White paper

The Context Window and the Context Layer

An architectural distinction for agentic AI

Keith Deutsch

Co-Founder, CTO · Clinical Context AI

October 2026

White paper

Contents

01	Introduction	3
02	The limitations of retrieval-based context	3
03	Systems of Record are not in the context business	5
04	Relationships as the foundation of context	6
05	The living Context Model	8
06	Context as shared infrastructure	11
07	Grounding the Context Layer in a domain ontology	12
08	Integrating human concerns and constraints	14
09	From request-driven reasoning to context-driven intervention	17
10	An architectural separation with broader implications	19

Figures

Fig. 1	Retrieval-based context assembly	4
Fig. 2	Relationships as the foundation of context	7
Fig. 3	The Context Layer as an agentic system	10
Fig. 4	Integrating human concerns and constraints	15

Author

Keith Deutsch

Co-Founder, CTO

Published

October 2026

Length

About 3,000 words

13-minute read

Section

01

Introduction

The Context Window and the Context Layer are distinct but complementary concepts in agentic AI architectures. A Context Window contains the information made available to an inference model for a particular reasoning task. A Context Layer is a persistent, semantically integrated information substrate that maintains the evolving relationships necessary to construct meaningful Context Windows.

A Context Window is assembled for inference; a Context Layer maintains contextual understanding independently of any particular inference.

The distinction is fundamental. A Context Window is assembled for inference; a Context Layer maintains contextual understanding independently of any particular inference. The former determines what information a model can consider at a given moment. The latter establishes and maintains the contextual relationships that give that information its significance.

This distinction has implications beyond the construction of better Context Windows. By maintaining contextual understanding independently of individual inference tasks, a Context Layer makes that understanding persistent, cumulative, and reusable across applications. In dynamic environments, it can also evaluate emerging conditions against domain-specific concerns and constraints, recognizing when changes in context warrant attention and initiating further reasoning or intervention. This shifts the role of contextual infrastructure beyond supporting responses to predefined questions, toward enabling more open-ended, exploratory, and evolving forms of agentic reasoning.

Section

02

The limitations of retrieval-based context

In many contemporary agentic architectures, Context Windows are assembled by retrieving information from multiple sources, often using semantic similarity as a proxy for relevance. Conventional Retrieval-

Augmented Generation (RAG) systems exemplify this approach, identifying information fragments related to a query and presenting them to an inference model.

Information that is semantically similar is not necessarily contextually relevant …

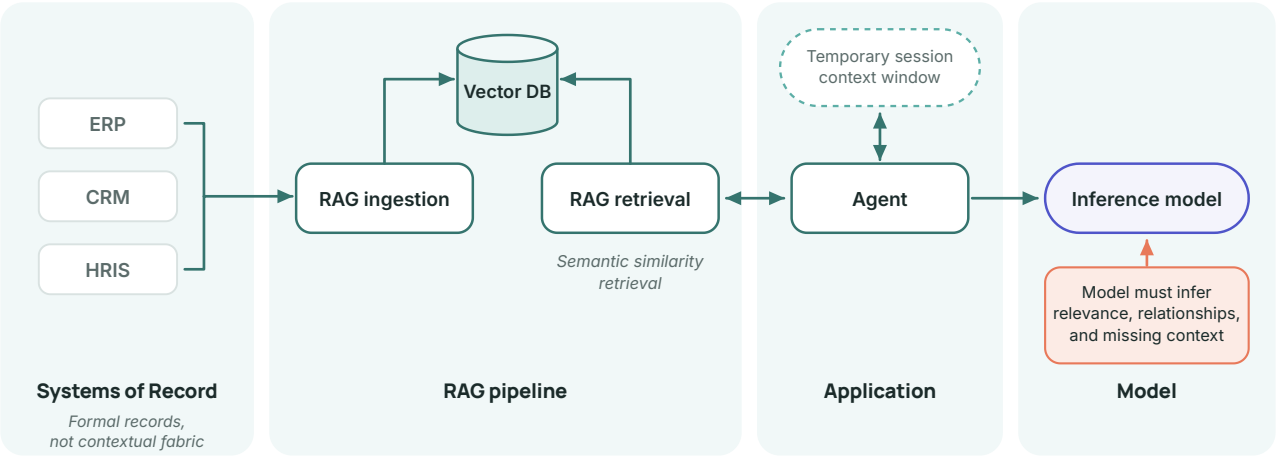
While effective in relatively bounded applications, retrieval alone does not establish the relationships among retrieved fragments, determine their significance, or identify what is missing. Information that is semantically similar is not necessarily contextually relevant, and information that is critically related may exhibit little semantic similarity.

The burden of contextualization therefore falls disproportionately on the inference model. The model must determine how retrieved information relates to the question, establish relationships among fragments, reconcile apparent contradictions, and infer what additional information may be needed. Much of this work must be repeated across successive inferences, even when the underlying information and its relationships have changed very little.

This approach can work reasonably well when the problem space is narrow, the structure of the underlying information is predictable, and the agent can impose useful constraints on the question and the resulting inference. However, as agentic applications move into more open-ended environments, both the range of possible questions and the heterogeneity of relevant information expand rapidly. The ability to constrain the inference diminishes, increasing reliance on the model's capacity to reconstruct context from disparate information fragments.

Figure 1

Retrieval-based context assembly



More sophisticated retrieval methods, larger Context Windows, and increasingly capable inference models can mitigate these limitations. But they do not eliminate the underlying architectural problem: retrieving information and maintaining contextual understanding are different responsibilities.

Section

03

Systems of Record are not in the context business

The problem is amplified by the dependence of many agentic systems on Systems of Record.

Systems of Record are designed to preserve structured, auditable representations of transactions, events, observations, and decisions. Their information models are organized around defined operational purposes, supporting consistency, accountability, and the reliable retrieval of formal records.

These systems contain valuable information and relationships, but those relationships are bounded by the purposes for which the systems were designed. They do not necessarily preserve the broader network of associations, temporal dependencies, provisional interpretations, uncertainties, and emerging signals required for open-ended contextual reasoning.

Much of the contextual understanding needed to interpret formal records has traditionally been supplied by humans operating within the associated processes. Human expertise connects information across organizational and system boundaries, recognizes the significance of incomplete evidence, and interprets formal records in light of circumstances that may never have been explicitly captured.

When agentic systems are introduced into these processes, this contextual responsibility does not disappear. If the underlying information infrastructure cannot supply it, the burden shifts to the inference model.

Accountability requires Systems of Record, Context requires Systems of Information.

The limitations of Systems of Record are not defects in their design. They reflect a different architectural responsibility. Preserving a dependable record of what has occurred is not the same as maintaining an evolving

understanding of what those records mean in relation to one another. Accountability requires Systems of Record, Context requires Systems of Information.

Section

04

Relationships as the foundation of context

The fundamental issue is that the significance of information resides not merely in individual facts, but in the relationships among them.

An observation, event, or document may have an identifiable meaning in isolation. Its significance within a particular situation, however, depends on its relationships to other information, the circumstances in which it arose, its history, and the questions or concerns against which it is being evaluated.

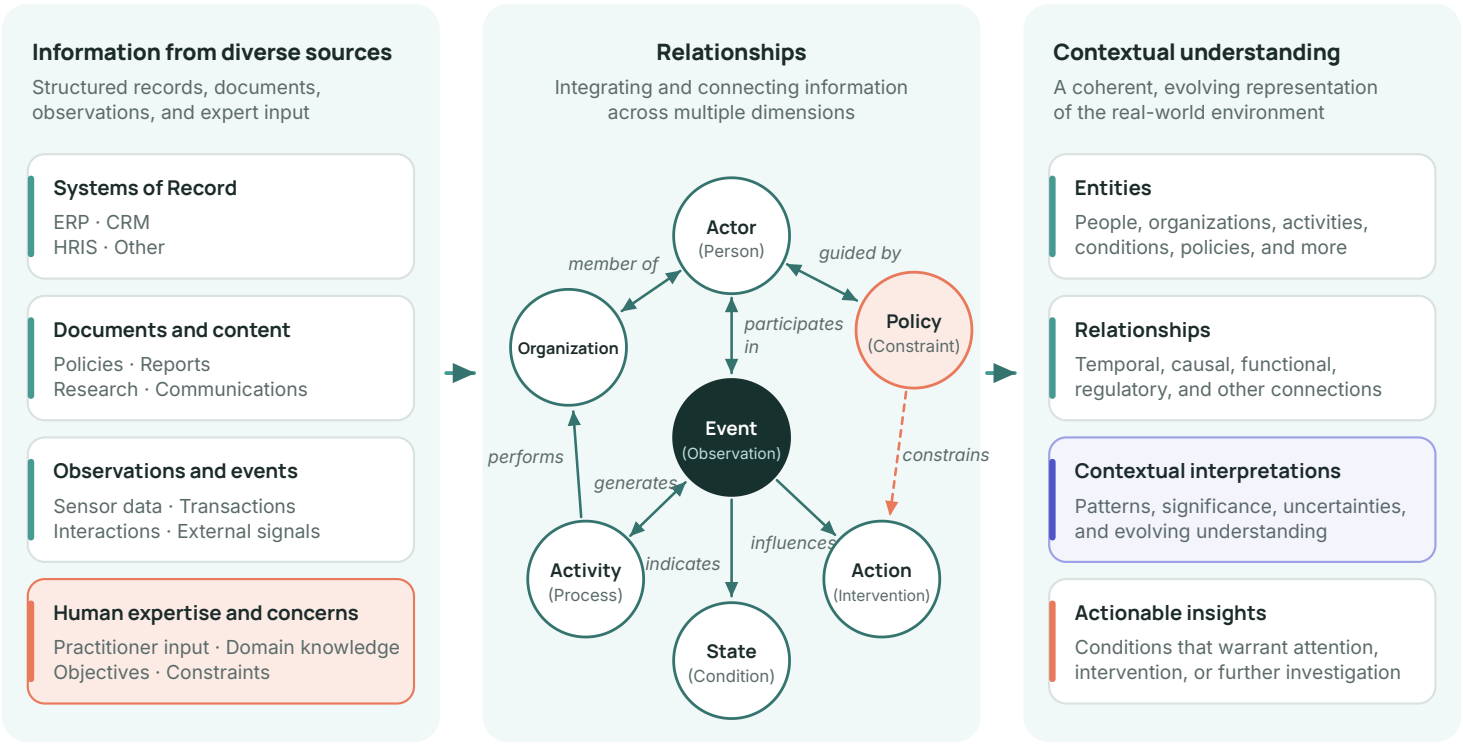
These relationships may be temporal, causal, functional, or inferential. They may cross system boundaries, connect formal records with less structured information, and evolve as new evidence becomes available.

The patterns that matter frequently emerge from these intersections rather than from individual observations. Information fragments that appear inconsequential in isolation may collectively reveal an important condition. Conversely, information that appears highly relevant when considered independently may lose significance when placed in its broader context.

Contextual understanding therefore requires more than retrieving and aggregating information fragments. It requires an integrated representation of their relationships, dependencies, and evolving significance.

Figure 2

Relationships as the foundation of context



> **The fundamental unit of contextual understanding is not the information fragment, but the relationships that give it significance.**

05

Section 05

The living Context Model

> The Context Layer is itself an agentic system.

In a relatively static environment, a sufficiently rich and well-maintained knowledge representation may provide much of the required contextual substrate. A Context Layer operating in such an environment might resemble a periodically updated Knowledge Graph, with agentic contextualization occurring primarily as new information is ingested. Even here, the preservation of meaningful relationships provides a richer foundation for inference than the retrieval of semantically similar information fragments alone.

But many real-world environments are dynamic. They generate ongoing flows of observations, events, changes, and interactions that must be integrated into an evolving understanding of the system. As the pace and complexity of change increase, the requirements for maintaining contextual understanding change with them.

In these environments, the Context Layer becomes an active, event-driven system that maintains a living Context Model as new signals arrive, circumstances change, and interpretations evolve. Its defining characteristic remains the maintenance of a coherent, semantically integrated contextual representation, but the mechanisms of maintenance must now operate at a frequency and level of responsiveness appropriate to the dynamics of the environment.

The Context Layer is itself an agentic system. Its living Context Model is maintained through successive cycles of event-driven interpretation and semantic integration.

When a new event or observation arrives, the existing Context Model provides the contextual basis for interpreting it. A relevant portion of that model is assembled into a Context Window, enabling an inference model to evaluate the new information in relation to prior observations, established relationships, evolving interpretations, and applicable human concerns and constraints.

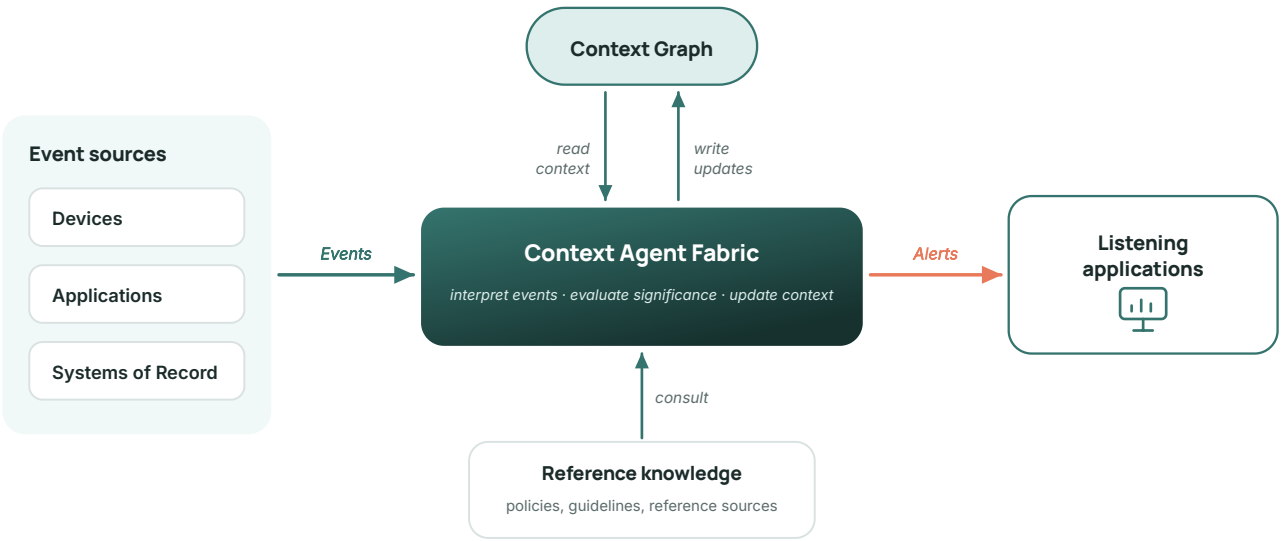
The Context Layer's agentic framework then determines what contextualization actions are required and integrates their results into the persistent Context Model. New relationships may be established, existing interpretations revised, uncertainties preserved or resolved, and emerging patterns updated. Each cycle builds upon the contextual understanding maintained through previous cycles.

But the agentic framework is not limited to maintaining contextual understanding. As an event is interpreted and integrated, the framework can also evaluate whether the resulting contextual state satisfies previously established conditions of significance. These may include threshold crossings, the completion of recognized patterns, or other conditions defined through domain-specific design or human-authored concerns and constraints. When such conditions are met, the Context Layer can generate and transmit a message or alert to an appropriate consuming application, initiating further reasoning or intervention.

Contextualization is therefore both cumulative and potentially actionable. The relationships established through successive inference cycles persist within the shared Context Model, while changes in those relationships can themselves become events that initiate application-level activity. The Context Layer does not merely maintain an evolving understanding of the environment; it can actively recognize and communicate when that understanding warrants attention.

Figure 3

The Context Layer as an agentic system



Incoming events are interpreted in light of existing context and reference knowledge; resulting contextual updates and significance assessments can trigger alerts to downstream applications.

A second inference cycle occurs when a consuming application is invoked, whether by an external request, an established workflow, or a contextually significant event identified by the Context Layer. The application

determines the scope of information relevant to its task and constructs a Context Window from the corresponding portion of the Context Model.

The application's inference model must still interpret the information it receives, but its task is fundamentally different. Rather than reconstructing the relationships among disparate source fragments, it interprets an already contextualized representation in light of a particular objective.

The architecture therefore separates two complementary forms of agentic reasoning: continuous, event-driven contextualization that maintains the shared understanding of a dynamic environment, and episodic, scope-driven interpretation that applies that understanding to particular decisions and actions.

The Context Layer does not replace inference. It makes the results of contextual inference persistent, cumulative, and reusable across applications.

Section

06

Context as shared infrastructure

The Context Layer occupies a distinct position between the dynamic systems being observed and the agentic applications that reason about them. It transforms information from multiple sources into a shared, continuously maintained contextual representation that exists independently of any particular application.

Contextual understanding becomes a reusable infrastructure capability ...

This separation allows multiple agents, pursuing different objectives, to draw upon the same underlying Context Model while constructing Context Windows appropriate to their individual tasks. Contextual understanding becomes a reusable infrastructure capability rather than something each application must independently develop and maintain.

The designation *layer* describes this architectural responsibility, not a particular technology or implementation. The Context Layer may employ knowledge graphs, databases, event processing, inference models, and other mechanisms. What defines it is its role in maintaining a coherent, evolving contextual representation of the environment and making that understanding available to the applications operating upon it.

Section

07

The Context Layer cannot arrive in a new domain as an empty framework ...

Grounding the Context Layer in a domain ontology

While the core machinery of a Context Layer may be largely independent of the domain it serves, meaningful contextualization requires a domain-specific semantic foundation. The Context Layer cannot arrive in a new domain as an empty framework and be expected to develop a coherent understanding of its environment from incoming signals alone.

That foundation is provided by a domain ontology: a representation of the principal entities, relationships, and semantic distinctions through which the domain is understood.

Certain concepts may be common across domains. Actors, occurrences, records, states, and relationships provide broadly reusable building blocks. But the specific meaning and structure of these concepts vary considerably. An Actor might represent a patient in a clinical environment or a machine within an industrial system. Each has its own identity, relationships, internal structure, and domain-specific significance.

The ontology establishes these essential distinctions, providing the semantic framework within which the Context Layer interprets incoming information and maintains the evolving Context Model.

Agentic inference introduces an important degree of flexibility into this architecture. Traditional information systems have often relied on extensively specified data models and attributes to support deterministic interpretation and processing. An agentic Context Layer can operate with a comparatively high-level ontology because much of the content associated with its entities can remain relatively free-form.

Rather than requiring every potentially meaningful attribute or relationship to be anticipated and formally defined, the inference machinery can interpret less structured content in relation to the entities, relationships, and domain semantics established by the ontology.

This does not eliminate the need for formal structure. Identity, essential relationships, operational constraints, and other critical distinctions must remain explicit. But it allows the ontology to concentrate on the semantic

foundations of the domain rather than attempting to exhaustively prescribe the structure of all the information that may become relevant.

The result is a separation between reusable contextualization machinery, a domain-specific semantic framework, and the living Context Model maintained through their interaction. This allows the Context Layer to accommodate evolving and unanticipated information without sacrificing the coherent domain understanding necessary for meaningful contextualization.

08

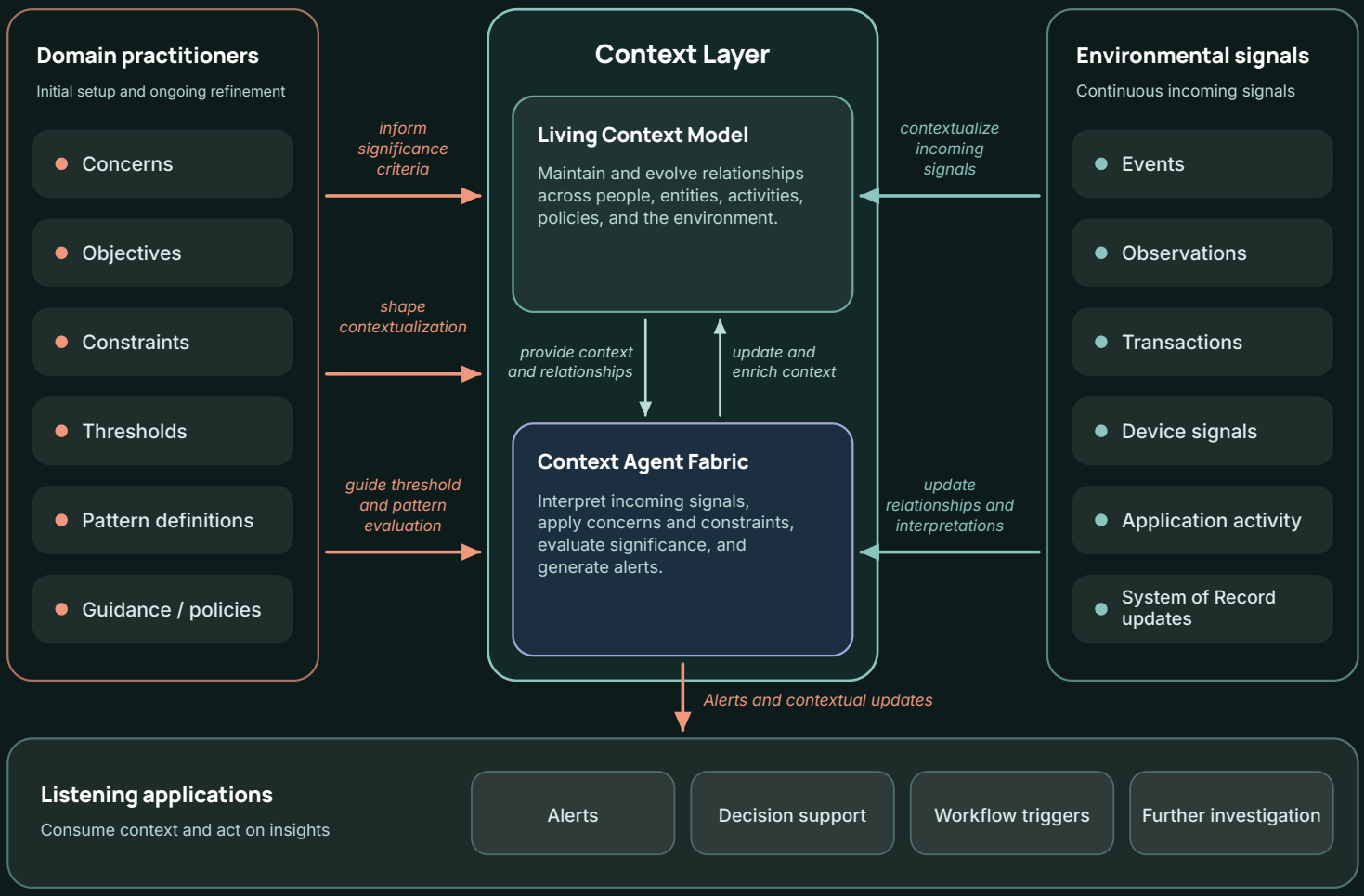
Section 08

Integrating human concerns and constraints

> The Context Layer must be accountable to the domain it serves, not merely responsive to the information sources it observes.

Figure 4

Integrating human concerns and constraints



Domain concerns and constraints help determine what matters; incoming signals are interpreted in that context to generate meaningful alerts and support responsive action.

The Context Model must not be limited to signals originating in the environment being observed. It must also incorporate the concerns, objectives, constraints, and judgments of the domain practitioners and other responsible actors whose decisions the Context Layer is intended to support.

These considerations are not supplementary information to be applied after observations have been interpreted. They are integral to the contextualization process itself, establishing what matters, why it matters, and under what circumstances changes in the observed environment warrant attention.

A Context Layer must therefore be grounded in two complementary dimensions: an evolving understanding of the environment being observed, and an evolving understanding of the domain-specific concerns and constraints that determine the significance of what is observed.

This grounding cannot be supplied by the signal sources alone. It requires engagement with the practitioners and other responsible authorities whose expertise, objectives, and judgments define the domain in which the system operates.

These concerns and constraints may be expressed through formal policies, operational requirements, expert assessments, statements of concern, or other human-authored artifacts. Rather than treating these artifacts as supplementary documents to be retrieved when an agent encounters a particular question, the Context Layer incorporates them into the living Context Model, maintaining their relationships to the observations, events, and interpretations to which they apply.

The significance of an emerging condition is therefore not determined by observations alone. It arises from the relationship between accumulating evidence and the domain-specific concerns and constraints against which that evidence is evaluated.

A particular pattern may be inconsequential in one operating context but demand immediate attention in another. The criteria governing that judgment may themselves change as objectives, circumstances, and human understanding evolve. The Context Layer must maintain these evolving criteria alongside the observed environment, so that changes in either can alter the significance of emerging patterns.

These criteria can also govern the event-driven behavior of the Context Layer itself. Thresholds, pattern classifications, and other conditions of significance may be established through domain-specific design or introduced through human-authored artifacts that express the concerns of particular practitioners or application users. Once incorporated into the Context Model, these criteria can be evaluated as new events are interpreted, enabling the Context Layer to recognize and signal when an emerging condition warrants attention.

In this way, domain practitioners do more than supply interpretive guidance. They can help determine what the Context Layer actively monitors for and when its evolving contextual understanding should initiate further reasoning or intervention.

This does not mean treating observations, interpretations, expert judgments, and policies as interchangeable. Their different origins, authority, uncertainty, and applicability must remain distinguishable within the Context Model. Preserving these distinctions is essential to maintaining a dependable contextual representation.

By integrating domain-specific concerns and constraints into the same living Context Model as the observed environment, the Context Layer establishes the foundation for recognizing when accumulated evidence becomes significant and for informing how that significance should be addressed.

> **The Context Layer must be accountable to the domain it serves, not merely responsive to the information sources it observes.**

Section

09

From request-driven reasoning to context-driven intervention

Most agentic applications today operate within a predefined problem space. They respond to questions, automate established processes, or exercise judgment at decision points already anticipated within an operational workflow.

A Context Layer can substantially improve these applications by providing the integrated contextual understanding that human participants have traditionally supplied. In this role, it enables agents to reason more effectively about known problems without repeatedly reconstructing the underlying context.

But the full potential of a living Context Model extends beyond improving established processes.

By continuously maintaining the relationships among emerging signals, accumulated evidence, and the concerns and constraints that govern their significance, the Context Layer can recognize when a developing condition warrants attention even before a question has been explicitly formulated.

Gaps, contradictions, and emerging patterns can become visible as they develop. Individually inconclusive signals may accumulate and interact until their combined significance crosses a contextually determined threshold.

Such thresholds need not be fixed numerical boundaries.

Such thresholds need not be fixed numerical boundaries. They may reflect the strength of accumulated evidence, the significance of changing relationships, the degree of uncertainty, the consequences of inaction, or the priorities and constraints established by human judgment.

This creates the possibility of fine-grained, event-driven decision support in which changes in context can themselves initiate agentic reasoning.

The Context Layer identifies conditions that may warrant investigation or intervention, while agentic applications undertake the task-specific reasoning necessary to determine what response is appropriate. Human-authored constraints and authority boundaries continue to govern how those responses may be formulated or executed.

The distinction is important. The Context Layer need not become an all-purpose decision-making system. Its responsibility is to maintain the contextual understanding necessary to recognize significant developments and provide the basis for an informed response.

In this architecture, the evolving environment becomes more than a source of information for answering predefined questions. **It can become the source of new questions, emerging concerns, and opportunities for intervention.**

This moves agentic systems beyond the automation of established workflows toward the dynamic identification and interception of emerging problems and opportunities.

Section

10

An architectural separation with broader implications

The distinction between the Context Window and the Context Layer is therefore more than a matter of how information is organized or retrieved.

It represents a separation of architectural responsibilities.

The Context Window provides a bounded, task-specific representation of information for inference. The Context Layer maintains the larger, evolving contextual fabric from which that representation is drawn.

The Context Layer also extends beyond supporting inference on demand. By integrating environmental signals with human concerns, constraints, and judgments, it provides the foundation for recognizing when evolving conditions have become significant enough to warrant new reasoning or action.

This architecture does not depend on a particular inference model, retrieval technique, or knowledge representation. Its value remains relevant as models become more capable and Context Windows become larger, because the responsibility for maintaining shared contextual understanding persists independently of any individual inference.

The broader proposition is that contextual understanding should be treated as an enduring infrastructure capability, not something reconstructed independently within every agentic interaction.

In closing

A Context Window is a temporary assembly of information for inference. A Context Layer is the living fabric of relationships that makes that information meaningful, recognizes when its significance changes, and enables agentic systems to respond.

Context is a fabric, not a list.





AI needs context.
Context needs AI.

Get in touch

b.ratnadeep@gmail.com · cliniciancontext.com

Specified · Prototype next · Not in clinical use

© 2026 Clinical Context AI