

VEILFORGE V4

VeilForge V4 Executive Brief

Deterministic Privacy-Readiness Analysis and Verifiable Evidence for Solidity on Arc

DOCUMENT	Executive brief
STATUS	Grant Candidate / evidence package
DATE	August 2026
SOURCE DIGEST	sha256:ffafb30639b265d41944a136182017eadad23396054b48ef9f995c2f2c0f5185

Evidence-first release engineering for Solidity teams building on Arc.

Deterministic Privacy-Readiness Analysis and Verifiable Evidence for Solidity on Arc

Status: Grant Candidate brief Product version: 4.0.0-gc.1 Technical candidate: V4 RC1 Grant status: V4 Grant
Candidate Implementation boundary: Arc Testnet; mainnet deployment is not claimed

VeilForge is an independent open-source project. This brief does not claim Circle or Arc endorsement, a grant award, customers, revenue, audit assurance, universal correctness, or confidentiality.

Problem

Financial Solidity applications can expose information through their own behavior even when teams are attentive to chain-level privacy. Public calldata, events, storage and getters, return or revert data, metadata, and arguments passed to external calls can make payment, treasury, borrower, lender, collateral, or terms information observable. Some disclosure is intentional and necessary; some conflicts with application policy. Teams need a repeatable way to distinguish the two before deployment.

Manual review remains important, but it is difficult to reproduce continuously. Review notes may not provide stable identities or machine-readable evidence, and periodic audits do not cover every development change. A tool that silently skips unsupported paths can also create false confidence. Privacy readiness therefore needs compiler-backed provenance, explicit uncertainty, stable reports, and bounded claims.

Solution

VeilForge V4 is a local-first, deterministic privacy-readiness scanner for Solidity, initially focused on Arc Payments, Arc Treasury, and Arc Private Credit. It compiles exact Solidity 0.8.24 projects and builds an AST index, normalized intermediate representation, control-flow graphs, a call graph, and intra/interprocedural dataflow. Classification and domain detectors turn source-to-sink evidence into stable findings with severity, confidence, disposition, location, trace, context, remediation, and explicit incomplete-analysis states.

The same evidence model supports a browser scanner, CLI, SDK, JSON and Markdown exports, SARIF, GitHub Action integration, a policy release gate, verified local history, and an optional Arc Testnet proof workflow. Browser analysis runs in an isolated Web Worker and does not send source to a remote AI analysis service. Reports and exports can still be sensitive and must be handled accordingly.

Reports use schema 4.1.0 and canonical hash payload `veilforge.report.hash.v2`. Verification recomputes integrity rather than trusting a UI flag. "No finding" is not presented as proof of confidentiality. VeilForge is evidence for engineering review, not an audit or formal-verification replacement.

Current Evidence

The maintained benchmark has 60 oracle cases: 20 each for Arc Payments, Arc Treasury, and Arc Private Credit. The recorded release baseline is 60/60 cases passed, 56 true positives, zero false positives, zero false negatives, zero negative-case false positives, release gate passed / allow, and zero nondeterministic results. This result is bounded to the maintained corpus. It is not a statement that every Solidity project will produce the same performance.

The browser runtime uses exact solc 0.8.24, limits projects to 100 files, 512 KiB per file, and 1 MiB total, and supports one active scan per client. Abort, timeout, crash, cleanup, and restart behavior are explicit. Chromium and WebKit have documented local acceptance evidence; Edge and Firefox clean-CI acceptance remains a bounded limitation.

The production V4 feature flag defaults to false and V3 remains the default build.

VeilForge also has a real Arc Testnet proof-publication identity. Transaction

0xdb674c986195ed9b3950f34d058637fbb2b887f58ca724400225ba177884192c, block 55469453, published report hash sha256:fce5ffa529c79d185a6013a362e25658020d1691550557d59173c9acc6a417ea from publisher 0x60B6333a0722bBEA39d4026b284Ae1E142bEb914 to Registry V2 at

0x88B4055eaB061CEa9BdfeFF524f65ff461B5401d. The method was `publishReport`, value was 0 USDC, and the observed Testnet fee was 0.001175966 USDC. Receipt, event, publisher, registry, and report hash were verified. Duplicate reconciliation returns already-published and blocks a second transaction. The fee is not a mainnet estimate.

Why Arc

Arc provides a relevant environment for payment, treasury, private-credit, and USDC-oriented financial application workflows. VeilForge complements network capabilities by examining the application layer: what Solidity source makes observable, how that evidence travels, and whether the result matches policy. The focus is ecosystem-specific without claiming an official integration or endorsement.

Grant Purpose

Grant support would fund a measurable transition from a strong local/open candidate toward responsibly operated developer workflows. Planned work includes a secure hosted foundation, account and metering boundaries, private-CI pilots, cost measurement, privacy and retention controls, managed Developer workflows, Team workspace and tenant-isolation foundations, detector and corpus expansion, Arc onboarding and ecosystem integrations, and independent security validation.

The proposed allocation is 35% product engineering, 15% hosted CI infrastructure, 15% security validation, 10% documentation and onboarding, 15% Arc ecosystem integrations, and 10% developer support and operations, totaling 100%. Acceptance evidence would include threat-model and tenant-boundary tests, private-repository workflow evidence, retention controls, measured job costs, authorization tests, versioned benchmark expansion, integration documentation, and independent review findings. No fake delivery dates or guaranteed outcomes are claimed.

Commercial Sustainability

The local/open foundation is intended to remain free: scanner core, CLI, local web, basic SDK, public detectors, portable reports, verification, and documentation. Proposed paid value comes from managed private CI, hosted history, higher operational limits, multiple projects, team workspaces, shared policies, dashboards, scheduling, audit trails, support, private runners, and self-hosted delivery.

Developer pricing of \$19–39 per month, Team pricing of \$99–249 per month, and custom annual Enterprise pricing are hypotheses. Paid plans and production billing are not live. Month-12 planning scenarios are conservative \$588 MRR, base \$3,997 MRR, and upside \$14,705 MRR. These are not customers, revenue, bookings, first-year income, or guarantees.

Limitations

VeilForge's benchmark is bounded; exact solc 0.8.24 is the supported compiler baseline; browser projects are limited to 1 MiB; and unsupported constructs, unresolved boundaries, or budgets can produce incomplete analysis. Results outside the maintained corpus can include false positives or false negatives. Browser and host security remain outside the scanner's control. VeilForge does not replace an audit, provide formal verification, or guarantee confidentiality.

Registry V2 is immutable, public, non-payable, publisher-scoped, and has latest-record overwrite semantics. It has no owner, upgrade, or pause mechanism. Independent review, operational ownership, and fee validation remain required before mainnet use. Arc mainnet values and deployment are unresolved; `enabled=false`, `proofReadEnabled=false`, and `publishEnabled=false` remain the canonical mainnet state.

Next Milestones

1. Hosted Foundation: secure account, metering, private-CI pilot, cost measurement, privacy, and retention evidence.
2. Developer Workflow: private repository checks, hosted history, managed policy gates, and structured discovery or pilots without fabricated customer claims.
3. Team Foundation: workspaces, roles, tenant isolation, shared policies, and audit trails with cross-tenant denial evidence.
4. Arc Expansion: detector and corpus growth, onboarding, ecosystem integration, and independent review.

VeilForge merits consideration because the request builds on a working compiler-backed system, reproducible evidence, shipped developer tooling, a local privacy boundary, and a real Arc Testnet publication—not only a roadmap. The grant would unlock a defined next stage while the project continues to state its technical, operational, browser, mainnet, and commercial boundaries clearly.

Full whitepaper: `docs/whitepaper/veilforge-v4-whitepaper.md` Final evidence package:
`docs/grant/final/Canonical manifest: docs/grant/final/grant-evidence-manifest.json`