haidrun

TECHNICAL WHITEPAPER

A private Layer 1 for regulated digital money

A private Layer 1 for regulated digital money — architecture, consensus, security and compliance.

DRAFT V0.1 · NOT FOR DISTRIBUTION

Haidrun AB · Stockholm, Sweden · Reg. no. 559259-8550
2026-07-31 · Internal review draft · haidrun.io

Haidrun secures the issuance and movement of stablecoins, the tokenization of real-world assets, and institutional payments and settlement on a private, permissioned blockchain — deployed inside an institution's own security perimeter, with compliance and settlement native to the protocol rather than added at the application layer.

> **Draft status.** First technical draft for internal and engineering review. Two areas are deliberately held open and must be completed by their owners before publication: `[ENGINEERING]` the exact consensus mechanism and finality proof (the wording here is the agreed public framing — "evolved Proof-of-Vote + DPoS, deterministic finality"; the precise technical description is to be finalized by the Haidrun engineering team); `[LEGAL]` all regulatory/compliance statements, pending legal sign-off — nothing here is a claim of current licensure. `[VERIFY]` marks performance/market figures needing source confirmation. This document is informational and is not financial or legal advice.

1. Introduction & design goals

Existing infrastructure forces regulated institutions into a trade-off: the programmability and settlement speed of public blockchains, or the control, privacy, and compliance posture their obligations demand. Haidrun is designed to remove that trade-off. Its design goals are:

- **Sovereignty** — run entirely within the institution's perimeter, with no external blockchain dependencies.
- **Determinism** — final, irreversible settlement with no forks, reorganizations, or rollback logic.
- **Compatibility** — reuse the EVM toolchain and connect to existing bank, KYC/AML, and payment systems through standard APIs.
- **Compliance by construction** — embed KYB/KYC, AML, reserves, and audit into the core data model. `[LEGAL]`
- **Performance at institutional scale** — tens of thousands of transactions per second with sub-second finality.
- **Energy efficiency** — a consensus mechanism that avoids proof-of-work cost.

2. System architecture

Haidrun is a single platform organized in three layers, each independently addressable.

Engine / Layer 1. Haidrun's proprietary blockchain core is the settlement and execution engine — deterministic, energy-efficient, with native interoperability built directly into the protocol. Institutions can build directly on it through the SDK. The engine is the authoritative ledger for stablecoin supply: total supply is always derivable from chain state.

API layer (B2B). Industry-standard, versioned REST APIs secured with OAuth 2.0 + JWT and TLS 1.3 / mutual TLS. Core endpoints cover authentication, transactions, wallets, KYC/KYB, and complaints handling (MiCA Art. 71) `[LEGAL]`. The API layer lets banks, KYC/AML providers, and external payment rails connect plug-and-play, with no custom protocol development. OpenAPI specifications are generated from the API so documentation stays in sync with the implementation.

Application layer (B2C). Customer-facing apps, wallets, and payment experiences ship on the same foundation, built with the SDK. This is where an institution extends from B2B infrastructure to retail products without changing the underlying rails.

3. Consensus & finality — `[ENGINEERING: exact mechanism to be finalized]`

Haidrun runs an **evolved Proof-of-Vote + DPoS** consensus with a governed validator set, delivering **deterministic finality**: once a transaction is committed, it is final and irreversible — no forks, no reorganizations, no rollback logic.

- **Validator model** — a permissioned, governed set of validators (delegated proof-of-stake selection).
  [ENGINEERING: validator selection, staking/delegation parameters, rotation, governance of the set.]
- **Voting & commitment** —
  [ENGINEERING: describe the Proof-of-Vote round structure, quorum/threshold, message flow, and the point at which finality is achieved.]
- **Finality guarantee** — deterministic single-slot/round finality (no probabilistic confirmation depth).
  [ENGINEERING: state the exact finality condition and its safety/liveness assumptions.]
- **Fault tolerance** — *[ENGINEERING: Byzantine fault-tolerance threshold and behavior under validator failure/partition.]*
- **Energy profile** — vote-based agreement avoids proof-of-work computation.
  [ENGINEERING/VERIFY: quantify energy characteristics if a figure is to be published.]

4. Execution & the Haidrun Virtual Machine (HVM)

The HVM extends the EVM. Institutions deploy Solidity smart contracts and reuse standard EVM tooling — compilers, libraries, and auditing frameworks — with deterministic execution under load.

- **EVM compatibility** — existing contracts and developer tooling port with minimal change.
- **Parallel execution** — independent transactions execute in parallel for higher throughput, while preserving deterministic ordering guarantees.
  [ENGINEERING: concurrency model, conflict detection, and state-access scheduling.]
- **Upgradability** — an upgradable path to future EVM standards.

5. Interoperability

Interoperability is built into the core rather than added through external bridges or a Layer 2 — a deliberate choice to avoid the security and latency costs those introduce.

[ENGINEERING: describe the native interoperability model — supported message/asset transfer semantics, trust assumptions, and how cross-network settlement prese

6. Performance & scalability

METRIC	FIGURE
Throughput (production)	15,000–30,000 TPS
Throughput (optimized lab)	up to 65,000 TPS
End-to-end latency (p99)	< 350 ms
Transaction finality	< 500 ms
Availability	99.99%
Recovery time objective (RTO)	< 5 minutes

Throughput scales through infrastructure — additional capacity and network resources — rather than architectural redesign, with a documented, zero-downtime path to higher throughput. **[VERIFY: confirm all figures and test conditions before publication; footnote methodology.]**

7. Security architecture

Security is embedded in the architecture and data model.

- **HSM-first key management** — all signing keys reside in FIPS 140-3 Level 3 hardware security modules; private keys never leave the hardware boundary.
- **Network** — TLS 1.3 with mutual TLS for all API clients, a WAF with OWASP ruleset, DDoS protection, private subnets, and an isolated HSM network segment.
- **Data** — AES-256 encryption at rest, TLS 1.3 in transit, field-level encryption for PII, and automatic key rotation.
- **Access** — role-based access control, multi-factor authentication for administrative access, least privilege, and just-in-time elevation.
- **Audit** — an immutable audit trail with 7-year retention, real-time alerting on security events, and regular third-party penetration testing.
- **Data sovereignty** — deployment inside the institution's own perimeter (private cloud or on-premises); the institution's network, data, and rules, with no external blockchain dependencies.

8. Stablecoin model (HSC)

An institution mints and burns its own backed stablecoin (the **HSC** model) within its controlled environment, with reserves, token lifecycle, and compliance managed on-platform. Because the engine is the authoritative ledger for supply, **total stablecoin supply is always derivable from chain state**, supporting real-time reserve tracking and attestation. Multi-currency issuance is supported.

[ENGINEERING: mint/burn authorization flow, reserve-linkage and attestation mechanism.] *[LEGAL: reserve and redemption representations.]*

9. Tokenization & real-world assets (RWA)

Haidrun provides full, built-in functionality to issue and manage real-world assets on-chain — securities, funds, real estate, private credit, commodities, and carbon credits. Capabilities include compliant issuance and whitelisting, lifecycle management (distributions and redemptions), fractional ownership, programmable transfer rules, custody and registry, and secondary transfers — with compliance and settlement native to the platform, on the institution's own private Layer 1.

10. Compliance architecture — **[LEGAL: pending sign-off]**

Regulatory requirements are embedded in the core data model and workflows rather than implemented per application.

- **MiCA** — reserve requirements, on-chain reserve attestation, HSM key custody with full audit trail, treasury controls, complaints workflow (Art. 71), tiered client identification.
- **US GENIUS Act** — 1:1 reserve backing with real-time tracking, permitted reserve assets, monthly attestation, redemption at par, reserve segregation.
- **FATF** — real-time AML transaction monitoring, risk-based limits, sanctions screening (EU, UN, OFAC), case management with SAR generation, dynamic risk scoring with ongoing due diligence.
- **VARA and beyond** — engineered to support additional frameworks.
- **KYB / KYC / AML** — document verification with liveness checks, beneficial-owner verification, sanctions/PEP screening, ongoing monitoring with anomaly detection, and risk-based classification with enhanced-due-diligence triggers.

> Haidrun is in the process of applying for licenses. The platform is engineered to be compatible with, and to facilitate compliance with, these frameworks; this is **not a claim of current licensure**. All statements in this section are pending legal sign-off.

11. Deployment & operations

Deploy on all major clouds — Oracle Cloud, Google Cloud, Microsoft Azure, AWS — or on-premises / private cloud. Multi-availability-zone by default, with zero-downtime scaling and support for geographic expansion. Operations target 99.99% availability with an RTO under five minutes.

12. Technology stack

- **Blockchain core / Layer 1** — Haidrun's proprietary engine (consensus per §3).
- **Backend services** — Node.js, TypeScript, Fastify.
- **Event streaming** — Apache Kafka with exactly-once semantics.
- **Key management** — hardware security modules validated to FIPS 140-3 Level 3.
- **Application / UI** — React, Tailwind CSS.
- **Interfaces** — REST APIs (OAuth 2.0 + JWT, TLS 1.3 / mTLS), native SDK, HVM (EVM-compatible), OpenAPI specifications.

Appendix

Glossary. *StablePay* — the enterprise stablecoin platform on the Haidrun Network. *HSC* — Haidrun StableCoin. *HVM* — Haidrun Virtual Machine (EVM-compatible). *DPoS* — Delegated Proof-of-Stake. *HSM* — Hardware Security Module.

Open items before publication. `[ENGINEERING]` consensus/finality/interoperability/parallel-execution detail (§3–§5, §8); `[LEGAL]` all of §10 and reserve/redemption language; `[VERIFY]` performance figures (§6) and any market statistics; `[INPUT]` certifications (ISO 27001 / SOC 2) if held.

Footnotes & disclaimers. 15,000–30,000 TPS in production depending on infrastructure and network conditions; up to 65,000 TPS measured in optimized lab / closed-network environments. `[VERIFY]` This document is informational only and is not financial, investment, or legal advice. Regulatory statements are pending legal sign-off and are not a claim of current licensure. Forward-looking statements about capabilities and roadmap are subject to change.