Arbitrary Audit Query

Why Chandra Represents a Different Category of Compliance Evidence

General Reasoning, Inc. · April 2026

1. The Central Claim

> **Traditional audit systems capture what they expected to need. Chandra captures everything and lets you ask anything.**

This is not a feature distinction. It is a categorical distinction. Every audit system in current use -- log aggregators, SIEM platforms, workflow audit trails, compliance dashboards -- was designed around a fixed query model. The system's designers decided in advance which questions would need to be answered. The system captures data in a shape that answers those questions. When an examiner asks a question the designers did not anticipate, the answer either does not exist or must be reconstructed by hand from raw logs, if the raw logs were retained at all.

Chandra's design inverts this. The immutable hash-chained context unit (CU) record captures a complete, attributed, tamper-evident account of every action in the system at write time. The query layer is applied after the fact, against data that was not shaped to answer any particular question. An auditor or examiner can ask any question that is answerable from the recorded fields -- including questions that could not have existed when the data was recorded.

The word "could not" is precise. It does not mean the examiner had not thought of the question yet. It means the question was structurally impossible to ask -- because the regulatory framework, the threat category, or the technology that makes the question necessary did not yet exist.

When AI governance regulations are written in 2027, the question "show me every AI inference call made by this organization between January and March 2026 where a human approved the output and an agent acted on it within 60 seconds" could not have existed in 2024. The regulatory framework requiring that evidence did not exist. Traditional audit systems require you to decide what to capture before the question is asked. If you did not anticipate the question, the data is not there.

Chandra captures everything by construction. The chain grows with every action regardless of what questions anyone expects to ask. When the 2027 regulator arrives with

questions that could not have existed in 2025, the chain already has the answers. The law changes. The threat model evolves. New categories of evidence get required retroactively. Chandra is already capturing it.

This section describes what that capability means in practice, at what scope it operates, and why it represents a compliance evidence model that has no direct predecessor in enterprise software.

2. What Arbitrary Query Means

The term "arbitrary" requires precision. It does not mean unlimited. It means: not determined in advance by the system designer. The query is constrained by what fields were recorded on each CU, and by the computational expressiveness of the query layer. Within those constraints, the examiner constructs the query. The system did not anticipate it. The system answers it anyway.

Concrete examples clarify the scope:

Temporal reconstruction	Show every CU published by agent X between March 1 and March 15 where the artifact contained a field named "approval" with value "granted", and a subsequent CU on the same spoke was published by a different author within 60 seconds.
Attribution chain	Identify every action in the system that can be traced, through predecessor-id linkage, to a specific root CU published at system initialization. Show the full lineage.
Cross-spoke anomaly detection	Find every subject where a human-authored CU was followed by an agent-authored CU within the same hour, and the agent CU carried a subject-type that was not present in the spoke's prior history.
Completeness verification	For every spoke in domain X, verify that the hash chain is unbroken from CU-0001 to the current tail. Report any gap or hash mismatch with the CU-id and position of the first failure.

None of these queries requires that the system anticipated the question. They require only that the relevant fields were recorded on the CU at publish time -- which is a property of Chandra's write path, not its query layer.

3. The Five Layers of Query Scope

Chandra's query capability operates at five distinct scopes, each building on the one below it. The architectural significance of each layer is not just the query capability it provides but

the compliance surface it covers.

1	**Single Spoke -- Recent CUs** The operational layer. An authorized user browses the 25 most recent CUs in a spoke. No query planning required. O(1) tail lookup via spoke-last-cu-id.
2	**Single Spoke -- Arbitrary Query** The audit layer. An examiner issues a parameterized query against a single spoke's full CU chain. Field filtering, temporal windowing, sequence detection. This is where the query provider earns its value. AllegroGraph with SPARQL provides provable index utilization -- the query plan is itself an audit artifact.
3	**Cross-Spoke Query** The domain audit layer. Query across all spokes within a domain. "Show me every subject where approval was granted by a human and a subsequent agent action was taken within 60 seconds" -- executed across hundreds or thousands of spokes simultaneously. Graph traversal via property paths. No existing compliance platform offers this natively.
4	**Enterprise Scope (Hub and Domain)** The organizational audit layer. Multi-tenant scope with isolation guarantees. A domain owner queries across all spokes in their domain. A hub owner queries across all domains. The instance manifest establishes the scope boundaries. Cross-domain queries are controlled by the authorization layer, not by technical limitation.
5	**The Marshaller -- Federated Query** The regulatory layer. Query across physically separate Chandra deployments, regardless of where they run. A regulatory examiner issues a query against every regulated entity's Chandra instance simultaneously. Each instance answers independently. The Marshaller aggregates and returns a unified result. This is infrastructure for an entire regulated industry, not a product feature.

4. Why This Has No Predecessor

The claim that Chandra represents a different category of compliance evidence requires substantiation. The comparison class is: SIEM platforms (Splunk, IBM QRadar, Microsoft Sentinel), workflow audit systems (ServiceNow, Jira audit logs), compliance dashboards (Vanta, Drata, Tugboat Logic), and general-purpose audit log stores (AWS CloudTrail, Azure Monitor).

Property	Traditional Systems	Chandra
Query model	Fixed schema, anticipated questions	Arbitrary, post-hoc query against immutable chain
Tamper evidence	Access controls on mutable logs	Cryptographic hash chain -- structural, not operational
Attribution	Log entry with actor field	CU with human-author, model-id, predecessor-id -- full prove
Cross-entity query	Manual correlation across log sources	Native cross-spoke, cross-domain, federated
Retroactive questions	Not supported -- data not retained in queryable form	Supported -- chain is the record, query is applied after
Query auditability	Query logs, if retained	SPARQL query plan is itself an audit artifact
Scope	Single deployment	Single spoke to federated multi-instance (Marshaller)

The comparison is not that Chandra has better logs. It is that Chandra does not have logs in the traditional sense. It has chains. The distinction is not semantic. A log is a record of what the system chose to capture, in the shape the designer chose, queryable through the interface the designer provided. A chain is a cryptographically linked sequence of attributed, immutable records that can be queried against any question the fields support -- including questions the designer did not anticipate, asked by an examiner whose question could not have existed when the data was written.

5. Honest Limitations

This section applies the same standard of intellectual honesty that governs the parent whitepaper.

Query is bounded by recorded fields

An examiner can only ask questions answerable from what was recorded on the CU. If a field was not captured at publish time, it cannot be queried retroactively. The write path determines the query horizon.

Query performance is not guaranteed

Arbitrary queries against large chains require a performant query layer. The naive in-memory provider is correct but not scalable. AllegroGraph with SPARQL provides provable index utilization for supported query patterns. Novel query shapes may require query plan optimization.

The Marshaller is designed, not yet built

Federated query across Chandra instances (Layer 5) is an architectural design. The implementation is on the product roadmap. This section describes the capability as designed; the shipping product covers Layers 1 through 4.

Cross-instance query requires trust establishment

Federated query across independently operated Chandra instances requires mutual authentication and authorization. The protocol design includes this. The operational trust establishment between regulated entities and regulatory bodies is a governance problem, not a technical one.

6. Conclusion

The compliance evidence problem is not fundamentally a data storage problem. It is a query problem. Regulators and examiners ask questions. The question an examiner will ask five years from now -- about a decision made today -- cannot be anticipated today. Any audit system that requires anticipation of the question before the data is recorded is structurally inadequate for the compliance demands of AI-augmented regulated operations.

Chandra's architecture addresses this at the foundation. The chain is the record. Attribution, sequencing, and hash integrity are structural -- not operational assertions that can be disabled or misconfigured. The query layer is applied after the fact, against data that was not shaped to answer any particular question.

> **Traditional audit systems capture what they expected to need. Chandra captures everything and lets you ask anything. That is not a feature. That is a different category of system.**