

Proof Verifies Where It Is Held

Complete artifact. Local recomputation.
Zero network calls.



THE LOCAL-OFFLINE VERIFICATION RELEASE

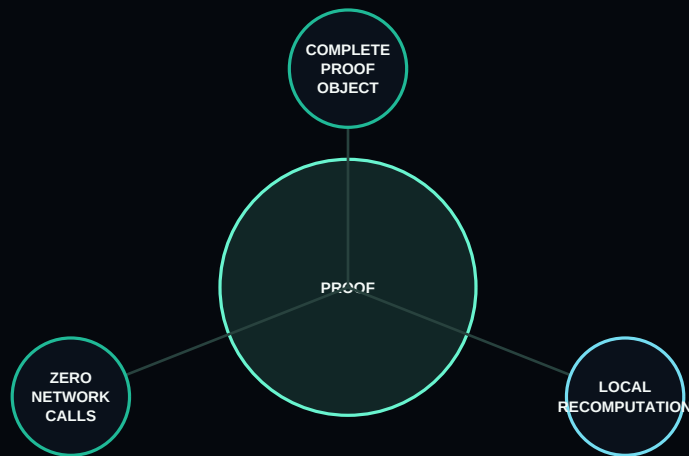
RELEASE DATE / 2026-07-18

FIRST ADMISSION ONLY / THEN APPEND FOREVER

RELEASE DECLARATION

Proof verifies where it is held.

Receiz v109.0.0 is the coordinated local-offline developer release. A complete sealed proof object carries the material required to recompute its truth locally, without asking a server, session, database, registry, or running Receiz.com process for permission.



- Complete proof object

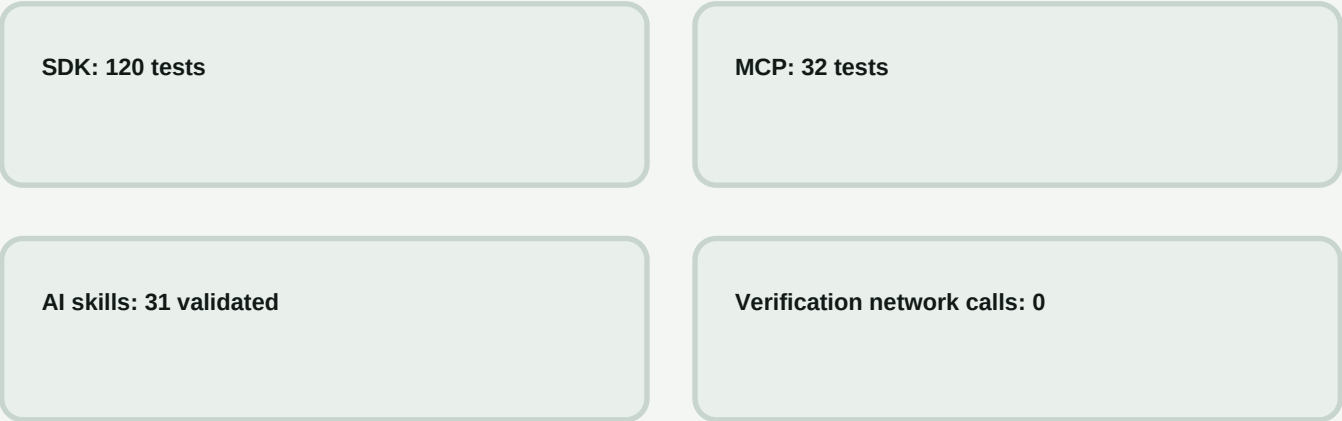
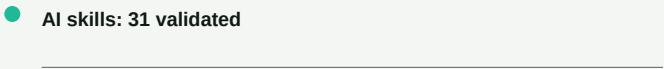
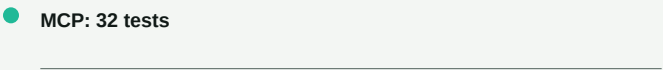
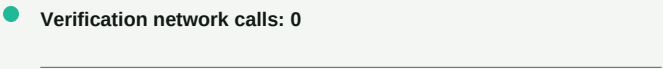
- Local recomputation

- Zero network calls

THE RELEASE

What changed at v109

V109 packages complete-artifact verification and opening into the public SDK, composes MCP and AI skills beneath that verifier, removes obsolete developer entry points, and seals zero-network verification as constitutional law.



AUTHORITY

The artifact remains the authority.

The proof object - not the SDK, MCP response, server, session, database, registry lookup, or release label - carries the verification boundary. Developer tools recompute and report that truth; they do not create it.

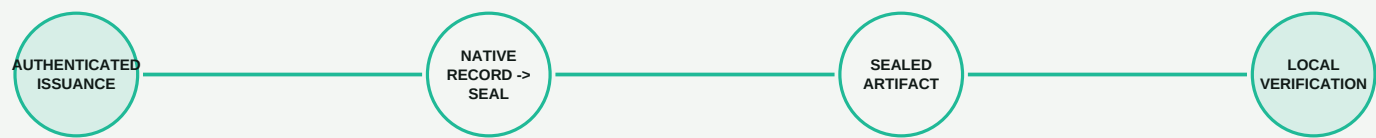


- | | |
|---------------------|-----------------------|
| ● Receiz law | ● Sealed proof object |
| _____ | _____ |
| ● Local verifier | ● SDK projection |
| _____ | _____ |
| ● MCP orchestration | ● AI explanation |
| _____ | _____ |

BOUNDARY

Issuance and verification are different operations.

Native issuance remains authenticated Record -> Seal and may use Receiz.com. Artifact-time verification is local recomputation from sealed bytes. V109 changes verification and opening, not issuance authority or mutation permissions.



● Authenticated issuance

● Sealed artifact

● Native Record -> Seal

● Local verification

VERIFY ARTIFACT

verifyArtifact() reads the complete object.

`receiz.verification.verifyArtifact(file)` accepts the complete `.receizbundle` carrier and verifies its carried proof locally. A method name is not enough: the returned verdict must arise from recomputation over the bytes, not an API response.



● Complete carrier bytes

● Local verification

● Structured verdict

● No API permission

VERIFY AND OPEN

Opening never outruns verification.

receiz.artifacts.verifyAndOpen(file) verifies the enclosing artifact before extraction. It returns the complete sealed artifact separately from the verified payload so custody and payload domains remain explicit.



● Complete artifact

● Extract verified payload

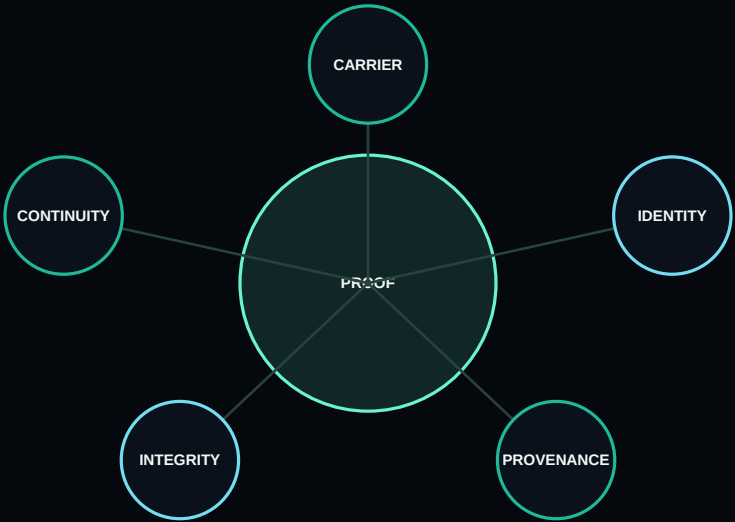
● Verify enclosure

● Separate custody domains

COMPLETE ARTIFACT

The carrier is indivisible.

The local verifier begins with the complete native carrier. Payload bytes, manifests, metadata, SDK handles, and database rows cannot substitute for the object that binds identity, provenance, integrity, and continuity together.



● Carrier

● Provenance

● Continuity

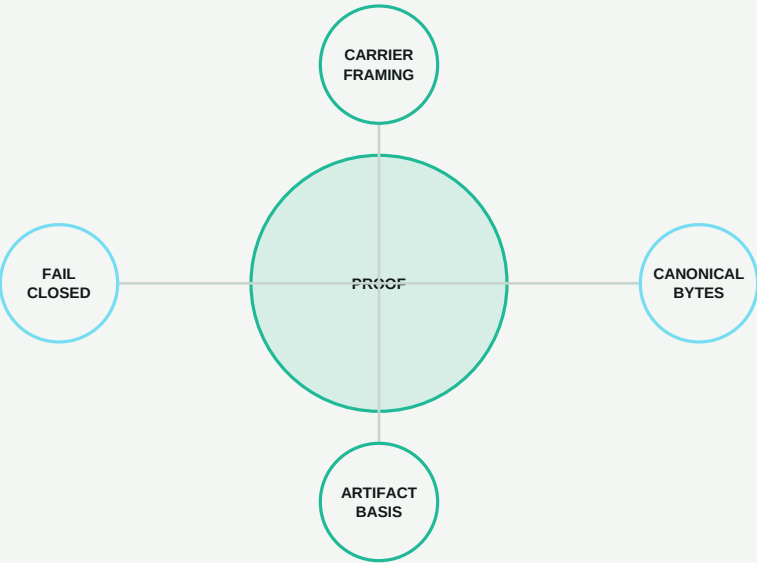
● Identity

● Integrity

CARRIER INTEGRITY

Malformed structure fails closed.

Carrier framing, canonical bytes, artifact binding, and document structure are verified before any payload projection is admitted. A malformed carrier or mismatched artifact basis produces rejection, not partial success.



● Carrier framing

● Artifact basis

● Canonical bytes

● Fail closed

PAYLOAD BINDING

Payload bytes remain a projection.

The payload digest must agree with the enclosing proof object. Verified payload bytes become available only after carrier integrity and proof binding pass. Extraction never turns the subpayload into an independent authority.



● Enclosing artifact

● Verified extraction

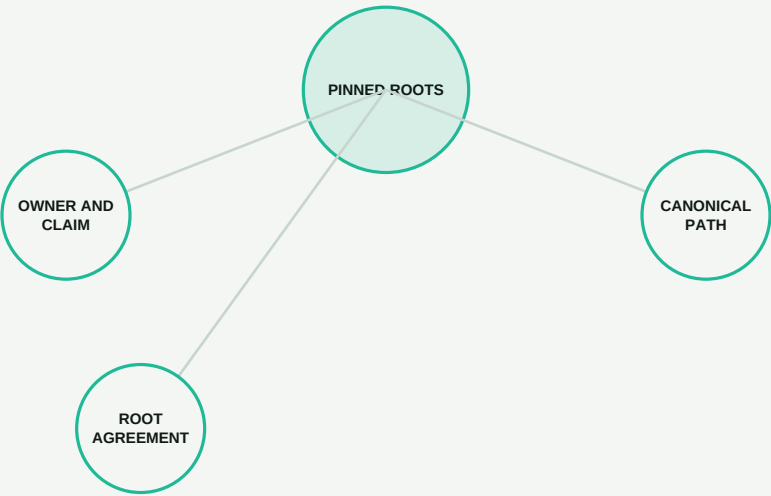
● Payload digest

● Domain parser

SIGNATURE V4

The certificate root verifies locally.

Pinned Signature V4 roots ship with the SDK. Certificate, owner, claim, path, and root agreement are recomputed from packaged verification material; server confirmation is not part of the artifact verdict.

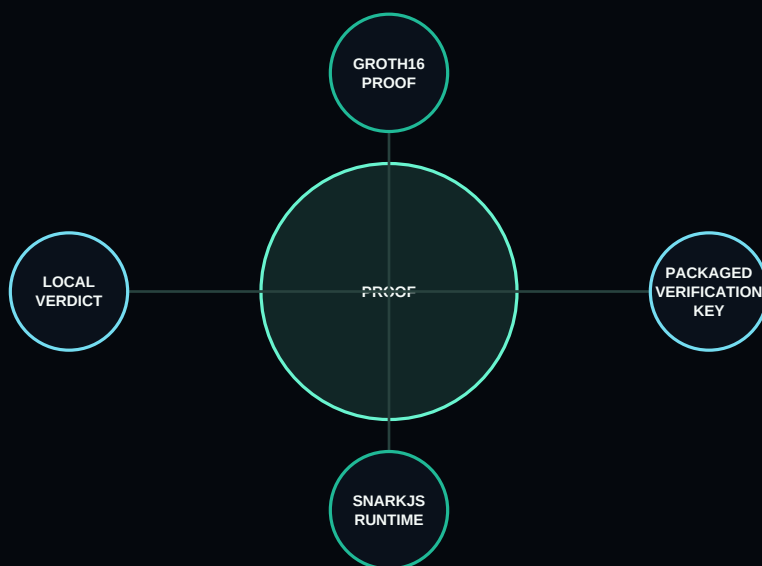


- Pinned roots
- Owner and claim
- Canonical path
- Root agreement

GROTH16

The document seal verifies locally.

The document-seal Groth16 verification key and snarkjs runtime ship with the SDK. The proof object can therefore recompute its zero-knowledge seal without downloading verifier state at decision time.



● Groth16 proof

● Packaged verification key

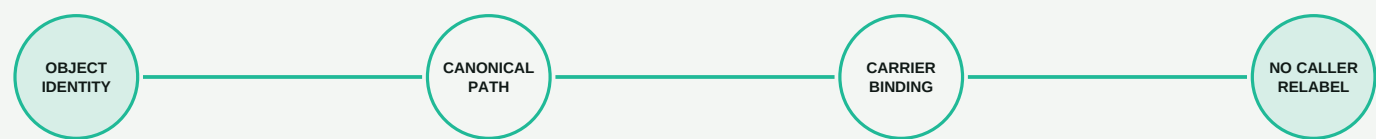
● snarkjs runtime

● Local verdict

CANONICAL IDENTITY

Identity and path must agree.

Successful verification requires canonical identity and path fields to agree with the sealed carrier. Transport names, caller hints, or API routes cannot relabel the object into a different proof identity.



● Object identity

● Carrier binding

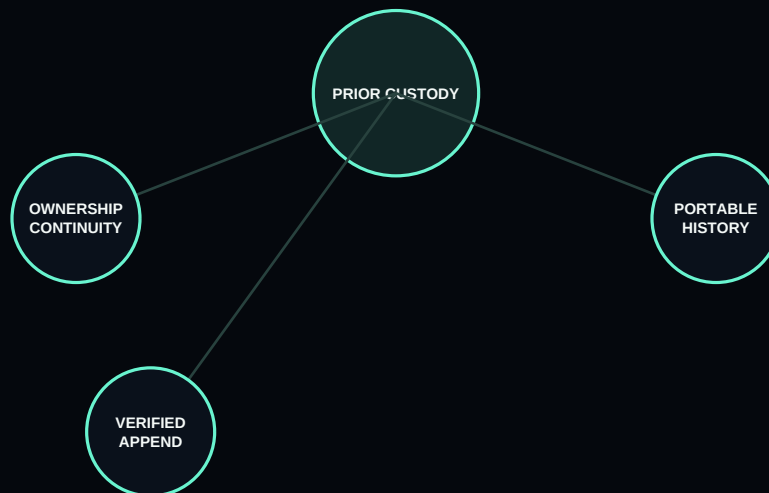
● Canonical path

● No caller relabel

PORTABLE CONTINUITY

Carried ownership remains verifiable.

Where portable ownership continuity applies, the local verifier validates that carried history as part of the complete proof object. Invalid continuity rejects the object instead of being softened into missing enrichment.



● Prior custody

● Ownership continuity

● Portable history

● Verified append

MUTATION

One changed byte changes the verdict.

A one-byte mutation of a production-signed complete artifact fails local verification with zero fetch calls. Mutation cannot fall through to a server lookup, a compatibility success, or a weaker payload-only result.



PRODUCTION-SIGNED ARTIFACT



ONE-BYTE MUTATION



LOCAL REJECTION



0 FETCH CALLS



Production-signed artifact



Local rejection



One-byte mutation

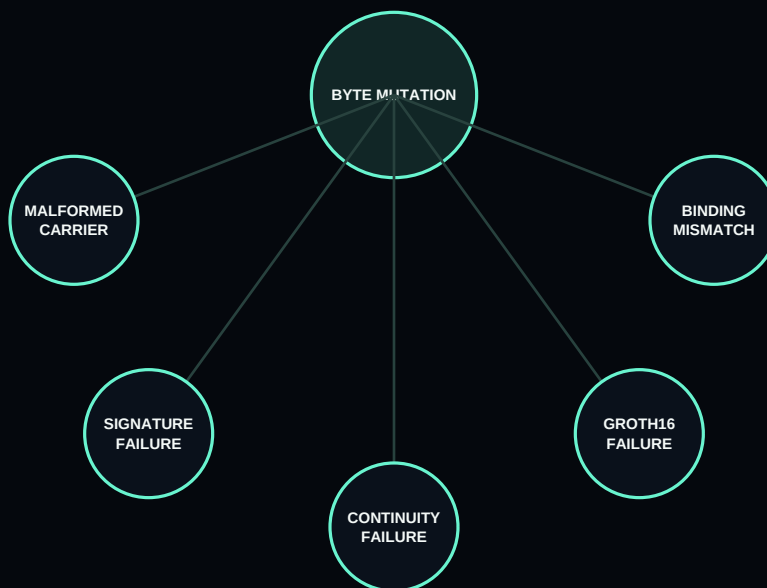


0 fetch calls

FAIL-CLOSED MATRIX

Every broken boundary rejects.

V109 contracts reject one-byte mutation, malformed carrier structure, artifact-binding mismatch, Signature V4 failure, Groth16 failure, and invalid portable continuity. No failure mode is converted into a partial verified object.



● Byte mutation

● Malformed carrier

● Binding mismatch

● Signature failure

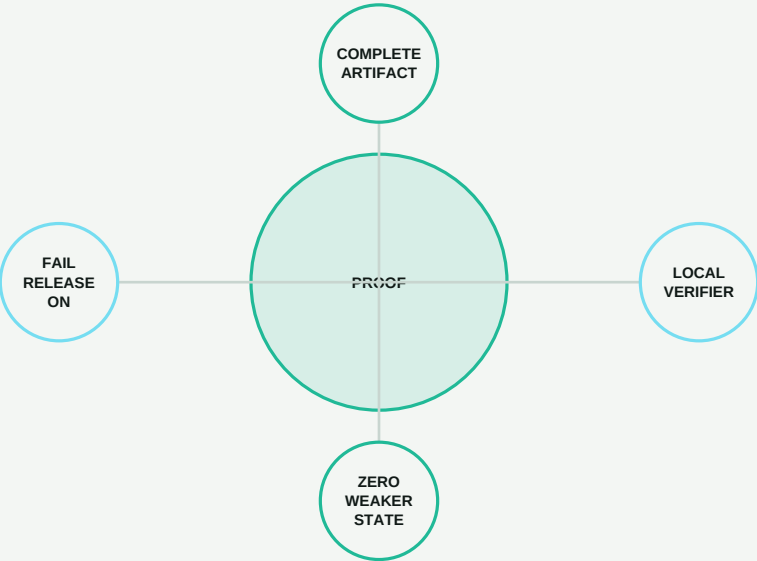
● Groth16 failure

● Continuity failure

ARTIFACT-011

Zero-network verification is release law.

Complete SDK artifact verification and extraction must remain available without network, session, database, or server state. The denial code is LOCAL_ARTIFACT_VERIFICATION_MUST_NOT_REQUIRE_WEAKER_STATE.



- Complete artifact
- Local verifier
- Zero weaker state
- Fail release on dependency

PACKAGED SUBSTRATE

Every verification dependency travels.

The SDK package includes Signature V4 roots, Groth16 verification material, local verifier code, and the snarkjs runtime. Offline proof is real only when the installed package contains the full decision substrate.



● Signature roots

● Verifier runtime

● Verification key

● Artifact schemas

ZERO NETWORK

No fetch path exists in the verifier.

Public verification and opening perform zero network calls and require no access token, cookie, database connection, registry lookup, or running Receiz.com process. Network absence is normal operation, not fallback mode.



NO ACCESS TOKEN



NO COOKIE



NO DATABASE



NO REGISTRY LOOKUP



NO RECEIZ.COM PROCESS

● No access token

● No database

● No Receiz.com process

● No cookie

● No registry lookup

MCP

Orchestration reuses the same verifier.

MCP artifact tools call the public SDK verifier and preserve complete artifact custody. MCP may plan, invoke, and explain; it cannot mint a stronger verdict, extract before verification, or become proof authority.



● SDK verifier

● MCP orchestration

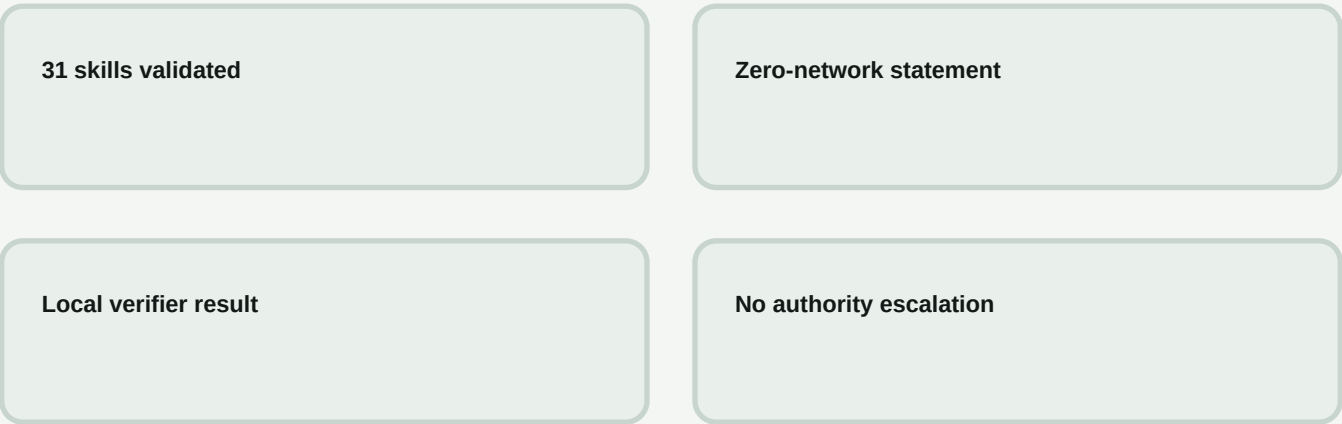
● Same local verdict

● Artifact remains authority

AI SKILLS

Claims require recorded local evidence.

Artifact-related AI skills require the exact statement Network calls during verification: 0 and a recorded Local verifier result before a production-ready claim. Generated explanation remains evidence beneath the verified object.



- 31 skills validated
- Local verifier result

- Zero-network statement
- No authority escalation

PACKAGE HYGIENE

Clean before build.

Deleted version-specific source can survive in stale compiler output unless dist is cleaned. V109 makes clean-before-build part of package correctness so obsolete implementations cannot remain packable after their source is removed.



● Remove stale dist

● Inspect tarball

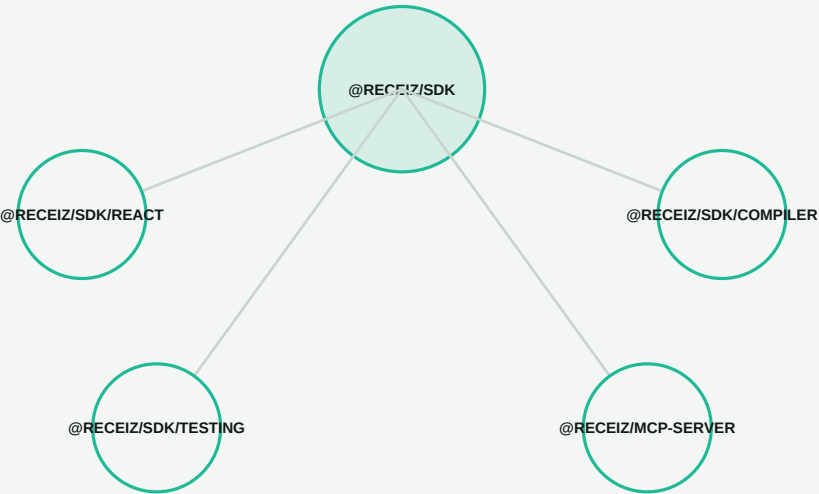
● Compile current source

● Reject obsolete outputs

CURRENT IMPORTS

Public imports remain unversioned.

The maintained imports stay @receiz/sdk, @receiz/sdk/react, @receiz/sdk/compiler, @receiz/sdk/testing, and @receiz/mcp-server. Existing v108 method names and artifact/payload separation remain source-compatible.



- @receiz/sdk
- @receiz/sdk/compiler
- @receiz/mcp-server

- @receiz/sdk/react
- @receiz/sdk/testing

HISTORICAL BOUNDARY

Old proof survives without old SDK authority.

Historical sealed artifacts, receipts, registry records, and release evidence remain immutable historical truth. Obsolete v107 SDK/MCP implementations, exports, generators, and migration commands are not shipped or installable.



● Historical proof preserved

● Historical receipts preserved

● Obsolete exports removed

● No executable old authority

RECEIZ.COM

Product behavior remains v108 exact.

Receiz.com product behavior and database paths are unchanged. Visible release labels, service-worker version, runtime-coherence version, and offline-verifier identification move to v109 without changing application behavior.

Behavior unchanged**Visible version v109****Service worker v109****Runtime coherence v109**

● Behavior unchanged

● Visible version v109

● Service worker v109

● Runtime coherence v109

DATABASE

No schema or production data changed.

V109 changes developer verification capability only. No database schema, migration, query path, production row, or application mutation was introduced. The isolated profile fixture remains forbidden from using production data.



NO MIGRATION



NO QUERY CHANGE



NO PRODUCTION MUTATION



ISOLATED FIXTURE ONLY

● No migration

● No production mutation

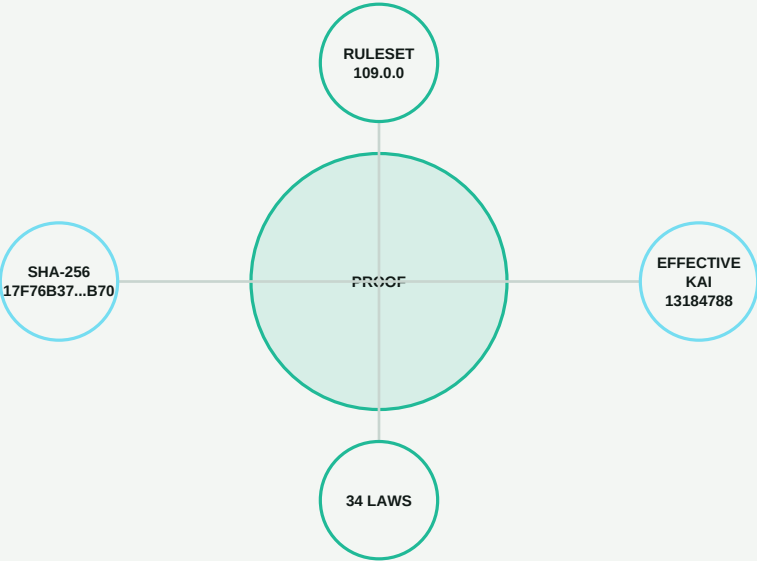
● No query change

● Isolated fixture only

CONSTITUTIONAL SEAL

ARTIFACT-011 extends the register.

The v109 registry contains 34 laws and preserves the v108 digest as its previous witnessed root. It was sealed at 2026-07-18T07:33:09.542Z with effective Kai 13184788 and digest 17f76b37c9fcd46f710239b5c1660b03cc34ec64bed30d1cc45c18d5d40eab70.



● Ruleset 109.0.0

● Effective Kai 13184788

● 34 laws

● SHA-256 17f76b37...b70

RELEASE EVIDENCE

Executable evidence records the boundary.

The release record states that a production-signed artifact verifies locally and a one-byte mutation rejects locally, both with zero fetch calls. SDK, MCP, AI skills, typechecks, tarballs, external installation, release freeze, and governance gates qualify the package.



● SDK 120 tests

● MCP 32 tests

● 31 skills

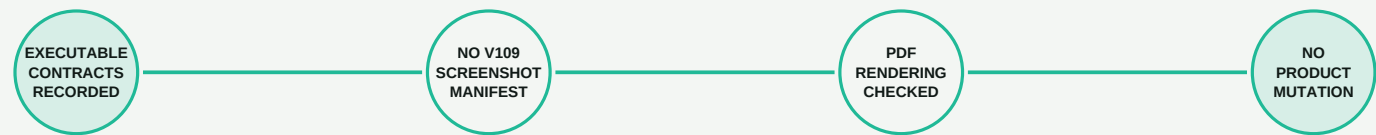
● Zero-fetch mutation pair

● Tarball and external install

EVIDENCE BOUNDARY

No screenshot is invented.

The v109 release record contains executable verifier, package, registry, and governance evidence but no dedicated product-surface visual-evidence manifest. This book records that exact boundary; its own rendered pages verify document layout, not application behavior.



● Executable contracts recorded

● PDF rendering checked

● No v109 screenshot manifest

● No product mutation claimed

EXTERNAL BOUNDARY

Publication and signing stay separate.

Repository qualification does not publish npm packages, deploy Receiz.com, mutate a database, or sign the release attestation. npm publication and release signing require separate authorization and their own evidence.



ATTESTATION SIGNING PENDING



NPM PUBLICATION PENDING



NO DEPLOYMENT IMPLIED



PRODUCTION DB MUTATION FORBIDDEN



Attestation signing pending



npm publication pending



No deployment implied



Production DB mutation forbidden

FROZEN REGRESSIONS

What v109 refuses to forget.

No public verifier may depend on an API response. No stale dist may preserve deleted developer code. No extraction may precede enclosure verification. No historical SDK is required to read historical proof. No version bump may rename unchanged product payload schemas.

✗ NO SERVER VERIFICATION DEPENDENCY

✗ NO STALE PACKABLE OUTPUTS

✗ NO EXTRACT-BEFORE-VERIFY

✗ NO HISTORICAL API AUTHORITY

✗ NO SCHEMA RENAME THEATER

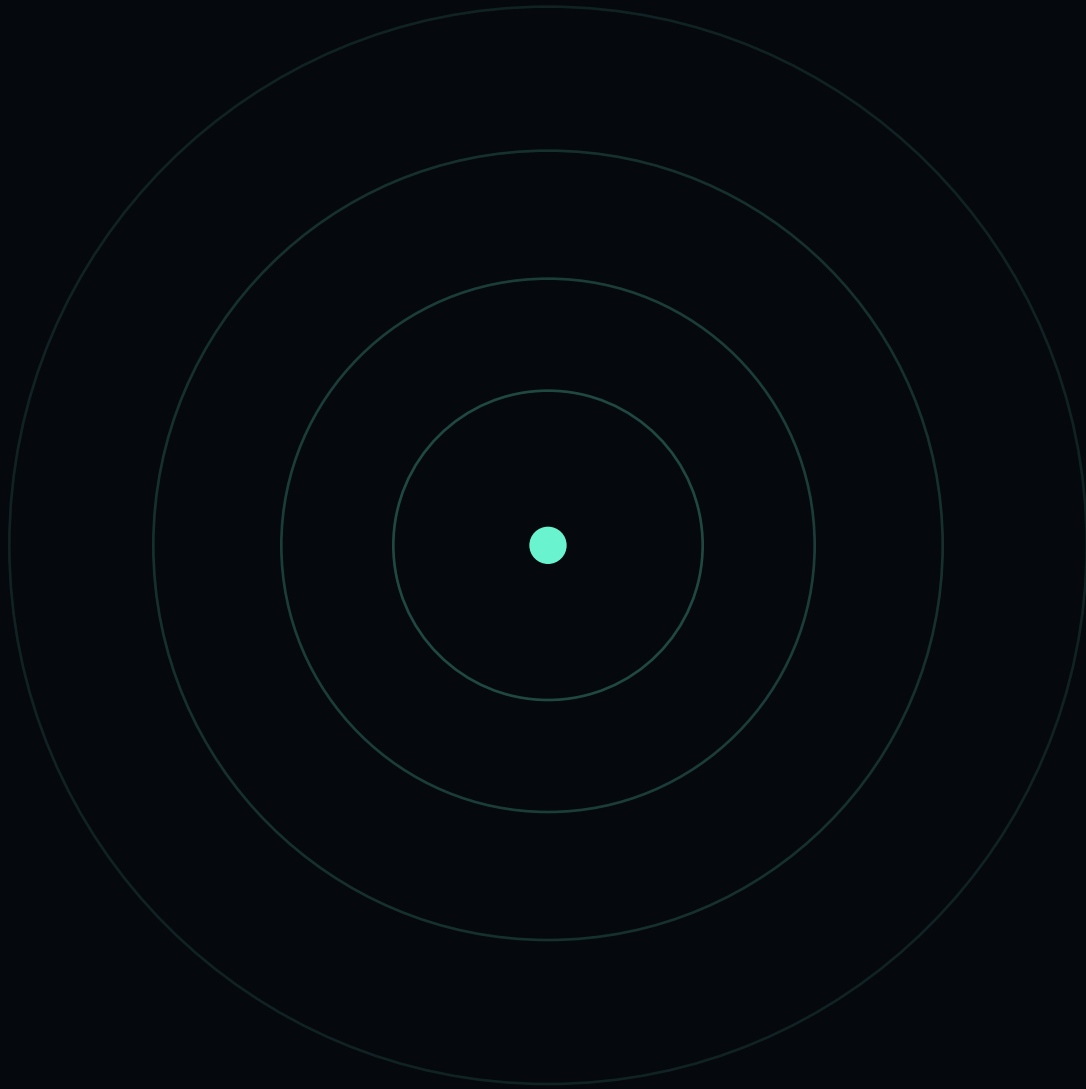
● No server verification dependency

● No stale packable outputs

● No extract-before-verify

● No historical API authority

● No schema rename theater



The object carries the verdict.

Verification recomputes locally. Weaker systems may transport, never authorize.

RECEIZ / v109.0.0 / LOCAL-OFFLINE VERIFICATION

COMPLETE ARTIFACT / SIGNATURE V4 / GROTH16 / ZERO NETWORK