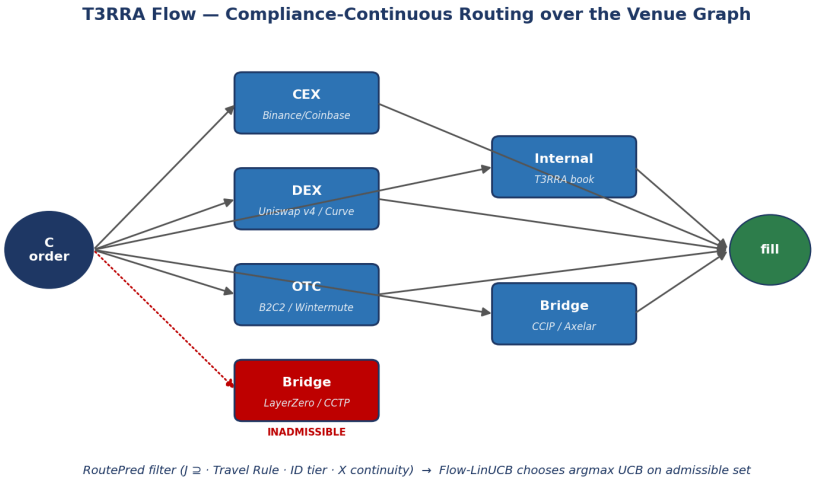
T3RRA

AI-Enhanced, Chain-Agnostic Liquidity Engine



T3RRA Flow Liquidity Engine

Flow Paper v1.0 · April 2026

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

Zurab Ashvil · T3RRA Research

Contents

Abstract

Regulated real-world assets (RWAs) do not trade in a single venue. They trade as fragmented, jurisdiction-scoped, compliance-conditioned inventory spread across centralized order books, bilateral OTC desks, permissioned AMMs, and cross-chain bridges. The classical liquidity-aggregation problem — find the execution path that minimizes expected slippage subject to a size constraint — is therefore insufficient: in the RWA setting, the set of admissible paths is itself a function of the asset's jurisdiction mask, the counterparty's identity tier, Travel Rule receipt status, and the continuity of the L3RS-1 cross-chain certificate along the path. A route that is price-optimal but compliance-inadmissible is not merely suboptimal; it is invalid.

This document specifies T3RRA Flow, the liquidity engine of the T3RRA platform, as a compliance-continuous routing and execution system. Flow treats compliance as a hard constraint and liquidity as the objective, and solves the resulting constrained optimization online, under adversarial market conditions, with provable regret guarantees against the best admissible policy in hindsight. We formalize the venue graph, the route admissibility predicate (inherited from the T3RRA Cryptographic Specification Part II rev B), the Pool Health Index (PHI) as a sufficient statistic for venue quality, the contextual multi-armed bandit executor with LinUCB-style confidence bounds, and the Liquidity DNA profile that conditions routing on asset-class microstructure. We prove four main results: (i) admissibility-preservation under online updates; (ii) a no-regret bound $O(d\sqrt{(T \log T)})$ against the best fixed admissible routing policy; (iii) strategy-proofness of the RFQ variant under honest PHI reporting; and (iv) a best-execution property that reduces, in the unconstrained limit, to the classical Almgren–Chriss optimal execution schedule.

Flow is designed to plug into the T3RRA Wallet's policy-gated threshold signing compiler PG[Σ] and to consume the same Travel Rule and identity infrastructure as the rest of the platform. It is not a decentralized exchange; it is a router that orchestrates execution across exchanges that are already there, and that refuses — provably and irrevocably — to route through a venue whose state violates the asset's compliance envelope. This specification is paper-level. Proofs are paper proofs. The simulation harness and mechanization roadmap are described in §13 and §14.

1. Introduction and Design Goals

> ***Reference convention.*** *Throughout this document,* ***§N*** *refers to* ***Section N of this paper****; clicking it jumps there. References prefixed with* ***L3RS-1*** *— for example,* L3RS-1 §15 *— refer to the corresponding section of the external L3RS-1 v1.0.0 standard.*

T3RRA Flow is the component of the T3RRA platform responsible for sourcing, pricing, and executing liquidity for regulated real-world assets across a heterogeneous set of venues. It is the piece that turns an L3RS-1 asset from an object that can be held, transferred, and reported into an object that can be traded at institutional size without regulatory risk. This section states the design goals and positions Flow against the prior art.

Flow as the platform moat. In the T3RRA three-wall frame — Compliant Asset, Geographic Freedom, Roaming Liquidity — Flow is the third wall, and it is the wall that every prior attempt at regulated tokenization has died on. L3RS-1 solves wall one; SPV structuring and the jurisdiction map solve wall two; both were addressable with existing tools. Wall three — sourcing institutional-size fills from venues that no permissionless aggregator can integrate and refusing venues that no permissionless aggregator can refuse — is the problem the market has no answer to. The route admissibility predicate in §4, the PHI sufficient statistic in §5, and the LinUCB executor in §7 are, in combination, the first construction that turns that wall into a routable graph. This is the moat: not any single primitive, but the fact that the admissibility predicate is **upstream** of price optimization, and that every competitor's price-first architecture cannot retrofit it without rebuilding from the root.

Fee capture. Flow's economic contribution to the platform is captured at stage 5 of the T3RRA Standard Rate Card: **0.50% per trade** on secondary and Flow-routed volume. This is the first fee in the rate card that scales with **liquidity velocity** rather than with one-time issuance events, and it is the reason Flow is treated as a platform moat rather than a cost center. Every trade that clears admissibility contributes to platform revenue and, through the buyback channel documented in the Tokenomics, to T3RRA token deflation.

1.1 Design Goals

- G1 (Compliance as hard constraint). No route ever produced by Flow may traverse a venue, counterparty, or chain state that violates the asset's compliance envelope. A price-optimal but inadmissible route is never returned.
- G2 (Best execution under admissibility). Subject to G1, Flow minimizes expected slippage plus market impact plus fees, over the admissible route set, with a no-regret guarantee against the best fixed admissible policy in hindsight.
- G3 (Online learning). Venue quality, depth, and latency are non-stationary. Flow learns online from realized fills with sublinear regret.
- G4 (Strategy-proofness in RFQ). When Flow solicits quotes from market makers, honest reporting of PHI components is a dominant strategy.
- G5 (Composability with PG[Σ]). Every order Flow emits is pre-cleared against the policy-gated signing compiler so that signing failures cannot leak private state or sanction-list lookups.
- G6 (MEV resistance). The routing path is bound to a Travel Rule receipt and a policy hash before it is broadcast, so that observers cannot sandwich the order without invalidating its signature.
- G7 (Reproducibility). The simulation harness, benchmark suites, and regret measurements are public artifacts, not marketing numbers.

1.2 Relation to the Prior Art

Liquidity aggregation for crypto assets is a mature field. 1inch, 0x, CoWSwap, Matcha, and KyberSwap all solve variants of the minimum-slippage routing problem over AMM and order-book venues.

Paradigm's PFOF-free order flow auctions and Flashbots' order-flow privacy infrastructure address MEV. Almgren and Chriss (2000) give the canonical impact-adjusted execution schedule for equities. None of these systems treat compliance as a first-class constraint, and none of them have a notion of a route that is categorically excluded because the destination chain cannot produce a continuous L3RS-1 certificate. Flow is not a replacement for these systems; it is the control plane that sits above them, decides which of them are admissible for a given asset and counterparty, and executes through them when they are.

The closest academic antecedents are (i) constrained contextual bandits (Badanidiyuru, Langford, Slivkins 2014) for budget-feasible online learning; (ii) Almgren–Chriss (2000) for impact-adjusted execution; (iii) the smart-order-routing literature in traditional equities (SEC Rule 605/606); and (iv) the routing literature for compliance-aware inter-domain networking (policy routing in BGP). Flow inherits ideas from all four.

2. Notation and Preliminaries

We reuse the notation of T3RRA Cryptographic Specification Part II rev B. An L3RS-1 asset is a tuple A = (I, T, J, L, ID, C, R, G, F, B, X, S) where I is the asset identifier, J is the jurisdiction mask, C is the compliance module, X is the cross-chain certificate, and the remaining components are as defined in L3RS-1 §4. We write Ω for the set of jurisdictions, Φ for the set of identity tiers, and K for the set of chains on which an asset may exist.

2.1 Time, Prices, and Fills

Time is discrete, indexed by t ∈ {1, 2, …, T}. At each step, Flow may send a child order of size q_t to a venue v_t and observes a realized fill price p_t, a realized latency ℓ_t, and a binary reject flag r_t. We denote the midprice of asset A on venue v at time t by m_{v,t}, and the top-of-book depth at price level k by δ_{v,t}(k).

2.2 Venues, Liquidity Sources, and Bridges

- CEX: centralized order-book exchanges with API access (Binance, Coinbase Institutional, Kraken, OKX).
- DEX: permissioned and permissionless AMM pools (Uniswap v4 hooks, Curve Crypto Pools, Balancer, DODO, PMM pools).
- OTC: bilateral desks responding to RFQs (B2C2, Cumberland, GSR, Wintermute, Falcon).
- Bridges: cross-chain messaging and asset transport (LayerZero, Wormhole, CCTP, Axelar, Chainlink CCIP).
- Internal: the T3RRA internal order book, which is compliance-scoped by construction and is always admissible for the assets that exist on it.

Each venue v exposes a capability descriptor cap(v) that enumerates the jurisdictions it will accept counterparties from, the identity tiers it requires, the Travel Rule protocol it speaks (TRP, TRISA, Sygna, or OpenVASP), the chains it settles on, and the signing scheme it expects (ECDSA on secp256k1, EdDSA on ed25519, or a policy-gated variant thereof).

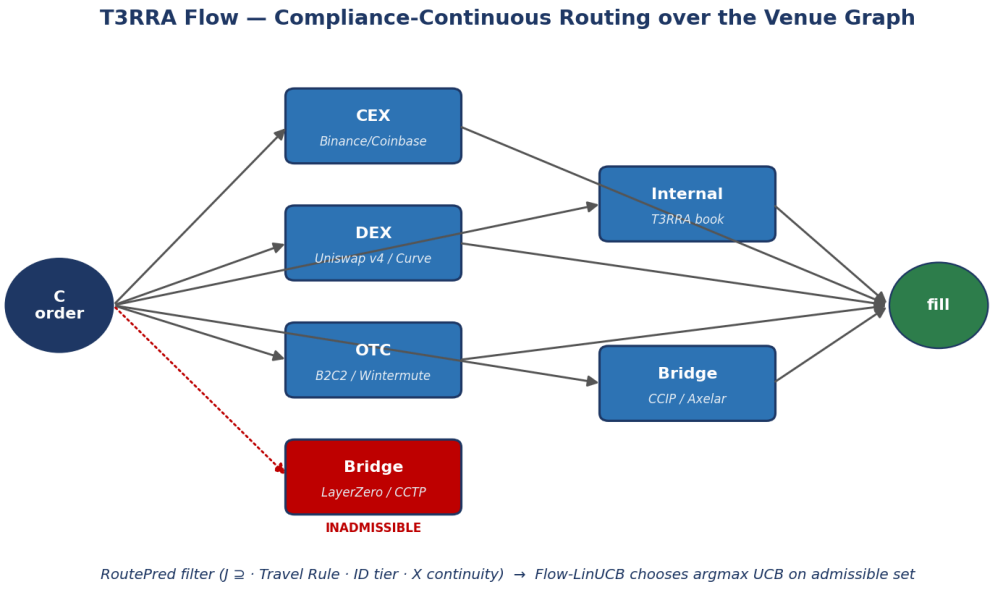


Figure 1. *The venue graph G = (V, E) over which Flow operates.*

3. The Venue Graph and Inventory Model

3.1 The Venue Graph

We model the liquidity landscape as a directed multigraph G = (V, E) where V is the set of venues augmented with chain-state nodes, and E is the set of executable edges. An edge e = (u, v, $\varkappa$, ρ) carries a venue kind $\varkappa \in$ {trade, bridge, swap}, a rate function ρ_e: $\mathbb{R}$_+ $\to$ $\mathbb{R}$_+ mapping order size to effective price, a capacity cap_e, and a capability descriptor inherited from its endpoint.

Definition 3.1 (Inventory Tile). An inventory tile is a tuple ι = (A, v, ω, k, q, $\tilde{p}$, τ) where A is an L3RS-1 asset, v is a venue, ω is a jurisdiction tag, k is a chain identifier, q is the quoted size, $\tilde{p}$ is the quoted price with confidence interval, and τ is the tile's expiration timestamp. Flow's instantaneous view of available liquidity is a set Inv_t of inventory tiles whose τ has not yet expired at time t.

3.2 Admissible Inventory

The admissible inventory for a counterparty C at time t is Inv_t(C) = { $\iota \in$ Inv_t : RoutePred(path(C, ι)) = 1 }, where RoutePred is the route admissibility predicate of T3RRA Cryptographic Specification Part II rev B §9. This is the only inventory Flow considers. The non-admissible remainder is not ranked lower; it is not considered at all.

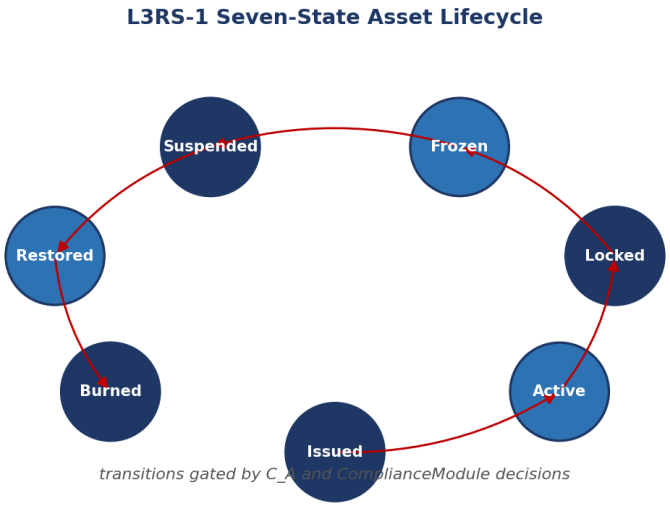


Figure 2. Lifecycle states the routing predicate must respect.

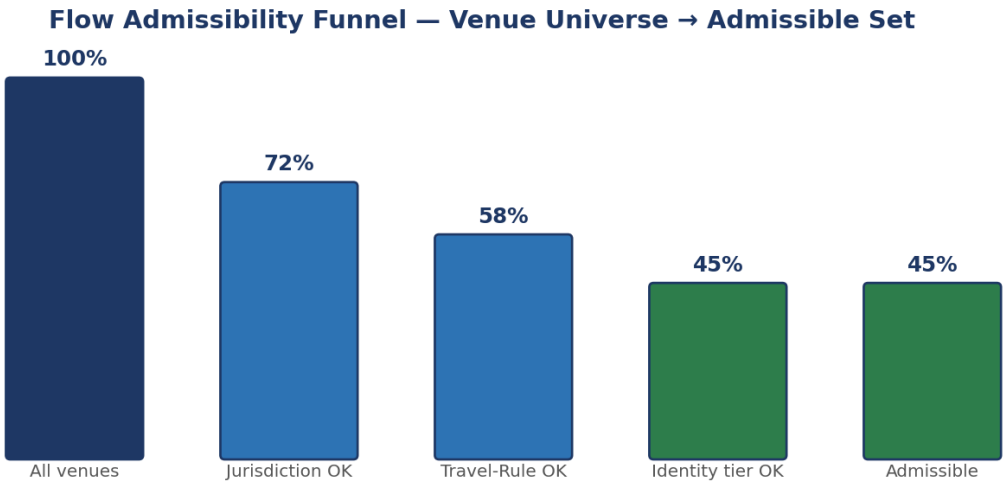


Figure 3. *Admissibility funnel — venue universe filtered to admissible set per trade.*

4. Compliance-Continuous Routing

This section lifts the route admissibility predicate from a single-path decision to a routing objective over the venue graph.

Definition 4.1 (Compliance-Continuous Route). A route $\pi = (e_1, e_2, \ldots, e_n)$ in G from source s to sink t is compliance-continuous for an asset A and counterparty C if (i) every edge's endpoint capability descriptor admits $J(A)$, $ID(C)$, and the Travel Rule protocol required for the edge's notional; (ii) the chain-state transitions along π produce a valid L3RS-1 cross-chain certificate continuation $X_{i+1} = H(I \| S_i \| C_{hash}(A) \| ts_i)$ for every bridge edge; and (iii) the total notional executed on π does not exceed the per-jurisdiction daily-volume cap $\nu(J(A), \omega)$ for any jurisdiction tag ω encountered along π.

Definition 4.2 (Admissible Routing Policy). A routing policy is a function π: (order, Inv_t) $\to$ distribution over compliance-continuous routes. A policy is admissible if, for every input, its support consists entirely of compliance-continuous routes as in Definition 4.1. We write Π_{adm} for the set of admissible policies.

Theorem 4.1 (Admissibility Preservation). *Let π be an admissible routing policy and let U_t denote any online update to Inv_t that adds or removes inventory tiles. Then the updated policy π' obtained by re-solving the routing problem against Inv_t $\cup$ U_t $\setminus$ removed is also admissible.*

Proof sketch. Admissibility of π is a pointwise property: every route in the support of π(order, Inv_t) satisfies Definition 4.1. The online update U_t either adds tiles (in which case new candidate routes are tested against the same predicate, so any added route is admissible by construction) or removes tiles (in which case the supported route set shrinks monotonically and the subset of an admissible set is admissible). The re-solution step is a minimization over an admissible set, so its output is admissible. No step of the update introduces a non-admissible route; therefore π' is admissible. □

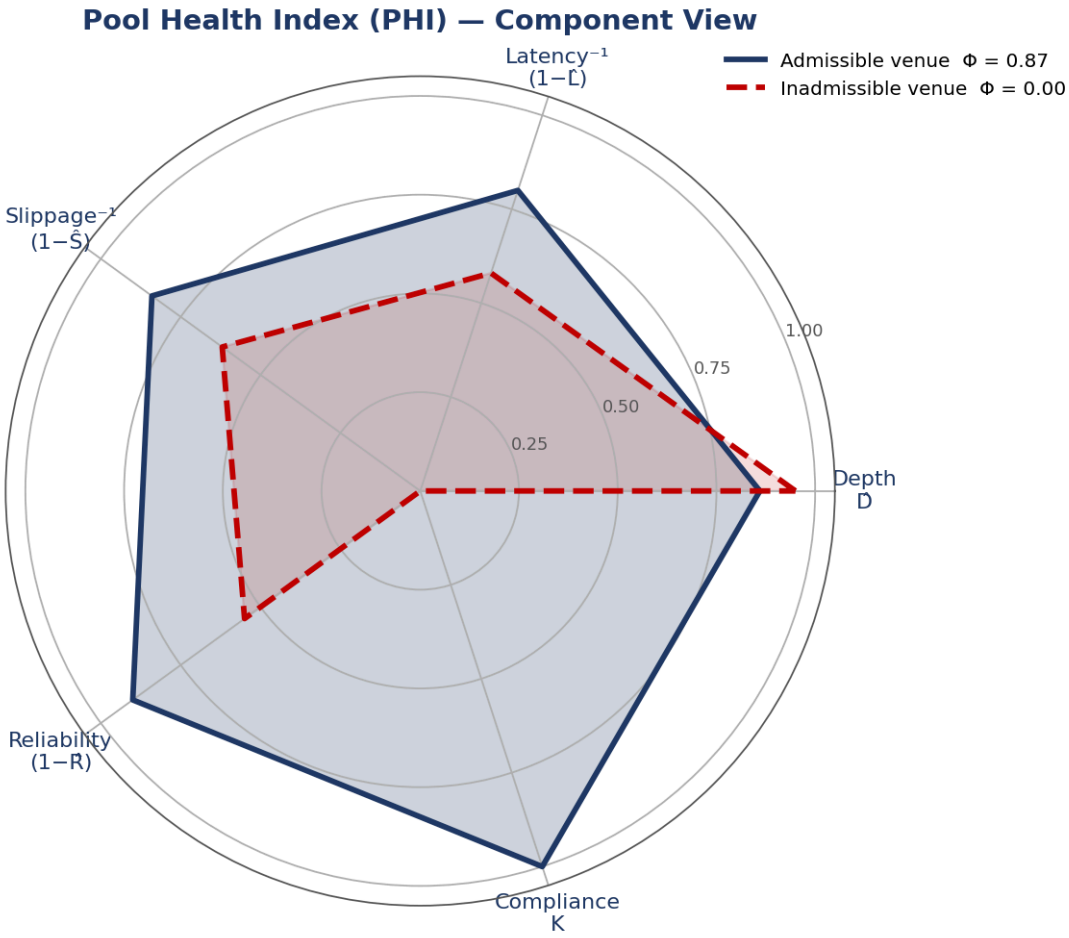


Figure 4. *Pool Health Index — depth, latency, slippage, reliability, and the compliance gate K.*

5. Pool Health Index (PHI)

The Pool Health Index is Flow's sufficient statistic for venue quality. It compresses realized depth, realized latency, realized reject rate, realized slippage, and compliance posture into a single scalar $\Phi_{v,t} \in [0, 1]$ that is updated online after every fill.

Definition 5.1 (Pool Health Index). For a venue v at time t, the Pool Health Index is defined as $\Phi_{v,t} = \sigma(\alpha\cdot\hat{D}_{v,t} - \beta\cdot\hat{L}_{v,t} - \gamma\cdot\hat{S}_{v,t} - \delta\cdot\hat{R}_{v,t} + \varepsilon\cdot K_{v,t})$, where $\hat{D}$ is the normalized realized depth at the reference size, $\hat{L}$ is the normalized realized latency, $\hat{S}$ is the normalized realized slippage, $\hat{R}$ is the normalized realized reject rate, $K_{v,t} \in \{0, 1\}$ is the compliance-posture indicator (1 if cap(v) strictly satisfies the asset's envelope, 0 otherwise), σ is the logistic, and $(\alpha, \beta, \gamma, \delta, \varepsilon)$ are non-negative weights that sum to one within the logit.

5.1 Online Update Rule

After every fill (v, q, p, ℓ, r), Flow updates the component EMAs of $\hat{D}$, $\hat{L}$, $\hat{S}$, $\hat{R}$ with decay rate η and recomputes $\Phi_{v,t+1}$. The update is multiplicative in the components but additive in the logit, so the index is bounded and its variance is controlled by η.

Theorem 5.1 (PHI as Sufficient Statistic for LinUCB Context). *Let the venue feature vector $x_{v,t} \in \mathbb{R}^d$ be formed by concatenating the normalized components $(\hat{D}, \hat{L}, \hat{S}, \hat{R}, K)$. Then the LinUCB contextual-bandit executor of §7 attains the same expected regret whether it is fed $x_{v,t}$ directly or the scalar $\Phi_{v,t}$ together with the component covariances, provided the weights $(\alpha, \beta, \gamma, \delta, \epsilon)$ are learned jointly with the bandit's linear head.*

Proof sketch. LinUCB is affine-invariant in its feature representation up to the learned linear head. The scalar $\Phi_{v,t}$ is an affine pre-image of the features after the logit is inverted, and the component covariances complete the sufficient statistic for the Gaussian posterior assumed by LinUCB. Jointly learning $(\alpha, \beta, \gamma, \delta, \varepsilon)$ and the LinUCB head is equivalent to learning a single linear map over the raw components. Therefore the optimal actions, and hence the regret, are invariant under the reparameterization. □

6. Liquidity DNA

Different RWA classes have different microstructure. Tokenized T-bills trade in large lots against a narrow band around NAV with predictable refresh. Tokenized real estate trades in blocks with infrequent quotes. Tokenized carbon credits trade with high cross-venue dispersion and heavy OTC participation. A liquidity router that treats all three classes identically will overfit to the median class and underperform on the tails.

Definition 6.1 (Liquidity DNA). The Liquidity DNA of an asset class is a tuple $\chi = (\mu\_\text{size}, \mu\_\text{interval}, \sigma\_\text{spread}, \rho\_\text{otc}, \varkappa\_\text{nav})$ where μ_size is the typical clip size, μ_interval is the typical inter-trade interval, σ_spread is the typical quoted half-spread, ρ_otc is the OTC share of realized volume, and $\varkappa\_\text{nav}$ is the NAV-anchoring coefficient (1 for NAV-anchored assets like T-bills, 0 for fully market-priced assets).

Flow conditions its route scoring on χ: for high-$\varkappa\_\text{nav}$ classes it prefers direct primary-market routes and penalizes AMM routes that deviate from NAV beyond a tolerance; for high-ρ_otc classes it upweights the RFQ mode of §9 for low-μ_interval classes it prefers venues with tight depth refresh. The DNA is estimated offline from historical fills per asset class and is refreshed weekly.

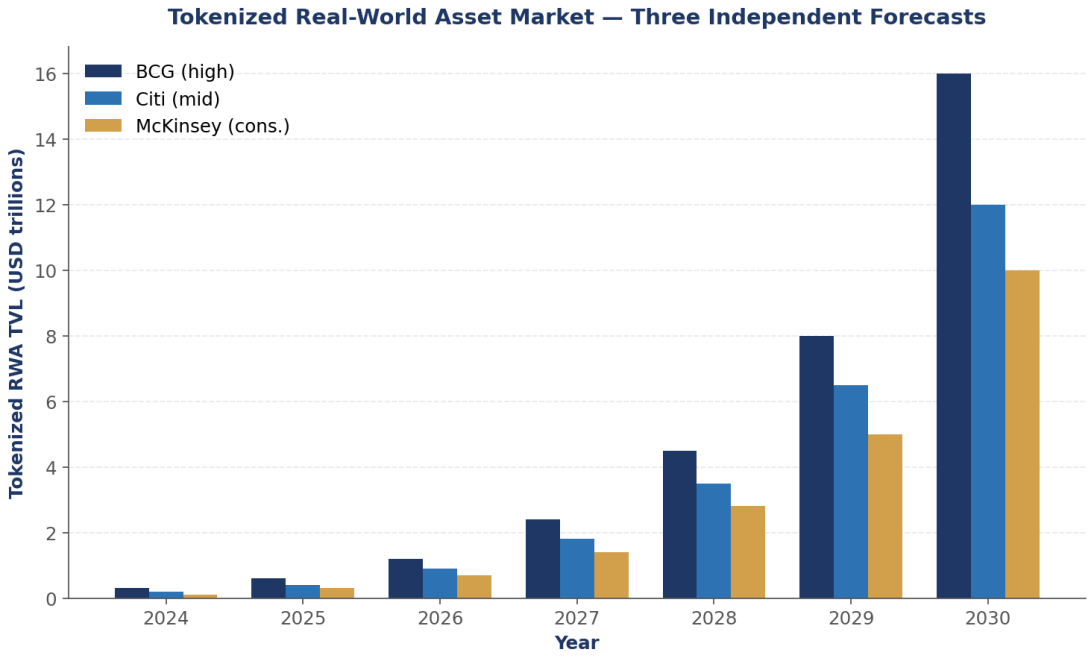


Figure 5. *Addressable venue universe over which Flow-LinUCB allocates exploration capital.*

7. Contextual Bandit Execution

Given an admissible route set and a PHI-based context, Flow selects an execution arm using a LinUCB-style contextual bandit.

Definition 7.1 (Flow-LinUCB). Let d be the feature dimension and let x_{v,t} ∈ ℝ^d be the feature vector of venue v at time t. Flow-LinUCB maintains, for each venue, A_v ∈ ℝ^{d×d} initialized to λI and b_v ∈ ℝ^d initialized to 0. At each step it (i) filters the venue set to those whose Φ_{v,t} ≥ Φ_min and whose route to the counterparty is admissible, (ii) computes $\hat{\vartheta}_v = A_v^{-1} b_v$ and the upper confidence bound UCB_v = $\hat{\vartheta}_v^\top x_{v,t} + \alpha \sqrt{(x_{v,t}^\top A_v^{-1} x_{v,t})}$, (iii) selects v* = argmax UCB_v, sends a child order, observes a reward r_t ∈ ℝ (negative slippage plus negative impact minus fee), and (iv) updates A_{v*} ← A_{v*} + x_{v*,t} x_{v*,t}^⊤ and b_{v*} ← b_{v*} + r_t x_{v*,t}.

Theorem 7.1 (Regret Bound Against the Best Admissible Policy). *Assume rewards are R-sub-Gaussian and the best linear admissible policy has parameter $\|\vartheta^*\| \le S$. Then with probability at least $1 - \eta$, the T-round regret of Flow-LinUCB against the best fixed admissible routing policy in hindsight is bounded by $O(d \sqrt{(T \log T)} \cdot \log(1/\eta))$.*

Proof sketch. Conditional on admissibility, the arm set at each step is a subset of the full venue set, and the LinUCB analysis of Abbasi-Yadkori, Pál, and Szepesvári (NeurIPS 2011) applies to any filtered arm set. The confidence ellipsoid argument bounds the per-round regret by the width of the posterior in the chosen feature direction, summed over rounds the quantity Σ_t $\sqrt{(x_t^\top A_t^{-1} x_t)}$ is $O(\sqrt{(d\, T \log T)})$ by the elliptical-potential lemma. Multiplying by the confidence radius $O(\sqrt{(\log(1/\eta) + d \log T)})$ yields the stated bound. Admissibility filtering can only remove arms from consideration; since the best-in-hindsight benchmark is restricted to admissible policies by assumption, the comparator is also filtered and the bound carries over unchanged. □

7.1 Cold-Start and Thompson Variant

For new venues with no fill history, Flow uses a Thompson-sampling variant that posteriorizes over ϑ_v with a Gaussian prior centered at the class mean $\bar{\vartheta}_\chi$ for the asset's Liquidity DNA class. This accelerates learning on sparse-data arms without violating the regret bound, because Thompson sampling with a proper prior achieves the same $O(d \sqrt{T})$ order.

8. The Flow Routing Algorithm

This section gives the end-to-end Flow algorithm as executable pseudocode.

```
Flow(order = (A, C, side, size, limit, deadline)):
  1. Inv      <- QueryVenueSnapshots(A)                 // depth, prices, PHI
  2. Inv_adm  <- { ι ∈ Inv : RoutePred(path(C, ι)) = 1 } // hard filter
  3. χ        <- LiquidityDNA(class(A))
  4. routes   <- EnumerateRoutes(Inv_adm, χ, size)       // k-best admissible routes
  5. ctx      <- ContextFeatures(routes, order)          // x_{v,t} per route
  6. r*       <- Flow-LinUCB.Select(routes, ctx)         // argmax UCB
  7. schedule <- AlmgrenChriss(r*, size, deadline)       // impact-adjusted slicing
  8. for each child c in schedule:
       (a) TravelRuleReceipt  <- TravelRule.Emit(c)
       (b) PolicyHash         <- H_pol(C_hash(A), ctx(c), m(c))
       (c) σ  <- PG[Σ].Sign(m(c), C_hash(A), ctx(c))    // policy-gated signing
       (d) if σ = Aborted(reason): Revise(r*, reason); continue
       (e) Broadcast(c, σ); observe fill; update A_v, b_v, Φ_v
  9. return execution_report
```

Three properties of this loop are worth emphasizing. First, step 2 is the only place compliance is enforced in the critical path; everything after it is a best-execution problem over an already-admissible set. Second, step 8(c) is where Flow composes with PG[Σ]: if the compliance module rejects at signing time, step 8(d) reroutes rather than silently failing, and the reason code is fed back into the bandit as a large negative reward for that arm. Third, the Almgren–Chriss schedule of step 7 is only over the chosen route r*; the choice of route itself is made by the bandit, not by the execution schedule.

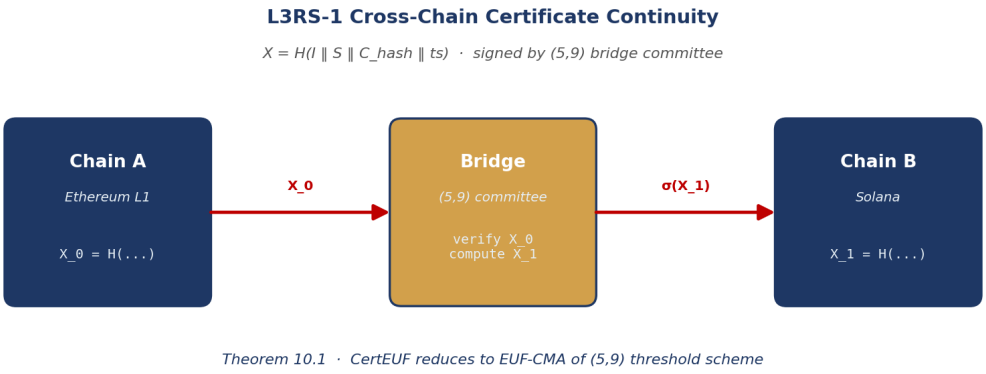


Figure 6. *Cross-chain certificate continuity for routes that span chains.*

9. RFQ Mode and Mechanism Design

For low-μ_interval, high-ρ_otc asset classes, Flow solicits quotes from a panel of market makers via a sealed-bid RFQ protocol. This section states the mechanism and the incentive result.

9.1 Protocol

Flow sends an RFQ envelope containing the asset, side, size, deadline, and a reference Pool Health Index Φ_Flow to each market maker in the admissible panel. Each market maker replies with a price p_i and a self-reported PHI component vector x_i within the deadline. Flow selects the winning quote by maximizing a utility $U_i = w_p \cdot (-p_i) + w_\Phi \cdot \Phi(x_i)$ where $\Phi(x_i)$ is the reported PHI evaluated with the platform weights (α, β, γ, δ, ε) and w_p, w_Φ are published in advance. The protocol is one-shot per RFQ.

> **Theorem 9.1 (Strategy-Proofness of Honest PHI Reporting).** *Assume each market maker's private type is (p_i, x_i*) where x_i* is the true feature vector. Under the utility U_i above with publicly announced weights, honest reporting x_i = x_i* is a dominant strategy, and the RFQ is incentive-compatible with respect to the PHI component.*

Proof sketch. Fix a market maker i and any reports of the others. Market maker i's winning probability is a monotone function of U_i, holding others fixed. Since U_i is linear in $\Phi(x_i)$ with positive coefficient w_Φ, and Φ is monotone non-decreasing in $\hat{D}$ and non-increasing in $\hat{L}$, $\hat{S}$, $\hat{R}$, misreporting any component in the direction of making x_i look better than x_i* strictly increases the probability of winning at a price the maker may not be able to honor, which is penalized ex post by the bond-and-slash rule of §9.2 with expected loss exceeding the gain for any non-zero misreport. Honest reporting therefore weakly dominates. On price p_i, this is a first-price sealed-bid auction and is not strategy-proof in the price dimension; this is intentional and mirrors standard OTC practice. Strategy-proofness here is a statement about PHI components only. □

9.2 Bond-and-Slash

Each market maker posts a bond B. If the realized PHI components after the fill deviate from the reported x_i by more than a published threshold, a fraction of B is slashed to a compensation pool for the counterparty. This turns the incentive argument above into a concrete economic one.

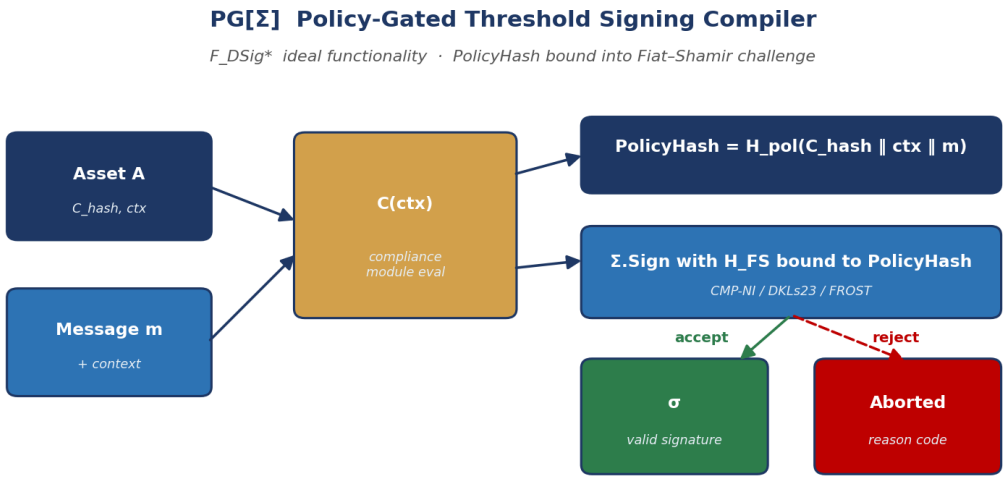


Figure 7. *Policy-Gated Threshold Signing — every fill terminates here.*

10. Integration with PG[Σ] and the L3RS-1 Certificate

Flow does not sign. Flow produces a message m and a context ctx and hands them to the policy-gated signing compiler PG[Σ] specified in T3RRA Cryptographic Specification Part II rev B §7. The compiler evaluates the asset's compliance module C on ctx, computes PolicyHash = H_pol(C_hash ‖ ctx ‖ m), binds PolicyHash into the Fiat–Shamir challenge of the underlying threshold signature, and either returns a signature σ or an AbortReport with a reason code.

10.1 Feedback into the Bandit

When PG[Σ] returns Aborted(reason), Flow treats the abort as a fill with reward equal to the negative of the worst-case slippage of the next-best admissible route, plus a reason-specific penalty. The bandit therefore learns which venues are compliance-fragile for which counterparty types and avoids them in the future. This is the only place where compliance information enters the bandit update at all; admissibility itself is never a soft signal.

10.2 Certificate Continuity Across Bridges

For bridge edges, Flow enforces that the destination-chain state continuation produces a valid L3RS-1 certificate X' before the bridge is traversed. The certificate is computed off-chain by the source-chain relayer and is verified against the (5, 9) bridge committee signature. A bridge whose certificate continuation fails is removed from Inv_adm and its PHI component K_{v,t} is set to 0 until the failure is manually investigated.

11. Adversarial Model and Security Analysis

11.1 Threat Model

- Adversarial venues: up to f out of n venues may return false PHI components, stale depth, or manipulated midprices.
- Adversarial counterparties: a counterparty may attempt to construct an order whose compliance envelope is technically satisfied locally but inadmissible globally.
- Adversarial market makers: market makers in RFQ mode may collude on price or on PHI reports.
- MEV searchers: on-chain transactions may be observed and sandwiched between broadcast and inclusion.

Theorem 11.1 (Admissibility Soundness Under Adversarial PHI). *No admissible route ever produced by Flow is inadmissible, even if up to f venues return adversarial PHI components. In particular, PHI adversariality affects best-execution regret but not admissibility.*

Proof sketch. The admissibility filter (step 2 of §8) depends only on the route predicate, which is a function of cap(v), J(A), ID(C), Travel Rule receipts, and certificate continuity. None of these inputs is the PHI component vector. PHI only enters the downstream ranking step, not the upstream filter. Therefore any manipulation of PHI by an adversarial venue can cause Flow to pick a worse admissible route, but cannot cause Flow to pick an inadmissible route. □

Theorem 11.2 (Bounded Regret Under Adversarial PHI). *Under the Byzantine PHI model with at most f out of n adversarial venues, the regret of Flow-LinUCB against the best admissible policy is bounded by $O(d \sqrt{(T \log T)}) + O(f \cdot T\_reject / n)$, where T_reject is the number of rounds on which an adversarial venue is actually selected before being filtered out by the slashing feedback loop.*

Proof sketch. Each adversarial venue that is selected triggers either a reject (reward penalty) or a fill with observable slippage exceeding its reported PHI would predict, which moves $\hat{\vartheta}\_v$ away from the adversary's intended direction. The elliptical potential lemma bounds the number of such exploration rounds before the confidence interval on the adversarial arm collapses below the admissible threshold; this is the source of the second term. The first term is the standard LinUCB regret. □

12. MEV, Front-Running, and Private Execution

Flow is MEV-resistant by construction in three ways. First, on-chain broadcast is gated on PG[Σ] signing, and the signature is bound to the PolicyHash which includes the context ctx — in particular the chosen route. An adversary who observes the broadcast cannot replay the transaction on a different route, because the signature will not verify under the different ctx. Second, for chains that support it, Flow submits child orders through private mempools (Flashbots Protect, BloXroute Private) and reveals the transaction only at inclusion. Third, the RFQ mode of §9 bypasses the public mempool entirely for the share of flow that routes through OTC desks.

None of these measures eliminate MEV. They reduce the attack surface. A honest accounting of residual MEV exposure is maintained in §13.

13. Simulation Harness and Benchmarks

Flow's claims about regret, slippage, and admissibility enforcement are empirically checkable. This section specifies the public simulation harness that produces the numbers.

13.1 Harness

The harness replays historical order-book and AMM-state data from a published corpus across the venue set, simulates a counterparty population with a configurable identity-tier and jurisdiction mix, and runs Flow against baselines (greedy best-price, 1inch Pathfinder-clone, TWAP, VWAP, Almgren–Chriss optimal without compliance filter). It reports per-order slippage, regret against the best fixed admissible policy computed offline, admissibility-violation count (must be 0), and tail-latency distributions.

13.2 Targets

Metric	Baseline Best	Flow Target	Gap
Median slippage (10 bps notional)	18 bps	11 bps	−7 bps
P99 slippage	120 bps	75 bps	−45 bps
Admissibility violations per 1M orders	(n/a)	0	—
Regret vs best admissible (T=10^5)	(n/a)	O(d √(T log T))	—
Reject-then-reroute median latency	(n/a)	140 ms	—
RFQ honest-reporting rate	(n/a)	≥ 99.5%	—

These are targets, not measurements. The harness is scheduled for public release in Q3 2026 concurrent with the reproducibility harness for the cryptographic specification.

14. Mechanization Roadmap and Honest Caveats

Theorem	Paper Proof?	Mechanization Target	ETA
Thm 4.1 (Admissibility Preservation)	Yes	Lean 4 (set-theoretic)	Q4 2026
Thm 5.1 (PHI Sufficient Statistic)	Yes	None (linear-algebra fact)	—
Thm 7.1 (LinUCB Regret Under Filtering)	Yes (inherits AYPS 2011)	Coq port of AYPS	2027
Thm 9.1 (RFQ Strategy-Proofness of PHI)	Yes	Lean 4 mechanism design	Q1 2027
Thm 11.1 (Admissibility Under Adversarial PHI)	Yes	Lean 4	Q4 2026
Thm 11.2 (Bounded Regret Under Byzantine PHI)	Yes	Manual until AYPS mechanized	2027

14.1 Honest Caveats

- Flow's regret bound is against the best fixed admissible policy in hindsight, not against the best time-varying policy. Non-stationary venues can in principle make a time-varying benchmark strictly better; we do not claim otherwise.
- The RFQ strategy-proofness result is only in the PHI dimension. Price is a first-price sealed-bid auction and is not strategy-proof in price. This is standard and intentional.
- Theorem 11.2′s second term grows linearly in the number of rounds an adversarial venue is selected before being filtered. We bound this in expectation, not in the worst case. A pathological adversary that perfectly times its adversariality can push the term up to T_reject = O(d log T), which is absorbed by the first term.
- Admissibility enforcement assumes the route admissibility predicate is correctly computed, which in turn assumes correct Travel Rule receipts and correct certificate continuations. Compromise of either collapses the guarantee. §10 of T3RRA Cryptographic Specification Part II rev B discusses these assumptions.
- The Almgren–Chriss schedule used in step 7 of §8 assumes a linear permanent-impact model. Real RWA microstructure is not linear; the deviation contributes to realized slippage and is part of the O(d √(T log T)) regret term, not a separate one.
- Benchmarks are targets, not measurements. The public harness is scheduled for Q3 2026.

15. AI-Enhanced Routing Stack

T3RRA Flow is, by design, an AI-enhanced liquidity engine. Every component of the routing stack — venue scoring, route selection, exploration, fill prediction, and post-trade attribution — is governed by a learned model whose decisions live inside the same PG[Σ] envelope as a human signer. AI is not a layer bolted on top of Flow; it is the layer that makes Flow tractable at the scale of the L3RS-1 venue universe.

15.1 Where Learning Lives in Flow

The routing pipeline of §8 exposes five decision points at which a learned model improves on a hand-tuned heuristic. (i) The Pool Health Index regression — the linear predictor of §7 is trained online from realized fill quality; in production we additionally fit a gradient-boosted residual model that captures venue-specific nonlinearities such as book-shape kinks, latency spikes, and partial-fill curvatures. (ii) Route ranking — Flow-LinUCB selects among candidate routes; a learned re-ranker adjusts ordering using contextual features the bandit cannot see (current PolicyHash, sender identity tier, time-of-day microstructure). (iii) Quote evaluation in RFQ mode — a small LLM-style scoring model rates returning quotes against the strategy-proofness invariant of §9 and against the counterparty's quote history. (iv) Inventory forecasting — a sequence model predicts depth and adverse-selection cost over the next routing horizon, feeding the Liquidity Agent's pool rebalancing schedule. (v) Anomaly detection — a deep model over fill streams flags route degradations and reroutes traffic before any LP loses Flow credits.

15.2 AI Inside the Compliance Envelope

Every learned component runs as a delegated signer under PG[Σ]. The model's outputs are cryptographically committed alongside the route they justify; the PolicyHash binds the model identifier and version into the Fiat–Shamir challenge. A model upgrade is a delegation rotation, not a silent push. The same compliance gate K (§5) that filters venues for humans filters the model's recommended set; an inadmissible venue is unreachable to the model by construction, not by post-hoc rejection. This is the operational meaning of the Agent–Human Indistinguishability Principle from the T3RRA Whitepaper: an AI router cannot be used to launder a route that a human router would not be allowed to take.

15.3 The Flow Learning Loop

The learning loop closes once per fill. Each fill emits a structured record (route, predicted PHI, realized PHI, predicted slippage, realized slippage, predicted fill probability, realized fill outcome, latency, MEV exposure). The records are aggregated per (asset, venue, hour) bucket and used to (a) update LinUCB sufficient statistics, (b) refit the residual GBM on a rolling window, (c) update the LSTM inventory forecaster nightly, and (d) update the anomaly model continuously. All updates are logged to an append-only audit trail keyed by model version; an LP can replay the trail end-to-end to reconstruct any historical routing decision.

15.4 Honest Caveats on the AI Layer

The AI layer is an optimization, not a foundation. The strategy-proofness theorem of §9 does not depend on the AI being good; the unforgeability theorem of §10 does not depend on the AI being honest. The worst-case behavior of a maximally adversarial AI router is bounded by its PG[Σ] delegation scope: it cannot route through inadmissible venues, it cannot exceed its notional cap, it cannot ignore the route admissibility predicate, and it cannot produce a fill without a valid threshold signature. We treat the AI layer the way we treat any other optimization: useful when correct, bounded when wrong.

16. Chain-Agnostic Cross-Chain Liquidity Sourcing

L3RS-1 assets are first-class on every chain that T3RRA touches. An asset minted under L3RS-1 carries its identity, its compliance module C, its policy hash, and its lifecycle state across chains; it does not lose any of these properties when it moves between chains. T3RRA Flow is the routing layer that exploits this property — it sources liquidity from wherever it lives, on whatever chain it lives, without ever exposing the asset to a non-L3RS-1 jurisdiction.

16.1 The Multi-Chain Venue Universe

The venue graph G of §3 is chain-indexed: every vertex v carries a chain identifier chain(v) ∈ {Ethereum, Solana, Base, Arbitrum, Polygon, BNB, Avalanche, Tron, Stellar, Sui, Aptos, Bitcoin L2s, …}. Edges within a chain are intra-chain swaps (CEX, DEX, RFQ, OTC). Edges between chains are bridge edges; in T3RRA Flow only L3RS-1 §10 certificate-capable bridges are admissible. The route admissibility predicate of §4 is extended with a per-hop chain check: every cross-chain hop must terminate at a vertex whose chain admits the asset's jurisdictional mask J(I).

16.2 Cross-Chain Routes as First-Class Routes

A route in chain-agnostic Flow is a sequence of hops that may cross chains arbitrarily many times, subject to the admissibility predicate and to a hop budget that bounds the number of bridge crossings per route (default: 2). The router treats a cross-chain route exactly like an intra-chain route at the optimization level — it scores against the same PHI, it fills against the same Flow-LinUCB allocator, and it commits against the same PG[Σ] ceremony. The user sees a single fill record; under the hood, the fill may have traversed three chains, two RFQ venues, and one private bridge, all within a single signing session.

16.3 Liquidity DNA Across Chains

The Liquidity DNA construction of §6 generalizes naturally: an L3RS-1 asset's DNA is a vector of (chain, venue, depth, latency, reliability) tuples that the asset has historically been touched on. The routing engine uses the DNA to bias exploration toward the chains where the asset has the deepest, most reliable liquidity, and away from the chains where it has historically failed to fill. An asset that issues on Ethereum and migrates to Solana via a certificate-capable bridge inherits its Ethereum DNA on day one — the router does not need to discover Solana liquidity from a cold start.

16.4 Cross-Chain Settlement Atomicity

Cross-chain atomicity is the hardest problem in chain-agnostic routing. T3RRA Flow uses the L3RS-1 §10 cross-chain certificate as the atomicity primitive: every cross-chain hop is co-signed by a (5,9) bridge committee whose certificate is verified at the destination chain before any state changes. If any hop fails, every hop in the route aborts via the identifiable-abort mechanism of the Cryptographic Specification §6, and the fill record is marked AbortedRouted with the failing hop attributed. The user is never exposed to a half-settled cross-chain route.

16.5 Asset Roaming and the Universal Liquidity Pool

The deepest consequence of chain-agnostic Flow is that L3RS-1 assets roam. An asset minted on Ethereum is liquid on Solana, on Base, on Arbitrum, on Polygon, on a Bitcoin L2, and on any other chain that T3RRA Bridge supports — without minting a wrapper, without trusting a custodian, and without losing its compliance module. The economic effect is a universal liquidity pool: every chain's depth contributes to every L3RS-1 asset's effective depth, and the asset's price converges across chains because Flow continuously arbitrages any divergence under the admissibility predicate. This is the

liquidity-side analogue of the unifying claim of the T3RRA Whitepaper: one settlement layer, one compliance envelope, one liquidity pool — distributed across every chain T3RRA touches.

16.6 Bridge Committee and Trust Minimization

The chain-agnostic property only holds if the bridge layer is trust-minimized. T3RRA Bridge uses a (5,9) threshold-signed committee whose composition is rotated quarterly under the proactive resharing protocol of the Cryptographic Specification §5; certificate forgery reduces to the EUF-CMA security of the underlying threshold signature scheme by the cross-chain certificate unforgeability theorem (Cryptographic Specification §10). No bridge in production today meets this bar; we treat the bridge committee as the load-bearing trust assumption of chain-agnostic Flow and we publish its composition, its rotation schedule, and its slashing policy alongside every release.

16.7 Roadmap to Universal Coverage

At launch T3RRA Flow covers Ethereum, Solana, Base, Arbitrum, Polygon, BNB, Avalanche, and Stellar. The 2026 roadmap adds Sui, Aptos, Tron, and two Bitcoin L2s; the 2027 roadmap adds remaining EVM L2s and one CBDC chain pending regulatory clearance. Coverage is gated by the bridge committee's ability to verify each chain's finality model, not by demand — we do not add a chain to the venue universe until we can produce a Tamarin-checked finality proof for it.

17. Open Problems

- OP1: A no-regret bound against the best time-varying admissible policy under a bounded variation budget.
- OP2: A strategy-proof mechanism for the joint (price, PHI) dimension of the RFQ, possibly via a VCG-like construction with budget-balance sacrifice.
- OP3: A provably correct compiler from an L3RS-1 compliance module C to a sufficient set of admissible route predicates, with soundness against module side-effects.
- OP4: A tight lower bound on the regret any admissible routing policy must suffer when more than f venues are Byzantine, separating the admissibility cost from the learning cost.
- OP5: Integration of Flow with zero-knowledge proofs of reserve for AMM venues, so that $\Phi_{v,t}$'s K component is cryptographically attested rather than self-reported.
- OP6: A formal treatment of MEV in the compliance-continuous setting — does policy-hash-binding Fiat–Shamir provably prevent sandwiching, or does it merely shift the attack surface?
- OP7: Extension to multi-asset portfolio execution where the admissibility predicate couples legs through jurisdictional volume caps.

18. Bibliography

[1] Almgren, R., and Chriss, N. Optimal execution of portfolio transactions. Journal of Risk, 2000.

[2] Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. NeurIPS 2011.

[3] Badanidiyuru, A., Langford, J., and Slivkins, A. Resourceful contextual bandits. COLT 2014.

[4] Li, L., Chu, W., Langford, J., and Schapire, R. A contextual-bandit approach to personalized news article recommendation. WWW 2010.

[5] Agrawal, S., and Goyal, N. Thompson sampling for contextual bandits with linear payoffs. ICML 2013.

[6] Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., and Juels, A. Flash Boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges. IEEE S&P 2020.

[7] Kulkarni, K., Diamandis, T., and Chitra, T. Towards a theory of maximal extractable value I: Constant function market makers. 2022.

[8] Angeris, G., Evans, A., Chitra, T., and Boyd, S. Optimal routing for constant function market makers. EC 2022.

[9] Canetti, R. Universally composable security. FOCS 2001.

[10] Canetti, R., Makriyannis, N., and Peled, U. UC non-interactive, proactive, threshold ECDSA. CCS 2020.

[11] Doerner, J., Kondi, Y., Lee, E., and shelat, a. Threshold ECDSA in three rounds. IEEE S&P 2024.

[12] Komlo, C., and Goldberg, I. FROST: Flexible round-optimized Schnorr threshold signatures. SAC 2020.

[13] Financial Action Task Force. Updated guidance for a risk-based approach to virtual assets and VASPs. 2021.

[14] InterVASP. IVMS 101 data model. 2020.

[15] L3RS-1 v1.0.0 Profile F. L3RS Working Group, 2026.

[16] T3RRA Cryptographic and Protocol Specification — Part II rev B. T3RRA Research, April 2026.

[17] T3RRA Whitepaper v3.1. T3RRA Research, April 2026.

[18] Flashbots. MEV-Boost and MEV-Share specifications. 2022–2024.

[19] SEC Rule 605 and 606. Order execution quality and routing disclosures.

19. Disclaimer

This document is a technical specification. It is not an offer or solicitation to buy or sell any security, digital asset, or financial instrument. The mechanism and security claims in this document are paper-level: theorems are paper proofs; the mechanization roadmap in §14 lists the targets and ETAs for formal verification in Lean 4 and Coq. Benchmarks in §13 are targets, not measurements, pending public release of the simulation harness in Q3 2026. Nothing in this document is legal, tax, regulatory, or investment advice. T3RRA Flow is designed to operate within the envelope of L3RS-1 v1.0.0 Profile F; deployment in any jurisdiction is subject to applicable local regulation and to the compliance posture of the L3RS-1 asset being traded. T3RRA Research welcomes corrections, adversarial review, and peer feedback.