

derien × Arc — Mainnet Settlement Reference Implementation

Status: v1.1 — September 2026

Audience: Arc builders, wallet developers, stablecoin issuers, settlement-network operators

Arc does not create the FX relationship. It settles the result of a resolution already determined across the graph.

1. The proposition

derien is a deterministic multicurrency resolution layer. Participants submit independently authorized monetary intentions. The derien engine evaluates the live population under global FIFO and full-graph implied-N logic and produces a complete ResolutionCandidate for atomic settlement.

Arc is the first reference settlement environment. Arc does not determine the FX relationship; it atomically settles the result determined by derien.

derien resolves. Arc settles.

2. What the reference implementation demonstrates

The reference implementation demonstrates independently authorized participant wallets resolving through the existing derien engine and settling atomically on Arc Mainnet. Phase 1 uses USDC and EURC. The architecture is multicurrency from inception and extends to genuine implied-N as additional canonical Arc assets become available.

derien does not custody participant assets and does not rely on a DEX, broker, market maker, or third-party FX execution layer.

3. Resolution domain

A derien resolution domain is the population of intentions capable of settling atomically inside the same programmable settlement environment. For the Arc reference implementation, the domain is Arc Mainnet plus one deployed DerienArcSettlement contract and one active asset registry.

The network is the shared ledger; derien is not itself the ledger.

4. Participant authorization

Each participant cryptographically authorizes its economic intention, including source asset, target asset, quantity bounds, recipient, price constraint, and expiry.

The participant's hard price constraint is an eligibility condition, not an execution-price engine. The assigned valuation matrix determines the internal resolution quantities.

Participant assets remain in their wallets until settlement.

The reference implementation uses EIP-712 for intention authorization and bounded Permit2 authorization for token movement at settlement.

5. Admission to resolution

Only valid, unexpired, independently authorized intentions referencing recognized assets are admitted to resolution.

6. Canonical derien intention

The existing derien engine remains authoritative. The Arc adapter conforms to the engine; it does not introduce alternative matching, FIFO, closure-selection, or quantity logic.

7. Valuation topology and price policy

The topology defines value. The engine resolves it. Arc settles the result.

The engine operates against one coherent valuation matrix. For the reference implementation, the matrix is fixed for each resolution run. Production market-data sourcing and publication are separate from the settlement architecture.

8. Resolution semantics

Phase 1 proves direct atomic settlement using independently authorized USDC/EURC intentions. It is deliberately bilateral and is not presented as an implied-N demonstration.

Phase 2 introduces three or more currencies and demonstrates a genuine multicurrency ResolutionCandidate generated by the existing derien engine. The target deterministic cycle is:

```
USDC → EURC → BRLA → MXNB → USDC
```

9. Resolution object

A ResolutionCandidate contains the complete participant set and quantities selected by the derien engine. The settlement adapter may translate its representation but may not alter its economics.

10. Arc settlement contract

DerienArcSettlement.sol validates the authorized settlement payload and atomically realizes the complete engine result. It performs no matching, pricing, oracle lookup, order selection, or implied-N calculation.

11. Atomicity and conservation

A resolution is indivisible for settlement. All member movements execute atomically in one Arc transaction. The settlement contract enforces conservation across the complete result; if the required checks fail, the entire transaction reverts.

ALL MEMBERS SUCCEED — OR THE ENTIRE SETTLEMENT REVERTS.

12. Settlement lifecycle

1. Participants independently authorize intentions.
 2. Valid intentions enter the existing derien engine.
 3. The engine deterministically produces a ResolutionCandidate.
 4. Participants authorize the token movements required by that result.
 5. The complete result settles atomically in one Arc transaction and is independently observable onchain.
-

13. Security model

The settlement adapter can verify and execute the engine result but cannot rewrite either the participant's authorized economic instruction or the engine's ResolutionCandidate.

It cannot change the assets, recipient, authorized quantity, participant set, or resolved economic quantities, and completed settlements cannot be replayed.

14. Privacy and custody

Participant private keys remain under participant control and assets remain in participant wallets until atomic settlement. derien takes no custody.

15. Economic properties

No principal position, no market-maker spread, no broker margin, and no derien execution or resolution fee for compatible intentions that close inside the graph.

Network and third-party infrastructure costs remain external to derien.

16. Reference implementation split

Component	Role
Existing derien engine	Matching, FIFO, partial fills, full-graph implied-N, deterministic ResolutionCandidate
Engine adapter	Deterministic mapping to/from the pinned native interface
Arc integration layer	Signatures, asset registry, candidate validation, settlement construction, submission, reconciliation
DerienArcSettlement.sol	Onchain verification, bounded pulls, conservation, atomic payouts, replay protection, events
Arc	EVM execution, consensus/finality, network-level compliance, USDC gas

17. Phase 1

Phase 1 is complete when two independently authorized USDC/EURC intentions are resolved by the existing derien engine and the complete resulting state transition settles successfully and atomically in one observable Arc Mainnet transaction. The finalized onchain state reconciles to the ResolutionCandidate produced by the

derien engine.

18. Phase 2

Phase 2 extends the same architecture to at least three currencies and demonstrates genuine implied-N resolution. Phase 2 targets USDC, EURC, BRLA and MXNB, subject to canonical Arc asset availability.

The Phase 2 acceptance gate is a genuine 3+ currency ResolutionCandidate produced by the existing engine and settled atomically through the same contract architecture used in Phase 1.

19. The proposition in one sentence

derien determines the resolution; Arc atomically realizes the resolved monetary state.

The Arc reference implementation is currently being built on Mainnet, beginning with direct USDC/EURC settlement and extending to multicurrency implied-N as additional canonical assets become available.