KERN

INSTITUTIONAL BRIEF
kern.brussels

The noyau. The kernel. The grain of state that endures.

A Layer-1 **settlement and registry layer** engineered for institutional legibility — deterministic finality, machine-checked contract invariants, and a documented on-chain evolution path, with EVM compatibility delegated to rollups so the base layer stays small and auditable.

THE STRUCTURAL GAP

Today's Layer-1 landscape forces a trade-off. Ethereum-family chains win on developer gravity and tooling, but evolve through contentious hard forks and leave properties like solvency or supply caps as claims in an audit rather than guarantees in the protocol. Rigour-first chains offer on-chain governance and verifiable contracts but lack developer mass. Neither makes **regulatory obligations machine-verifiable**. KERN is a deliberately narrow base layer built to close that gap: it settles, governs, attests and registers — and pushes execution out to rollups.

WHAT KERN IS

1 **Deterministic finality.** A four-phase BFT consensus in the Byzantine-fault-tolerant family with ~1s blocks and ~2s finality. No probabilistic reorganisations.

2 **Liquid Proof-of-Stake.** Delegate stake without surrendering custody — balances stay in the holder's account. No liquid-staking derivative, no lock-up.

3 **Verifiable contracts.** Skald is a small, statically typed language that enforces declared invariants at runtime. A lending market cannot enter an insolvent state — the runtime rejects the transition.

4 **On-chain evolution.** Protocol amendments — consensus parameters, fees, even Skald itself — pass through a dual-track governance cycle visible to all stakeholders. No hard forks.

5 **EVM via rollups.** Existing Ethereum applications run on a KERN-anchored optimistic rollup with standard tooling unchanged, while the L1 stays narrow and auditable.

\+ **Slashable attestations.** Equivocation accountability generalised beyond consensus to any signed off-chain claim — KYC, oracle prices, NAV, notary or ESG attestations.

WHY IT MATTERS TO AN INSTITUTION

> **Predictable finality, not probabilistic.** A settlement made at second T is final at T+2 — no rollback to explain to a supervisor. Banking-grade auditability of state.

> **Provable correctness, not asserted.** "This pool is always solvent" stops being an audit promise and becomes a protocol-enforced runtime guarantee.

> **Documented evolution, not coordination crisis.** Every change follows a multi-week, on-chain voting path; the route an upgrade took is itself part of the record.

> **Auditable by design.** The rules — who may transfer, under what conditions, with what limits — are readable at the source by regulators, auditors and counterparties, not buried in a 200-page report.

Compliance by construction — tokenised securities

Three Skald templates (startup equity, institutional fund, real-estate) encode the EU **securities** regime as machine-checked invariants: the **Prospectus Regulation**, **MiFID II** (Art. 16 & 24), **AIFMD** (depositary independence, segregation, diversification) and **MAR** (market-abuse blackout). Security tokens are financial instruments and are therefore excluded from MiCA under Art. 2(4); the applicable framework is securities law. A competent authority reads the invariants directly, supplementing rather than replacing the audit cycle.

WHERE IT FITS — FLAGSHIP USE CASES

Sovereign stablecoins & CBDC pilots. Issuance held by an institution-controlled multisig; per-address holding caps and reserve-attestation rules as invariants; monetary parameters amendable through governance with the issuer as sole proposer. ~2s finality matches payment expectations; the amendment path is friendlier to regulators than off-chain forks.

Regulated DeFi, tokenised securities & RWAs. A market declares invariant solvent { assets ≥ liabilities }; the runtime rejects any breaching transaction, including oracle manipulation and flash-loan attacks. Solvency becomes an on-chain property; issuers get a settlement layer they can credibly describe to auditors.

Public-sector registries. Land titles, civil registers, licensing, procurement — transfer conditions and persistence encoded in Skald, parameter changes through governance, records queryable by light clients. No notary need explain probabilistic reorgs; the registry's own evolution is part of the public record.

Settlement hub for EVM rollups. Multiple rollups post state commitments to KERN; cross-rollup transfers settle through it, bridge accounting is invariant-enforced, and rollup upgrades are themselves governed. The chain that settles should not be the chain optimised for execution — the narrowness is the feature.

HOW IT COMPARES

	ETHEREUM L1	TEZOS	SOLANA	KERN
Finality	Probabilistic (~12 min)	Deterministic (~30 s)	Deterministic (~12 s)	**Deterministic (~2 s)**
On-chain governance	No	Yes	No	**Yes (dual-track)**
Contract verifiability	Hard (Solidity)	Strong (Michelson)	Hard (Rust / BPF)	**Strong (Skald + runtime invariants)**
EVM compatibility	Native	Via rollup	No	**Via optimistic rollup**
Design intent	General execution	Self-amending L1	High throughput	**Settlement & registry**

STATUS — AN HONEST ACCOUNT

MATURITY
v1.1-rc
Reference implementation in Python; ~700 automated tests, all green.

NETWORK PATH
Devnet → Testnet
Yggdrasil (public testnet) → Midgard (mainnet), permissionless validators.

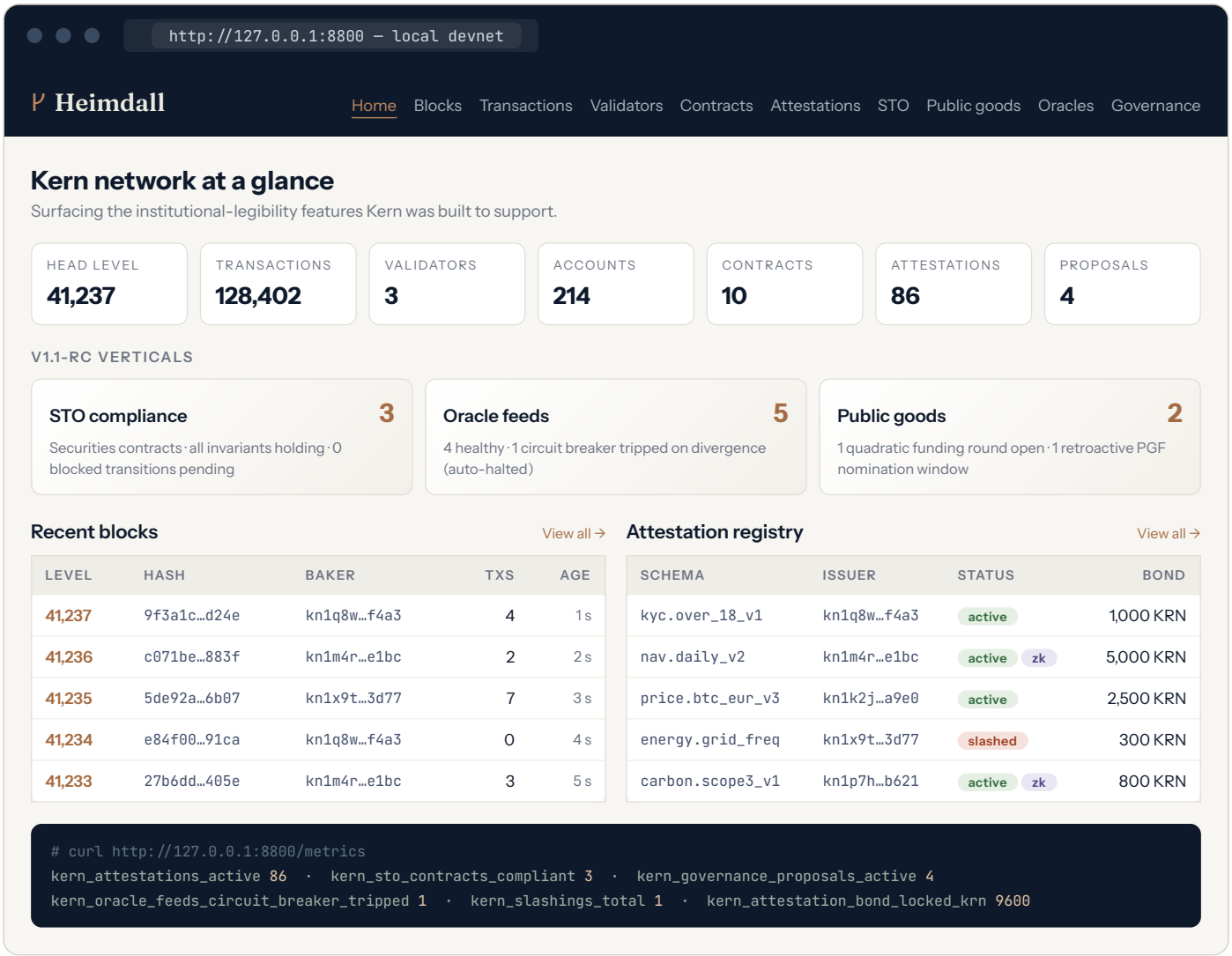
LICENSING
Apache-2.0
Code Apache-2.0 (explicit patent grant); docs CC-BY-SA-4.0.

KRN, the native token, pays fees, secures the validator set as Liquid-PoS collateral, weights governance, meters storage and funds an on-chain treasury. The implementation is **pre-audit**: a first external security audit precedes any public testnet carrying value — it is for research, education and building rollup applications against a stable settlement abstraction, **not** for securing real value today.

THE EXPLORER — HEIMDALL

Legibility needs a window. Heimdall is Kern's official block explorer and monitoring stack — it surfaces what generic explorers cannot: the live securities-compliance state of each STO, oracle health per feed, the active attestation registry, and the slashing economy. It ships with the v1.1-rc reference implementation and runs against a local node out of the box.

FastAPI + SQLite indexer · 19 HTML pages · 12 JSON API endpoints · 7 Grafana dashboards · 10 Prometheus alerting rules ·
`pip install -e ".[explorer]" && heimdall`



What is shown. The Heimdall home view as shipped in the v1.1-rc reference implementation, populated with illustrative local-devnet data. The interface, navigation, metric names, and table structures are those of the actual software; figures are representative of a development network, not a live public chain. Operator guide: `docs/setup-heimdall-operator.md`.

COMPLIANCE, LIVE

Per-contract dashboards

Each STO contract is auto-classified by Skald template; its invariants are displayed with their current truth value, block by block.

ACCOUNTABILITY, PRICED

The slashing economy

Active bonds, equivocation evidence, and slashing events are first-class objects — the cost of lying on-chain is a queryable figure.

OPERATIONS, MONITORED

Prometheus → Grafana

~36 exported metrics, 7 dashboards, 10 alerting rules — the same observability surface an institutional operator would demand.

LEGAL NOTICE

This brief is a standing notice from the author of the KERN reference implementation. By reading or relying on it, you accept the terms below. They summarise, and are subordinate to, the full disclaimer published in the repository and on kern.brussels. Nothing here overrides the licences under which the project is released.

01 **No offer or solicitation.** This document is provided for information only. It is not, and must not be construed as, an offer, invitation, inducement or solicitation to buy, sell or subscribe to any security, financial instrument, token or investment, nor as advice to enter into any transaction. KRN is described solely as a protocol utility and governance unit; nothing here is an offer of KRN or of any instrument associated with it.

02 **No professional advice.** Nothing in this brief constitutes legal, financial, investment, tax, accounting, regulatory or fiduciary advice. Any party considering a deployment in a regulated context must engage its own licensed counsel, compliance officer, auditor and competent national supervisory authority (e.g. FSMA, AMF, BaFin, CSSF) before relying on any property described here.

03 **No assurance of compliance.** References to EU regulation (MiFID II, the Prospectus Regulation, AIFMD, MAR, MiCA, DORA) are informational, illustrative and forward-looking design discussions — not regulatory determinations and not assurances of compliance. No regulator has approved, endorsed or certified the software, the Skald templates, the attestation primitive or any artifact described. “Compliance by construction” denotes a design approach; it is not a representation that any given deployment is compliant.

04 **Forward-looking statements.** The roadmap, the network path (Devnet → Yggdrasil → Midgard) and any token-related design are statements of intent, not commitments, and may change without notice in light of audit findings, regulatory dialogue, legal advice, technical feasibility and the author’s capacity. No party should acquire, transact in, or take any position in the expectation that a milestone will occur, on any timeline, at any price, or with any outcome.

05 **Pre-audit software; no warranty.** The reference implementation has not undergone an independent professional security audit; such an audit is a gating item before any public testnet carrying value. It is published “AS IS” under the Apache-2.0 licence, without warranty of any kind — including merchantability, fitness for a particular purpose, non-infringement, security, accuracy or regulatory compliance — and is unsuitable for production use with material value at stake.

06 **Limitation of liability.** To the maximum extent permitted by applicable law, Nicolas Van Eeckhout and any KERN contributor shall not be liable for any direct, indirect, incidental, special, consequential or punitive damages — including loss of profits, revenue, data or goodwill, regulatory penalties, or third-party claims — arising from or in connection with the use of, reliance on, or inability to use the software or this brief, regardless of the form of action and even if advised of the possibility of such damages.

07 **No endorsement; no agency.** The author does not control, operate or endorse any third-party network, deployment or representation, nor any token sale, listing, airdrop or derivative purporting to be associated with KERN. Use of the project creates no agency, partnership, joint venture, fiduciary duty or other relationship beyond the applicable licences (Apache-2.0 for code; CC-BY-SA-4.0 for documentation). KERN is an open-source project, not an incorporated entity; until a foundation is incorporated, the author makes no representations on behalf of any third party.

08 **Trademarks.** “KERN”, “Skald”, “Heimdall”, “Yggdrasil” and “Midgard” are reserved as marks of the KERN project. Third-party names (Ethereum, Tezos, Solana and others) are the trademarks of their respective owners, used here under nominative fair use / comparative reference solely to identify the subjects of factual comparison. No affiliation, sponsorship or endorsement is asserted or implied, and no third-party logo or trade dress is reproduced.

09 **Governing terms & distribution.** The licence terms govern any dispute relating to the software; the author is based in Belgium. This brief is not directed at any person in any jurisdiction where its publication or distribution would be unlawful, and recipients are responsible for observing any applicable restrictions. The full disclaimer in the repository prevails over this summary in case of any inconsistency.