LARRI
VERIFICATION PROTOCOL

Ledger of Attested Records and Receipts for Integrity

Volume I of V

LARRI Core Protocol Specification

Version 0.1.0

Purpose. LARRI - Ledger of Attested Records and Receipts for Integrity - is a domain-agnostic integrity and attribution protocol for application-defined results. This volume defines the minimum public assurance and verification semantics required for independently verifiable receipts and logical bundles without prescribing the private process, software architecture, or proprietary methods used to produce an application assertion.

Property	Value
Document type	Normative core protocol specification
Publication set	LARRI 0.1.0, Volume I of V
Protocol	LARRI - Ledger of Attested Records and Receipts for Integrity
Version	0.1.0
Status	Experimental
Use boundary	Synthetic laboratory and supervised evaluation only
Publication date	27 July 2026
Official website	https://larriverification.org

Assurance boundary. LARRI verifies defined integrity, attribution, binding, scope, and deterministic-verification properties under explicitly supplied trust inputs. It does not establish factual truth, human or organizational identity, authority, completeness, independently trusted time, clinical correctness, legal sufficiency, regulatory compliance, production readiness, universal absence of sensitive information, or correctness of the private decision process. This pre-pilot publication is intended only for synthetic laboratory and supervised evaluation.

Normative specification

Contents

Document relationship. This volume defines core assurance meaning, logical roles and trust boundaries, verification targets, the normative dependency order, result meaning, and the source-authority hierarchy. Volume II owns exact public data shapes, lexical forms, canonical bytes, cryptographic messages, trust objects, bundle contracts, resource limits, refusal representation, and normalized-result bytes. Volume III owns optional profile and extension semantics. Volume IV owns conformance execution, evidence, reports, claims, and publication gates. Volume V is informative and creates no conformance requirement.

1. Status, scope, and normative language

LARRI 0.1.0 defines a narrow, domain-agnostic protocol for issuing and independently verifying authenticated receipts and closed logical bundles. A receipt carries an application-defined assertion and cryptographic commitments to material that the Issuer elects to bind. Verification can complete without access to the Issuer/Core implementation's private decision process, source code, private evidence, generator state, or continued cooperation.

LARRI standardizes only the observable public boundary required for interoperable verification. Every property visible to a Verifier is publicly defined by the normative publication set. The protocol does not prescribe an internal software architecture, programming language, cryptographic library, deployment model, product workflow, model, policy engine, prompt, threshold, or decision algorithm.

1.1 Normative terms and requirement identifiers

The terms MUST, MUST NOT, SHOULD, SHOULD NOT, and MAY indicate requirement levels. A conforming implementation satisfies every applicable MUST and MUST NOT in Volumes I through IV and in the authoritative machine-readable contracts for the exact target, profiles, extensions, and configuration it claims.

LARRI-010-CORE-SCOPE-001 A conformance or verification claim MUST identify its verification target and selected profiles and MUST NOT present a narrower target result as satisfying a broader target.

LARRI-010-CORE-SCOPE-002 A feature not defined by the selected contracts and profiles MUST be treated as unsupported and MUST NOT silently influence PASS or REFUSE.

LARRI-010-CORE-SCOPE-003 A Verifier MUST fail closed. PASS is emitted only when every check required by the selected target and profiles succeeds.

LARRI-010-CORE-SCOPE-004 Normative requirement identifiers MUST be unique across the LARRI 0.1.0 publication set. A duplicate requirement definition is a publication defect and prevents a conformance claim.

1.2 Goals

- Exact, portable binding of application assertions and identified artifact bytes.
- Authenticated receipts whose signed context cannot be altered without detection.
- External trust that cannot be established or replaced by the artifact being verified.
- Signed, closed-scope logical bundles with deterministic verification semantics.
- Implementation-independent verification outcomes and bounded assurance statements.
- Offline verification for every base verification target.
- Black-box conformance without disclosure of proprietary decision logic.

1.3 Non-goals

- Disclosure or evaluation of private decision mathematics, models, model weights, thresholds, policies, prompts, rules, evidence, internal state, or orchestration.
- Factual truth, accuracy, quality, clinical correctness, or legal sufficiency of asserted content.
- Human or organizational identity proofing, authority, delegation, consent, or permission to act.
- Completeness of any source universe or proof that omitted records do not exist.
- Independently trusted time, ordering, inclusion, non-equivocation, or long-term renewal unless a selected profile defines and verifies a narrower property.
- Proof that an artifact or scope contains no sensitive information.
- Regulatory compliance, production approval, or authorization of a substantive decision.
- A mandatory external organization, online service, integration, pilot participant, or independent implementation.
- A general-purpose transport, workflow, record system, policy language, or application-specific semantic standard.

2. Terminology, roles, and deployment boundaries

2.1 Core Protocol and implementation roles

Role or system	Protocol function	Boundary
Core Protocol	Public assurance principles, trust boundaries, target meanings, dependency order, result meaning, and source-authority hierarchy defined by Volume I.	Does not prescribe private decision production or product architecture.
Issuer/Core implementation	Application-side component that may evaluate private evidence, policies, rules, models, thresholds, prompts, or proprietary decision mathematics and produce an assertion and commitments.	The protocol constrains emitted artifacts, not how the decision is reached.
Issuer	Selects the public assertion, commitments, limitations, signer policy, and packaging context and forms a receipt or logical bundle.	Its private process is outside base verification.
Attestor	Controls a private signing key and produces a protocol attestation.	May be combined with the Issuer/Core implementation, but key possession is not assumed.
Operator	Selects the verification target and supplies the artifact or subject, semantic contracts, external trust, profiles, limits, and configuration.	Operator choices are public protocol inputs.
Verifier	Performs only public deterministic protocol behavior.	Requires no private Core logic, private evidence, generator state, or network access.
Conformance Harness	Controls public test inputs and compares a Verifier with authoritative contracts, vectors, and frozen expected results.	Evidence is implementation-scoped, not an artifact-verification result.
Relying Process	Interprets a bounded result in a separate authorized decision process.	LARRI does not authorize its substantive decision.

2.2 Deployment composition

Roles describe logical responsibilities, not mandatory products, services, organizations, or processes. One deployment may combine several roles, or isolate them across separate components, provided artifact-independent trust, signing-key control, protocol-visible determinism, and relying-decision authorization remain distinct.

LARRI-010-CORE-ROLE-001 Operational combination of LARRI roles MUST NOT permit artifact-contained material to establish its own verification trust, make private decision behavior part of Verifier semantics, or make a protocol result directly authorize a substantive decision.

2.3 Trust and proprietary boundaries

LARRI-010-CORE-TRUST-001 The verified artifact MUST NOT establish, replace, or override the external trust inputs used to verify it.

LARRI-010-CORE-TRUST-002 All artifact bytes MUST be treated as untrusted until the applicable input-safety, identity, scope, commitment, and signature checks succeed.

LARRI-010-CORE-TRUST-003 Core verification targets MUST be capable of completing without network access.

LARRI-010-CORE-TRUST-004 Receipt validity MUST NOT depend on disclosure of proprietary decision logic, source code, private evidence, software architecture, or internal state.

3. Assurance boundary and non-goals

3.1 Minimum observable assurances

Assurance	Established property	Not established
Integrity	Signed or committed bytes have not changed relative to verified identities and digests.	Truth, quality, or completeness of the underlying material.
Attribution	A required attestation verifies under public key material supplied through external trust.	Human identity, organizational identity, authority, licensure, or legal role.
Binding	The assertion, commitments, limitations, signer declarations, and policy context are bound as the artifact contracts define.	Correctness of the private decision process.
Scope	A bundle PASS applies to the exact closed logical bundle; a profile PASS applies to the explicit profile scope and evidence.	Unlisted records, universal source completeness, or unselected assurances.
Determinism	Identical public protocol inputs under one semantic contract set produce identical normalized result bytes.	Identical internal architecture or implementation source.

3.2 Meaning of PASS

PASS means: For the identified semantic contracts, supplied artifact or target subject, selected verification target, selected profiles, external trust, limits, and configuration, every check required by that operation passed and the corresponding bounded LARRI assurance was established.

PASS does not establish factual or clinical correctness, completeness, human or organizational identity, authority, independently trusted time, legal validity, regulatory compliance, universal absence of sensitive information, or correctness of the private decision process.

3.3 Meaning of REFUSE

REFUSE means: The Verifier did not establish the selected target-scoped LARRI assurance. REFUSE does not prove fraud, falsity, negligence, malicious conduct, or substantive invalidity. It is protocol data for controlled investigation, retransmission, or exception handling.

3.4 Limitations and completeness

A signed receipt and a normalized result may carry registered limitations that qualify the assertion or verification claim. The absence of a record from a receipt or bundle is not evidence that the record does not exist elsewhere.

LARRI-010-CORE-LIMIT-001 A receipt or normalized result that makes no completeness claim MUST represent that limitation using the registered completeness limitation.

4. Artifact concepts and binding model

4.1 Public artifact concepts

Concept	Core meaning
Receipt	A byte-addressable protocol object containing closed signed content and the outer attestations required by its signed policy.
Application assertion	An application-defined result in a closed public wrapper. It may expose a public value or a digest commitment to a private result. Base LARRI treats its substantive meaning as opaque.
Commitment	A typed relationship between an identifier and a digest of exact bytes or registered canonical bytes. It establishes correspondence, not correctness, completeness, legality, or availability.
Attestation	A registered signature object that authenticates the exact signed message under a selected suite and policy.
Logical bundle	The prepared-directory verification subject governed by manifest.larri.json and manifest.larri.sig.json, together with every in-scope file. Base LARRI defines no mandatory archive or canonical directory serialization.
External trust input	Public verification material, pins, fingerprints, and target permissions supplied independently of the artifact.
Semantic contracts	Authoritative schemas, registries, requirement inventory, profile and extension records, and non-release-specific gate definitions that determine public protocol behavior.
Normalized result	A closed deterministic projection of target context, identities when computable, checks, refusals, limitations, and outcome.

4.2 Application assertion

LARRI-010-CORE-ASSERT-001 The signed receipt content MUST bind the assertion namespace, assertion type, and the exact public value or value commitment presented by the Issuer.

LARRI-010-CORE-ASSERT-002 Base protocol verification MUST NOT require disclosure or re-execution of the decision process that produced the assertion.

4.3 Commitments

LARRI-010-CORE-COMMIT-001 A Verifier MUST recompute a commitment when the required bytes are supplied and the selected target requires recomputation.

LARRI-010-CORE-COMMIT-002 When committed bytes are not supplied and are not required by the selected target, the Verifier MUST distinguish the signed commitment from an independently recomputed digest.

4.4 Content-derived identity and canonical binding

Receipt identity is non-recursive. It is derived from the canonical signed-content object with the receipt identity value omitted and before outer attestations are attached. Exact fields, lexical forms, and byte sequences are owned by Volume II.

LARRI-010-CORE-ID-001 Receipt identity computation MUST exclude the receipt identity value itself and MUST NOT depend on a final bundle manifest that inventories the receipt.

LARRI-010-CORE-ID-002 The Verifier MUST recompute the receipt identity from the parsed signed content and compare it using the exact lexical form defined by the contracts.

LARRI-010-CORE-ID-003 Array order is significant unless a field is expressly defined as an order-independent set commitment.

LARRI-010-CORE-CANON-001 A canonical protocol object MUST produce exactly the bytes defined by its registered canonicalization identifier.

LARRI-010-CORE-CANON-002 Unknown or disabled canonicalization identifiers MUST produce a registered unsupported-feature refusal and MUST NOT be interpreted heuristically.

LARRI-010-CORE-BIND-001 Signer and signature-policy metadata outside the signed content MUST exactly match the signed values where the artifact format duplicates them.

LARRI-010-CORE-BIND-002 A Verifier MUST reject a digest, signature, or identity calculation performed over a byte sequence other than the one selected by the registered protocol algorithms.

LARRI-010-CORE-TIME-001 A core PASS message MUST NOT represent an issuer-asserted time as independently trusted time.

4.5 Logical bundle integrity

The manifest declares one closed inventory using normalized paths, registered content roles, exact lengths, and digests. The manifest binds receipts and other files; receipts do not bind the later manifest that inventories them.

LARRI-010-CORE-MANIFEST-001 Every in-scope file MUST appear exactly once in the manifest, and every manifest entry MUST correspond to exactly one in-scope file.

LARRI-010-CORE-MANIFEST-002 Unexpected, missing, duplicate, role-invalid, size-mismatched, or digest-mismatched files MUST produce a registered refusal for the bundle or profiled-bundle target.

LARRI-010-CORE-MANIFEST-003 Before accepting the detached manifest signature, the Verifier MUST recompute and compare the manifest identity and final manifest digest, then verify that signature under the external trust selected for the bundle.

```
application material -> receipt
receipt and bundle files -> bundle manifest
bundle manifest -> detached manifest attestation
external trust -> verification
```

LARRI-010-CORE-GRAPH-001 Protocol artifact identities and bundle relationships MUST form an acyclic dependency graph.

LARRI-010-CORE-BYTES-001 Verification MUST compare the exact bytes declared by the manifest and MUST NOT repair, normalize, or rewrite received bundle content unless an explicitly selected canonicalization is part of the declared commitment.

LARRI-010-CORE-ARCHIVE-001 Unsafe paths, ambiguous duplicate names, prohibited link behavior, or exceeded resource limits MUST prevent target-scoped PASS without unsafe traversal, materialization, or uncontrolled allocation.

LARRI-010-CORE-PACKAGE-001 Base bundle PASS MUST NOT require or imply a canonical directory byte stream, byte-identical packaging, or equal carrier digests for independently constructed realizations of the same verified logical bundle.

LARRI-010-CORE-PACKAGE-002 A normalized result for a prepared-directory bundle MUST identify the declared logical bundle subject defined by the artifact contracts; file checks determine whether the observed directory realizes that declaration. A byte-stream bundle identity requires a selected registered carrier extension.

5. External trust and signature policy

5.1 Three kinds of independence

Independence	Meaning
Trust independence	Verification trust is supplied independently of the artifact. It may be fully internal to one organization and fully offline.
Verification independence	The Verifier reproduces all required public behavior without Issuer/Core source, private evidence, policy logic, generator state, or network services.
Organizational independence	An optional property of an Independent Interoperability claim. It is not required for ordinary artifact verification or Protocol Conformance.

5.2 External trust meaning

A valid pin or fingerprint establishes continuity with the supplied public key bytes and the target permissions encoded in the trust input. It does not establish who controls the corresponding private key or whether that controller is authorized.

LARRI-010-CORE-TRUST-010 The Verifier MUST refuse when required external trust material is absent, malformed, or inconsistent with its supplied fingerprint or pin.

LARRI-010-CORE-TRUST-011 A key label, issuer label, or organization annotation MUST NOT establish trust or affect PASS unless a selected profile expressly binds it to an independently trusted authority mechanism.

LARRI-010-CORE-TRUST-012 The normalized result MUST identify the fingerprints of the external trust material actually used.

5.3 Signature suites and policies

Baseline suite. The identifier larri/ed25519-sha256-v1 remains Ed25519 over the raw 32-byte SHA-256 digest of the final canonical signed content. The exact key, signature, message, and acceptance rules are owned by Volume II and MUST NOT be changed under this identifier.

Post-quantum, hybrid, multi-signature, or other alternative capability may be added only through new registered suite and policy identifiers with exact public semantics, deterministic downgrade behavior, and conformance vectors. LARRI defines no heuristic cryptographic negotiation or fallback.

LARRI-010-CORE-SIG-001 A Verifier MUST apply the exact registered signature-suite semantics and MUST refuse invalid, non-canonical, malformed, or unsupported required signatures.

LARRI-010-CORE-SIG-002 All verification paths that claim the same signature suite MUST produce equivalent acceptance behavior for the published adversarial vectors.

LARRI-010-CORE-POLICY-001 A Verifier MUST refuse when a required attestation is absent, invalid, or uses an unsupported required suite.

LARRI-010-CORE-POLICY-002 A recognized but non-required attestation MAY be reported as UNVERIFIED when it is outside the selected target and does not weaken the required policy result.

6. Verification targets and explicit selection

Target	Minimum subject	Bounded assurance
receipt	One byte-addressable receipt, one semantic contracts identity, selected external trust, limits, and configuration.	Receipt identity, signed assertion binding, required commitment treatment, signer binding, and required attestations.
bundle	One prepared-directory logical bundle governed by the fixed manifest control objects.	Manifest identity, detached manifest attestation, closed scope, path and content-role rules, exact file lengths and digests, and required contained receipts.
profiled-bundle	The complete bundle target plus one explicit enabled profile or enabled closed profile combination.	Bundle assurance plus every additional artifact, trust input, check, claim boundary, and limitation selected by the profile contract.

Receipt and bundle operations use an empty profile set. Profiles are never inferred, silently dropped, merged ad hoc, or used as fallback behavior.

LARRI-010-CORE-TARGET-001 A normalized result MUST identify exactly one enabled verification target.

LARRI-010-CORE-TARGET-002 A PASS for one target MUST NOT be represented as a PASS for a broader target.

7. Normative verification order

Verification is a deterministic evaluation of public protocol inputs. A conforming Verifier may use any internal architecture, concurrency model, execution language, or storage strategy, provided its observable behavior follows the normative dependencies below.

Order	Stage	Required effect
1	Input and subject safety	For the established enabled target, apply the selected limits and reject unsafe subject forms before unsafe processing.
2	Strict decoding and schema recognition	Apply the strict decoding policy and identify exactly one applicable top-level schema.
3	Contract and feature resolution	Validate the applicable schema and resolve protocol version, artifact type, canonicalization, signature policy, selected profiles, and required extensions.
4	External trust	Validate independently supplied trust material, target permissions, and fingerprints.
5	Manifest canonical form and identity	For bundle targets, validate the manifest form and recompute its non-recursive identity and final digest.
6	Detached manifest attestation	For bundle targets, verify the detached manifest signature only after the recomputed identity and digest match.
7	Closed bundle scope and file integrity	For bundle targets, verify paths, prohibited links, closed file set, content roles, lengths, and digests.
8	Receipt identities and signed bindings	Recompute contained receipt identities and validate signed and duplicated metadata relationships.
9	Receipt attestations	Verify signer binding, signature policies, and required receipt attestations.
10	Commitments and selected profiles	Recompute required commitments and evaluate selected extension or profile conditions.
11	Normalized result	Select the deterministic primary refusal when applicable and emit the canonical target-scoped result.

An inapplicable stage produces no refusal. A failed or unsafe prerequisite blocks dependent downstream checks. Primary-refusal precedence follows this order.

LARRI-010-CORE-VERIFY-001 A later check that is blocked by a failed, missing, or unsafe prerequisite MUST NOT be executed merely to collect additional failures.

LARRI-010-CORE-VERIFY-002 Verification of the core targets MUST complete without Issuer-private code, generator state, or network access.

LARRI-010-CORE-VERIFY-003 Artifact defects MUST be handled as protocol data and MUST NOT be represented as PASS because an internal error interrupted verification.

LARRI-010-CORE-RESOURCE-001 A resource-exhaustion input MUST produce the deterministic REFUSE or execution-class result required by its applicable public contract, without unsafe traversal, uncontrolled allocation, code execution, or partial PASS.

LARRI-010-CORE-DETERMINISM-001 Identical protocol inputs under the same semantic contracts, target, profiles, limits, configuration, and external trust MUST produce byte-identical normalized result bytes.

8. Invocation rejection and target-scoped verification

An invocation establishes the operation context before a target-scoped larri/verifier-result can exist. At minimum, the Operator selects a target and supplies the semantic contracts, artifact or subject, external trust, profiles, limits, and configuration through their respective public input channels.

Invocation boundary. An unsupported operator-selected target is not a target-scoped artifact defect because no enabled target contract has been selected. It is rejected before target-scoped verification, is never mapped to another target, and does not produce a fabricated larri/verifier-result. Volumes II and IV define the exact bounded invocation-refusal codes, public representation, and invocation_refuse conformance exit class.

Once an enabled target is established, malformed artifacts, missing or defective trust inputs, unsupported required features, and other target-applicable defects are handled under the target-scoped refusal contracts whenever a truthful normalized result can be constructed.

LARRI-010-CORE-INVOKE-001 An unsupported operator-selected target MUST be rejected before target-scoped verification, MUST NOT be silently mapped to an enabled target, and MUST be represented through the public invocation-refusal mechanism defined by Volumes II and IV.

9. Result, refusal, and evidence meaning

9.1 Check statuses

Status	Meaning
PASS	The check executed and its requirement was satisfied.
REFUSE	The check executed and produced a blocking target-scoped protocol refusal.
SKIP	The check was not executed because a prerequisite failed or execution would be unsafe.
NOT_PRESENT	An optional subject was not supplied and the selected target did not require it.
UNVERIFIED	A signed assertion, commitment, or optional attestation was present, but required bytes or evidence were unavailable under the selected target.

9.2 Primary and additional refusals

The refusal registry is authoritative for code, stage, rank, applicability, and safe public description. The primary refusal is the first blocking refusal in the normative verification order. Additional refusals may include only safely completed sibling checks and are ordered deterministically by the applicable registry.

LARRI-010-CORE-REFUSAL-001 Every refusal emitted by a conforming path MUST be active for the selected target or profile in the applicable refusal registry.

LARRI-010-CORE-REFUSAL-002 A Verifier MUST NOT execute blocked or unsafe downstream checks solely to produce more refusal codes.

9.3 Normalized result principles

A normalized result identifies the selected target and profiles, semantic contracts, artifact or logical subject identity when computable, external trust identity when computable, actual key fingerprints used, baseline limits, every selected non-baseline configuration, outcome, primary and safe additional refusals, ordered checks, and applicable limitations.

Receipt JSON remains byte-addressable. When every input byte was actually received, its received-byte digest may be available even if strict decoding later refuses. Prepared-directory bundle identity is the declared logical bundle identity derived from the manifest when that identity is computable. Exact nullability branches are owned by Volume II.

LARRI-010-CORE-OUTPUT-001 PASS requires a null primary refusal and an empty refusal list. REFUSE requires a registered primary refusal.

LARRI-010-CORE-OUTPUT-002 The normalized result MUST exclude absolute paths, hostnames, process identifiers, local timestamps, execution durations, diagnostic output, exception text, stack traces, temporary directories, and nondeterministic ordering.

LARRI-010-CORE-OUTPUT-003 Human diagnostics MUST be separated from the normalized result and MUST NOT alter its canonical bytes.

LARRI-010-CORE-OUTPUT-004 The normalized result MUST identify the selected target and profiles and MUST bind every selected non-baseline verification configuration as required by Volume II, whether or not its stricter thresholds change the outcome.

LARRI-010-CORE-OUTPUT-005 A normalized result MUST NOT contain a zero digest, empty-input digest, fixed placeholder, or other fabricated identity. An identity may be null only on an exact REFUSE branch where the applicable public contract determines that it cannot be computed truthfully.

9.4 Distinct result classes

Evidence class	Meaning	Boundary
Artifact verification result	Evidence about one selected receipt, bundle, or profiled-bundle operation.	Does not establish implementation conformance or authenticate a release package.
Conformance report	Evidence that one implementation reproduced required public behavior for one exact scope.	Does not verify a particular application assertion.
Release authentication	Evidence that a release envelope authenticates exact files and reports.	Does not turn failed conformance evidence into PASS and does not expand the underlying claim.

10. Profiles and extensions

Profiles add bounded public assurance conditions to the profiled-bundle target. Extensions add registered artifact types, algorithms, carriers, trust-input forms, evidence forms, signature suites or policies, or public checks. Neither mechanism can waive a failed core requirement, enable artifact self-trust, reinterpret target meanings, or depend on undocumented implementation behavior.

LARRI-010-CORE-PROFILE-001 A selected profile MUST be resolved through a closed public contract and MUST NOT depend on undocumented implementation behavior.

LARRI-010-CORE-PROFILE-002 Profiles MUST NOT rely on implicit inheritance, open-ended composition, unregistered constraint merging, or more than one enabled combination record for the same exact ordered profile set under one semantic contracts identity.

LARRI-010-CORE-PROFILE-003 An unknown or unselected extension artifact MAY remain subject to generic bundle byte-integrity checks, but it MUST NOT contribute semantic assurance, a profile claim, or execution of the extension unless its exact semantics are selected and registered.

LARRI-010-CORE-PROFILE-004 A profile requiring additional operator-controlled trust or policy input MUST NOT be enabled until a registered public contract defines the exact input schema, independent supply boundary, validation, target and profile binding, and protocol-visible identity or bound evidence that retains its use.

Application and domain specifications may define assertion vocabularies, public policy commitments, semantic validators, privacy-screening evidence, lineage, time evidence, or other assurance conditions. They remain separate from the domain-agnostic base protocol. A profile may require disclosure only when its selected public claim expressly depends on that public evidence.

LARRI-010-CORE-EXT-001 Base LARRI conformance MUST remain achievable by independently engineered implementations that share no proprietary decision engine, source code, internal data model, or software architecture.

Cryptographic agility. Post-quantum and hybrid policies remain available through new registered suite and policy identifiers with exact public verification semantics, deterministic downgrade behavior, and vectors. No alternative construction changes the baseline suite under its existing identifier.

11. Security, privacy, and relying-process safety

11.1 Security considerations

Threat	Core control	Residual limitation
Post-attestation alteration	Signed content, non-recursive identities, commitments, and closed manifests.	False or incomplete material supplied before attestation is not detected by integrity verification alone.
Artifact self-trust	Independently supplied trust and target permissions.	Base LARRI does not identify or authorize the key controller.
Parser and subject exploitation	Strict encodings, path rules, closed scope, and resource limits.	Implementation defects remain possible and require conformance testing.
Serialization drift	Registered canonicalization, schemas, registries, and frozen expected results.	Only registered encodings and algorithms are interoperable.
Signer metadata substitution	Signed signer binding and exact duplicated-field matching.	Identity and authority remain external.
Time overclaim	Issuer time is untrusted without selected profile evidence.	Base LARRI provides no independent ordering or deadline assurance.
Algorithm obsolescence	Explicit suite identifiers and suite-specific contracts.	Long-term renewal is outside base assurance.

11.2 Privacy considerations

LARRI can reduce disclosure by supporting opaque assertions and digest commitments. It does not provide encryption, access control, transport confidentiality, automatic minimization, anonymization, or proof that sensitive information is absent. A public digest may reveal information when its preimage has low entropy or belongs to a small guessable set.

LARRI-010-CORE-PRIVACY-001 Normalized results and public refusal descriptions MUST NOT include raw private evidence, sensitive values, host paths, private keys, secrets, or free-form exception text.

LARRI-010-CORE-PRIVACY-002 A privacy-related profile PASS MUST be limited to the identified checks, rules, result, and exact scope and MUST NOT be represented as proof of universal absence of sensitive information.

11.3 Relying-process safety

Decision boundary. Neither PASS nor REFUSE directly authorizes a substantive decision. A Relying Process remains responsible for its own authority, policy, exceptions, and review requirements outside LARRI.

LARRI-010-CORE-RELY-001 A Relying Process MUST NOT translate REFUSE directly into a substantive denial, adverse action, or loss of rights without a separate authorized decision process and appropriate exception handling.

LARRI-010-CORE-RELY-002 A user-facing result statement MUST identify the verified target and selected profiles and MUST NOT describe the assertion as verified true, legally valid, clinically correct, authorized, complete, anchored, compliant, free of sensitive information, or otherwise established beyond the selected assurance.

12. Conformance, authority, and release architecture

12.1 Five-volume authority hierarchy

Source	Authority
Volume I	Assurance meaning, roles and trust boundaries, verification targets, normative dependency order, result meaning, non-goals, safety principles, and source-authority hierarchy.
Volume II	Exact fields, schemas, lexical forms, canonical bytes, signature messages, trust objects, limits, bundle rules, refusal representation, and normalized-result bytes.
Volume III	Optional profile and extension semantics.
Volume IV	Conformance execution, vectors, reports, claims, evidence identity, and publication gates.
Volume V	Informative implementation and operational guidance only.
Schemas	Exact expressible structure, closed objects, field types, constants, bounds, and conditional requirements.
Registries	Identifier status, applicability, ordering, rank, parameters, and protocol meaning.
Normative prose	Assurance meaning and requirements not reducible to schema structure.
Vectors and frozen expected results	Concrete consequences of public rules; they cannot invent or override normative semantics.

LARRI-010-CORE-AUTHORITY-001 Public schemas, registries, algorithms, refusal rules, and normalized-result contracts MUST remain authoritative public protocol sources and MUST NOT be demoted to informative mirrors of reference implementation code or undocumented validator behavior.

Conflict rule. A conflict among normative sources is a publication defect. Implementations do not resolve it through aliases, document-order precedence, fallback parsing, or private tie-breakers.

12.2 Three one-direction identity layers

Layer	Identifier	Scope
Semantic contracts identity	contracts_sha256	Authoritative schemas, registries, normative requirements, profile and extension records, and gate definitions that do not contain release-specific evidence digests.
Conformance evidence identity	vector_set_sha256	Vector index, fixtures, frozen expected results, and their exact applicability to one contracts_sha256.
Release authentication identity	Outer release manifest	Exact semantic contracts, conformance evidence, reports, publication metadata, and release files.

```
semantic contracts
    -> conformance evidence
        -> conformance report
            -> release envelope
```

Semantic contracts do not contain vector-set or release-envelope digests that create a reverse dependency. Reference implementation output does not define expected results; expected results are derived from public rules and then frozen.

LARRI-010-CORE-RELEASE-001 Semantic contract, conformance evidence, report, and release-envelope identities MUST be non-circular and MUST follow the one-direction dependency architecture defined in this section.

12.3 Protocol Conformance and optional claims

Protocol Conformance is black-box agreement with the exact public behavior for one scope. Clean-Room Reproduction, Independent Interoperability, and Reproducible Release or other reproducible-build claims are optional additional claims. They do not follow from ordinary artifact PASS or ordinary Protocol Conformance.

LARRI-010-CORE-CONF-001 A conformance statement MUST identify the protocol version, semantic contracts digest, claimed verification targets, selected profiles, selected extensions, conformance evidence identity, platforms or environments claimed, and result.

LARRI-010-CORE-CONF-002 Basic Protocol Conformance MUST NOT require publication of proprietary source code, decision logic, internal evidence, implementation architecture, or product release procedures.

LARRI-010-CORE-CONF-003 Every normative MUST and MUST NOT MUST map to a machine-contract assertion, vector, harness assertion, or clearly identified publication gate.

12.4 Registries and supported versions

LARRI-010-CORE-REGISTRY-001 An identifier not enabled for the selected target and profiles MUST produce a registered unsupported-feature or unknown-feature refusal.

LARRI-010-CORE-REGISTRY-002 Generated registry views MUST be derived from one canonical authored source and verified for exact equality where published.

LARRI-010-CORE-VERSION-001 A Verifier MUST refuse an unsupported protocol, artifact, registry, or semantic contracts version rather than applying heuristic compatibility.

LARRI-010-CORE-VERSION-002 Unsupported versions and fields MUST NOT be interpreted through fallback parsing, aliases, punctuation substitution, or implementation-private compatibility rules.

A. Core assurance statements

Result	Permitted statement
Receipt PASS	The selected external trust validates every required attestation over the exact signed receipt content; receipt identity, assertion binding, required commitment treatment, and limitations validate for the receipt target.
Bundle PASS	Receipt assurance is established for every required receipt, and the signed logical bundle declaration matches every in-scope file and required content-role rule.
Profiled-bundle PASS	Bundle assurance is established and every additional artifact, trust input, and check required by the explicitly selected profile or combination succeeds.
REFUSE	The selected target-scoped assurance was not established. The result identifies the deterministic primary refusal and safely completed check statuses without making a substantive determination about the application assertion.
Invocation refusal	No enabled target-scoped verification contract was established. The invocation is rejected without mapping to another target and without fabricating a target-scoped result.

Required qualification. Every user-facing statement identifies the target and selected profiles and retains all applicable limitations.

A.1 Proprietary-information boundary

Boundary	Content
LARRI requires public specification of	Artifact formats, canonicalization rules, digest and signature inputs, external trust requirements, commitment forms, verification order, refusal behavior, result semantics, and claim limitations.
LARRI does not require disclosure of	Decision algorithms, business rules, policy logic, model weights, prompts, training data, private evidence, thresholds, internal state, source code, software architecture, operational databases, or confidential ruleset contents.

B. Core requirement summary

Area	Core rule
Scope	Target-specific claims; unsupported features cannot affect the result; fail closed.
Roles	Deployment roles may combine, but trust, signing, deterministic verification, and substantive-decision boundaries remain.
Trust	External trust cannot be established or overridden by the artifact, and base verification remains offline.
Assertion	Bind the exact public assertion without requiring private decision logic.
Commitments	Recompute supplied required commitments and distinguish absent or unrecomputed material.
Identity	Use non-recursive content-derived identities; never fabricate an identity.
Canonicalization	Use only registered exact-byte algorithms; reject unknown identifiers.
Signatures	Apply exact suite and policy semantics without prescribing a library or private architecture.
Bundles	Use one prepared-directory logical subject with explicit fixed control objects, non-recursive manifest identity, detached attestation, and closed file scope.
Verification	Follow the normative dependency order; recompute manifest identity before accepting its signature; block unsafe downstream work.
Invocation	Reject unsupported target selection before target-scoped verification.
Output	Emit deterministic normalized bytes with honest identities, registered refusals, selected non-baseline configuration binding, and no host diagnostics.
Profiles	Add only closed registered public conditions, retain trust-input traceability, and never weaken the core.
Privacy and safety	Avoid truth, completeness, identity, authority, trusted-time, clinical, legal, regulatory, and privacy-absence overclaims.
Authority	Keep public machine contracts authoritative; maintain separate non-circular semantic, evidence, report, and release identities.
Conformance	Judge public behavior without requiring proprietary source, architecture, or decision logic; treat additional reproduction claims as optional.
Versioning	Refuse unsupported versions and prohibit heuristic compatibility.

End of LARRI Core Protocol Specification 0.1.0