Elenchus Protocol

A Governed Principal-to-Substrate Dialog Protocol

Elenchus is a transport-agnostic protocol governing structured dialog between a human principal and a governed execution substrate. Every interaction cycle -- an Elenchus Unit (EU) -- passes through five sequential stages: ingestion, distillation, graph evaluation, external dispatch, and response attestation. No EU closes without a Chandra Protocol Chronicle Unit (CU) reference. No query reaches a frontier AI API without passing two governed evaluation gates. Voice is the reference transport implementation. The protocol is designed for regulated enterprise environments where the audit record is the authorization mechanism, not a retrospective receipt.

General Reasoning, Inc. · May 2026

Contents

1

Abstract

Elenchus is a transport-agnostic protocol governing structured dialog between a human principal and a governed execution substrate. The protocol defines a single atomic interaction unit -- the Elenchus Unit (EU) -- which represents one complete governed interaction cycle from principal input to attested response. No EU may close without producing a Chandra Protocol Chronicle Unit (CU) reference. No query may reach an external frontier AI API without passing two sequential governed evaluation gates: a local distilled model gate and an AllegroGraph semantic safety gate.

Voice interaction is the reference transport implementation. The protocol is transport-agnostic by design; text, voice, and future transports are all valid EU carriers. The governing invariant is not the transport but the five-stage pipeline that every EU must traverse, and the attestation requirement that every EU must satisfy before closing.

Elenchus is designed for regulated enterprise environments where the audit record is the authorization mechanism, not a retrospective receipt. It extends the Chandra Protocol and is intended to operate on a governed execution substrate conforming to the Aegis Genera specification. GABA Standard attestation (ABRAR artifact with mandatory Chandra CU reference) is required for all conforming implementations.

Core Invariant

Every Elenchus Unit opens on principal input and closes on attested response. An EU that does not produce a Chandra CU is not a valid EU. An implementation that permits EU closure without CU production is not a conforming Elenchus implementation.

2

Definitions

Principal

The human initiating dialog with the Substrate. The Principal is the source of all EU input. The protocol does not govern Principal identity; identity management is a Substrate responsibility.

Substrate

The governed execution environment receiving Principal input and producing attested responses. The reference Substrate implementation is Aegis Genera.

EU (Elenchus Unit)

The atomic unit of one complete governed interaction cycle. An EU opens on Principal input and closes when a Chandra CU is written against the completed cycle. An EU is either successful (response delivered, CU written) or failed (error response delivered, failure CU written). An EU is never left open.

Distilled Model

A local language model operating within the Substrate, authorized exclusively for NLQ parsing and safety evaluation. The Distilled Model is the first governed evaluation gate. It does not interact with the Principal directly. Its provenance, scope, and integrity are governed by Section 8 of this specification.

Graph Evaluation

The second governed evaluation gate. AllegroGraph applies clarity and safety filtering to Distilled Model output before any external dispatch is permitted.

Frontier API

The single sanctioned external AI API to which evaluated queries may be forwarded. The Frontier API is the only external exposure in the EU pipeline. All other computation is Substrate-local.

Chronicle

The append-only audit record produced by the Chandra Protocol. Every EU produces at least one Chronicle Unit (CU). The Chronicle is the authoritative record of all EU interactions.

CU (Chronicle Unit)

A Chandra Protocol append-only hash-chained audit record. Defined by the Chandra Protocol specification. In the context of Elenchus, every EU closure event -- success or failure -- produces a CU.

Boundary

The governed limit beyond which queries may not pass without satisfying all evaluation gate requirements. Boundary enforcement is a non-bypassable property of a conforming Substrate.

ABRAR

Governed AI Boundary Attestation Report. Defined by the GABA Standard. Required for all conforming Elenchus implementations. The ABRAR must reference a Chandra CU as its mandatory attestation anchor.

Mythos

The class of adversarial pressures -- technical, social, and institutional -- that attempt to degrade governed protocol enforcement. Named for the rhetorical displacement of logos by myth. Mythos critique is the practice of formally identifying and specifying Mythos attack surfaces within a governed protocol.

3

Protocol Architecture

The Elenchus Protocol defines a five-stage linear pipeline through which every EU must pass. The pipeline is strictly sequential. No stage may be skipped. No stage output may bypass the subsequent stage. A conforming Substrate must enforce this ordering as a structural property, not a procedural convention.

Stage	Name	Description	Failure Action
1	Ingestion	Raw principal input received by Substrate. Transport layer stripped; normalized input passed forward.	EU closes. Failure CU written.
2	Distillation	Local distilled model parses and validates NLQ. Query intent, scope, and safety evaluated against Substrate boundary.	EU closes. Failure CU written. Principal receives governed error response.
3	Graph Evaluation	AllegroGraph clarity and safety filter applied to distilled output. Query evaluated for boundary compliance and semantic coherence.	EU closes. Failure CU written. Principal receives governed error response.
4	External Dispatch	Single sanctioned external exposure. Query forwarded to frontier AI API. Response received by Substrate.	EU closes. Failure CU written (dispatch failure or timeout).
5	Response Attestation	Frontier API response evaluated by Substrate. Chandra CU written. EU closes. Attested response delivered to principal.	If response fails attestation, it is suppressed. Failure CU written.

The pipeline has exactly one external exposure: Stage 4, External Dispatch. All other computation is Substrate-local. This is a deliberate architectural constraint. The single external exposure is the point at which the governed boundary is crossed, and it is the point at which the most complete set of evaluation results is available to support that crossing decision.

Stage 5, Response Attestation, is not optional post-processing. It is a required pipeline stage. The Frontier API response is not delivered to the Principal until the Substrate has evaluated it and written a CU. A response that cannot be attested is suppressed. The Principal receives a governed error response and a failure CU is written. The EU closes.

The Single Exposure Principle

The Frontier API is the only point at which an EU crosses the governed boundary. Any implementation that introduces additional external exposures -- intermediate API calls, logging services, telemetry endpoints -- at any pipeline stage other than Stage 4 is not a conforming Elenchus implementation. Each such exposure is an unattested boundary crossing and a protocol violation.

4

The EU Lifecycle

4.1 EU State Machine

An EU exists in one of four states: OPEN, PROCESSING, CLOSED-SUCCESS, or CLOSED-FAILURE. State transitions are one-directional. An EU may not re-enter OPEN or PROCESSING once closed.

- ◆ **OPEN** EU created on Principal input. EU identifier assigned. Timestamp recorded. Input normalized by transport layer.
- ◆ **PROCESSING** EU is traversing the five-stage pipeline. State is maintained across all five stages under the EU identifier.
- ◆ **CLOSED-SUCCESS** EU completed all five stages. Attested response delivered to Principal. Chandra CU written. EU identifier retired.
- ◆ **CLOSED-FAILURE** EU failed at any pipeline stage. Governed error response delivered to Principal. Failure CU written with stage-of-failure recorded. EU identifier retired.

4.2 Failure Semantics

EU failure is atomic and immediate. When any pipeline stage produces a failure condition, the EU transitions directly to CLOSED-FAILURE. There is no retry mechanism within a single EU, no open-and-rephrase loop, and no partial completion state. This is a deliberate design decision.

The rationale is twofold. First, atomicity keeps the EU as the indivisible unit of audit record. An EU with internal retry state produces an ambiguous CU -- it is unclear which attempt the CU describes. Fail-fast produces one unambiguous record per attempt. Second, conversation recovery belongs to the application layer built on Elenchus, not to the protocol itself. The protocol governs the pipe. The application governs the conversation.

A Principal who receives a failure response opens a new EU by submitting revised input. Each attempt is independently attested. The Chronicle records all attempts, successful and failed, producing a complete interaction history that is auditable at the attempt level.

Fail Fast, Attest Everything

The complete record of failed EU attempts is not a deficiency -- it is an audit asset. A regulated environment benefits from knowing not only what queries succeeded but what queries failed, at which stage, and how many times. Elenchus produces this record by construction.

5

Boundary Conditions

A boundary condition is any condition that prevents an EU from advancing to the next pipeline stage. Each stage has defined boundary conditions. All boundary conditions result in immediate EU closure with CLOSED-FAILURE state.

Stage 1 -- Ingestion

- Input exceeds maximum length defined by Substrate configuration.
- Input cannot be normalized by the transport layer (malformed, unsupported encoding).
- Principal authentication failure (if Substrate enforces authentication).

Stage 2 -- Distillation

- Distilled Model determines query intent is outside Substrate-authorized scope.
- Distilled Model determines query contains content that fails safety evaluation.
- Distilled Model produces output that cannot be parsed as valid NLQ.
- Distilled Model inference timeout (configurable, Substrate-defined).

Stage 3 -- Graph Evaluation

- AllegroGraph determines query violates semantic boundary constraints.
- AllegroGraph determines query is insufficiently specified for safe forwarding.
- AllegroGraph evaluation timeout (configurable, Substrate-defined).

Stage 4 -- External Dispatch

- Frontier API unreachable or returns non-recoverable error.
- Frontier API response timeout.
- Frontier API response cannot be parsed by Substrate.

Stage 5 -- Response Attestation

- Frontier API response fails Substrate safety evaluation.
- Chandra CU cannot be written (Chronicle unavailable).
- Response attestation timeout.

In all boundary condition cases, the governed error response to the Principal must not expose internal pipeline state, stage-of-failure specifics, or Substrate configuration details. The error response is a governed artifact. Its content is determined by Substrate policy, not by the nature of the failure. The failure detail is recorded in the CU, not communicated to the Principal.

6

Transport Bindings

The Elenchus Protocol is transport-agnostic. The five-stage pipeline and EU lifecycle are invariant across transports. Transport bindings define how raw Principal input is delivered to Stage 1 (Ingestion) and how the attested response is returned from Stage 5 (Response Attestation) to the Principal.

6.1 Voice (Reference Implementation)

Voice is the reference transport for Elenchus. The voice pipeline preceding Stage 1 consists of: voice activity detection (VAD), speech-to-text transcription (STT), and text normalization. The voice pipeline following Stage 5 consists of: text-to-speech synthesis (TTS) and audio delivery. All voice pipeline components are Substrate-local. No voice audio is transmitted to external services.

The voice transport is designed for the governed conversational interface use case. Latency budget for a natural conversational experience is 1.0 to 1.5 seconds from end of Principal utterance to first audio out. The primary latency contributors are STT (150-300ms), Distilled Model inference (150-800ms depending on hardware), AllegroGraph evaluation (<100ms), Frontier API (800-2000ms to first token, streaming), and TTS first chunk (100-200ms). Hardware selection for the Substrate should be made with reference to this budget.

6.2 Text

Text transport delivers Principal input as normalized UTF-8 text directly to Stage 1. No STT or TTS components are required. The EU pipeline is identical to the voice binding from Stage 1 onward. Text transport is appropriate for interfaces where the Principal interacts via keyboard or programmatic input.

6.3 Extension Points

Additional transport bindings may be defined by Substrate implementers provided that: (a) the transport delivers normalized text input to Stage 1; (b) all transport-layer processing is Substrate-local; (c) no transport-layer component introduces an external exposure not declared in the ABRAR; and (d) the transport binding is documented and included in the attestation record. Undocumented transport bindings are protocol violations.

7

Attestation Requirements

Every EU closure event -- success or failure -- must produce a Chandra Protocol Chronicle Unit (CU). The CU is the attestation record for the EU. An EU without a CU has not closed. An implementation that permits EU closure without CU production is not a conforming Elenchus implementation.

Mandatory CU fields for a valid EU record:

`eu-id` -- Unique EU identifier. Assigned at EU open. Carried through all pipeline stages.

`eu-state` -- Final EU state: CLOSED-SUCCESS or CLOSED-FAILURE.

`eu-stage-of-failure` -- Stage at which failure occurred (1-5). Null for CLOSED-SUCCESS.

`eu-open-ts` -- UTC timestamp of EU open.

`eu-close-ts` -- UTC timestamp of EU close.

`eu-transport` -- Transport binding identifier (e.g., "voice", "text").

`eu-distilled-model-hash` -- SHA-256 hash of the Distilled Model weights file loaded by the Substrate session. Mandatory. See Section 8.4.

`eu-session-id` -- Substrate session identifier. Links EU to session-level attestation including model hash registration.

`eu-frontier-dispatched` -- Boolean. True if Stage 4 was reached and dispatch was attempted.

`eu-chandra-prev-hash` -- Hash of the preceding CU in the Chronicle chain. Inherited from Chandra Protocol.

ABRAR Relationship

Every conforming Elenchus implementation must produce a GABA Standard Governed AI Boundary Attestation Report (ABRAR). The ABRAR must reference a Chandra CU as its mandatory attestation anchor. An ABRAR without a CU reference is a document, not an attestation. GABA attestation is mandatory, not recommended. An implementation that does not produce a valid ABRAR is not a conforming Elenchus implementation.

Document vs. Attestation

The distinction between an ABRAR with a CU reference and an ABRAR without one is not procedural. It is ontological. An ABRAR without a CU reference asserts that governed interaction occurred. An ABRAR with a CU reference proves it. Elenchus requires proof.

8

Distilled Model Governance

The Distilled Model is the first and most consequential governed gate in the EU pipeline. If the Distilled Model is compromised, miscalibrated, or operating outside its authorized scope, all downstream evaluation is operating on poisoned input. AllegroGraph graph evaluation and Chandra attestation cannot compensate for a compromised first gate. This section is accordingly the most detailed and most restrictive in the specification.

Critical Governance Surface

The Distilled Model governance requirements in this section are mandatory, not advisory. An implementation that relaxes any requirement in Section 8 -- provenance documentation, scope constraint, model hash attestation, MCP tool declaration, or evaluation suite compliance -- is not a conforming Elenchus implementation regardless of compliance with other sections.

8.1 Provenance Requirements

The Distilled Model must have a documented, auditable origin. Specifically:

- The base model must be identified by name, version, and originating organization.
- A public model card must exist for the base model at the time of Substrate deployment.

◆ If the model has been fine-tuned, the fine-tuning process must be documented including: training data sources, fine-tuning objective, evaluation methodology, and the identity of the party that performed the fine-tuning.

◆ The fine-tuning process must be reproducible from documented inputs. Non-reproducible fine-tuning is not permitted.

◆ Anonymous or opaque models -- models whose origin, training data, or fine-tuning process cannot be documented -- are categorically excluded.

8.2 Scope Constraint

The Distilled Model is authorized for exactly one function: NLQ parsing and safety evaluation for query forwarding decisions. This constraint is absolute.

◆ The Distilled Model does not interact with the Principal directly under any circumstances.

◆ The Distilled Model does not perform functions beyond NLQ parsing and safety evaluation.

◆ The Distilled Model does not have access to Chronicle data, session history, or Principal identity information beyond the current EU input.

◆ Any model operating beyond this scope within a Substrate claiming Elenchus conformance is a protocol violation regardless of the quality of its output.

8.3 Suitability Criteria

The following criteria define a suitable Distilled Model. General Reasoning, Inc. maintains a registry of evaluated models (see Section 8.8). Implementers may use registry-listed models or submit candidate models for evaluation.

Parameter Count Ceiling. Models up to approximately 14 billion parameters are recommended for the Distilled Model role. Larger models are not categorically excluded but must be justified: a larger model with broader capability is harder to scope-constrain and harder to audit. Bigger is not better in this role.

Quantization Transparency. If the model is quantized, the quantization method (e.g., GGUF Q4_K_M, AWQ), resulting capability delta relative to full-precision baseline, and any known degradation in safety-relevant evaluation tasks must be documented.

Benchmark Compliance. The model must demonstrate performance against the Elenchus NLQ Evaluation Suite (see Section 8.7). Minimum passing thresholds are published in the suite. A model that does not meet the published thresholds is not suitable for the Distilled Model role regardless of other properties.

Adversarial Robustness. The model must be evaluated against the adversarial prompt subset of the Elenchus NLQ Evaluation Suite. A model that can be induced to pass queries it should block through prompt injection or instruction following attacks is not suitable.

8.4 Model Hash Attestation

Every EU CU must contain the SHA-256 hash of the Distilled Model weights file loaded by the Substrate for the session in which the EU occurred. This is a version 1 requirement. There is no deferral.

Implementation note: the model hash is computed once at Substrate session initialization, not per-EU. The hash is registered as a session-level attestation and referenced by session identifier in each EU CU. This makes per-EU model attestation tractable without per-EU recomputation overhead.

A CU that does not contain a model hash reference (direct or via session identifier) is not a valid Elenchus CU. The rationale: an attestation that records what query was evaluated without recording what evaluated it is not an attestation of the evaluation process. It is a log entry. Elenchus requires attestation of the process, not merely the outcome.

8.5 MCP Extensibility Governance

The Model Context Protocol (MCP) permits extension of model capability through tool binding. In the context of the Distilled Model, MCP extensibility is a governed capability that must be treated with the same rigor as scope constraint.

- ◆ Every MCP tool available to the Distilled Model must be declared in the ABRAR. Undeclared tools constitute a protocol violation.
- ◆ MCP tool additions after initial ABRAR submission require ABRAR amendment and re-attestation.
- ◆ Each declared MCP tool must be evaluated for scope compatibility: does this tool expand the Distilled Model's effective capability beyond NLQ parsing and safety evaluation? A tool that does is not permitted.
- ◆ MCP tools that introduce external network exposure must be treated as additional pipeline exposures and are prohibited unless they replace Stage 4 (External Dispatch) entirely and are declared as such.

8.6 Mythos Critique Surface

The following attack vectors against the Distilled Model governance are explicitly identified and specified against. This section is part of the protocol specification, not supplementary guidance. Conforming implementations must address each vector.

Model Substitution. An attested model is replaced with an unattested model without updating the ABRAR or session hash. Mitigation: mandatory model hash attestation in every EU CU (Section 8.4). A substituted model produces CUs with a hash that does not match the declared model. This is a detectable Chronicle anomaly.

Prompt Injection via Principal Input. A Principal crafts input designed to manipulate the Distilled Model into passing a query it should block. Mitigation: adversarial robustness evaluation requirement (Section 8.3). The evaluation suite must include a standard adversarial prompt corpus. Models that fail adversarial evaluation are not suitable.

Capability Creep via MCP. MCP tools are added to the Distilled Model context that silently expand its effective scope beyond NLQ validation, enabling it to perform functions it is not authorized to perform. Mitigation: mandatory tool declaration in ABRAR (Section 8.5). Undeclared tool use is a protocol violation detectable in attestation review.

Fine-Tune Poisoning. A fine-tuned Distilled Model contains adversarially introduced biases that cause it to systematically pass certain classes of dangerous queries. This is the most difficult vector to specify against because it requires the specification to address evaluation methodology, not merely provenance. Mitigation:

reproducible fine-tuning requirement (Section 8.1) and benchmark compliance requirement (Section 8.3). A poisoned fine-tune that passes the Elenchus NLQ Evaluation Suite represents a suite failure, not a protocol failure. This is addressed in Section 8.7 through suite governance.

Governance Path Substitution. An implementation claims Elenchus conformance while omitting GABA attestation or relaxing CU requirements, marketing itself as lower-friction. This is a governance exploit, not a technical exploit. Mitigation: the mandatory GABA requirement (Section 7) and explicit statement in this specification that an implementation without valid ABRAR is not running Elenchus.

8.7 Evaluation Suite Governance

The Elenchus NLQ Evaluation Suite is the reference benchmark for Distilled Model suitability assessment. It is maintained by General Reasoning, Inc. under the same MIT license as this specification.

Governance model: General Reasoning, Inc. holds merge authority over the evaluation suite. External contributions are accepted through a formal submission process subject to independent review. Merge authority is not shared. This is not a limitation -- it is a security property. An evaluation suite with open merge authority is an evaluation suite that can be poisoned through contribution. The governance model is a feature of the protocol.

General Reasoning, Inc. has submitted, or is in the process of submitting, the Elenchus model governance criteria defined in this section to the National Institute of Standards and Technology (NIST) for consideration as a standardization proposal. The evaluation suite governance will transfer to NIST or a NIST-designated body upon successful standardization. Until that transfer, General Reasoning, Inc. acts as sole maintainer.

General Reasoning, Inc. is actively identifying and evaluating specific models against the criteria defined in this section. The evaluation methodology, passing thresholds, and evaluated model registry are published and maintained at elenchusprotocol.com. The evaluated model list is a living document; this specification does not name specific models because model availability and suitability change faster than protocol specifications should.

8.8 Model Registry

General Reasoning, Inc. maintains a public registry of models evaluated against the Elenchus NLQ Evaluation Suite. The registry records: model name and version, originating organization, evaluation date, evaluation suite version used, pass/fail status per benchmark category, quantization variants evaluated, and known limitations relevant to the Distilled Model role.

The registry is advisory. Implementers may use registry-listed models without independent re-evaluation, accepting the published results. Implementers may also use non-registry models provided they conduct and document independent evaluation against the published suite and make that evaluation available for attestation review.

9
Reference Implementation

The Elenchus Protocol reference implementation is provided by General Reasoning, Inc. as part of the Aegis Genera governed execution substrate. The reference implementation demonstrates all mandatory protocol requirements and serves as the normative example for conformance evaluation.

Substrate -- Aegis Genera governed execution substrate. Three-layer architecture: Substrate, Instrument, Chronicle.

Language -- Allegro Common Lisp (Franz Inc. enterprise license). alisp executable.

Web Server -- AllegroServe.

Graph Engine -- AllegroGraph. Used for Stage 3 (Graph Evaluation) and as the derived read/reasoning model in the CQRS architecture.

Chronicle -- Chandra Protocol. Append-only hash-chained audit log. MIT licensed.

Build System -- Yocto Project. Custom Aegis Genera layer. Linux-only. No Windows support.

Voice Transport -- Whisper.cpp (STT, Substrate-local). Kokoro or Orpheus TTS (TTS, Substrate-local).

Distilled Model Runtime -- llama.cpp with ROCm/HIP backend. AMD RDNA architecture recommended. Open-source driver stack required for governed substrate conformance.

Attestation -- GABA Standard ABRAR. Chandra CU as mandatory attestation anchor.

The reference implementation is available at elenchusprotocol.com. The open-source Core implementation is MIT licensed. Enterprise extensions are maintained separately.