: without a single cryptographic compliance object that travels intact across signing, matching, routing, and settlement, the asset is unsellable to regulated capital. The second wall is **jurisdiction**: without SPV structuring, legal opinions, and a jurisdiction map, the asset cannot leave the country it was born in. The third wall is **liquidity**: without a route admissibility predicate that can source fills from regulated venues no permissionless aggregator can touch, the asset is stranded on whichever venue first lists it. T3RRA is the only stack built to clear all three walls — as a Compliant Asset (L3RS-1), with Geographic Freedom (T3RRA Structuring), and Roaming Liquidity (T3RRA Flow). The five theses below are the technical defense of that three-wall frame.

2.1 Five Theses

This paper makes five claims and defends each in the body. Stated up front:

Thesis 1 — The Compliance Continuity Problem. The fundamental obstacle to regulated tokenized markets is not the absence of compliant issuance, custody, or venues. It is the absence of a single compliance object that travels intact across signing, matching, routing, and cross-chain settlement. Every existing stack rebuilds compliance from scratch at each layer; every rebuild is a place where regulators lose the audit trail and where adversaries find the seam.

Thesis 2 — Compliance Belongs in the Cryptography. The right place to enforce compliance is not the user interface, not the smart contract, and not the policy engine. It is inside the signing protocol itself, bound into the Fiat–Shamir challenge of the underlying zero-knowledge proofs, so that a signature whose context violates the compliance predicate is not merely refused — it is impossible to forge.

Thesis 3 — Compliance-Gated Matching Is the Right Microstructure for RWAs. An order in a regulated market is not a (price, size) tuple. It is a (price, size, compliance precondition) triple. The matching engine that fills these orders is not a 2D constraint solver but a 3D constraint solver, and the third dimension must be evaluated at fill time — not at order time — because the compliance state of a counterparty can change between order placement and fill.

Thesis 4 — Liquidity for Regulated Assets Requires a Routing Predicate, Not Just an Aggregator. Existing aggregators (1inch, CoW Swap, Paraswap, Uniswap X) optimize price × gas. Aggregators for regulated assets must optimize price × gas subject to a hard four-conjunct admissibility predicate that operationalizes FATF Recommendation 16 at the route level. The predicate is the moat: it is what allows T3RRA to source liquidity from regulated venues that no permissionless aggregator can integrate, and to refuse liquidity from venues that no permissionless aggregator can refuse.

Thesis 5 — Cross-Chain Movement Must Carry Compliance Evidence, Not Just Value. The cross-chain certificate $X = H(I \parallel S \parallel C\_hash \parallel ts)$ is the cryptographic object that lets a regulated asset travel between chains while preserving its compliance envelope. T3RRA's bridge is the first production implementation of this certificate as an unforgeable, replay-protected, EUF-CMA-rooted artifact.

2.2 What T3RRA Owns

T3RRA owns four pieces of technology that, in combination, no competitor has assembled. Each is treated formally in Part II:

#	Construction	Where
1	Policy-gated threshold signing — a generic compiler from F_DSig to a policy-gated F_DSig*, instantiated over CMP-NI, DKLs23, and FROST	§6
2	Compliance-gated matching — a 3D constraint solver for order books with a strategy-proofness theorem under continuous trading	§7
3	The route admissibility predicate — a four-conjunct constraint over jurisdictional masks, Travel Rule receipts, identity tiers, and cross-chain continuity, computable in $O(\|V\| \cdot \|E\| \cdot \log\|V\|)$	§9
4	A cross-chain certificate unforgeability game — EUF-CMA reduction with a 5-of-9 quorum bridge committee against a static polynomial adversary	§10

2.3 Why Now

L3RS-1 is the first published meta-standard with sufficient formal rigor to be implemented as cryptography rather than as policy text. Its publication in February 2026 created the substrate on which T3RRA's claims become verifiable rather than aspirational. T3RRA's certification as a Profile F conformant implementation makes it the reference platform for the standard. The market opportunity (Section 3) is large. The technical opportunity (Sections 6–9) is unique. The window in which a single platform can become the reference implementation of a normative standard is short.

Liquidity fragmentation is the unsolved problem. Every year of tokenized RWA growth since 2020 has made the liquidity wall taller, not shorter: more venues, more chains, more bridges, more aggregators, each re-implementing compliance in a different way. The route admissibility predicate in §9 is the first construction that turns that wall into a routable graph. This is the capability the market does not yet have and cannot build without a compliance-continuous substrate underneath it.

3. Market Opportunity and the Compliance Continuity Problem

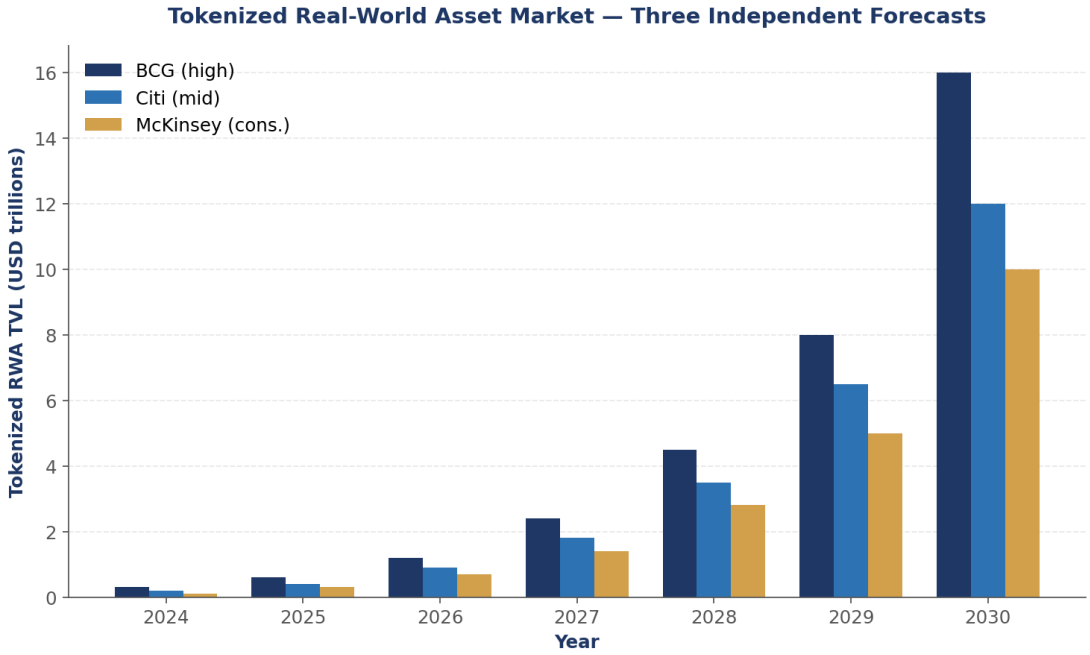


Figure 1. *Tokenized RWA market — three independent forecasts (BCG, Citi, McKinsey) for 2024–2030.*

3.1 Trapped Capital and the Unlock

The $16T tokenized-RWA forecast and the $379T global real-estate stock are not the interesting numbers. The interesting number is how much of that capital is **trapped** — owned by regulated institutions that cannot deploy it into any venue that fails the three walls. A compliant asset with geographic freedom and roaming liquidity is the first construction that lets that capital move without leaving the regulated perimeter. Every number in the table below should be read as trapped capital waiting for a route, not addressable market waiting for a product.

3.1.1 The Numbers

Sector	Size	Source / Year	Relevance
Global Real Estate	$379.7T	Savills 2023	Foundational vertical
Tokenized RWAs (2030 forecast)	$16T	BCG / 21Shares 2023	Direct TAM
Tokenized Treasuries (2030 forecast)	$2.4T	BlackRock / Citi 2024	INDUSTRY_STABLE assets
Stablecoin Float	$340B	DefiLlama Q1 2026	DvP cash leg
DeFi TVL	$160B	DefiLlama Q1 2026	Permissionless source
Institutional AUM	$100T+	Boston Consulting 2024	L4 onboarding tier
FATF VASPs registered	1,200+	FATF 2025 plenary	Travel Rule counterparties

3.2 The Compliance Continuity Problem

We name the central problem of regulated tokenized markets the Compliance Continuity Problem, defined as follows:

> **Definition 3.1 (Compliance Continuity Problem).** Given a regulated asset A with compliance predicate C, an originating party s, and a beneficiary r, design a system in which every state transition that touches A — issuance, custody, signing, listing, matching, routing, cross-chain hop, settlement, and reserve attestation — evaluates the same predicate C against the same context, with no semantic drift across system boundaries, and with the property that any transition for which C returns reject is impossible to perform, not merely refused at the UI.

The existing stack fails this definition at every boundary. Issuance platforms (Securitize, Tokeny, Polymath) embed compliance at the smart-contract level but cannot extend it to wallets they do not control. Custody providers (Fireblocks, Anchorage, BitGo) enforce compliance at the policy-engine level but treat the asset as an opaque token. DeFi venues (Uniswap, Curve, Balancer) ignore compliance entirely. Aggregators (1inch, CoW, Paraswap) re-implement KYC checks in a fourth way. Cross-chain bridges (LayerZero, Wormhole, CCIP, Axelar) carry value but not compliance evidence. Each layer has a different model of what 'compliant' means, and each boundary is a place where the audit trail discontinues and where regulators must trust the operator's word that the next layer will re-evaluate the same predicate.

T3RRA's organizing thesis is that this problem is solved by collapsing the layers around a single compliance object — the L3RS-1 ComplianceModule of the asset — and binding that object cryptographically into every transition. The remainder of this paper is a proof of construction for this thesis.

4. System Architecture

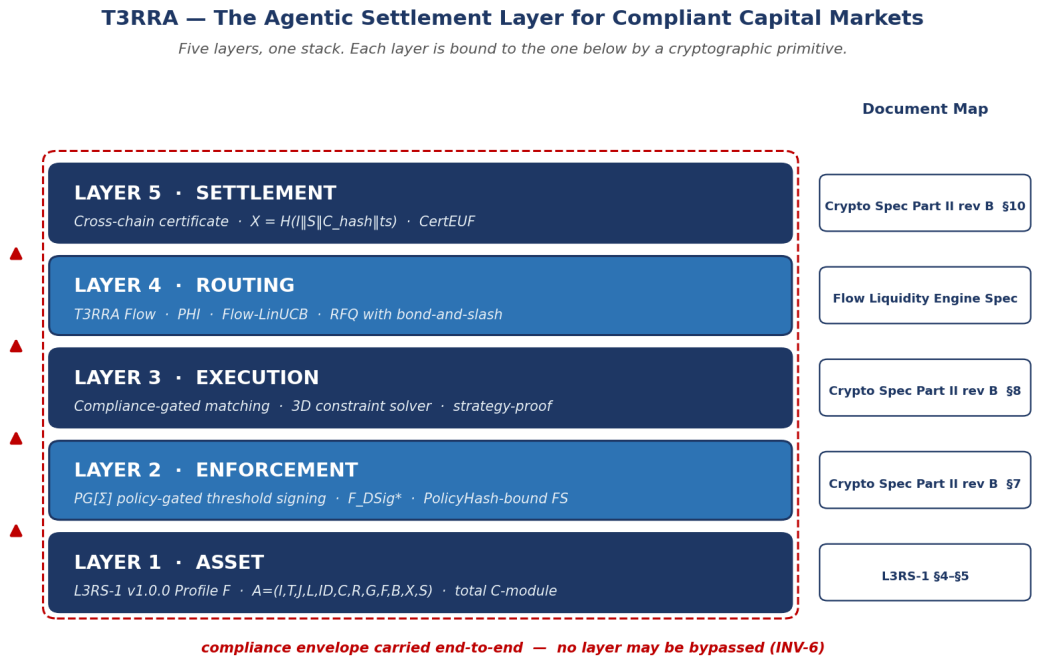


Figure 2. *The T3RRA five-layer stack. Each layer is necessary; none is sufficient on its own.*

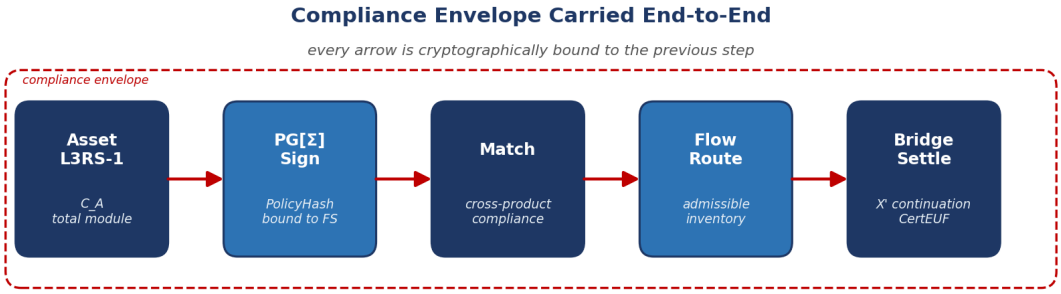


Figure 3. *Compliance continuity — the envelope is carried end-to-end and every arrow is cryptographically bound to the previous step.*

4.1 Five Subsystems, One Object Model, One Predicate

T3RRA is composed of five subsystems. Each is independently auditable but none is independently deployable: removing any one breaks compliance continuity.

Subsystem	Responsibility	L3RS-1 Anchor	Section
Issuance	Mints L3RS-1 Asset objects, registers LegalMirror artifacts	§4, §13	§5
Wallet	Threshold MPC custody with policy-gated signing	§5, §7	§6
Marketplace	Compliance-gated CLOB and RFQ venue	§6, §7	§7
Flow	Liquidity aggregation under route admissibility predicate	§7, §10	§9
Settlement	Cross-chain certificate, reserve attestation, fee routing	§8, §9, §10	§10–12

4.2 The Single Object Model

Every subsystem operates on the same L3RS-1 Asset object:

```
A = (I, T, J, L, ID, C, R, G, F, B, X, S)
```

where I is the immutable Asset_ID, computed at issuance as I = H(pk_issuer ‖ ts ‖ nonce); T is the asset type drawn from {CBDC, INDUSTRY_STABLE, REGULATED_SECURITY, UTILITY, GOVERNANCE, STORAGE_BACKED}; J is the jurisdictional mask, a bitset over ISO 3166-1 codes; L is the legal binding, a content-addressable pointer into the LegalMirror; ID is the minimum identity binding level required of any counterparty; C is the ComplianceModule, a deterministic decision function realized as version-pinned bytecode whose hash C_hash is part of the asset's identity; R is the ReserveInterface; G is the GovernanceOverride; F is the four-way fee policy; B is the burn policy; X is the cross-chain certificate state; and S is the lifecycle state, drawn from the seven-state machine of L3RS-1 §6.

There is exactly one representation of an asset across the entire platform. Marketplace listings, Flow quotes, cross-chain certificates, fee receipts, reserve attestations, and wallet signatures all carry I as a primary key and read C from the same bytecode artifact. There is no second model, no abstraction layer, no internal token wrapper.

5. The L3RS-1 Asset Lifecycle on T3RRA

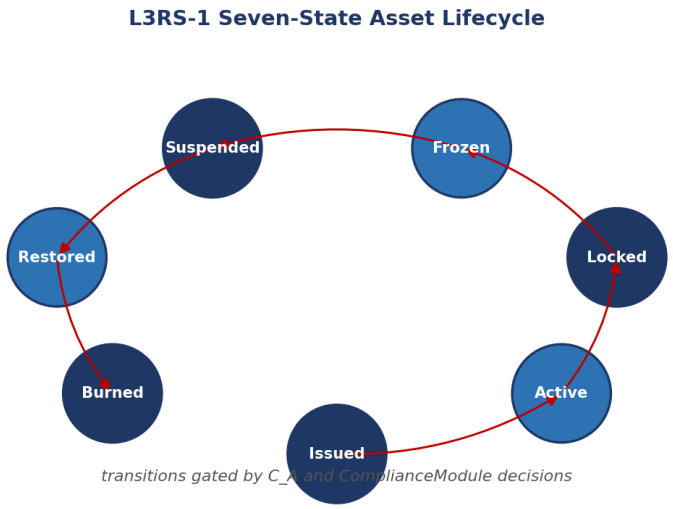


Figure 4. *L3RS-1 seven-state asset lifecycle. Every transition is gated by the asset's compliance module C_A.*

5.1 The Seven-State Machine

State	Meaning	Permitted Transitions Out
ISSUED	Minted, not yet activated	→ ACTIVE (issuer-signed)
ACTIVE	Fully transferable subject to C	→ RESTRICTED, FROZEN, REDEEMED, BURNED
RESTRICTED	Whitelisted recipients only	→ ACTIVE, FROZEN
FROZEN	All transfers blocked, balances visible	→ ACTIVE, SUSPENDED (governance)
SUSPENDED	Issuer-paused, no view	→ ACTIVE (governance + issuer)
REDEEMED	Burned against reserve withdrawal	Terminal
BURNED	Burned without reserve claim	Terminal

5.2 Issuance Pipeline

An issuer onboarded under L4 KYB publishes a draft Asset object through the Issuance subsystem. The draft passes through a deterministic validator that checks ten conditions — legal binding L is anchored in the LegalMirror, ComplianceModule C compiles to bytecode whose hash matches the declared C_hash, ReserveInterface R is reachable and returns a verifiable attestation, GovernanceOverride G specifies a 2/3 quorum, jurisdictional mask J is well-formed, fee policy F sums to 1, identity tier ID is in {L0, L1, L2, L3, L4}, asset type T is in the L3RS-1 enumeration, the issuer's pk_issuer is bound to a verifiable credential with a recognized root, and the nonce is fresh. Only after the draft passes the validator does the Issuance contract compute $I = H(\text{pk\_issuer} \parallel \text{ts} \parallel \text{nonce})$, bind it into the asset, and write the asset into ISSUED state. There is no out-of-band path to ISSUED.

6. Policy-Gated Threshold Signing

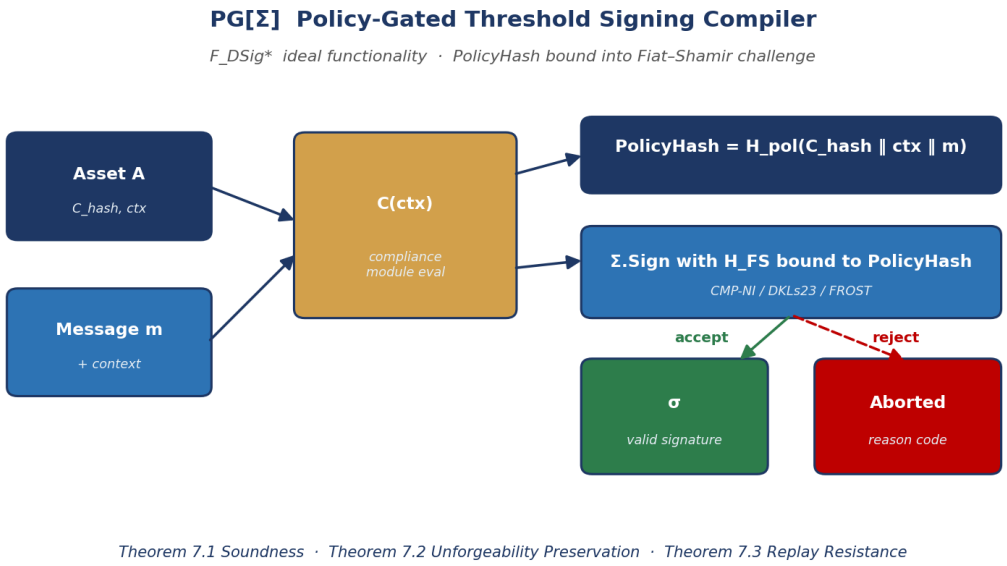


Figure 5. *PG[Σ] compiler — the compliance module C is evaluated, its decision binds the PolicyHash into the Fiat–Shamir challenge of the underlying threshold signature, and a non-compliant action produces an Aborted reason rather than a signature.*

This section is the cryptographic heart of the T3RRA Wallet. We define policy-gated threshold signing as a strengthening of the standard threshold-signature ideal functionality F_DSig, give a generic compiler from any UC-secure threshold scheme to a policy-gated variant, instantiate the compiler over CMP-NI, DKLs23, and FROST, and prove that the compiler preserves unforgeability and adds policy soundness.

6.1 Setting and Notation

6.2 The Policy-Gated Ideal Functionality F_DSig*

The ideal functionality F_DSig* strengthens the standard threshold-signature ideal functionality F_DSig of Canetti (FOCS 2001) with a deterministic compliance precondition. The signing transcript is bound to the compliance decision via the PolicyHash domain separator:

> **Definition 6.1 (F_DSig*).** F_DSig* is identical to F_DSig with the following addition. On input (Sign, sid, m, ctx, C, C_hash) from at least t parties: F_DSig* first verifies that Hash(bytecode(C)) = C_hash; if not, it ignores the request. F_DSig* then evaluates b ← C(ctx); if b = reject, F_DSig* outputs (Aborted, sid, reason) to all parties and halts. Only if b = accept does F_DSig* proceed to compute and output a signature σ on m. Furthermore, the signature transcript binds C_hash and ctx into the random-oracle query as PolicyHash = H(C_hash ‖ ctx ‖ m), and the signing operation is parameterized by PolicyHash as a domain separator in the Fiat–Shamir challenge of every zero-knowledge proof inside Σ.

6.3 The Generic Compiler

We give a generic compiler that turns any UC-secure threshold signature scheme into a realization of F_DSig*. The compiler is intentionally minimal — it does not change the underlying signing protocol;

it adds a deterministic prelude that every party runs locally before contributing any round-1 message, and it modifies the Fiat–Shamir challenge of any zero-knowledge proof inside the scheme to bind PolicyHash.

```
Compiler PG[Σ](C, C_hash):
  On (Sign, sid, m, ctx) from party P_i:
    1. P_i checks Hash(bytecode(C)) = C_hash.       // policy authenticity
       If not, ABORT(InvalidPolicy).
    2. P_i evaluates b_i ← C(ctx).                   // local prelude
       If b_i = reject, EMIT ComplianceAbort(sid, reason_i)
       and refuse to participate in round 1.
    3. P_i computes PolicyHash = H(C_hash ‖ ctx ‖ m).
    4. P_i runs Σ.Sign with the modification that
       every Fiat–Shamir challenge in Σ is computed
       as H_FS(transcript ‖ PolicyHash) instead of
       H_FS(transcript).
    5. Output σ on success.
```

6.4 Security Theorems

Theorem 6.1 (Policy Soundness). *For any compliance predicate C realized as deterministic bytecode with hash C_hash, the protocol PG[Σ] UC-realizes F_DSig*. In particular, if any honest party P_i computes b_i = reject in step 2, no signature σ verifying under the joint public key can be produced by any subset of t parties of which P_i is a member. Consequently, the existence of a verifying signature on (m, ctx) under PolicyHash implies that at least t parties evaluated b_i = accept.*

Proof sketch. By construction, an honest party that evaluates reject does not contribute to round 1 and emits no protocol message that is usable by the remaining parties. Since the threshold is t, any signing coalition includes at least one honest party in the dishonest-majority model assumed by Σ. Therefore at least one accept-evaluating honest party is required for a signature to exist. The PolicyHash domain separator in the Fiat–Shamir challenge prevents replay of a transcript produced under one (C_hash, ctx) against a different policy or context, since the challenges differ by a uniformly random oracle output. □

Theorem 6.2 (Unforgeability Preservation). *If Σ is EUF-CMA secure as a (t,n) threshold signature scheme realizing F_DSig, then PG[Σ] is EUF-CMA secure as a (t,n) threshold signature scheme realizing F_DSig*. Concretely, for any PPT adversary A against PG[Σ] there is a PPT adversary B against Σ such that Adv^EUF-CMA(A, PG[Σ]) ≤ Adv^EUF-CMA(B, Σ) + q_H · 2^(−λ), where q_H is the number of random-oracle queries made by A.*

Proof sketch. B simulates PG[Σ] for A by relaying signing queries to its own Σ oracle. The only addition is the PolicyHash binding in the Fiat–Shamir challenge, which is simulated by lazy programming of H_FS at each query. A forgery on PG[Σ] yields a forgery on Σ except with probability q_H · 2^(−λ) accounting for hash collisions on PolicyHash. □

Theorem 6.3 (Policy Replay Resistance). *A signature produced under PG[Σ] with parameters (C_hash, ctx) is computationally unforgeable as a signature for any (C_hash', ctx') ≠ (C_hash,*

ctx). In particular, an upgrade of the ComplianceModule that produces a new C_hash invalidates all prior signatures as authorizations under the new policy.

Proof sketch. Direct from the random-oracle property of H_FS over PolicyHash. The challenge bits depend on (C_hash, ctx); changing either yields a uniformly fresh challenge with probability $1 - 2^{(-\lambda)}$. □

6.5 Instantiation: CMP-NI for ECDSA

We instantiate PG[Σ] over CMP-NI (Canetti, Makriyannis, Peled, ACM CCS 2020) for ECDSA on secp256k1. CMP-NI is a UC-secure threshold ECDSA protocol with three online rounds, identifiable abort, and proactive resharing. The Fiat–Shamir challenges of the Paillier-encryption ZK proofs and the discrete-log proofs in CMP-NI are modified per the compiler in §6.3 to bind PolicyHash. The protocol round complexity, message complexity, and bandwidth are unchanged.

6.6 Instantiation: DKLs23 for ECDSA

For deployments where bandwidth dominates, we instantiate PG[Σ] over DKLs23 (Doerner, Kondi, Lee, shelat, IEEE S&P 2024). DKLs23 has two online rounds after a reusable preprocessing phase. The PolicyHash binding is added to the Fiat–Shamir challenges of the OT extension and consistency proofs. As with CMP-NI, the underlying protocol structure is preserved.

6.7 Instantiation: FROST for EdDSA

For ed25519 chains we instantiate PG[Σ] over FROST (Komlo and Goldberg, SAC 2020). FROST has two rounds and produces standard EdDSA signatures verifiable by any EdDSA verifier. The PolicyHash is bound into the Schnorr challenge.

6.8 Identifiable Abort and the AbortReport

All three instantiations support identifiable abort: any deviation from the protocol by a party P_j is attributed by the honest parties to P_j and surfaced as an AbortReport(P_j, evidence). Policy aborts in step 2 of the compiler are surfaced as ComplianceAbort(P_i, reason) where reason is a structured object that names the L3RS-1 clause that the predicate cited in returning reject. Both forms of abort produce verifiable artifacts that regulators and auditors can replay.

6.9 Hardware Attestation Chain

Each party P_i is a process running in an attested hardware environment: iOS Secure Enclave or Android StrongBox for mobile shares, FIPS 140-3 Level 3 HSMs inside AWS Nitro Enclaves with measured-boot attestation chains rooted in the Nitro hypervisor for backend shares, and independently-operated HSMs for recovery custodian shares. Attestation evidence is bound into the DKG transcript at key generation and re-verified at every signing ceremony. The attestation chain itself is anchored as a LegalMirror artifact under L3RS-1 §13.

7. Compliance-Gated Matching

This section formalizes the matching engine of the T3RRA Marketplace. We define compliance-gated matching as a market-microstructure construction in which the matching engine solves a three-way constraint over price, inventory, and a time-of-fill compliance evaluation. We give the formal model and prove a strategy-proofness theorem for honest counterparties under continuous trading.

7.1 The Standard Order Book

In the standard literature an order is a tuple (side, price, size). The matching engine maintains a price-time-priority order book and at each event time produces a fill if the best bid crosses the best ask. Compliance, where it exists, is a UI-level filter applied either at order placement (KYC at onboarding) or at withdrawal (sanctioned-address screening at exit). The matching engine itself is compliance-blind.

7.2 The Compliance-Gated Order

Definition 7.1 (Compliance-Gated Order). A compliance-gated order on T3RRA is a tuple ω = (side, price, size, asset_id I, owner pk, identity_credential vc, jurisdiction j, compliance_proof π) where π is a verifiable proof that, at order placement time, C_I(ctx_self) = accept where ctx_self is the order owner's self-context against the asset's ComplianceModule C_I. The order is admitted to the book only if Verify(π) succeeds.

7.3 The Three-Way Constraint Solver

Definition 7.2 (Compliance-Gated Match). Given two orders ω_b = (buy, p_b, s_b, I, pk_b, vc_b, j_b, π_b) and ω_s = (sell, p_s, s_s, I, pk_s, vc_s, j_s, π_s) such that p_b $\geq$ p_s, the matching engine constructs the prospective transfer context ctx = (pk_s, pk_b, I, min(s_b, s_s), (j_s, j_b), (vc_s, vc_b), travel_rule_payload, ts, nonce) and evaluates b $\leftarrow$ C_I(ctx). The match commits if and only if b = accept; otherwise the engine emits a structured reject reason and the orders remain on the book.

7.4 Strategy-Proofness

Theorem 7.1 (Strategy-Proofness Under Honest Compliance). *Assume the standard market-microstructure assumptions (price priority, time priority, no front-running by the venue) and assume the ComplianceModule C is deterministic and total. Then compliance-gated matching is strategy-proof for any honest counterparty: no honest counterparty can improve its expected fill outcome by misrepresenting its identity_credential, its jurisdiction, or by submitting a compliance_proof π for a context other than its true self-context.*

Proof sketch. Suppose for contradiction that an honest counterparty P with true credential vc strictly improves its fill outcome by submitting credential vc' $\neq$ vc. Since C is deterministic, the outcome of any prospective match against P is a function of the context constructed at fill time, which the matching engine constructs from the on-file credential. The on-file credential is bound to pk by the KYC vendor's signature over (pk, vc) at registration; substituting vc' requires either (a) producing a forged KYC signature, contradicting the unforgeability of the KYC vendor's signing key, or (b) re-onboarding under a different pk', in which case any fill accrues to pk' rather than pk and is not an improvement to P. Therefore no improvement is possible. The same argument

applies to jurisdiction misrepresentation and to compliance_proof substitution, since the verifying signature on π includes the credential identifier. □

Theorem 7.2 (Continuity). *Strategy-proofness in Theorem 7.1 is preserved under continuous trading. That is, if a sequence of orders {ω_k} is processed under compliance-gated matching, no honest counterparty can improve its outcome on order ω_k by deviating in its prior order ω_{k-1}, ω_{k-2}, …, ω_1.*

Proof sketch. Each order in the sequence is independently bound to the counterparty's credential at submission time, and the credential cannot be silently changed without re-onboarding. Therefore deviation in any prior order does not affect the credential bound to the current order, and Theorem 7.1 applies independently at each k. □

7.5 Sanctioned-Address Refusal Property

A practical consequence of compliance-gated matching is the property every regulator asks for: a sanctioned address cannot fill on T3RRA, even if it bids the best price. Concretely, if a counterparty's credential vc is updated by the KYC vendor to indicate sanctioned status, the next match attempt against that counterparty constructs a context with a sanctioned-status flag, and any reasonable C_I returns reject. The order remains on the book — visible, but unfillable. There is no way to bypass the check by routing around the matching engine, because the only path to settlement is through a wallet signature, which is also gated by the same C_I via the policy-gated signing protocol of §6.

7.6 Order Types and Auction Models

Order Type	Microstructure	C Evaluated
Limit (CLOB)	Price-time priority continuous matching	At each fill
RFQ Block	Quote disclosed to counterparty after C_I check	Before quote disclosure and at fill
Periodic Auction	Discrete clearing at intervals	At clear time for each pair
Primary Issuance	Subscription against issuer's ComplianceModule	At subscription doc validation
Redemption	Burn against ReserveInterface attestation	At burn + at attestation reconciliation

7.7 Settlement Atomicity

A T3RRA fill settles via on-chain atomic delivery-versus-payment. The asset leg is a policy-gated signature from the seller's wallet; the cash leg is a policy-gated signature against an INDUSTRY_STABLE asset (USDC, USDT, EURC, or a CBDC where available). Both signatures are produced inside a single MPC session whose abort semantics guarantee that neither leg can settle without the other. T+0 atomic, irrevocable, no on-platform credit risk.

8. The T3RRA Agent Mesh

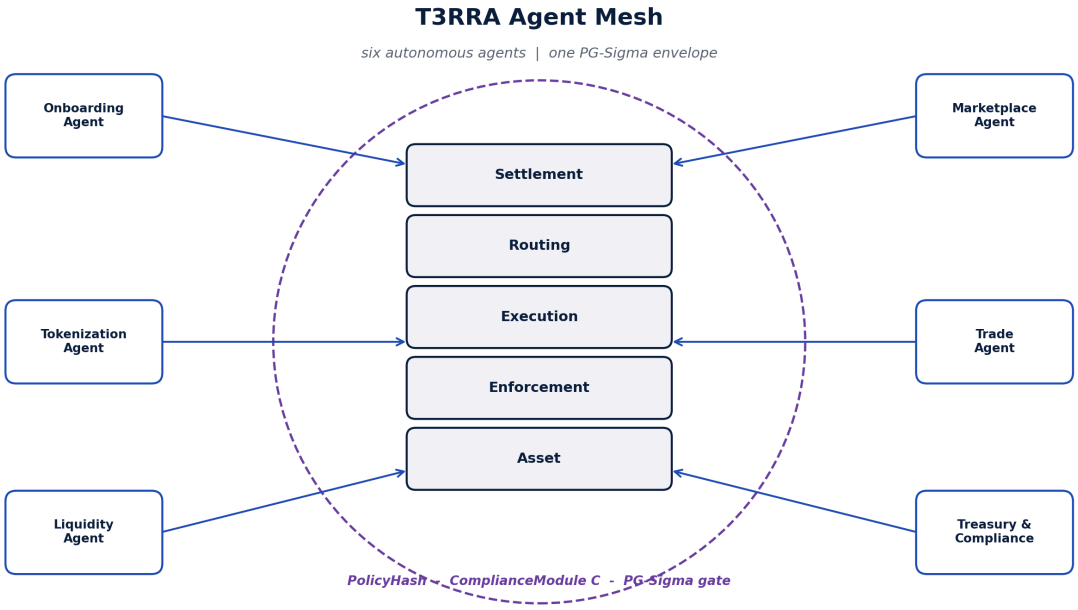


Figure 6. *The T3RRA Agent Mesh — six classes of autonomous agents operate against the same L3RS-1 + PG[Σ] primitives as human users, with every action gated by the ComplianceModule C and the PolicyHash.*

Capital markets are a coordination problem at scale: thousands of counterparties, thousands of jurisdictions, hundreds of asset classes, and a regulatory perimeter that changes weekly. The thesis of this section is that the L3RS-1 + PG[Σ] substrate is not only a settlement substrate for humans — it is also the first substrate on which autonomous software agents can act inside a regulated capital market without weakening any guarantee that humans rely on. We call the resulting system the T3RRA Agent Mesh.

The Agent Mesh is a permissioned set of autonomous agents that operate across every layer of the T3RRA stack — onboarding, tokenization, marketplace, trade execution, liquidity, and treasury — and whose every action is indistinguishable from a human action at the protocol layer. The same ComplianceModule C, the same PolicyHash, the same policy-gated signing ceremony, the same route admissibility predicate, the same cross-chain certificate. Agents do not bypass the regulatory envelope; they live inside it.

8.1 Design Principle: Agents Are First-Class Citizens, Not Privileged Bypasses

There is a tempting alternative architecture in which agents have a privileged channel — an admin API, a service account, a 'machine user' flag — that lets them act faster or with weaker checks than humans. We reject this architecture categorically. Every privilege gap between humans and machines becomes a privilege gap an attacker can exploit, and every privilege gap is a place where the formal guarantees of L3RS-1 stop holding. Instead, every agent in the T3RRA Agent Mesh is a delegated signer in the same sense as any human keyshare holder: it holds a share, it participates in PG[Σ] ceremonies, its actions evaluate against the same C, its identity tier is verifiable, and its compromise is bounded by the same threshold-security argument that bounds the compromise of any other party.

> **Principle 8.1 (Agent–Human Indistinguishability).** *For every agent action a and every human action h that target the same Asset I, the protocol-level transcript of a is indistinguishable from the*

transcript of h. In particular: (i) the same ComplianceModule C_I evaluates both; (ii) the same PolicyHash is bound into the Fiat–Shamir challenge; (iii) the same route admissibility predicate gates any liquidity touched by either; (iv) compromise of a is bounded above by the standard (t, n)-threshold compromise bound of the underlying signing scheme.

8.2 The Six Agent Classes

The Agent Mesh is composed of six canonical agent classes, each tied to a layer of the stack. The classes are not exhaustive — third parties may register additional agents through the same delegation interface — but they cover the workflows that, in our experience, dominate the operating cost of a regulated capital-markets venue.

Class	Layer	Primary Workflow
Onboarding Agent	Asset / Identity	KYC/KYB intake, jurisdiction routing, ID-tier provisioning, ComplianceModule C selection
Tokenization Agent	Asset	Asset structuring, lifecycle init, document binding, PolicyHash derivation, LegalMirror anchoring
Marketplace Agent	Execution	Listing curation, counterparty discovery, RFQ generation, quote evaluation under strategy-proofness
Trade Agent	Execution / Settlement	Order construction, route admissibility check, PG[Σ] co-signing as delegated signer, certificate verification
Liquidity Agent	Routing	Flow pool participation, PHI-aware rebalancing, Flow-LinUCB exploitation/exploration on behalf of LPs
Treasury / Compliance Agent	Settlement / Governance	TAF accounting, MBSR triggers, regulatory reporting, anomaly escalation, governance vote preparation

8.3 Delegation Under PG[Σ]

Agents become signers through a one-time delegation ceremony in which a principal (an institutional account, a DAO multisig, or a retail wallet at a sufficient identity tier) authorizes the agent as an additional party in a (t, n)-threshold quorum. The delegation is itself a PG[Σ] signature whose PolicyHash binds the agent's role, its capability scope, its expiry, and the set of asset classes it is permitted to touch. The formal treatment of this construction lives in Cryptographic Specification Part II rev B §11 here we record only the security claim:

Theorem 8.2 (Agent Delegation Soundness, informal). *Let an agent A be delegated as a signer with capability scope σ via a PG[Σ] delegation ceremony bound to PolicyHash h_σ. Then for every signing ceremony in which A participates, the existence of a valid output signature implies (i) an accepting transcript of the underlying ComplianceModule C on the transfer context, and (ii) the transfer context is consistent with σ. In particular, A cannot produce a valid signature for any action outside σ, and a compromise of A's keyshare is bounded above by the standard (t, n)-threshold compromise bound.*

8.4 Onboarding and Tokenization

The Onboarding Agent is the first point of contact for institutional and retail users. It collects KYC and KYB artifacts using OCR-based document verification and ML identity matching against sanctions and PEP lists, routes the user to the correct jurisdictional ComplianceModule, provisions an identity tier (L0–L4) bound as a verifiable credential, and hands the user off to a wallet provisioning ceremony in which the user's keyshares — and any agent shares the user authorizes — are generated. The agent never holds the user's identity documents in plaintext beyond the verification window; documents are encrypted to the LegalMirror and discarded from agent memory.

The Tokenization Agent automates the construction of an L3RS-1 Asset object A = (I, T, J, L, ID, C, R, G, F, B, X, S) from issuer inputs. Given an asset description, target jurisdictions, and a legal binding, the agent runs ML-driven valuation models against comparable market data, extracts material terms from deeds, prospectuses, and financial reports via NLP, selects the appropriate L3RS-1 type T, derives the jurisdictional mask J, anchors the legal binding L into the LegalMirror, instantiates the ComplianceModule C from a versioned library of bytecode artifacts, derives the PolicyHash, initializes the seven-state lifecycle in DRAFT, and submits the Asset for issuer review. A token issuance that historically required twelve weeks of legal and engineering work converges to a same-day workflow with cryptographic provenance for every step.

8.5 Marketplace and Trade Agents

The Marketplace Agent operates inside the compliance-gated CLOB and RFQ venues defined in §7. Its workflow is curation rather than execution: it uses LLM-based matchmaking over structured listing metadata and unstructured counterparty preferences to surface listings that match a counterparty's identity tier and jurisdictional mask, it generates RFQs to a permissioned set of liquidity providers with the correct VASP credentials, and it evaluates returning quotes against the strategy-proofness invariant of Theorem 7.1. The Marketplace Agent never executes a trade on behalf of a user without an explicit human-in-the-loop confirmation, unless the principal has signed a PG[Σ] delegation with explicit autonomous-execution scope.

The Trade Agent does the actual execution under such a delegation. It constructs the order, runs the route admissibility check (J ∧ T ∧ ID ∧ X) over the candidate venue set, participates in the PG[Σ] signing ceremony as a delegated signer, verifies the cross-chain certificate continuity for any cross-chain hops, and reports the structured fill record to the Treasury Agent. Because the Trade Agent's PG[Σ] binding includes an expiry and a per-asset notional cap, even a fully compromised Trade Agent cannot exfiltrate value beyond its delegation scope.

8.6 Liquidity Agents and the Flow Bandit

The Liquidity Agent is the agent class that interacts most directly with T3RRA Flow (§9). It wraps the Flow-LinUCB contextual bandit with an LP-facing strategy: given an LP's risk preferences and capital, it allocates liquidity across pools — including AMM-style constant-product and concentrated-liquidity strategies wrapped as admissible Flow venues — monitors the per-venue Pool Health Index,

rebalances when PHI components drift outside tolerance, and harvests Flow credits as returns. The bandit's exploration budget is itself bounded by a PG[Σ] delegation parameter, so even an aggressively-tuned Liquidity Agent cannot route LP capital through inadmissible venues. The full treatment of bandit-as-agent is in the Flow Liquidity Engine paper §6.

8.7 Treasury and Compliance Agents

The Treasury / Compliance Agent runs the back office. It accounts for the Trade Activity Fee (TAF) at 0.125% of every fill, attributes each TAF receipt to the four-way fee policy of L3RS-1 §8, monitors the MBSR (minimum buy-side reserve) ratio against the deflation curve of the Tokenomics paper §10, generates regulatory reports against the configured set of authorities, runs deep-learning anomaly detection over fill streams and on-chain blockchain analytics over counterparty wallets, proposes AI-driven risk hedges (delta, duration, FX) for the treasury book, and escalates anomalies to a human compliance officer. The Treasury Agent has no execution capability — its delegation scope is read-only over asset state and write-only over reporting artifacts.

8.8 Why Agents Make T3RRA Cheaper, Not Just Faster

The first-order effect of the Agent Mesh is throughput: a tokenization workflow that would take a small team a quarter to complete converges to a single day. The second-order effect is cost: human operating cost is the dominant line item in every regulated capital-markets venue, and removing it from the workflow without removing any of the regulatory guarantees lets T3RRA quote a Trade Activity Fee one or two orders of magnitude below incumbent venues while still capturing healthy unit economics. The third-order effect — and the one we believe will dominate the long run — is composability: an Agent Mesh exposes a clean machine interface to every other agent in the broader ecosystem. The first capital market that is fully agent-addressable will, we expect, attract a disproportionate share of agent-driven flow.

8.9 Risks and Bounded Authority

Autonomous agents introduce operational risk. We bound that risk through four mechanisms. (i) Every agent's authority is scoped by a PG[Σ] delegation with an explicit capability set, an explicit notional cap, and an explicit expiry. (ii) Every agent action evaluates against the same ComplianceModule C as a human action, so an agent compromise cannot bypass compliance. (iii) Every agent action is bounded by the (t, n)-threshold security of the underlying signing scheme; compromise of a single agent is no worse than compromise of a single keyshare holder. (iv) The Treasury / Compliance Agent monitors every other agent for anomalies and escalates to human review on first deviation from baseline. We treat agents as we treat any other party: powerful, useful, and never trusted unconditionally.

8.10 AI Capability Map

The table below maps the AI capabilities exposed across the T3RRA stack to their owning agent class. Every capability runs inside the same PG[Σ] envelope as its host agent — there are no off-protocol AI side channels.

AI Capability	Owning Agent Class
OCR document verification	Onboarding
Sanctions / PEP identity matching	Onboarding
ML-based asset valuation	Tokenization
NLP deed and prospectus parsing	Tokenization
LLM counterparty matchmaking	Marketplace
Quote evaluation under strategy-proofness	Marketplace
Autonomous order construction and PG[Σ] co-signing	Trade
Flow-LinUCB contextual bandit	Liquidity
AMM strategy wrapping (CPMM, CLMM)	Liquidity
PHI-aware rebalancing	Liquidity
Deep-learning anomaly detection	Treasury / Compliance
On-chain blockchain analytics	Treasury / Compliance
AI-driven risk hedging proposals	Treasury / Compliance
Regulatory report generation	Treasury / Compliance

9. The Route Admissibility Predicate

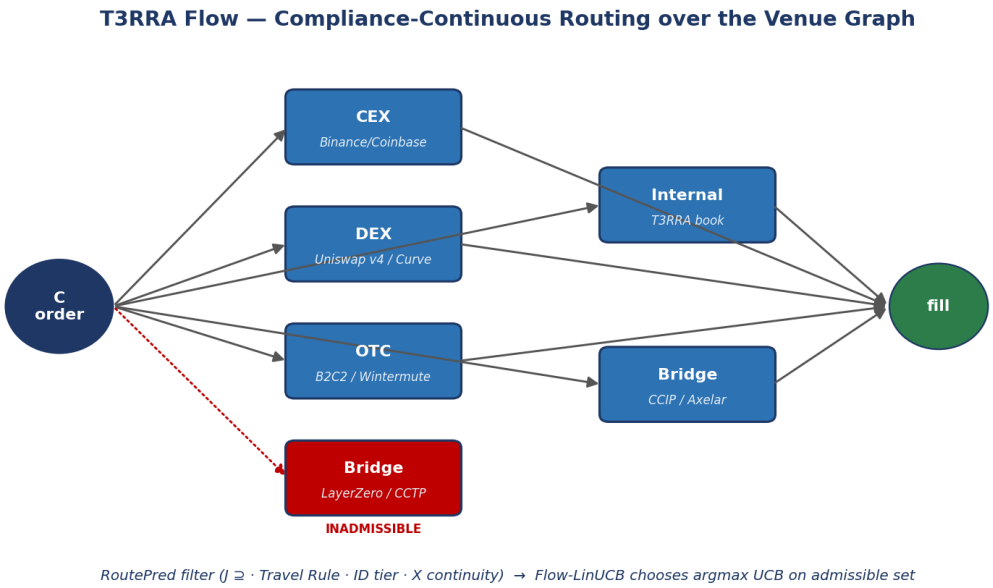


Figure 7. *T3RRA Flow over the venue graph — the route predicate filters inadmissible venues before any best-execution scoring runs.*

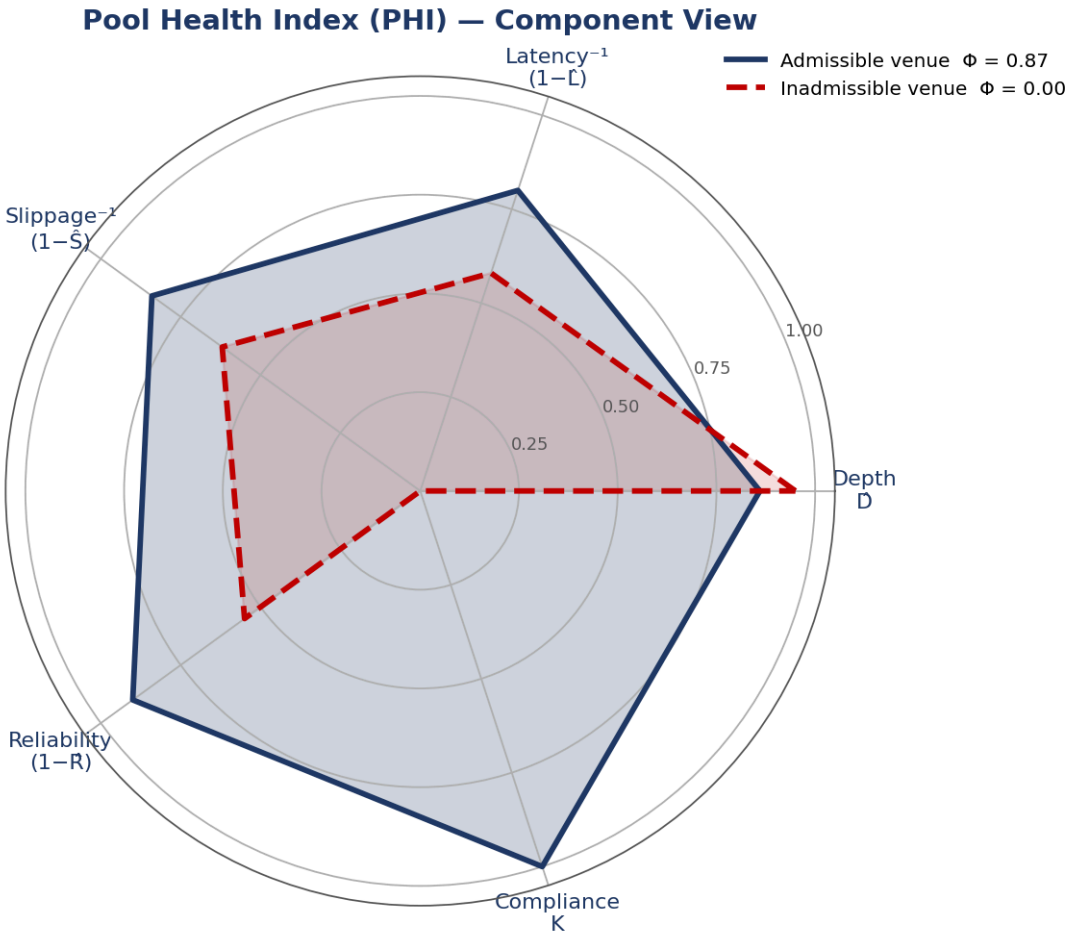


Figure 8. *Pool Health Index (PHI) — depth, latency, slippage, reliability, and the compliance gate K. A venue with K = 0 (Venue C) is structurally inadmissible regardless of the other components.*

This section formalizes T3RRA Flow. Where the marketplace provides native liquidity for assets listed on T3RRA, Flow goes outward — discovering, scoring, and routing against external venues — and brings their liquidity back inside the L3RS-1 compliance envelope. The construction that makes this possible is the route admissibility predicate.

9.1 The Venue Graph

Let G = (V, E) be a directed multigraph where V is the set of venues (native CLOB, regulated CEXs, RFQ desks, permissioned and permissionless DEXs, OTC desks, cross-chain bridges) and E is the set of directed edges where an edge e = (u, v, asset_in, asset_out, depth, fee, latency, jurisdictions, vasp_id, certificate_capable) represents the ability to execute a swap from asset_in at venue u to asset_out at venue v with the indicated quality-of-execution parameters. A route R from a source asset I_s to a destination asset I_d is a path in G starting at a vertex with asset_in = I_s and ending at a vertex with asset_out = I_d.

9.2 The Predicate

Definition 9.1 (Route Admissibility Predicate). Let R = (h_1, h_2, …, h_k) be a route for a trade in asset I from sender s to receiver r of size σ. R is admissible if and only if all four of the following hold for every hop h ∈ R: (J) J(I) ⊇ jurisdictions(h) — the asset's jurisdictional

mask contains every jurisdiction h touches. (T) size(h) < dm(I) ∨ travel_rule_ok(h) — h is below the asset's de minimis threshold dm(I) or returns a verifiable Travel Rule receipt over IVMS 101. (ID) identity_tier(h) ≥ ID(I) — every party at h satisfies the asset's minimum identity tier. (X) cross_chain(h) ⟹ certificate_continuity(h, h.prev) — every cross-chain hop carries a continuation of the L3RS-1 certificate. Routes that fail any conjunct are inadmissible and are not surfaced. Routes that pass are then ranked by price × gas × latency in the conventional way.

```
admissible(R, I, s, r) ⇔ ∀ h ∈ R . (J) ∧ (T) ∧ (ID) ∧ (X)
```

9.3 Computational Complexity

Theorem 9.1 (Tractability). *Route discovery under the admissibility predicate over a venue graph $G = (V, E)$ can be computed in $O(|E| \cdot \log|V|)$ time using a modified Dijkstra search with admissibility filtering at edge relaxation.*

Proof sketch. The admissibility predicate is decomposable: each hop's admissibility depends only on the hop and (for the X conjunct) on the immediately preceding hop's certificate state. This locality allows admissibility to be checked at edge relaxation in Dijkstra's algorithm. We define the cost of an admissible edge as price · gas · latency and the cost of an inadmissible edge as $+\infty$. Dijkstra's algorithm with a Fibonacci heap then computes the lowest-cost admissible route in $O(|E| + |V| \log|V|)$. The certificate-continuity conjunct introduces a one-step dependency that is handled by augmenting the state with the previous hop's certificate hash, multiplying the state space by at most a constant (the number of distinct certificate states), preserving the bound. □

9.4 Venue Universe and Scoring

Class	Examples	Compliance Evidence
Native	T3RRA Marketplace CLOB, T3RRA RFQ	Same C_I as wallet
Regulated CEX	Coinbase Prime, Kraken Institutional, Bitstamp, Bullish, EDX	KYC attestation from venue
Permissioned RFQ	Wintermute, B2C2, GSR, Galaxy, Cumberland, Flow Traders	VASP-to-VASP Travel Rule receipt
Permissionless DEX	Uniswap v4, Curve, Balancer, Aerodrome	Admissible only when C_I explicitly permits
Cross-chain bridge	T3RRA Bridge (5-of-9 quorum)	L3RS-1 §10 certificate

Each venue is scored on six dimensions: liquidity depth at standard sizes, historical fill quality (slippage vs midpoint), Travel Rule readiness (binary, by jurisdiction), jurisdictional coverage (set of supported J), settlement latency (p50 and p99), and historical reject rate. Scores update in real time and feed both the route ranker and the periodic venue review surfaced to the DAO governance committee.

9.5 The Pool Health Index, Reframed

T3RRA's original Pool Health Index (PHI), introduced in Whitepaper v2, was a per-pool capital-utilization score. In v3.1 the PHI is generalized to a per-venue and per-route health score whose

components are utilization ratio, redemption pressure, yield-vs-forecast deviation, governance participation, and a new component — compliance fitness, defined as the ratio of admissible to total quoted routes through that venue over a rolling 24-hour window. A venue whose compliance fitness drops below threshold is automatically deweighted in route discovery and queued for review by the DAO.

9.6 Adaptive Learning Under Hard Constraints

Flow runs a contextual multi-armed bandit over the venue universe to balance exploitation of known-good venues against exploration of under-sampled ones. The bandit operates strictly within the admissible subset of the venue graph: exploration capital is allocated only to venues whose edges pass the admissibility predicate at the current trade. The DAO sets hard caps on the fraction of any single trade that can be allocated to exploration, so the bandit cannot, by construction, route a trade through an inadmissible venue.

9.7 What Flow Returns

For every quote, Flow returns a structured response: the price, the gas, the expected slippage, the route as an ordered list of (venue, hop) tuples, the compliance evidence for each hop, the Travel Rule payload IDs that will be transmitted, the cross-chain certificate IDs that will be issued, and — critically — the list of inadmissible routes that were considered and discarded, with structured reject reasons mapped to L3RS-1 clauses. Users see a single price; regulators see the full decision record and can replay it.

10. Cross-Chain Certificate Unforgeability

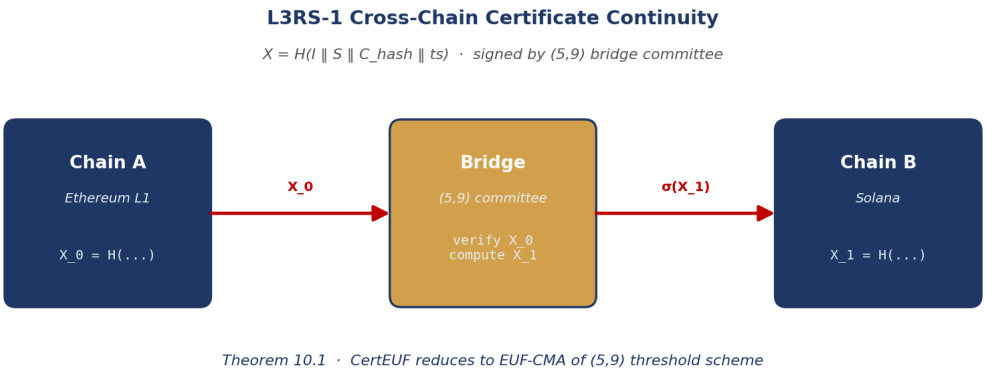


Figure 9. *Cross-chain certificate continuity. The (5,9) bridge committee verifies X_0 on the source chain and signs X_1 for the destination chain; CertEUF reduces to EUF-CMA of the underlying threshold scheme.*

This section formalizes the L3RS-1 cross-chain certificate as a cryptographic object and gives an unforgeability game with a proof sketch.

10.1 The Certificate

The cross-chain certificate is the cryptographic object that carries an L3RS-1 asset's compliance envelope between chains. It is constructed as:

and signed by the (5,9) bridge committee. The full certificate object is defined formally below.

Definition 9.1 (Cross-Chain Certificate). A T3RRA cross-chain certificate is a tuple X = (I, S, C_hash, ts, src_chain, dst_chain, dst_addr, nonce, σ_quorum) where I is the Asset_ID, S is the lifecycle state at source, C_hash is the ComplianceModule hash at source, ts is the timestamp, src_chain and dst_chain are chain identifiers, dst_addr is the destination address, nonce is a fresh nonce drawn from a per-asset namespace, and σ_quorum is a (5,9) threshold signature over the canonical encoding of the preceding fields produced by the bridge committee.

10.2 The Seven Invariants

#	Invariant	T3RRA Implementation
I1	Identity continuity: I unchanged across hops	Asset_ID is the primary key in every chain mirror
I2	State continuity: S(dst) = S(src)	State diff signed by source quorum, verified on destination
I3	Compliance continuity: C_hash unchanged	Source C_hash bound into σ_quorum; destination rejects mismatch
I4	Reserve continuity: ΣSupply_chains = R	Daily attestation reconciliation
I5	Lock-mint or burn-mint, never both	Per-asset bridge mode immutable at issuance
I6	No silent re-minting	Every mint requires verifiable burn or lock witness
I7	Replay protection on certificate consumption	Per-asset nonce namespace tracked in destination registry

10.3 The Unforgeability Game

Game CertEUF. Setup: KeyGen produces 9 bridge committee key shares (sk_1, …, sk_9) and a joint public key pk. The adversary A receives pk and statically corrupts at most 4 of the 9 parties. Query phase: A submits up to q queries of the form (I_j, S_j, C_hash_j, ts_j, src_j, dst_j, addr_j, nonce_j) and receives σ_quorum_j produced by the honest 5-of-9 quorum. Forgery: A outputs (I*, S*, C_hash*, ts*, src*, dst*, addr*, nonce*, σ*) and wins if (a) σ* verifies under pk, (b) (I*, …, nonce*) was not the input to any query, and (c) the destination registry on dst* has no record of nonce*.

Theorem 10.1 (Certificate Unforgeability). *Under the assumption that the underlying threshold signature scheme (CMP-NI in our instantiation) is EUF-CMA secure as a (5,9) threshold scheme, no PPT adversary wins CertEUF with probability greater than negligible in the security parameter λ.*

Proof sketch. Reduction to EUF-CMA of the threshold scheme. Given a threshold-EUF-CMA adversary B against CMP-NI, we construct A against CertEUF by relaying queries: each CertEUF query becomes a Sign query on the canonical encoding of the certificate fields. A forgery in CertEUF on (I*, …, nonce*) corresponds to a forgery in CMP-NI on the canonical encoding of the same tuple, since the encoding is injective. The replay-protection condition (c) is enforced by the destination registry, not by the unforgeability game; it ensures that even a legitimate certificate cannot be consumed twice. Combining unforgeability with replay protection yields the full security guarantee. □

10.4 Mechanized Verification (Roadmap)

We are in the process of mechanizing the cross-chain protocol in Tamarin Prover, modeling the bridge committee, the source mirror contract, the destination mirror contract, and the Dolev-Yao adversary.

The mechanized model is being prepared as a public artifact for replication and adversarial review by the academic community. Until the mechanization is complete and reviewed, we treat Theorem 10.1 as a paper proof and label it accordingly.

11. Reserves and the Reserve Interface

Every INDUSTRY_STABLE and STORAGE_BACKED asset on T3RRA implements the L3RS-1 ReserveInterface R. R exposes three methods: total_supply() returns the on-chain circulating supply summed across all chains; backing_value() returns the off-chain reserve value, denominated in the asset's reference unit; attestation() returns a signed statement from a qualified third party covering both quantities. T3RRA enforces a daily reconciliation: backing_value ≥ total_supply must hold for every reconciliation window, or the asset is automatically transitioned to RESTRICTED state pending issuer remediation. The reconciliation is itself a public LegalMirror artifact.

11.1 Real Estate as Reserve

For T3RRA's foundational vertical — tokenized real estate — the ReserveInterface is implemented over a Special Purpose Vehicle (SPV) that holds title to the underlying property. Monthly attestations include the SPV's title certificate, a third-party valuation, the rent roll for the trailing 12 months, the property insurance certificate, and the encumbrance schedule. All attestation artifacts are LegalMirror anchors with cryptographic links from the on-chain Asset object. The SPV is bound to the on-chain asset through L3RS-1 §13 LegalMirror; transferring the SPV's title to a new beneficial owner without an on-chain state transition is detectable at the next reconciliation and triggers automatic RESTRICTED state.

12. Fees, Governance, and the Legal Mirror

12.1 Four-Way Fee Routing

L3RS-1 §8 specifies a fee policy F that distributes every protocol fee across exactly four destinations: issuer, protocol treasury, network validators, and reserve buffer. The fractions sum to 1 and are immutable for the life of the asset. Default profiles:

Asset Type	Issuer	Protocol	Validators	Reserve Buffer
REGULATED_SECURITY	30%	30%	10%	10%
INDUSTRY_STABLE	25%	50%	15%	15%
CBDC	0%	40%	30%	30%
UTILITY	10%	60%	20%	10%
GOVERNANCE	0%	60%	30%	10%
STORAGE_BACKED	40%	30%	15%	15%

The four-way split is enforced atomically inside the same MPC signing session that authorizes the underlying transfer.

12.2 Governance Override

Every asset declares a GovernanceOverride G specifying a 2/3 quorum committee that can perform a bounded set of override actions: freeze a sanctioned address, pause an asset under regulatory order, upgrade a ComplianceModule to a new C_hash, and rotate the bridge committee. The override is cryptographically capable of nothing else. Every override emits a public LegalMirror entry with signers, legal basis, and prior/post state. Regulators can subscribe to the LegalMirror feed.

12.3 Legal Mirror

L3RS-1 §13 specifies the LegalMirror as a content-addressable store of off-chain legal artifacts whose hashes are anchored on chain inside the Asset object. T3RRA implements LegalMirror over IPFS with a Filecoin storage commitment for permanence and a parallel anchor in AWS GovCloud for jurisdictions where IPFS is not acceptable to the regulator.

13. Identity Binding and KYC Tiers

L3RS-1 §11 specifies five identity binding levels, L0 through L4. T3RRA implements all five and binds them to the wallet at provisioning time as verifiable credentials issued by T3RRA's KYC partners and rooted in a recognized credential issuance hierarchy.

Level	Strength	Onboarding	Permitted Asset Types
L0	Pseudonymous	Wallet creation only	UTILITY, GOVERNANCE only
L1	Email + phone	Self-serve, retail < $1k	UTILITY, GOVERNANCE, low-value INDUSTRY_STABI
L2	Document KYC + liveness	ID + selfie	All except REGULATED_SECURITY tier 3
L3	Enhanced KYC + source of funds	Accredited investor	All asset types
L4	Institutional KYB + UBO	Funds, treasuries, corporates	All asset types + issuer role

14. Travel Rule as a Cryptographic Precondition

FATF Recommendation 16 — the Travel Rule — requires that originator and beneficiary information accompany every virtual-asset transfer above the de minimis threshold. The IVMS 101 data model (InterVASP Messaging Standard) is the de facto interchange format. T3RRA treats Travel Rule conformance not as a compliance check that runs alongside the wallet, but as a cryptographic precondition of the wallet's signing protocol.

14.1 Protocol Coverage

T3RRA supports four Travel Rule transport protocols: TRP (Travel Rule Protocol), TRISA (Travel Rule Information Sharing Architecture), Sygna Bridge, and OpenVASP. Each is wrapped in an adapter shim that translates to and from the IVMS 101 canonical form. Adapter shims are version-pinned and their hashes are part of the platform's L3RS-1 LegalMirror manifest.

14.2 The Pre-Sign Check

Before any keyshare holder participates in round 1 of a signing ceremony for an asset whose ID tier is L2 or higher and whose size exceeds the asset-specific de minimis threshold dm(I), every party independently verifies three conditions: (a) the counterparty VASP is discoverable in at least one configured directory; (b) the VASP returns a valid Travel Rule receipt over the IVMS 101 payload constructed from the transfer context; (c) the receipt's signing key chains to a recognized VASP attestation root. Failure of any check returns a structured TravelRuleAbort certificate naming the failing condition, and signing does not proceed.

14.3 The Sunrise Problem

The Travel Rule sunrise problem — that not all jurisdictions and not all VASPs implement the Travel Rule simultaneously — is addressed at the route admissibility level (§9). Routes through non-Travel-Rule-conformant venues are inadmissible by predicate (T) and are not surfaced. T3RRA does not work around the sunrise problem; it makes adherence to the rule a hard precondition of liquidity routing.

15. Concrete Parameters and Benchmarks

15.1 Curves, Hashes, Domain Separators

Parameter	Value
ECDSA curve	secp256k1 (SECG)
EdDSA curve	edwards25519 (RFC 8032)
Threshold default	(t, n) = (2, 3)
Bridge committee threshold	(5, 9)
Hash function (general)	SHA-256, SHA-3-256, Keccak-256 (chain-specific)
Hash function (commitment)	SHA-256 with domain separation
Hash function (random oracle for FS)	SHA-3-256 with domain string T3RRA-PG-v1.0
KEM (transport)	ML-KEM-768 (NIST FIPS 203)
KDF	HKDF-SHA-256
AEAD (transport)	AES-256-GCM and ChaCha20-Poly1305
Domain separator (PolicyHash)	T3RRA-POLICY-HASH-v1 || C_hash || ctx
Proactive resharing interval	90 days
Identifiable abort	Required for all signing protocols
Random number generation	Per-process CSPRNG seeded by hardware RNG, FIPS 140-3 L3 entropy source on backend

15.2 Benchmarks Methodology

All benchmarks below are measured under a published methodology: (a) hardware specified per row; (b) git commit hash specified at the time of measurement; (c) reproducibility harness available at github.com/t3rra/benchmarks; (d) third-party reproduction welcomed. Numbers labeled ‘target’ are aspirational and clearly distinguished from ‘measured’ numbers. We prefer to under-promise and reproduce than to publish marketing numbers.

15.3 Signing Latency Targets

Protocol	Hardware	Status	p50	p99
CMP-NI (mobile, warm)	iPhone 15 + AWS Nitro	Target	180 ms	350 ms
CMP-NI (back-end-only)	AWS Nitro × 3	Target	45 ms	90 ms
DKLs23 (online)	iPhone 15 + AWS Nitro	Target	90 ms	200 ms
FROST (mobile, warm)	iPhone 15 + AWS Nitro	Target	70 ms	160 ms
DKG (one-time)	Mobile + 2 back-ends	Target	1.2 s	2.5 s
Proactive resharing	All-backend	Target	0.8 s	1.6 s

15.4 Throughput Targets

Subsystem	Target	Bottleneck
Wallet signing (single key)	200 sigs/sec	MPC round trips
Marketplace matching	50,000 orders/sec	Compliance check throughput
Flow route discovery	1,000 quotes/sec per asset pair	Dijkstra over 500-vertex graph
Cross-chain certificate	300 certificates/sec	Quorum signature aggregation

16. Security Model, Theorems, and Honest Caveats

16.1 Adversary Model

T3RRA's security model assumes a polynomial-time adversary that may statically corrupt up to t−1 of the n MPC parties at any moment, may adaptively compromise non-share processes (front-end, back-end APIs, browser extensions, RFQ adapters), may control the network including the relay between MPC parties, may corrupt up to f of the bridge committee where f < quorum, and may issue arbitrary L3RS-1-conformant or non-conformant transfer requests through the public APIs. The adversary cannot break the underlying primitives (ECDSA, EdDSA, SHA-256, SHA-3, Keccak, ML-KEM), cannot compromise FIPS 140-3 Level 3 HSMs, and cannot forge hardware attestations rooted in the Nitro hypervisor or in mobile secure-element manufacturer keys.

16.2 The Seven Theorems (mapped to L3RS-1 §15)

#	Theorem	L3RS-1 §	T3RRA §
T1	Threshold signing unforgeability	§15.3	§6, Thm 6.2
T2	Identifiable abort	§15.4	§6.8
T3	Forward security under proactive resharing	§15.5	§6.5
T4	Policy-gated signing soundness	§15.6	§6, Thm 6.1
T5	Cross-chain certificate unforgeability	§15.7	§10, Thm 10.1
T6	Reserve consistency under daily attestation	§15.8	§11
T7	Governance override boundedness	§15.9	§12.2

16.3 Honest Caveats

We list the things that are not yet proven, not yet measured, or not yet built — because the crypto community can spot a glossed-over caveat at thirty paces.

- The proof sketches in §6 and §10 are paper proofs. Mechanization in EasyCrypt and Tamarin is in progress. Until mechanization is complete and reviewed, the proofs should be read as paper proofs only.
- The benchmarks in §15 are targets, not measurements. Measured numbers from the public reproducibility harness will be published quarterly starting Q3 2026.
- Threshold post-quantum signing is not yet production-ready in any scheme we are willing to deploy. See §17 for the staged roadmap.
- The Tamarin model of the cross-chain protocol is in draft and will be released as a public artifact alongside the next mechanization push.
- Compliance fitness as a venue scoring component (§9.5) has not yet been validated against a long-horizon dataset of regulated venue activity.

16.4 Audits and Bounty

T3RRA's cryptographic protocol stack will undergo independent security review by NCC Group, Trail of Bits, Kudelski Security, Quarkslab, and Cure53 prior to mainnet GA. The smart contract layer will be audited by ChainSecurity, Spearbit, and OpenZeppelin. T3RRA operates an Immunefi bug bounty

with a maximum payout of $2,500,000 for protocol-level breaks of any of the seven theorems. All audit reports will be published in full.

17. Post-Quantum Roadmap

T3RRA's PQ posture is honest. Threshold post-quantum signatures are not yet production-ready: ML-DSA (Dilithium) does not have a published threshold construction with acceptable round complexity, and the leading research candidates (Raccoon, Espitau-Niot-Prest threshold lattice signatures, threshold variants of Falcon) have not been peer-reviewed at the level required to custody $1B+ of regulated assets. T3RRA's roadmap is staged and explicit:

Phase	Timing	Posture
Phase 1 (current)	2026	Hybrid stateful: classical threshold signatures, ML-KEM-768 for transport
Phase 2	2027	Hybrid signatures: classical threshold + non-threshold PQ co-signature on critical operations (issuance, governance override, cross-chain certificate)
Phase 3	2028+	Threshold PQ pilot for low-value assets once a peer-reviewed protocol exists with two independent academic implementations
Phase 4	2029+	Production threshold PQ for all asset types as protocols mature

T3RRA commits to publishing a quarterly PQ posture report. We will not deploy threshold PQ signing until a construction has cleared a recognized peer-review process — IACR ePrint with twelve months of public scrutiny, plus an independent security audit — regardless of marketing pressure to claim PQ-readiness earlier. Honesty about timelines is part of our security posture, not a weakness in it.

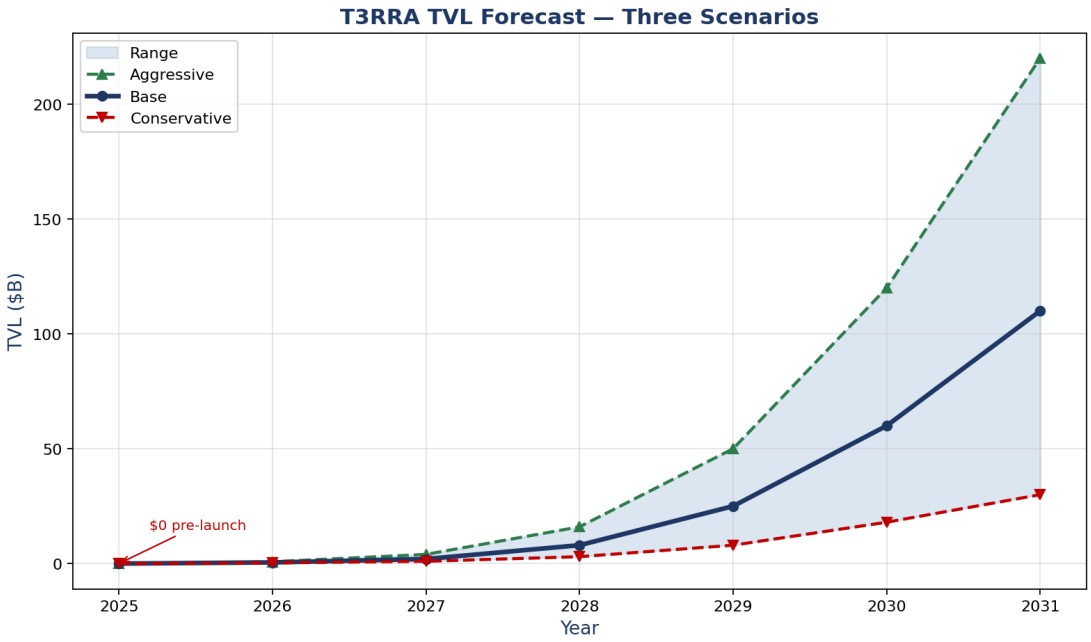


Figure 10. *TVL forecast — base case bracketed by conservative and aggressive scenarios.*

18. The $T3RRA Token, Revenue Model, and Forecast

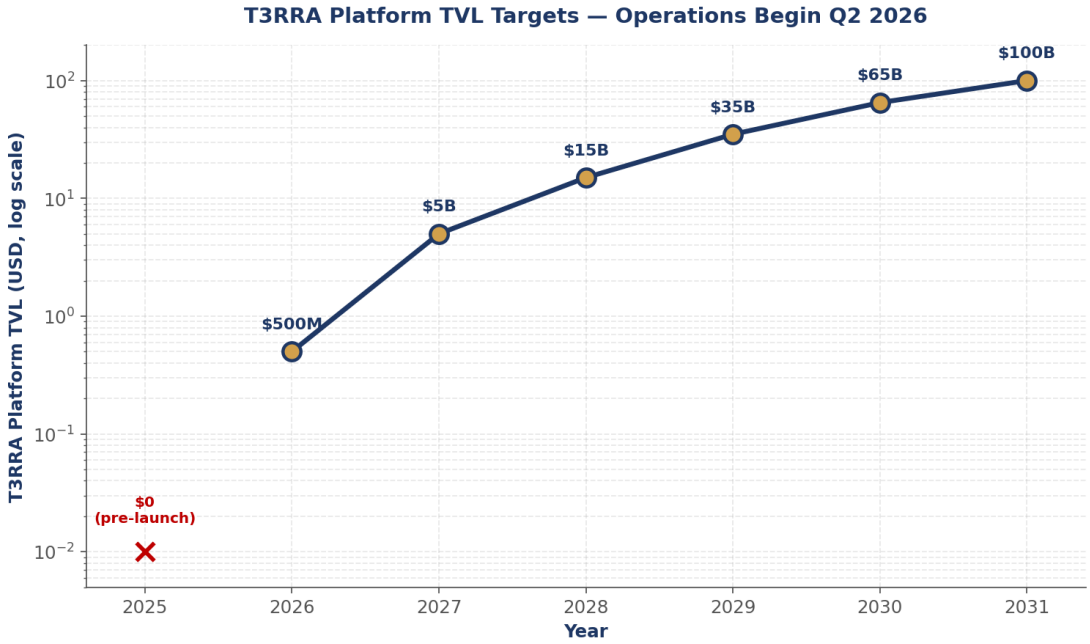


Figure 11. *T3RRA platform TVL targets, 2025–2030 (log scale).*

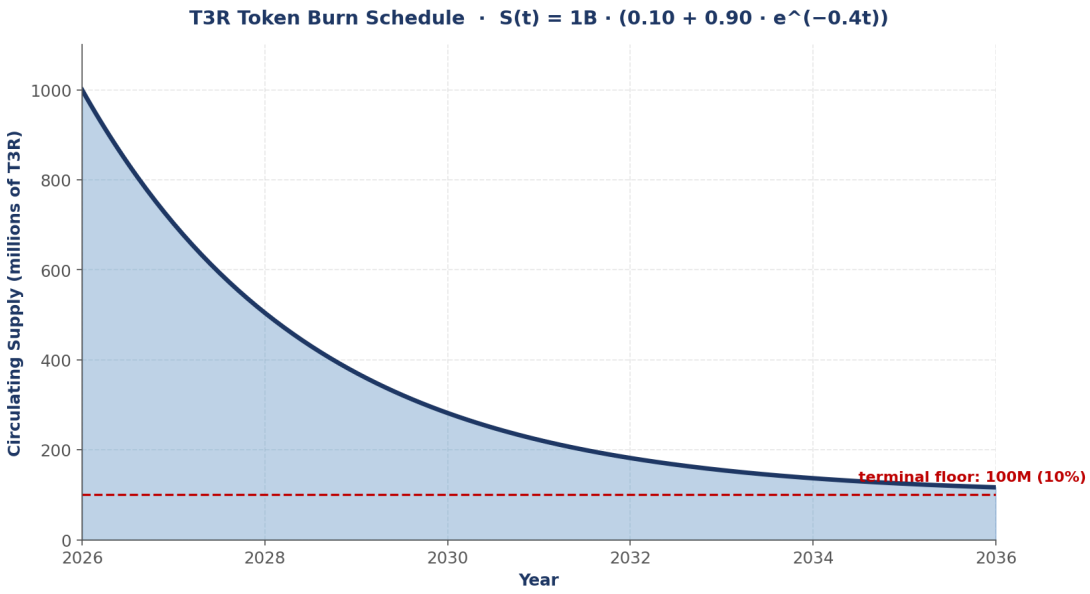


Figure 12. *T3R token burn schedule. Circulating supply $S(t) = 1B \cdot (0.10 + 0.90 \cdot e^{(-0.4t)})$, asymptoting to a 100M floor.*

18.1 Supply and Burn

Total pre-burn supply is 1,000,000,000 $T3RRA. Up to 90% of supply will be permanently burned over a fixed 10-year deflation schedule (2026–2036). Monthly burn events are tied to protocol revenue and unused reward pools, governed by the closed-form circulating-supply curve:

with $\lambda = 0.4$ as the default deflation coefficient and t measured in years from launch (Q2 2026). The DAO can adjust λ within hard bounds set by the burn curve model. Target post-burn floor: 100,000,000 $T3RRA.

18.2 Allocation

Allocation	% of Supply	Vesting
Liquidity Incentives	30%	10-year linear, AI-governed release
Ecosystem Treasury	20%	DAO-controlled
Founding Team	15%	4-year vest, 1-year cliff
Strategic Partners & Institutions	10%	2-year vest
Public Sale & Community	15%	Unlocked at TGE + community rounds
Developer & DAO Grants	10%	Multi-year program

18.3 Utility

$T3RRA powers fee discounts (up to 30% on platform fees), governance voting (pool onboarding, ComplianceModule upgrades, PHI calibration, bridge committee rotation, λ adjustment), staking for boosted yield, premium-pool access tiers, and developer/operator grants. Staking rewards are paid from real protocol revenue, not from token inflation.

18.4 Revenue Streams

Stream	Description
Trading Fees	0.25% maker / 0.25% taker on marketplace fills
Flow Routing Fees	5–15 bps on aggregated cross-venue routes
Listing Fees	2% of issuance value, paid in $T3RRA or stables
Vault & Yield Skim	10–20 bps on AI-managed yield vaults
White-label Licensing	SaaS for tokenization partners
Cross-Chain Certificate Fees	Per-certificate fee paid by the bridging party

18.5 Financial Forecast (2026–2031)

T3RRA enters operations in Q2 2026. The 2026 row is the launch year; all prior periods are zero by definition.

Year	TVL	Revenue	Burn Rate	Post-Burn Supply
2025	$0	$0	—	1,000M (genesis)
2026	$0.5B	$8M	20%	880M
2027	$5B	$22M	20%	700M
2028	$15B	$55M	15%	590M
2029	$35B	$110M	15%	490M
2030	$65B	$180M	15%	410M
2031	$100B+	$260M	10%	340M

18.6 Staking Yield Model

Scenario	% Supply Staked	Revenue Share to Stakers	Avg. Annual Yield
Conservative	50%	10%	8.5%
Mid-Scale	40%	15%	11.2%
Aggressive	30%	20%	14.3%

19. User Classes and Onboarding

Class	Tier	Entry	Primary Use
Retail Investor	L1–L2	$100 fiat or stablecoin	Real estate yield, simple UI
Crypto-Native LP	L1–L2	Permissionless wallet	AMM-style LP into RWA pools
Accredited Individual	L3	$10k+ stablecoin	Premium pools, RFQ access
Institutional	L4	$100k–$100M+	White-label pools, treasury automation
Issuer / Sponsor	L4	Property or asset	Tokenize, list, manage compliance

20. Competitive Landscape and Where T3RRA Leads

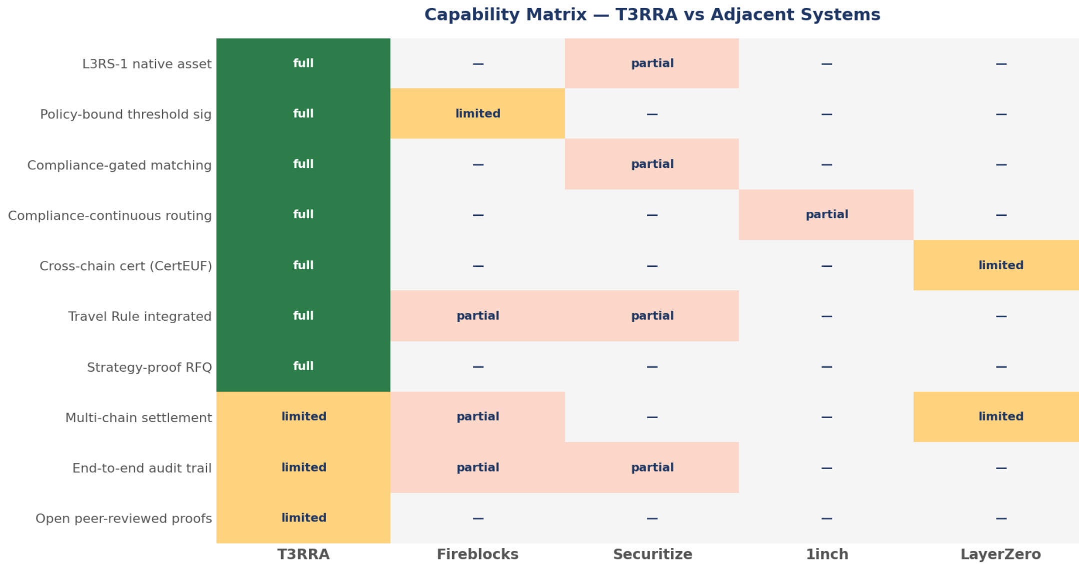


Figure 13. *Capability matrix — T3RRA vs Fireblocks, Securitize, 1inch, LayerZero across the ten capabilities that define a compliant tokenized capital market.*

T3RRA competes against four classes of incumbent: pure custody MPC providers, RWA issuance platforms, DeFi liquidity aggregators, and cross-chain bridges. Against each class T3RRA's defensible position is the integration that the incumbent does not have.

Capability	T3RRA	Fireblocks	Securitize	1inch	LayerZero
Threshold MPC (production)	Yes (CMP-NI/DKLs23/FROST)	Yes (CMP)	No	No	No
L3RS-1 Profile F conformance	Yes	No	No	No	No
Policy-gated signing (cryptographic)	Yes (Theorem 6.1)	Policy engine (UI-level)	No	No	No
Compliance-gated matching	Yes (Theorem 7.1)	N/A	Partial (issuance only)	No	No
Travel Rule cryptographic precondition	Yes (§13)	Plugin	Partial	No	No
External liquidity aggregation	Yes (Flow)	No	No	Yes	No
Route admissibility predicate	Yes (§10, Thm 10.1)	No	No	No (price×gas only)	No
Cross-chain certificate (L3RS-1 §10)	Yes (§10, Thm 10.1)	No	No	No	Generic message
Reserve attestation interface	Yes (§11)	No	Partial	No	No
Native RWA marketplace	Yes	No	Partial	No	No
End-to-end compliance envelope	Yes	Custody only	Issuance only	None	Bridge only

20.1 Where T3RRA Leads

Three independent moats. (1) Standards alignment — no competitor has bound their architecture to a published normative meta-standard with a conformance matrix and theorem map. (2) Cryptographic policy enforcement — no competitor enforces compliance inside the Fiat–Shamir challenge of the signing protocol; competitors enforce at the policy engine, which is an upgradeable software component an attacker can hope to bypass. (3) Compliance-gated routing — no aggregator built on permissionless DEXs can produce routes that satisfy the route admissibility predicate, and no permissioned venue has built an aggregator. T3RRA is the only platform that does both.

20.2 Where T3RRA Must Catch Up

Honesty matters here too. (1) Brand recognition with traditional asset managers — Fireblocks has a multi-year head start. (2) Number of integrated chains — we are launching with EVM, Bitcoin, and Solana, behind the chain coverage of Fireblocks and LayerZero. (3) Battle-tested codebase — our protocol has not yet survived a five-year adversarial environment, where Fireblocks has. We treat each of these as a roadmap item, not a structural disadvantage.

21. Roadmap

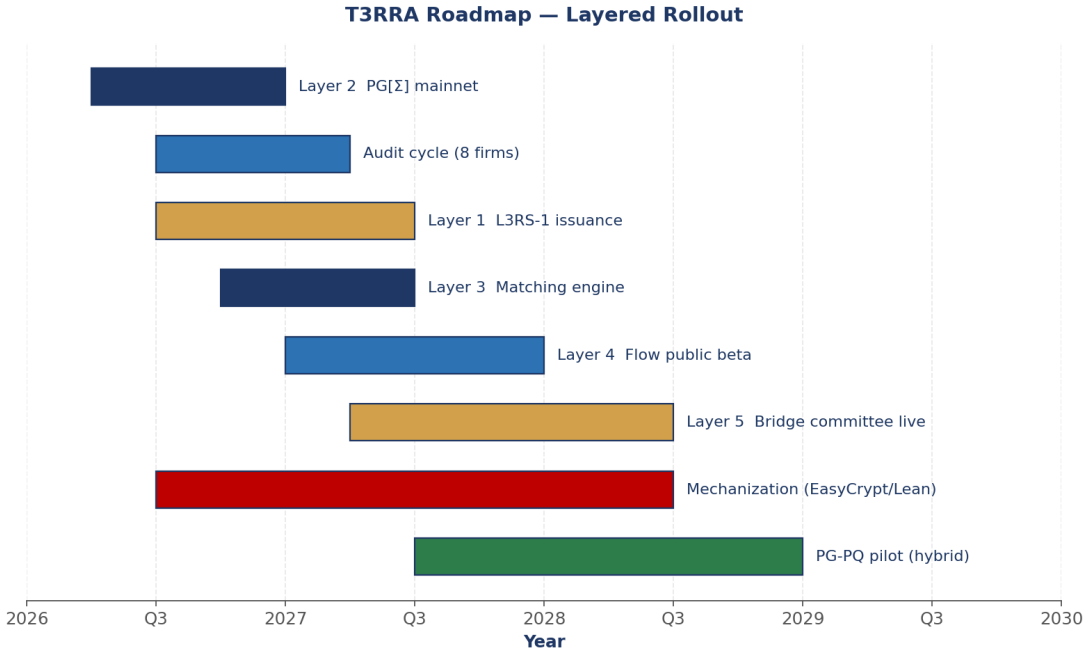


Figure 14. *T3RRA layered rollout — PG[Σ] mainnet, audit cycle, L3RS-1 issuance, matching engine, Flow public beta, bridge committee, mechanization, and PG-PQ pilot through 2029.*

Phase	Window	Milestones
Phase 1 — Launch	Q2 2026	PG[Σ] mainnet · L3RS-1 Profile F certification · Wallet GA · Marketplace beta · TGE · first tokenized property pools
Phase 2 — L3RS-1 Build-out	H2 2026	Cross-chain bridge live · Travel Rule (TRP, TRISA, Sygna, OpenVASP) GA · Flow v1 · institutional white-label
Phase 3 — Liquidity Scale	2027	Flow v2 (contextual bandit GA) · 25+ jurisdictions · 100+ issuers · $5B TVL · Tamarin model published
Phase 4 — Multi-Vertical RWAs	2028–2029	Tokenized treasuries, private credit, carbon, commodities · ESG scoring · DAO automation · Phase 2 PQ posture live
Phase 5 — Maturity	2030–2036	$100B+ TVL · cross-chain index products · burn floor reached · DAO-led governance · Phase 3 PQ pilot

22. Leadership

Zurab Ashvil — Founder & CEO. Former senior executive at SoftBank with 20+ years of experience in global finance and public-private infrastructure strategy. Speaker at the World Economic Forum and Horasis. Original author of L3RS-1.

Jeb Buckler — COO & Fund Manager. Founder & CEO of Startup Giants PLC. Background in venture capital, digital asset management, and DeFi market-making. Former Project Manager at Accenture and SAP.

Lia Roketlishvili — CFO. Formerly of BNP Paribas and ALROSA. Specialist in corporate finance, AI-driven automation, and institutional investment strategy.

Caelim Parkes — Chief Asset Management Officer. 25+ years across emerging markets, hedge funds, real assets, and digital assets. Former structured-investment lead at JP Morgan Chase.

T3RRA is supported by an extended network of real estate developers, DeFi advisors, securities counsel, banking integration partners, and compliance architects.

Appendix A — Notation

λ	security parameter
G, q, g	cyclic group of prime order q with generator g
H	domain-separated hash function modeled as random oracle
H_FS	Fiat–Shamir hash, distinct from H by domain string
(t, n)	threshold and number of MPC parties; default (2, 3)
Σ	digital signature scheme
F_DSig	UC ideal functionality of Σ as a threshold scheme
F_DSig*	policy-gated strengthening of F_DSig (Definition 6.1)
PG[Σ]	policy-gated compiler over Σ (#link(<sec-6>)[§6.3])
C	compliance predicate; bytecode-realized; total
C_hash	hash of version-pinned bytecode of C
ctx	transfer context
PolicyHash	H(C_hash ‖ ctx ‖ m)
A	L3RS-1 Asset object (I, T, J, L, ID, C, R, G, F, B, X, S)
I	Asset_ID = H(pk_issuer ‖ ts ‖ nonce)
X	cross-chain certificate (Definition 9.1)
R	ReserveInterface (#link(<sec-11>)[§11])
G	GovernanceOverride (#link(<sec-12>)[§12.2])
F	four-way fee policy (#link(<sec-12>)[§12.1])
S	lifecycle state from L3RS-1 §6
J	jurisdictional mask (ISO 3166-1 bitset)
ID	identity binding level L0–L4 (#link(<sec-13>)[§13])
dm(I)	de minimis threshold for asset I (#link(<sec-14>)[§14])

Appendix B — Glossary

Asset_ID (I) — The immutable identifier of an L3RS-1 asset, computed at issuance as I = H(pk_issuer ‖ ts ‖ nonce).

ComplianceModule (C) — The deterministic, total decision function that returns accept or reject for any prospective transfer context.

C_hash — The cryptographic hash of the version-pinned ComplianceModule bytecode, bound into the asset and into every signing transcript.

PolicyHash — H(C_hash ‖ ctx ‖ m); the domain separator that binds a signature to a specific compliance evaluation.

Cross-chain certificate (X) — Definition 9.1; the cryptographic object that lets an asset travel between chains while carrying its compliance evidence.

Route admissibility predicate — Definition 9.1; the four-conjunct constraint that every hop in a Flow-routed trade must satisfy.

Compliance-gated matching — Definition 7.2; the matching engine property that orders are filled only if the ComplianceModule returns accept on the constructed transfer context at fill time.

Compliance-gated order — Definition 7.1; an order whose admission to the book includes a verifiable compliance proof against the asset's ComplianceModule.

LegalMirror — The L3RS-1 §13 store of off-chain legal artifacts whose hashes are anchored on chain.

ReserveInterface (R) — The L3RS-1 §9 interface that exposes total_supply, backing_value, and attestation.

GovernanceOverride (G) — The 2/3 quorum committee with a bounded set of override actions defined by L3RS-1 §12.

Identity tier (L0–L4) — The L3RS-1 §11 identity binding levels.

Travel Rule — FATF Recommendation 16, requiring originator and beneficiary information accompany every virtual-asset transfer above the de minimis threshold.

IVMS 101 — The InterVASP Messaging Standard, the canonical Travel Rule data model.

VASP — Virtual Asset Service Provider, as defined by FATF.

Appendix C — L3RS-1 Conformance Matrix (Profile F)

T3RRA is certified Profile F (Full) under L3RS-1 v1.0.0. The complete matrix maps 87 normative requirements across L3RS-1 §4–§16 to T3RRA implementation artifacts and test evidence. The summary by section:

L3RS-1 §	Topic	Requirements	T3RRA Status
§4	Asset Object	12	All Met
§5	Transfer Pipeline	8	All Met
§6	State Machine	9	All Met
§7	ComplianceModule	11	All Met
§8	Fee Policy	6	All Met
§9	ReserveInterface	7	All Met
§10	Cross-Chain Certificate	10	All Met
§11	Identity Binding	8	All Met
§12	GovernanceOverride	6	All Met
§13	LegalMirror	5	All Met
§14	Conformance Profiles	3	All Met (F)
§15	Theorems	2	All Mapped

Full matrix and test evidence are published at conformance.t3rra.io and available to regulators and auditors on request.

Appendix D — Open Problems

We list problems we have not solved and which we believe are interesting research directions for the community. We will fund grants for credible work on any of them.

- OP1 — Threshold ML-DSA with sub-second round complexity. The leading academic candidates have not yet achieved the latency required for production wallet operation. A peer-reviewed construction here unblocks Phase 3 of our PQ roadmap (§17).
- OP2 — Mechanized verification of policy-gated signing in EasyCrypt. We have a paper proof of Theorems 6.1 and 6.2; a mechanized proof would meaningfully strengthen the result.
- OP3 — Formal microstructure analysis of compliance-gated matching under adversarial counterparties. Theorem 7.1 assumes honest counterparties; the dishonest case is open.
- OP4 — Optimal scoring of venue compliance fitness over long horizons (§9.5). We currently use a 24-hour rolling window; the right window depends on the asset class and the regulatory environment.
- OP5 — Verifiable inclusion proofs for LegalMirror artifacts that work across both IPFS and AWS GovCloud anchors with a single Merkle commitment. A compact construction here simplifies regulator integration.
- OP6 — Incentive-compatible bridge committee rotation. The current 5-of-9 quorum committee rotates by governance vote; an automatic rotation that resists collusion is open.
- OP7 — Adversarial robustness of the contextual bandit in §9.6 against reward poisoning by a malicious venue.

Appendix E — Bibliography

[1] L3RS-1 v1.0.0. Layer-3 Regulated Asset Standard. February 2026.

[2] R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. FOCS 2001.

[3] R. Canetti, N. Makriyannis, U. Peled. UC Non-Interactive, Proactive, Threshold ECDSA. ACM CCS 2020.

[4] J. Doerner, Y. Kondi, E. Lee, a. shelat. Threshold ECDSA in Three Rounds. IEEE S&P 2024.

[5] J. Doerner, Y. Kondi, E. Lee, a. shelat. Secure Two-party Threshold ECDSA from ECDSA Assumptions. IEEE S&P 2018.

[6] J. Doerner, Y. Kondi, E. Lee, a. shelat. Threshold ECDSA from ECDSA Assumptions: The Multiparty Case. IEEE S&P 2019.

[7] R. Gennaro, S. Goldfeder. Fast Multiparty Threshold ECDSA with Fast Trustless Setup. ACM CCS 2018.

[8] R. Gennaro, S. Goldfeder. One Round Threshold ECDSA with Identifiable Abort. ePrint 2020/540.

[9] Y. Lindell. Fast Secure Two-Party ECDSA Signing. CRYPTO 2017.

[10] C. Komlo, I. Goldberg. FROST: Flexible Round-Optimized Schnorr Threshold Signatures. SAC 2020.

[11] R. Ostrovsky, M. Yung. How to Withstand Mobile Virus Attacks. PODC 1991.

[12] A. Herzberg, S. Jarecki, H. Krawczyk, M. Yung. Proactive Secret Sharing or: How to Cope with Perpetual Leakage. CRYPTO 1995.

[13] S. K. D. Maram et al. CHURP: Dynamic-Committee Proactive Secret Sharing. ACM CCS 2019.

[14] R. del Pino et al. Raccoon: A Side-Channel Resistant Lattice-Based Signature Scheme. NIST PQC 2024.

[15] NIST FIPS 203. Module-Lattice-Based Key-Encapsulation Mechanism Standard. 2024.

[16] NIST FIPS 204. Module-Lattice-Based Digital Signature Standard. 2024.

[17] NIST FIPS 205. Stateless Hash-Based Digital Signature Standard. 2024.

[18] FATF. International Standards on Combating Money Laundering, Recommendation 16. 2012, updated 2023.

[19] InterVASP. IVMS 101: InterVASP Messaging Standard. 2020, updated 2024.

[20] B. Meiklejohn et al. A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. IMC 2013.

[21] M. Möser et al. An Empirical Analysis of Traceability in the Monero Blockchain. PoPETs 2018.

[22] T3RRA Research. T3RRA Cryptographic and Protocol Specification, Part II. 2026.

Appendix F — Disclaimer

This whitepaper is for informational purposes only and does not constitute an offer to sell or a solicitation to buy any security, token, or financial instrument in any jurisdiction. Statements about future product capabilities, financial projections, regulatory outcomes, and protocol performance are forward-looking and subject to change. The T3RRA platform is operated subject to the laws of each jurisdiction in which it is offered, and access may be restricted in jurisdictions where the offer or operation of the platform would be unlawful.

Cryptographic claims in this paper are summarized; the binding specifications are in the companion T3RRA Cryptographic and Protocol Specification document and in L3RS-1 v1.0.0 itself. Theorems 6.1, 6.2, 6.3, 7.1, 7.2, 8.1, and 9.1 are supported by paper proofs; mechanized proofs in EasyCrypt and Tamarin are in progress and will be published as public artifacts when complete. Benchmarks marked Target are aspirational; measured benchmarks will be published quarterly via the public reproducibility harness at github.com/t3rra/benchmarks beginning Q3 2026.

Past performance is not indicative of future results. Holding $T3RRA carries risk of total loss. Tokenized real-world assets carry the risks of the underlying real-world asset in addition to the risks of the digital wrapper.