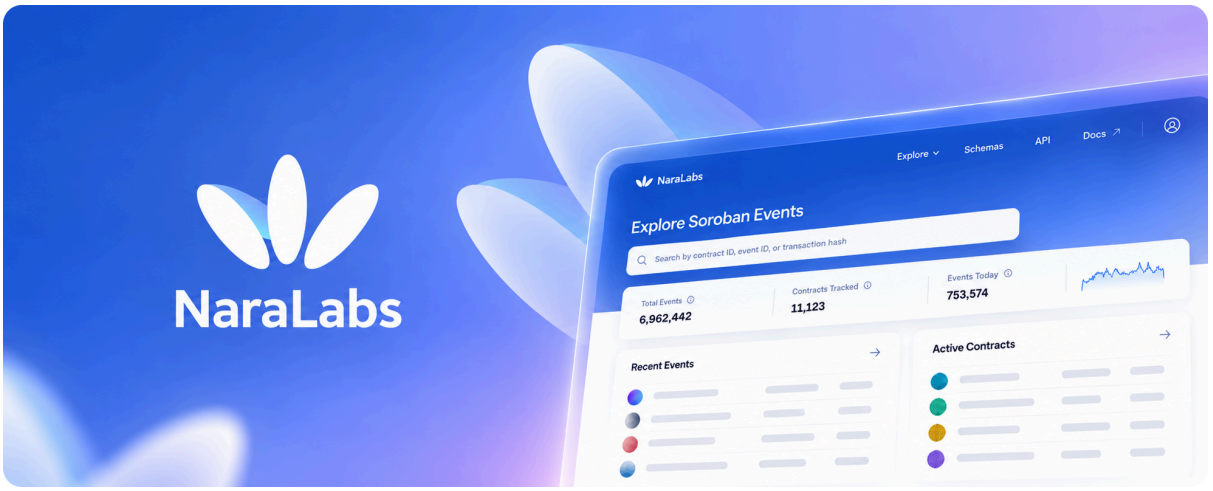
NaraLabs - Instaward Completion Report



Project & Team Information

Project Name:	NaraLabs
Builder / Team Name:	NaraLabs Foundation
Primary Contact:	Audri Mona Najogi (monanajogi02@gmail.com)
Ambassador Chapter:	Indonesia
Ambassador Chapter Lead:	Kenny Rivaldi
Date Completed:	01 October 2026

Summary

Soroban contracts emit large volumes of opaque contract events: payloads arrive as XDR and generic tagged JSON, while most contracts ship without on-chain event metadata. Explorers and RPC expose raw bytes; there is no shared, author-controlled layer that turns those payloads into stable field names, summaries, and queryable decoded vs raw status at scale.

The gap is semantic infrastructure: durable indexing, a registry where authors publish SEP-0048 event schemas, an API that decodes RPC-style payloads on demand, and a product surface where developers and reviewers can browse and try decoding without ad hoc scripts.

NaraLabs (https://naralabs.io) closes that gap on Stellar. Contract authors publish naralabs.schema.yaml to the registry; Atlas indexes activity and resolves schemas per contractId; POST /v1/decode returns level-3 semantic JSON for integrators. NaraLabs lists events and contracts with honest Decoded / Raw labels, hosts the schema registry, and

offers a Decode Playground where users pick a registry-backed contract, paste JSON or XDR, and run the same decode path in the browser.

Deliverables

Deliverable 1: SEP-0048 Schema Registry Core

Deliverable: Core registry backend where Soroban event schemas can be published, versioned, listed, and retrieved by contract ID, network, event name, and schema version (aligned with SEP-0048 event specifications).
Evidence type: Source code repository + API documentation
Status: **Completed**
Tracking issue: https://github.com/naralabsdev/naralabs-atlas · commit https://github.com/naralabsdev/naralabs-atlas/commit/39d56cb · production Atlas deployed (testnet registry live)

NaraLabs Atlas implements Instawards Deliverable 1 as the SEP-0048 Schema Registry Core: a Postgres-backed registry (event_schemas) exposed over REST (/v1/schemas), with publish-token auth for authors, contract bundle listing for explorers, and versioned retrieval per contract, network, and event name. Schema bodies follow SEP-0048-aligned JSON (name, prefix_topics, params with topic/data locations), validated on publish.

The live testnet catalog is built from real indexed Soroban activity (production ClickHouse events): contracts were chosen from the highest event volume on testnet, and each registered event name matches the dominant on-chain topic symbol (for example transfer, settled, create_order). Schema bodies are community-sourced (mapped from observed Soroban topic symbols by the NaraLabs operator, not official deployer-published SEP-0048 via wallet verify). Rows are status: published, version: 1. On production Postgres, operator bulk promote (https://github.com/naralabsdev/naralabs-atlas/blob/main/scripts/promote-community-schemas-to-verified.sql) sets trust_tier to verified with a placeholder verified_wallet for catalog-wide decode and benchmarks; this is not POST /v1/schemas/verify. Live GET /v1/schemas/summary?network=testnet reports publishedContracts: 10, eventSchemas: 19, verifiedContracts: 10, communityContracts: 0, exceeding the SOW minimum of 5 contracts and 10 event types. The public explorer at https://naralabs.io/schemas reads the same registry.

Repository: https://github.com/naralabsdev/naralabs-atlas

Evidence provided:
Registry implementation (publish, list, retrieve, versioning). The registry module is the Deliverable 1 code path in Atlas: HTTP routes register publish, paginated list, get-by-contract, get-by-event (optional version), version history, registry summary, and contract bundle reads used by the product UI. Publish auto-increments version per contract, network, and event name for a publisher; retrieve returns the latest published body or a

specific version when requested. The same surface is documented in OpenAPI on a running Atlas server.

Registry source:

- Routes: https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/registry/routes/routes.go
- HTTP Handlers:
- HTTP handlers: internal/module/registry/handler/schema_handler.go - https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/registry/handler/schema_handler.go
- Registry service: https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/registry/service/schema_service.go
- Validation: https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/registry/service/validate.go
- Bundle ingest: https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/registry/service/schema_bundle_service.go
- Postgres migration: https://github.com/naralabsdev/naralabs-atlas/blob/main/db/migrations/postgres/004_event_schemas.up.sql

SEP-0048-aligned schema format documentation. The SOW asks for schema format documentation, not only code. The repository includes a dedicated guide explaining args vs params, prefix_topics, publish/list/retrieve examples, and alignment with SEP-0048 event specifications. Product-facing registry browsing is documented on the public docs site. Interactive request and response shapes for registry endpoints are available from the deployed Atlas OpenAPI and Scalar UI.

Schema format documentation:

- SEP-0048 format doc: https://github.com/naralabsdev/naralabs-atlas/blob/main/docs/sep-0048-naralabs-schema-format.md
- Explorer docs: https://naralabs.io/docs/explorer/schemas

Production testnet catalog:

- Manifest: https://github.com/naralabsdev/naralabs-atlas/blob/main/testdata/registry/samples/manifest.json
- Sample schemas: https://github.com/naralabsdev/naralabs-atlas/tree/main/testdata/registry/samples/schemas
- Seed script: https://github.com/naralabsdev/naralabs-atlas/blob/main/scripts/seed-production-registry.sql

- Example schema (transfer): https://github.com/naralabsdev/naralabs-atlas/blob/main/testdata/registry/samples/schemas/transfer__CDLZFC3S.json
- Registry UI: https://naralabs.io/schemas
- API summary: https://naralabs.io/api/atlas/v1/schemas/summary?network=testnet
- API retrieve (transfer example): https://naralabs.io/api/atlas/v1/schemas/CDLZFC3SYJYDZT7K67VZ75HPJVIEUVNIXF47ZG2FB2RMQQVU2HHGCYSC/transfer?network=testnet
- Operator bulk promote (trust_tier only, not wallet verify): https://github.com/naralabsdev/naralabs-atlas/blob/main/scripts/promote-community-schemas-to-verified.sql
-
- Catalog provenance vs trust_tier. Community-sourced: schemas map observed on-chain topic symbols; they are not deployer wallet-verified official protocol SEP-0048. trust_tier=verified in production reflects operator bulk promote for decode benchmarks and explorer badges, with placeholder verified_wallet per script comments—not POST /v1/schemas/verify. Wallet-backed verification remains a separate registry feature for contract deployers.

Automated tests (publish, list, retrieve). Unit tests cover validation and service behavior. Deliverable 1 acceptance loads the same manifest as the production export and walks publish, list-by-contract, get-event, and list-versions in the service layer. CI runs the full Go test suite on push. Reproducing locally does not require the production database: clone the repository and run the acceptance test named below, or make test.

Automated tests and demos:

- SOW acceptance test: https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/registry/service/schema_registry_sow_test.go
- Service unit tests: https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/registry/service/schema_service_test.go
- Validation unit tests: https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/registry/service/validate_test.go
- CI workflow: https://github.com/naralabsdev/naralabs-atlas/blob/main/.github/workflows/go-test.yml
- Demo script: https://github.com/naralabsdev/naralabs-atlas/blob/main/scripts/registry-demo-operations.sh

Publish path for integrators (same registry). Authors publish with POST /v1/schemas and Bearer nl_live publish tokens, or the public CLI package at https://www.npmjs.com/package/@naralabs/cli

---

Deliverable 2: Event Decoder API

Deliverable: Build a decoder service that accepts raw Soroban event topics and values, matches them against registered schemas, and returns readable JSON. Supports single and batch decode, ScVal/XDR structures, and raw fallback when no schema matches.
Evidence type: Source code repository + test fixtures
Status: **Completed**
Tracking issue: https://github.com/naralabsdev/naralabs-atlas · commit https://github.com/naralabsdev/naralabs-atlas/commit/75d2487 · k6 benchmarks https://github.com/naralabsdev/naralabs-perf · commit https://github.com/naralabsdev/naralabs-perf/commit/ca3e30a · production Atlas (testnet decode live at https://atlas.naralabs.io)

NaraLabs Atlas implements Instawards Deliverable 2 as the Event Decoder API in the same repository as the registry (Deliverable 1). The decoder loads SEP-0048 schemas from the registry (Postgres), matches prefix_topics and params, and returns level-3 semantic JSON when a schema matches. When no schema matches or prefix validation fails, the API returns honest level-2 tagged raw JSON (decodeStatus: raw) rather than inventing field names.

SOW completion targets for the demo environment are verified with k6 against the live Atlas API (https://atlas.naralabs.io). Measured 2026-10-01: p95 http_req_duration ~73 ms (939 requests, 5 VUs, 30 s); ≥90% fixture success as 100% checks on 9/9 production-published HTTP fixtures. Committed k6 summary JSON artifacts are linked under Automated tests and demos (naralabs-perf evidence/instawards-sow-d2-demo-env-2026-10-01/). Ten distinct decoded event types are covered across the fixture set and Atlas manifest.

Repository: https://github.com/naralabsdev/naralabs-atlas

Evidence provided:
Decoder implementation (HTTP API + decode engine). The decoder module resolves schemas by contractId and network, supports optional eventName hints, batch decode, and playground-facing decode for the product (Deliverable 3). Core matching and field extraction live in lib/decoder; HTTP wiring in internal/module/decoder.

Decoder source:

- Routes: https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/decoder/routes/routes.go
- HTTP Handlers: https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/decoder/handler/decode_handler.go
- Decode service: https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/decoder/service/decode_service.go
- Decode engine: https://github.com/naralabsdev/naralabs-atlas/blob/main/lib/decoder/decode.go

HTTP fixtures and k6 benchmarks (primary SOW evidence). Prepared POST /v1/decode bodies live in naralabs-perf (exported from naralabs-atlas testdata/decoder). k6 scripts exercise production testnet decode with API key auth and assert SOW targets on http_req_duration and per-fixture decodeStatus checks.

Fixtures and k6 benchmarks:

- Repository: https://github.com/naralabsdev/naralabs-perf
- Fixture index (HTTP bodies): https://github.com/naralabsdev/naralabs-perf/tree/main/fixtures
- p95 load test: https://github.com/naralabsdev/naralabs-perf/blob/main/k6/decode-single.js
- Fixture rotation test: https://github.com/naralabsdev/naralabs-perf/blob/main/k6/decode-fixtures.js
- Run scripts: https://github.com/naralabsdev/naralabs-perf/tree/main/scripts
- Source manifest (Atlas): https://github.com/naralabsdev/naralabs-atlas/blob/main/testdata/decoder/manifest.json

API documentation and live decode surface. Integrators call decode with Bearer API keys. Public docs on naralabs.io describe the decode API; k6 benchmarks use the direct Atlas host below.

API documentation:

- Web decode API doc: https://naralabs.io/docs/api/decode-api
- OpenAPI (Atlas): https://naralabs.io/docs/api (Scalar; decode-event and decode-events-batch operations)
- Live decode endpoint (API key / k6): https://atlas.naralabs.io/v1/decode
- Proxied decode (web BFF): https://naralabs.io/api/atlas/v1/decode (Authorization: Bearer nl_api_…)

Automated tests and demos:

- naralabs-perf (k6, primary SOW evidence): https://github.com/naralabsdev/naralabs-perf
- Written metrics (report sections 1–2): https://github.com/naralabsdev/naralabs-atlas/blob/main/docs/deliverable3-benchmark-report.md
- Archived k6 JSON (demo environment, 2026-10-01):
- Evidence index: https://github.com/naralabsdev/naralabs-perf/blob/main/evidence/instawards-sow-d2-demo-env-2026-10-01/README.md
- p95 JSON artifact: https://github.com/naralabsdev/naralabs-perf/blob/main/evidence/instawards-sow-d2-demo-env-2026-10-01/k6-decode-p95.json
- Fixtures JSON artifact: https://github.com/naralabsdev/naralabs-perf/blob/main/evidence/instawards-sow-d2-demo-env-2026-10-01/k6-decode-fixtures.json

- Manual curl demo: https://github.com/naralabsdev/naralabs-atlas/blob/main/scripts/decoder-demo-operations.sh
- Atlas CI engine test: https://github.com/naralabsdev/naralabs-atlas/blob/main/lib/decoder/decoder_sow_test.go
- CI workflow: https://github.com/naralabsdev/naralabs-atlas/blob/main/.github/workflows/go-test.yml

---

Deliverable 3: Explorer Dashboard & Developer Documentation

Deliverable: Lightweight dashboard on Stellar testnet to search contracts and events, view registered SEP-0048 schemas, test raw event decoding in the browser, and read integration documentation for the registry and decode API.
Evidence type: Screenshots, Documentation, Demo Video
Status: **Completed**
Tracking issue: https://github.com/naralabsdev/naralabs-web · commit https://github.com/naralabsdev/naralabs-web/commit/2699653 · benchmark report https://github.com/naralabsdev/naralabs-atlas/commit/2ec4412

NaraLabs is the public explorer and developer surface for Soroban on testnet: indexed events and contracts, the schema registry UI, and a Decode Playground that runs Atlas decoding without an end-user API key. Nextra docs at /docs cover API credentials, the decode API, explorer guides, and playground usage for integrators.

The demo video walks a reviewer through the workflow on the live product: browse activity on naralabs.io, inspect registered schemas, run a decode in the playground, and open API documentation for at least three core endpoint groups (decode, explore, registry). Decode success rate and response time benchmarks are summarized in the linked Atlas report.

- **Public URL (live)**: https://naralabs.io
- **Decode playground:** https://naralabs.io/playground
- **Developer documentation:** https://naralabs.io/docs
- **Explorer repository:** https://github.com/naralabsdev/naralabs-web
- **Demo video:** https://youtu.be/rl66t7cHr14

Figure 1: screenshot of Contract Search: Explorer Dashboard

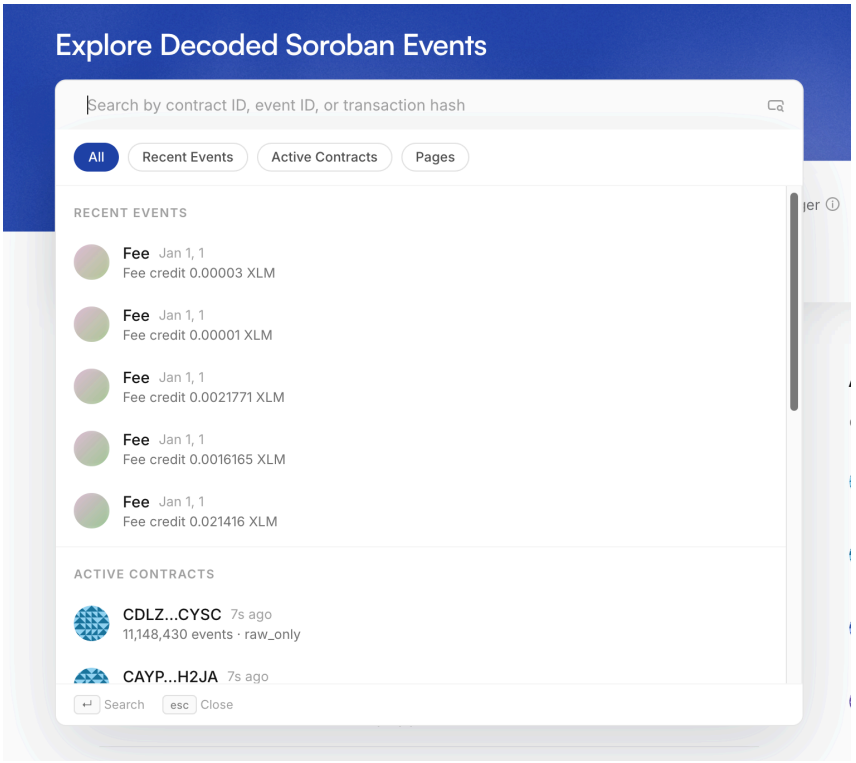


Figure 2: screenshot of Registered Schema List

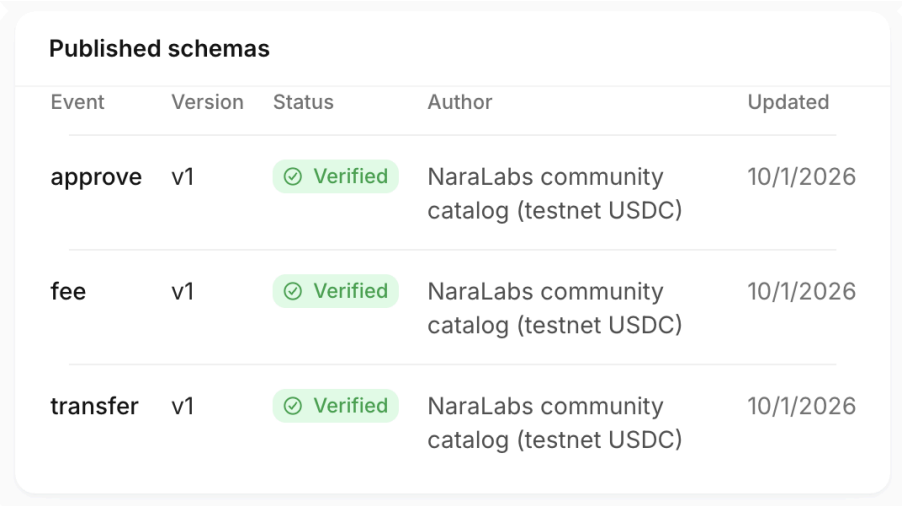


Figure 3: screenshot of Decode Playground

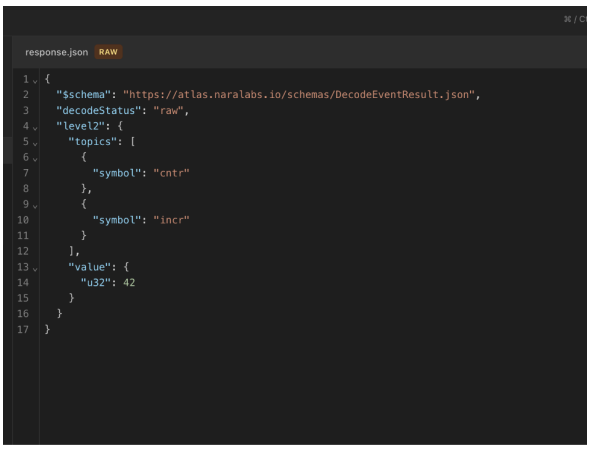


Figure 4: screenshot of Developer Documentation

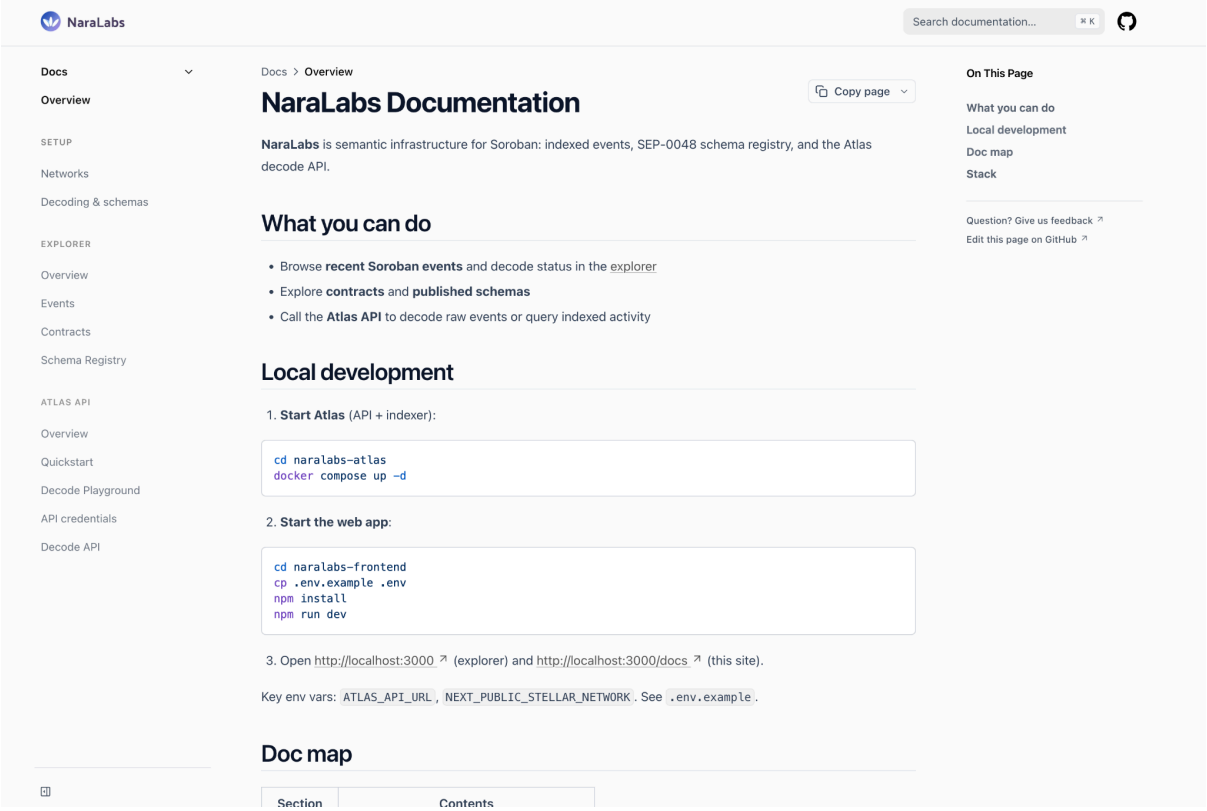
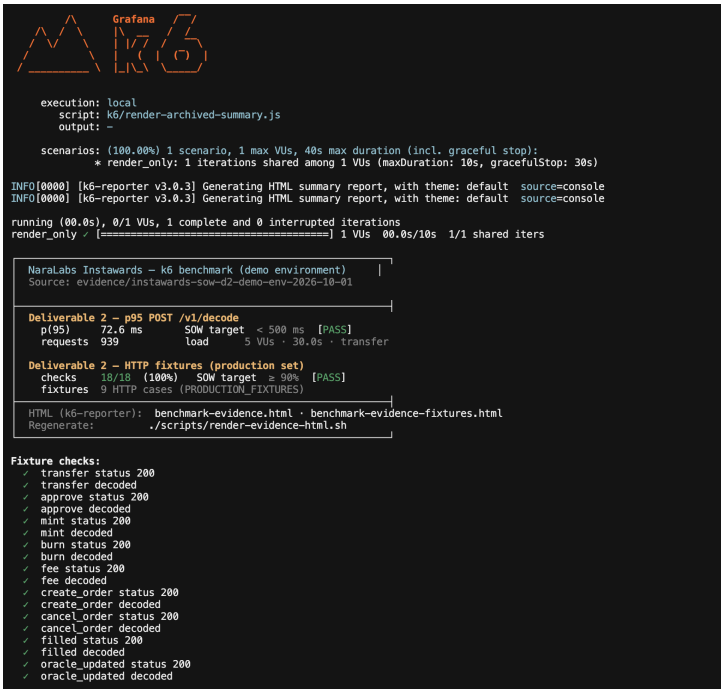


Figure 4: screenshot of Benchmark Report: Decode Success Rate and Response Time



Supporting Files

Name	Link
Web app explorer, registry, playground	https://naralabs.io
Events explorer	https://naralabs.io/events
Contracts explorer	https://naralabs.io/contracts
Schema registry UI (SEP-0048)	https://naralabs.io/schemas
Decode Playground	https://naralabs.io/playground
Developer documentation (Nextra)	https://naralabs.io/docs
API overview (decode, explore, registry)	https://naralabs.io/docs/api/overview
Decode API reference	https://naralabs.io/docs/api/decode-api
Playground integration doc	https://naralabs.io/docs/api/playground
Demo video (Deliverable 3 walkthrough)	https://youtu.be/rI66t7cHr14
Atlas API base (production)	https://atlas.naralabs.io
POST /v1/decode (API key)	https://atlas.naralabs.io/v1/decode
Proxied Atlas health (BFF)	https://naralabs.io/api/atlas/health
Registry summary testnet (proxied)	https://naralabs.io/api/atlas/v1/schemas/su

	mmary?network=testnet
Schema retrieve example USDC transfer	https://naralabs.io/api/atlas/v1/schemas/CDLZFC3SYJYDZT7K67VZ75HPJVIEUVNIXF47ZG2FB2RMQQVU2HHGCYSC/transfer?network=testnet
OpenAPI / Scalar (via docs)	https://naralabs.io/docs/api
Explorer schemas doc	https://naralabs.io/docs/explorer/schemas
GitHub organization	https://github.com/naralabsdev
Repository naralabs-web	https://github.com/naralabsdev/naralabs-web
Repository naralabs-atlas (indexer, registry, decoder)	https://github.com/naralabsdev/naralabs-atlas
Repository naralabs-perf (k6 benchmarks)	https://github.com/naralabsdev/naralabs-perf
Repository naralabs-cli	https://github.com/naralabsdev/naralabs-cli
npm package @naralabs/cli	https://www.npmjs.com/package/@naralabs/cli
Benchmark report (markdown)	https://github.com/naralabsdev/naralabs-atlas/blob/main/docs/deliverable3-benchmark-report.md
SEP-0048 NaraLabs schema format doc	https://github.com/naralabsdev/naralabs-atlas/blob/main/docs/sep-0048-naralabs-schema-format.md
Stellar SEP-0048 specification	https://github.com/stellar/stellar-protocol/blob/master/ecosystem/sep-0048.md
k6 evidence index (2026-10-01 demo env)	https://github.com/naralabsdev/naralabs-perf/blob/main/evidence/instawards-sow-d2-demo-env-2026-10-01/README.md
Archived k6 p95 JSON	https://github.com/naralabsdev/naralabs-perf/blob/main/evidence/instawards-sow-d2-demo-env-2026-10-01/k6-decode-p95.json
Archived k6 fixtures JSON	https://github.com/naralabsdev/naralabs-perf/blob/main/evidence/instawards-sow-d2-demo-env-2026-10-01/k6-decode-fixtures.json
k6-reporter HTML p95 (repo root)	https://github.com/naralabsdev/naralabs-perf/blob/main/benchmark-evidence.html
k6-reporter HTML fixtures (repo root)	https://github.com/naralabsdev/naralabs-perf/blob/main/benchmark-evidence-fixtures.html

k6 decode p95 script	https://github.com/naralabsdev/naralabs-perf/blob/main/k6/decode-single.js
k6 decode fixtures script	https://github.com/naralabsdev/naralabs-perf/blob/main/k6/decode-fixtures.js
HTTP decode fixtures directory	https://github.com/naralabsdev/naralabs-perf/tree/main/fixtures
Registry module source	https://github.com/naralabsdev/naralabs-atlas/tree/main/internal/module/registry
Registry routes	https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/registry/routes/routes.go
Production registry sample manifest	https://github.com/naralabsdev/naralabs-atlas/blob/main/testdata/registry/samples/manifest.json
TestDeliverable1 registry acceptance test	https://github.com/naralabsdev/naralabs-atlas/blob/main/internal/module/registry/service/schema_registry_sow_test.go
Registry demo script	https://github.com/naralabsdev/naralabs-atlas/blob/main/scripts/registry-demo-operations.sh
Decoder module source	https://github.com/naralabsdev/naralabs-atlas/tree/main/internal/module/decoder
Decoder engine lib/decoder	https://github.com/naralabsdev/naralabs-atlas/blob/main/lib/decoder/decode.go
Decoder fixture manifest	https://github.com/naralabsdev/naralabs-atlas/blob/main/testdata/decoder/manifest.json
TestDeliverable2 decoder acceptance test	https://github.com/naralabsdev/naralabs-atlas/blob/main/lib/decoder/decoder_sow_test.go
Decoder demo script (curl)	https://github.com/naralabsdev/naralabs-atlas/blob/main/scripts/decoder-demo-operations.sh
Atlas CI workflow (Go tests)	https://github.com/naralabsdev/naralabs-atlas/blob/main/.github/workflows/go-test.yml
Atlas tracking commit Deliverable 1 (example)	https://github.com/naralabsdev/naralabs-atlas/commit/39d56cb
Atlas tracking commit Deliverable 2 (example)	https://github.com/naralabsdev/naralabs-atlas/commit/75d2487

naralabs-perf tracking commit (example)	https://github.com/naralabsdev/naralabs-perf/commit/ca3e30a
naralabs-web tracking commit (example)	https://github.com/naralabsdev/naralabs-web/commit/2699653
Playground source (naralabs-web)	https://github.com/naralabsdev/naralabs-web/tree/main/src/app/(playground)/playground
Soroban testnet RPC (network default)	https://soroban-testnet.stellar.org