



ISSN 23939729

CLAWS

No. **146**

2026

MANEKSHAW PAPER

**The Model Should Be the
Smallest Part of the System:
Hierarchy-Aware Axiomatic-
Agentic Intelligence for
Military Decision Support**

T Eswara Rao

CENTRE FOR LAND WARFARE STUDIES

Field Marshal Sam Hormusji Framji Jamshedji Manekshaw, better known as Sam “Bahadur”, was the 8th Chief of the Army Staff (COAS). It was under his command that the Indian forces achieved a spectacular victory in the Indo-Pakistan War of 1971. Starting from 1932, when he joined the first batch at the Indian Military Academy (IMA), his distinguished military career spanned over four decades and five wars, including World War II. He was the first of only two Field Marshals in the Indian Army. Sam Manekshaw’s contributions to the Indian Army are legendary. He was a soldier’s soldier and a General’s General. He was outspoken and stood by his convictions. He was immensely popular within the Services and among civilians of all ages. Boyish charm, wit and humour were other notable qualities of independent India’s best known soldier. Apart from hardcore military affairs, the Field Marshal took immense interest in strategic studies and national security issues. Owing to this unique blend of qualities, a grateful nation honoured him with the Padma Bhushan and Padma Vibhushan in 1968 and 1972 respectively.



**Field Marshal SHFJ Manekshaw, MC
1914-2008**

CLAWS Occasional Papers are dedicated to the memory of Field Marshal Sam Manekshaw

The Model Should Be the Smallest Part of the System:

Hierarchy-Aware Axiomatic-Agentive Intelligence for Military Decision Support

T Eswara Rao



Centre for Land Warfare Studies
New Delhi



Editorial Team : CLAWS

ISSN : 23939729



Centre for Land Warfare Studies

RPSQ Complex, Parade Road, Delhi Cantt, New Delhi 110010

Phone +91-11-25691308 Fax: +91-11-25692347

Email: landwarfare@gmail.com, website: www.claws.co.in

CLAWS Army No.33098

The Centre for Land Warfare Studies (CLAWS), New Delhi, is an independent Think Tank dealing with national security and conceptual aspects of land warfare, including conventional & sub-conventional conflicts and terrorism. CLAWS conducts research that is futuristic in outlook and policy-oriented in approach.

CLAWS Vision: To be a premier think tank, to shape strategic thought, foster innovation, and offer actionable insights in the fields of land warfare and conflict resolution.

CLAWS Mission: Our contributions aim to significantly enhance national security, defence policy formulation, professional military education, and promote the attainment of enduring peace.

© 2026, Centre for Land Warfare Studies (CLAWS), New Delhi.

Disclaimer: The contents of this paper are based on the analysis of materials accessed from open sources and are the personal views of the author. The contents, therefore may not be quoted or cited as representing the views or policy of Government of India, or the Ministry of Defence (MoD), or the Centre for Land Warfare Studies.

Published in Bharat by



Sabre & Quill Publishers, New Delhi, India

www.sabreandquill.com/sabreandquill@gmail.com

Contents

• Abstract.....	5
• Introduction	6
Existing Approaches and Motivation.....	7
• The Common Starting Point.....	10
• Four Structural Limits	12
Context Saturation.....	12
Schema Proliferation	12
Retrieval is Not Analysis	13
Security Cannot Live in a Prompt.....	13
• Design Principles.....	14
• Why Two Systems Rather Than One.....	14
• The Intelligence Assessment Engine	16
Layer 0: Hierarchy Aware Security Governance.....	18
Layer 1: The Reasoning Gate.....	19
Layer 2: The Deterministic Enrichment Engine.....	19
Layer 3: The Agent Orchestrator.....	20
Layer 4: Specialist Agents and Multi-Lens Analysis.....	20
Conflicting and Contested Data.....	21
The Institutional Knowledge Agent	21
Layer 5: The Response Calibrator	22
Layer 6: Human Feedback, Persona and Organisational Memory	23
• The Geospatial Intelligence Agent.....	25
What the Agent Does	25
The Nineteen Tool Problem.....	26
The Two-Mode Redesign.....	26
Natural Language to SQL, and How It Scales	28

Operator Trust: Visible SQL, Honest Failures, Guided Dead Ends.....	29
Conversation, Context, and Follow-Up.....	30
• Discussion	31
Security by Architecture, Not by Prompt.....	31
An Explainable Hierarchy	31
Why Layer 6 Matters Most	32
Running Inside a Closed Network	33
The Model Should be the Smallest Part of the System	34
• Operational Status and Early Use	35
• Limitations and Future Directions	36
• Conclusion	37
• References	39

The Model Should Be the Smallest Part of the System: Hierarchy-Aware Axiomatic-Agentic Intelligence for Military Decision Support

Abstract

Most military experiments with large language models begin from one template. A model turns a commander's question into a database query, fetches rows, and writes a summary of them. The pattern demonstrates well and fails predictably. It breaks when a result set runs to thousands of records, when many different intelligence sources must be read at once, when the question calls for judgement rather than retrieval, and when access must follow a chain of command. This paper sets out a design doctrine for trustworthy military AI, drawn from the rebuild of a real system, which the author calls *hierarchy-aware axiomatic agentic intelligence*. Its central claim is simple: artificial intelligence should not establish truth; it should reason over truth established by deterministic computation and authoritative data. Authoritative databases and fixed algorithms decide what is true. Bounded AI agents then reason over that settled material inside a security boundary computed from the command hierarchy, so that the model cannot mishandle data it was prevented from ever seeing. The doctrine is realised as two complementary systems. The Intelligence Assessment Engine produces deep analytical products through a seven-layer pipeline that adds a parallel institutional knowledge agent and a human review layer for inference-time correction without retraining. The Geospatial Intelligence Agent answers

direct operational and geospatial queries in seconds and hands the heavier questions off to the first. The paper explains why two systems rather than one, sets out four structural limits of the conventional approach and the principles drawn from them, details both systems and their behaviour under conflicting data, doctrinal tension and closed network deployment, and argues that the model should be the smallest part of a trustworthy military system rather than its centre.

***Author note.** This paper describes two working systems in generalised form. The agentic assessment framework is referred to throughout as the Intelligence Assessment Engine (IAE), and the lighter geospatial system as the Geospatial Intelligence Agent (GIA). These are working designations used for this paper. The programmes carry official names which, together with specific data schemas, geographic sectors, named formations and computing infrastructure, are withheld for security*

Introduction

The volume of operational data available to modern military headquarters continues to grow faster than staff capacity to analyse it. Sensors, intelligence reports, electronic surveillance systems, geospatial products and operational databases generate far more information than can be manually fused within available decision timelines.

The wish to decide and act faster than an opponent is old. What has changed is the sheer scale of data that now stands between the question and the answer. In an era when digitisation has made data itself a primary operational asset, the bottleneck is no longer collection. It is the conversion of data into understanding at the speed of command. Any future military AI architecture must therefore be evaluated not only by the quality of its answers, but also by its ability to operate securely, transparently and at a tempo relevant to command decision-making.

Reaching for artificial intelligence to close this gap is natural, and reaching first for a large language model is just as natural, because it is the most visible form of the technology. That first instinct deserves scrutiny when it comes to high-stakes command. This paper argues that a trustworthy system should fetch verified facts and compute over them, rather than generate fluent text and hope it is right. This contribution comes from the build side. It reports what happened when a working intelligence assistant was built the conventional way, observed under operational demands, and then rebuilt around a different understanding of where truth should come from.

Existing Approaches and Motivation

Large language models have rapidly become central to decision-support systems through retrieval-augmented generation (RAG), natural language interfaces to structured databases, and, more recently, agentic AI frameworks. These approaches have significantly improved access to information and automated many reasoning tasks. However, they address different aspects of the decision-support problem and generally optimise language understanding, retrieval, or autonomous task execution rather than the design of a trustworthy military intelligence architecture.

Natural language-to-SQL systems translate operator questions into database queries and express the returned results in natural language. Such systems are effective when the requirement is to retrieve specific information from structured databases, but they remain fundamentally retrieval-oriented. They do not perform deterministic analytical processing across multiple intelligence sources or produce structured intelligence assessments beyond the retrieved records.

Retrieval-Augmented Generation extends this capability by grounding language models in external knowledge before

generation, reducing hallucinations and improving factual consistency. More recent agentic AI architectures coordinate multiple specialised agents capable of planning, tool use, and collaborative reasoning across complex tasks. Parallel developments in explainable AI, human-centred AI, and zero-trust security architectures have likewise emphasised transparency, human oversight, and security as essential properties of high-consequence AI systems.

These developments represent substantial progress, yet they leave several challenges unresolved for military intelligence systems. Existing architectures typically place language models at the centre of analytical reasoning, while deterministic computation, access governance, and institutional knowledge act as supporting components. Similarly, hierarchical command relationships, governed human feedback, and explainable analytical workflows are often treated as independent capabilities rather than parts of a unified architectural framework.

The architecture presented in this paper builds upon these developments while addressing their limitations. Rather than treating the language model as the centre of the system, it combines hierarchy-aware security governance, deterministic analytical processing, institutional knowledge grounding, specialist agent collaboration, and human-governed organisational memory into a unified architecture. The contribution of this paper is therefore not another agentic framework or retrieval architecture. It is an architectural doctrine that deliberately confines artificial intelligence to bounded reasoning over deterministic truth established by authoritative operational data, secured through hierarchy-aware access governance, and refined through human authority.

That doctrine emerged through the practical redesign of an operational intelligence assistant rather than from theory alone. That redesign did not produce one system. It produced two, because operators ask two fundamentally different kinds of questions, and a single design cannot serve both well.

Consider a divisional intelligence staff attempting to assess rising activity across its area of responsibility. The GSO-2 first asks the Geospatial Intelligence Agent (GIA) to plot incidents from the previous ninety days. Within seconds, hotspots, movement corridors and changes in activity density appear on the map. The officer then asks a deeper question: "Generate a comprehensive assessment of enemy activity across the division over the last year." The request is handed to the Intelligence Assessment Engine (IAE), which retrieves authorised data, computes deterministic analytics, deploys specialist agents and produces a structured assessment identifying emerging threats, shifts in deployment patterns, intelligence gaps and recommended areas for further collection.

The commander receives not only a conclusion, but also the evidence trail, source confidence, contradictory indicators and the reasoning path that produced the assessment. If the staff identifies an incorrect assumption or local nuance, the assessment can be corrected and the correction preserved for future analyses. The architecture described in this paper was developed to support precisely this transition from data collection to trusted decision support. Together, the two systems are designed to move a headquarters from information availability to decision readiness.

Most requests are immediate and concrete, such as showing an activity on a map or counting events in a sector, and these need speed and certainty. A smaller set is analytical, such as assessing an area of responsibility over a year, and these require deep reasoning across multiple intelligence sources. The two systems are the

Intelligence Assessment Engine (IAE), which produces analytical intelligence products, and the Geospatial Intelligence Agent (GIA), which answers direct operational and geospatial queries. They share a common philosophy, which this paper terms Hierarchy-Aware Axiomatic-Agentic Intelligence, and cooperate through a structured hand-off mechanism that routes each question to the architecture best suited to answer it.

The sections that follow begin with the conventional natural language-to-SQL architecture from which both systems evolved, examine the four structural limitations that made that approach unsuitable for operational intelligence, derive the design principles that emerged from those failures, explain why the work naturally divided into two complementary systems, and then describe each architecture in detail before concluding with the role of human governance in trustworthy military AI. The thread running through the paper can be stated in a single sentence: artificial intelligence should not establish truth; it should reason over truth established by deterministic computation and authoritative data. That statement is less a slogan than a design doctrine, and the architecture presented in the following sections is the practical expression of that principle.

The Common Starting Point

The architectural concepts discussed above were not developed in isolation. They emerged from practical experience with one of the most common contemporary approaches to AI-enabled decision support: natural language interfaces to operational databases. Both systems therefore trace back to a single first attempt that followed this familiar template. A commander question was read for intent. Salient features of the question were matched, by approximate string matching, against a taxonomy of database tables. The matched tables decided which schema to query. The

model wrote the query. The database returned rows. A second model call read those rows and produced prose. Two model calls bracketed a deterministic database step, one to write the query and one to read the result.

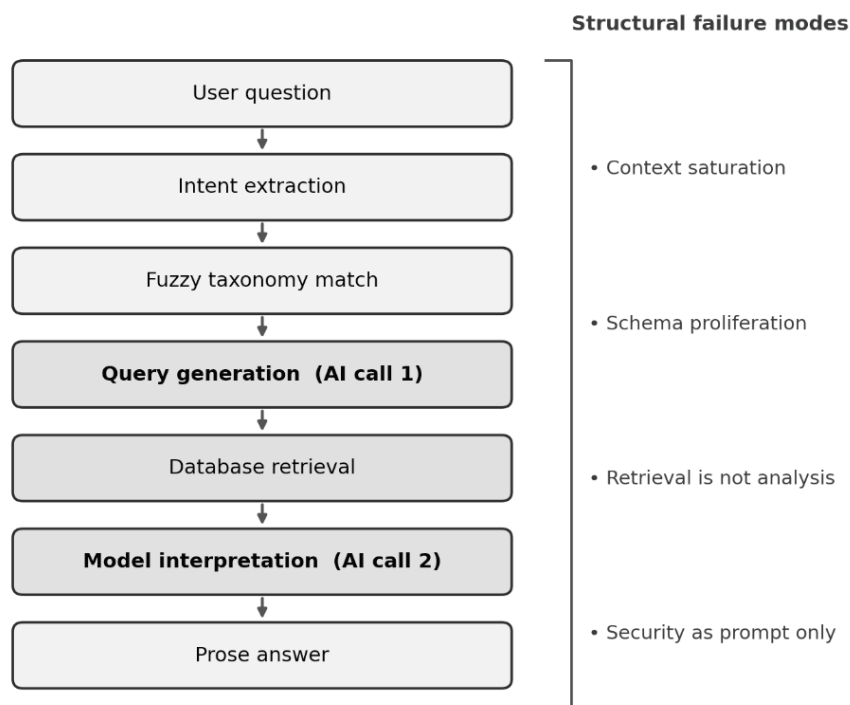


Figure 1: The conventional natural language to SQL pipeline, with the four structural failure modes that defeated it at operational scale

Source: Author's own

On narrow questions it behaved exactly as intended. A request to show enemy activity in a named sector over the previous month resolved cleanly. The taxonomy sent enemy activity to the enemy activity store, daily reporting to the situation report store, and electronic intelligence to the ELINT store. A query was written and run, a manageable set of rows came back, and the model summarised them. The demonstration convinced people, and it stayed convincing for exactly as long as the questions stayed small.

THE MODEL SHOULD BE THE SMALLEST PART OF THE SYSTEM

Four Structural Limits

The system did not fail because of a defect or a weak prompt. It failed because the template rests on four assumptions that do not hold for intelligence work. Each limit is structural, which is to say that no amount of prompt tuning removes it.

Context Saturation

The first failure arrived the moment a question stopped being narrow. A request to produce a detailed assessment of an area of responsibility for the past year is one sentence, but the evidence behind it is not. A query of that scope returned several thousand events across several intelligence domains. The database fetched them faithfully. The model could not hold them. There is no context window large enough to let a model reason reliably over that volume. Relevant detail is diluted, early evidence is effectively forgotten by the time the late evidence is read, and answer quality falls as input grows. The architecture had made the model the holder of the entire evidence base, and that responsibility does not scale.

Schema Proliferation

The second failure built up slowly and was a maintenance failure. Every new intelligence source meant new taxonomy entries, new prompt fragments and new query logic, and each addition interacted with the others. The work needed to add the next source did not stay flat. It rose, because each new source had to be reconciled against everything already present. There were also quieter traps. One live field carried a legacy name that implied one kind of intelligence while holding another, and the prompt had to spell out the truth so the model did not misread it. A design that ties the model tightly to the schema cannot take on new sources

gracefully, and an intelligence system that cannot take on new sources is out of date the day a new sensor is fielded.

Retrieval is Not Analysis

The third failure was the most important, and it was a failure of concept rather than of code. The template answers one question well: what is in the database. A commander rarely asks that. A commander asks what changed, what the threat is, what is trending, what is unusual, and what needs command attention. None of these are retrieval questions. Retrieval can report that forty-one engagements occurred along an axis last month. It cannot, by itself, say that this marks an escalation, that the escalation lines up with a shift in electronic activity, or that the shape of it resembles a preparatory phase. A system tuned to fetch rows will answer the wrong question well.

"Intelligence is not a database query. That single realisation is the hinge on which the whole redesign turned."

Security Cannot Live in a Prompt

The fourth failure concerned access. Armies run on hierarchy, and intelligence is compartmented along it. A brigade user may not require to see corps or peer formation intelligence, and no user should be able to reach up or sideways through the order of battle. The obvious first answer is to tell the model which formations a user may see. That answer must be rejected. An instruction to a model is a request, not a guarantee. It can be talked around, confused, or ignored under awkward input. Security written as a prompt is not security. The only sound design is one in which the model never receives unauthorised data at all, so that it cannot leak, summarise or invent what it was never handed. That property cannot be added downstream of retrieval. It has to be enforced before retrieval happens.

Design Principles

The four limits translate into six working rules that govern both systems. They read as instructions because that is how they behaved in practice. Each one closed off a class of design that had already been tried and found wanting.

- **Authoritative Data Before AI.** Fact starts in trusted operational databases, never in a generative model.
- **Computation Before Interpretation.** Counts, trends, correlations, and scores are calculated by fixed algorithms before any model is called.
- **Analysis Before Narrative.** A system meant to assess is built to produce an assessment, not to fetch and paraphrase rows.
- **Security Before Retrieval.** Authorisation boundaries are computed before any data is read, so the model is never shown data outside the user remit.
- **Human Governance Before Learning.** A system adapts only through validated human input, and never retrain itself on its own.
- **Many Lenses Over One Reasoner.** Distinct analytical tasks are handled by distinct specialists, the way an intelligence staff already divides its work.

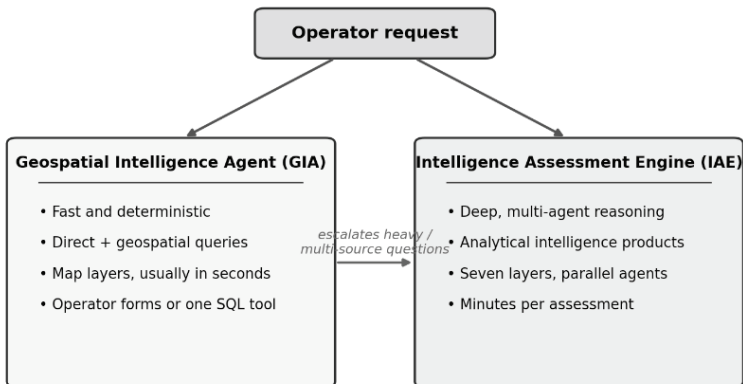
Why Two Systems Rather Than One

The single most important design decision was to stop trying to serve every question with one mechanism. Two systems answer two kinds of questions, and conflating them is what made the original tool unreliable.

Most of what an operator asks is immediate and concrete. Show this activity on a map. Draw a heatmap of the last week. Count

events in a sector. These questions want speed and certainty. They do not want a paragraph of reasoning, and they certainly do not want to wait. A smaller but vital share of questions are analytical. Assess my area for the year. Project the threat along this axis. Correlate what the different sources are saying. These want precisely the deep, many-sided reasoning that would be wasteful for a simple map request.

Trying to serve both with one design fails in both directions. Running a multi-step, multi-minute assessment pipeline to plot a single layer is heavy overkill, and overhead is the enemy of trust and speed at the tactical edge. Running a fast single query tool against an analytical question produces a thin, shallow answer that a commander cannot rely on. So the work is split.



Shared philosophy: authoritative data, deterministic computation, and bounded AI under hierarchy-aware governance.

Figure 2. Two systems built on one philosophy. The Geospatial Intelligence Agent answers direct queries quickly. The Intelligence Assessment Engine produces deep analytical products. The geospatial agent recognises heavier, multi-source questions and hands them off to the assessment engine

Source: Compiled by Author

The Geospatial Intelligence Agent handles direct operational and geospatial queries and renders them, usually in seconds. The

Intelligence Assessment Engine produces analytical intelligence products through the seven-layer pipeline described in the next section. The two are not rivals and not duplicates. They are a tiered design that puts the right amount of machinery behind each kind of question. The geospatial agent recognises when a request has outgrown its scope, for example, a question that asks to summarise across all sources, and hands it off to the assessment engine rather than attempting a fragile imitation of it. The operator therefore gets the fast path by default and the deep path only when the question genuinely needs it. Both systems sit behind the same hierarchy-aware access governance, so the separation is one of depth and speed, never of who is allowed to see what.

The table below places the approach against the two designs it grew out of. The rows name properties explained in the sections that follow, and together they are the contribution: not any single capability, but the combination of all of them in one governed pipeline.

Table 1. The conventional approaches and the architecture proposed here, by capability

Capability	Natural language to SQL	Generic agentic AI	This architecture
Retrieval of stored facts	Yes	Yes	Yes
Security computed before retrieval	No	Partial	Yes
Deterministic analytics over the data	No	Partial	Yes
Multi lens specialist analysis	No	Yes	Yes
Institutional knowledge grounding	No	Partial	Yes
Human governed memory	No	Partial	Yes
End to end audit trail	Limited	Limited	Yes

Source: Compiled by Author

The Intelligence Assessment Engine

The Intelligence Assessment Engine is the deep system. It has

seven layers, numbered zero to six. Layer zero settles who may see what. Layers one to five turn an authorised question into a finished product. Layer six captures human judgement so the engine improves with use without learning on its own. The order is the point. Security and deterministic computation sit upstream of every piece of model reasoning, which is what makes the engine defensible rather than merely capable. A parallel institutional knowledge agent grounds its output against doctrine; procedures and the formation own past assessments.

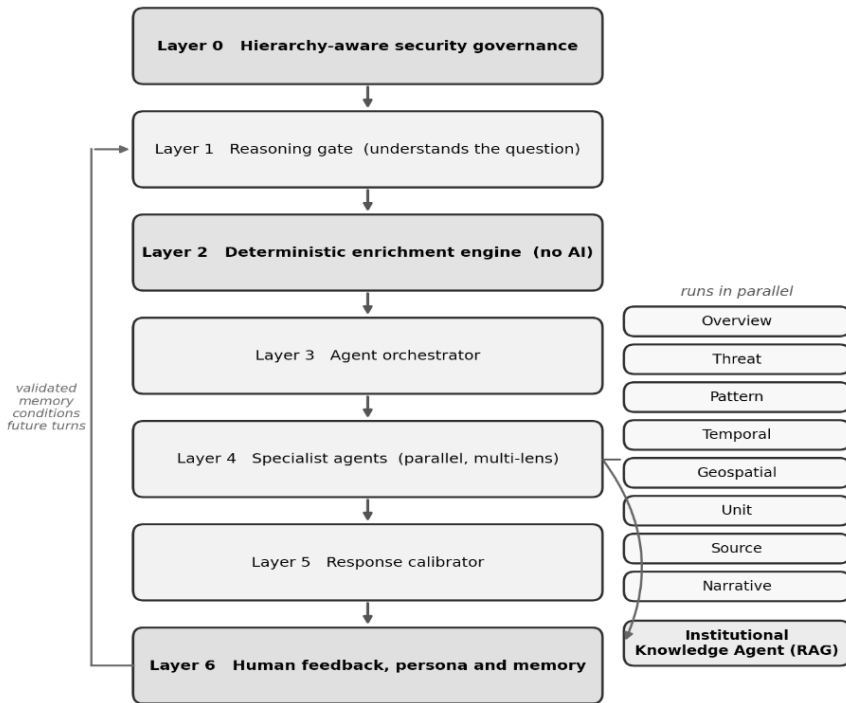
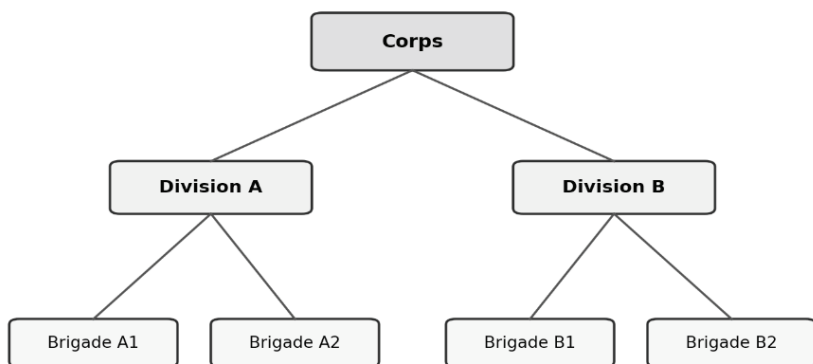


Figure 3. The seven layer Intelligence Assessment Engine. Specialist agents run in parallel over precomputed evidence. An institutional knowledge agent runs alongside them, grounding the draft in doctrine, procedures and past assessments. Validated memory from Layer 6 conditions later turns without retraining the model

Source: Compiled by Author

Layer 0: Hierarchy Aware Security Governance

Every request begins with the questioner, not the question. The user identity resolves to a unit. The unit is placed in the command hierarchy. From that position, the engine computes the set of formations the user may query, which is the user's own formation and everything subordinate to it, but never parent or peer formations. Only after this set is fixed is any retrieval allowed. A query that reaches outside the set is refused before a single row is read and before the model is called at all. This is stronger than role-based access control, which attaches permissions to roles and does not natively capture the parent-to-child structure of an order of battle. Because authorisation is computed from position in the hierarchy rather than from a fixed table of role permissions, it is better called hierarchy-aware access governance. Its value is structural. The model cannot mishandle data it was prevented from ever seeing. The same governance protects the geospatial agent described in Section 7.



A user sees own formation and everything below it, never parent or peer formations.

Figure 4. Downward visibility in the access governance layer. A brigade user sees only that brigade, a division user sees the division and its brigades, and a corps user sees the whole structure. Upward and sideways requests are refused before retrieval

Source: Compiled by Author

Layer 1: The Reasoning Gate

The first model call retrieves nothing. Its only job is to understand the authorised question and to pull out the time period, the intelligence domains involved, the depth of analysis wanted, and the sector and formation in scope. A request for a detailed assessment of an area of responsibility over the last year becomes a structured plan: period of one year, all domains, comprehensive depth, scope limited to the authorised area. No data is touched. The layer reasons about the question, and in doing so turns a loose sentence into a precise plan for retrieval and analysis.

Layer 2: The Deterministic Enrichment Engine

This is the layer that sets the design apart, and it contains no artificial intelligence at all. Once the authorised data has been retrieved, fixed algorithms turn it into analytical evidence. The categories produced include basic statistics, period-over-period comparison, temporal patterns, geospatial distribution, force strength measures, source reliability scoring, cross-domain correlation, discipline-specific analysis for electronic intelligence and for situation reports, narrative coverage, appearance and co-occurrence measures, unit attribution, axis analysis and source grading. Every number a commander will later read, every count, trend, hotspot, correlation, and reliability score, is computed here.

The rule is plain, and it matters: *numbers are computed, not generated*. A language model is a poor calculator over thousands of records and offers no guarantee that its arithmetic is right. A fixed routine is exact and repeatable. By moving all the quantitative work out of the model and ahead of it, the engine lays down a factual base that the reasoning above cannot corrupt. This layer, more than any other, is what earns the word axiomatic. It is where truth is fixed before interpretation starts. These figures are not hidden

inside the pipeline. The operator can inspect them directly through an intelligence dashboard, described in Section 8.2.

Layer 3: The Agent Orchestrator

The second model call decides how much analysis the question deserves and which specialists to send in. A short question may need only a couple of agents. A full annual assessment may need the entire set. The orchestrator works like a commanding officer assigning tasks. It picks the agents and gives each one a tailored brief, made up of a shared baseline of common statistics plus the specialised enrichment and the representative events that bear on that agent particular mission. It does not analyse. It allocates.

Layer 4: Specialist Agents and Multi-Lens Analysis

The specialist agents run in parallel, each looking at the same picture through a different lens. A representative set includes an overview agent that writes the executive picture, a threat agent that grades threat by sector, a pattern and anomaly agent that surfaces trends, escalation and the unusual, a temporal agent that finds peak periods and tempo, a geospatial agent that locates hotspots and movement corridors, a unit agent that reads enemy composition and disposition, a source correlation agent that cross-checks findings across disciplines, and a narrative agent that mines the text of reports for qualitative detail.

The intent is to imitate an intelligence staff rather than to replace one. A real staff does not ask a single analyst to do threat assessment, pattern detection, geospatial work, temporal work, source validation, and narrative reading all at once. It splits these among specialists who later converge. Asking one model to do everything at once is a convenience of implementation, not a reflection of how the work is organised. No agent invents fact. Each

one reasons over the precomputed evidence handed to it by Layer 2, which keeps every agent anchored to the same settled material.

Conflicting and Contested Data

A real intelligence picture is rarely clean. Reports contradict one another, sensors degrade, feeds drop, and an adversary actively tries to deceive. The deterministic enrichment of Layer 2 does not assume the incoming data is true. It assumes only that the data is authoritative in provenance, which is a different thing, and it carries that distinction forward through source reliability scoring and source grading. Each record enters analysis with a computed measure of how far it should be trusted, derived from the grading of its source and its agreement or disagreement with other reporting on the same event.

Two mechanisms then handle contradiction explicitly. Within Layer 2, co-occurrence and correlation measures expose where sources agree and where they diverge, so a single unreliable feed cannot quietly dominate a count. Within Layer 4, the source correlation agent exists precisely to cross-check findings across disciplines and to surface disagreement rather than average it away. When electronic, imagery, and human reporting tell different stories about the same activity, the system does not silently pick a winner. It presents the divergence, weighted by reliability, and marks the affected conclusion as lower confidence. Suspected deception is treated the same way, as a reliability and correlation signal that lowers confidence and is flagged for the analyst, because the judgement that a feed is being spoofed is a command judgement, not one the system should make on its own.

The Institutional Knowledge Agent

Alongside the specialist agents runs a retrieval agent of a different kind. Where the specialists reason over operational data,

this agent reads the institutional library, which is wider than doctrine alone. It covers standing operating procedures, operational doctrine, tactics, techniques and procedures, threat templates, and the formation's own validated past assessments. It works by retrieval-augmented generation, finding the passages relevant to the current question and using them in two ways. First, it supplements the assessment with the institutional context a seasoned analyst would bring, so that a pattern is read against the procedures and precedents that should govern the response to it. Second, and more usefully, it cross-checks the draft. If a specialist conclusion sits at odds with a standing procedure, or if a recommended action runs against doctrine or against how the formation has handled a similar situation before, the agent flags the conflict rather than letting it pass silently. This fuses three different kinds of input, operational data, institutional knowledge and human memory, into a single assessment, which is richer than retrieval over documents alone. Everything the agent contributes is grounded in named, citable sources, so it can be traced back to the procedure or assessment it came from.

Layer 5: The Response Calibrator

A final model call gathers the specialists and the institutional knowledge agent into one coherent product: an executive summary, a threat picture, an assessment of enemy disposition, activity trends, identified intelligence gaps, a commander assessment, and recommended actions. The calibrator retrieves nothing and recomputes nothing. It composes validated findings into a single readable assessment, playing the part of the senior analyst who turns the staff's separate contributions into the brief the commander actually receives.

The calibrator also decides how to present disagreement, and this is where the institutional knowledge agent earns its place.

When an operational recommendation from the specialists contradicts a standing procedure, the calibrator does not quietly choose one side and hide the other. It surfaces the tension as an explicit, flagged item in the product, setting the operational indicator and the doctrinal position side by side, for example, noting that the activity pattern argues for one course while standing procedure advises another. The resolution is left to the commander, with both the evidence and the doctrine in view. The same is done for low confidence conclusions arising from conflicting data. The principle is that the system makes tensions visible and leaves their resolution to human authority, rather than manufacturing a false consensus.

The architecture is designed to degrade gracefully when data is incomplete, delayed or unavailable. Missing sources do not halt the assessment process. Instead, affected agents operate on the available evidence and explicitly report reduced confidence levels and coverage limitations. The final assessment identifies these gaps rather than attempting to conceal them. This ensures that commanders remain aware not only of what is known, but also of what remains uncertain.

Layer 6: Human Feedback, Persona and Organisational Memory

The finished product is not the end of the pipeline. The analyst reviews it, corrects it, may ask for it to be regenerated, and can shape how it is written. Two mechanisms make this concrete.

The first is an editable persona. The operator can state who they are, how they want the assessment written, and what they care about most, for example, a brigade intelligence officer who wants brief, threat-first reporting with map layers, or a planner who wants longer prose with explicit assumptions. This persona overrides the default system prompt for the current question. If the operator chooses to keep it, it is stored and applied to later

questions as well. The stored guidance is not fixed in stone. The operator can open it, edit it, and switch each entry *active* or *inactive*, so guidance that suited one operation can be set aside for the next without being lost.

The second is a visible correction and regeneration loop. After submitting recommendations, the operator can regenerate the assessment and see what changed, rather than receiving a fresh answer with no way to tell what the feedback did. Validated corrections and preferences are written into organizational memory, a governed store that conditions future answers.

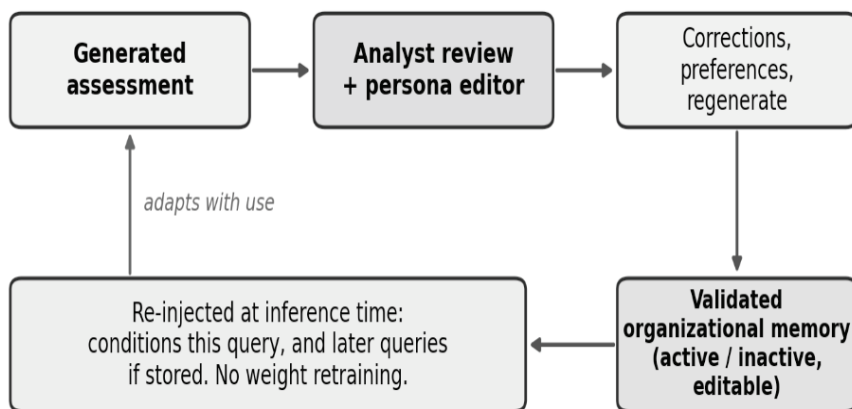


Figure 5. The human review loop. Corrections and persona edits become validated, toggleable memory that is re-injected at inference time. The model weights are never retrained

Source: Author's own

The distinction at the heart of this layer is that it is *not* autonomous retraining. There are no uncontrolled learning loops and no model drift introduced through use. Validated guidance is re-injected at inference time, as conditioning on the prompt and on retrieval, so the engine adapts to the operator and to the operational picture through use, while the underlying model and the data

beneath it stay untouched. In effect, the operator trains the behaviour of the engine at the point of use, without altering its weights and without the risks that autonomous retraining would bring. Human authority over the product is, and remains, final.

The Geospatial Intelligence Agent

The Geospatial Intelligence Agent is the fast system. It answers the direct operational and geospatial questions that make up most of an operator's daily traffic, and it renders them, usually as a map layer, in seconds. Its own evolution is instructive, because it reaches the same lesson as the assessment engine from a different direction: handing a model too much freedom is the source of failure, and removing that freedom in the right places is what restores trust.

What the Agent Does

Behind the agent sits a suite of nineteen analytical tools. These include plotting a layer, building a heatmap, computing statistics and charts, animating a timeline, detecting threat corridors, assembling a composite situation report, gap analysis, pattern of life, sensor what-if studies, multi-source fusion, report generation, regional statistics, buffer analysis, spatial clustering, sensor coverage, proximity analysis and convex hulls. Each tool is a precise, deterministic capability. Together they cover most of what an operator wants to see on a map in a hurry.

The language model serves a different role from the tools themselves. Once a map layer, chart, or analytical product has been generated, a model call examines the output and highlights observations that may not be immediately apparent to the operator. It does not merely describe what is visible on the screen. Rather, it attempts to identify patterns, concentrations, anomalies, correlations or operational implications that would otherwise require deliberate analysis by a trained user. For example, after

plotting activity on a map, the model may point out an emerging hotspot, an unusual shift in activity along an axis, gaps in sensor coverage, or a pattern that resembles known threat behaviour. In this role, the model functions as an analytical aide rather than a rendering engine. The map layer is produced deterministically by the selected tool; the model contributes interpretation of the generated output, helping the operator move more quickly from visualisation to understanding.

The Nineteen Tool Problem

In its first form, the agent exposed all nineteen tools to the language model and asked it to do three hard things at once: read the operator intent, pick the right tool, and fill in between three and ten parameters per call, all over a production schema of more than two hundred tables. That is far too much surface for any model to handle reliably. Tool calls failed often. Parameters were guessed wrongly. Operators lost trust in the output, which is fatal, because a map an operator does not trust is worse than no map at all. The failure was not the fault of any single model. It was the fault of asking a probabilistic component to make a long chain of brittle choices, where one wrong link produced a wrong picture.

The Two-Mode Redesign

The fix was to split the agent cleanly into two modes, divided by a rule a human can see at a glance. If the message begins with a slash, it is an operator-driven command. If it does not, it is a natural language question. The two modes have completely different failure surfaces, which is the whole point.

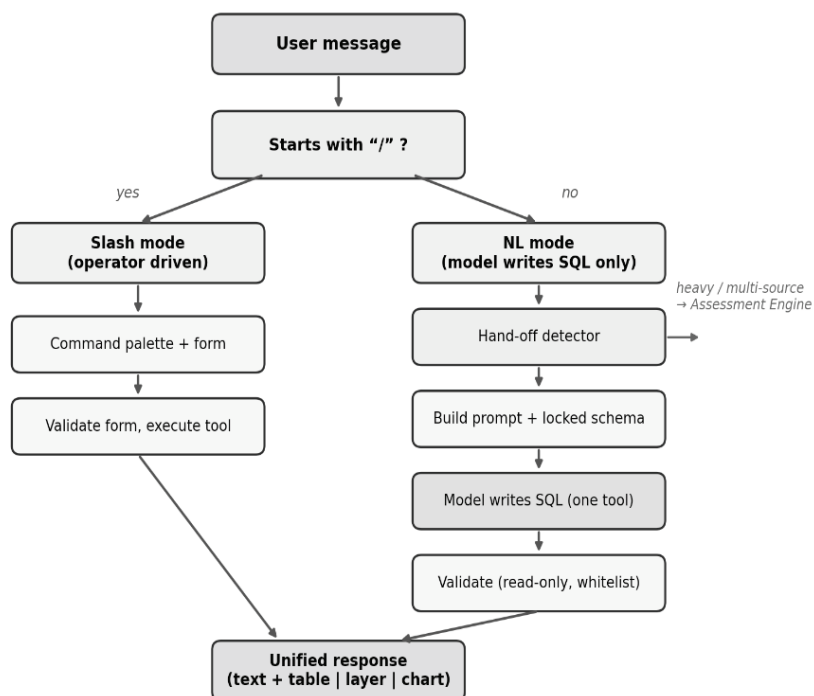


Figure 6. The Geospatial Intelligence Agent in two modes. A leading slash routes to operator-driven tools through a form. Plain language routes to a single SQL writing path. Heavy or multi-source questions are handed off to the Assessment Engine

Source: Author's own

In slash mode, the operator, not the model, chooses the tool. Typing a slash opens a command palette that lists the nineteen tools and filters as the operator types. Selecting one, say a heatmap, renders a form built from that tool parameter specification, with the right fields, sensible defaults, and validation. The operator fills the form and runs it. The request goes straight to the tool with no model in the loop, so the only thing that can go wrong is form validation, which is visible and fixable. Every tool that used to be model-callable still exists and is still reachable, but now through a deterministic form rather than through a guess.

In natural language mode, the model is given exactly one tool: write a read-only SQL query. Nothing else. It no longer chooses among nineteen capabilities. Reducing the model to a single tool, with a single class of error, is what turned an unreliable component into a dependable one.

Nineteen tools could in principle add their own friction under stress, so the palette is built to keep cognitive load low. Natural language is the default path, which means an operator never has to remember a command unless they want one. The tools are not presented as a flat list of nineteen. They are grouped by intent into plotting and visualisation, statistics and charting, spatial analysis, and fusion and reporting, and the palette filters by typing and surfaces an operator's most-used command first. In practice, the operator reaches for a slash command when they know exactly what they want and want it run deterministically, and falls back to plain language otherwise, so the two modes share the load rather than forcing a choice up front.

Natural Language to SQL, and How It Scales

The natural language path is a single model call per turn, wrapped in deterministic checks on both sides. Before the model runs, a cheap heuristic looks for questions shaped like assessment work, the ones that ask to summarise across all sources or that are heavy and narrative. Those are handed off to the Assessment Engine rather than attempted. If the question is in scope, the prompt is built from a short block of rules, the schema description, and a handful of worked examples that show good query patterns, including the joins to master tables that a correct query needs. The model writes the query. A validator then checks it, enforcing read-only access and the authorised table set, so a query that names a table outside that set is rejected before it runs. The query executes. If the result carries a geometry column, it is drawn as a map layer.

Otherwise, it is shown as a table, and a short summarising call turns the numbers into a sentence or two of plain answer. If the database returns an error, the error, the query, and the relevant slice of schema are fed back once for a corrected attempt, and the retry budget stops there.

The agent works over a curated working set of tables rather than the whole production schema. At present that set numbers a few dozen, chosen so that the full schema description fits inside a single prompt: the live operational sources, their master lookup tables, the order of battle hierarchy, and the user and access tables. This is a deliberate current scope, *not* a ceiling of the design. The agent already operates over more tables as they are curated in, and extending it to a much larger schema is an engineering exercise rather than a redesign. Once the schema grows past what a single prompt can hold, table selection moves to semantic retrieval, where only the tables relevant to a given question are pulled into the prompt for that turn. Adding a table in the present design needs an edit to one configuration file and a single refresh call, with no code change and no restart. The principle stays the same as the schema grows. Only the means of presenting the schema to the model changes, from a fixed list to retrieval, and that is the additional engineering the larger scale requires.

Operator Trust: Visible SQL, Honest Failures, Guided Dead Ends

Reliability is necessary but not sufficient. Operators also need to understand how the system arrived at an answer. The redesign therefore made the agent reasoning process visible rather than exposing technical implementation details. For each response, the operator can inspect what the system attempted to do, including the data sources consulted, the categories of information searched, the analytical tools invoked, and the sequence of processing steps that produced the result. The objective is not to expose database

queries or internal mechanics, but to provide sufficient transparency for the operator to judge whether the system followed a sensible path to the answer.

The access boundary is also visible, so the operator knows that the response reflects only information they are authorised to access. When a query returns no results, the agent does not stop at an empty response. Instead, it proposes concrete next steps, such as widening the time window, expanding the geographic area, or refining the search criteria. When a failure occurs, the system explains what stage failed, such as retrieval, analysis, visualisation, or interpretation, and provides options to modify and retry the request rather than hiding the problem behind a generic apology.

This visibility serves a second purpose. It allows operators to verify that the agent searched the expected information domains and followed a reasonable analytical process. A system that can explain its actions, acknowledge its limits, and expose its reasoning path is far more likely to earn trust under operational conditions than one that simply produces an answer without explanation.

Conversation, Context, and Follow-Up

The agent is not a one-shot query box. It holds the context of the exchange, so an operator can ask a question, see the answer and the map, and then refine it in the next turn without restating everything. It also proposes useful follow-up questions and can answer them, which helps an operator who knows the ground better than the query language to explore a picture rather than guess at a single perfect question. The same interface handles three kinds of request without the operator having to declare which is which: ordinary conversation, questions answered from the databases with results plotted on the map, and mixed exchanges that move between the two within a single conversation. This conversational behaviour sits on top of the deterministic

machinery, not in place of it. Every database answer still passes through the same validator and the same access boundary, so holding a conversation never loosens the guarantees that the rest of the design depends on.

Discussion

Security by Architecture, Not by Prompt

The most important property shared by both systems is also the easiest to miss, because it is defined by what the model never sees. In a conventional design, the model sits between the user and the data, and access control is wrapped around it, often after the model has already touched the data. Here the hierarchy engine sits between the user and the data, and the model sits below an already bounded set. The model cannot leak division intelligence to a brigade user, cannot summarise a peer formation reporting, and cannot invent the contents of a compartment it was never shown, because the offending data was excluded before retrieval. The guarantee is structural rather than behavioural, and structural guarantees are the kind that hold up under adversarial pressure.

An Explainable Hierarchy

Because fact is fixed before interpretation, every conclusion in an assessment can be unwound through a clear chain. Source records sit at the base. The deterministic metrics computed from them form the next level. The specialist assessments that reason over those metrics, checked against doctrine by the retrieval agent, form the third. The commander consolidated brief sits at the top. Any line in the brief can be traced down to the agent that wrote it, to the metric that supported that agent, and to the record the metric came from. For a system meant to support command decisions, and to be accredited for that role, this traceability is not a feature. It is a precondition.

This chain is not left implicit. The assessment engine exposes it through an intelligence dashboard, where the operator sees what was computed and consumed as visual infographics rather than raw tables. Each analytical lens has its own view, presenting the enrichment categories that fed it and the charts built from them, so an operator can look behind a threat picture or a unit assessment and see the figures it rests on. A separate audit view lists every specialist agent, whether it ran or was held back by the orchestrator, which enrichment categories it drew on, what it concluded, and the confidence it reported. Because every line of the final brief maps to a row in that view, and every row to a computed enrichment and the records beneath it, the dashboard turns the explainability chain from a design claim into something an operator can open and inspect. That matters as much for trust as for accreditation, since an analyst is far more willing to act on an assessment whose workings can be seen than on one delivered as an opaque paragraph.

Why Layer 6 Matters Most

It would be easy to treat the human review layer as a courtesy bolted on at the end. In the field it is the layer that decides whether the engine is usable at all. An assessment produced from general patterns can be wrong in ways that only a person standing on the ground can catch. It may read a piece of terrain incorrectly, because the model has no feel for that ground. It may draw an inference that is technically supported by the data and yet wrong in context, because it does not know what the formation knows. It may recommend an action that doctrine, or local experience, rules out. None of these are arithmetic errors that Layer 2 can prevent. They are errors of interpretation, and interpretation is exactly where a probabilistic component is fallible.

Layer 6 is the mechanism that catches and corrects them. The analyst who knows the ground edits the interpretation, and that

correction does not evaporate when the session ends. Held as validated, toggleable memory and re-injected at inference time, it shapes the next assessment, so the engine grows more suited to its formation and its terrain the more it is used. The operator is, in a real sense, training the engine at the point of use, conditioning its behaviour without touching its weights and without the risks that autonomous retraining would bring. A system that improves this way stays auditable, because every piece of learned behaviour is a human-authored, inspectable memory entry rather than an opaque shift in a model. The combination of correction, persona and governed memory is what turns a clever assistant into one a formation can come to rely on.

Running Inside a Closed Network

None of this assumes a connection to the open internet, and that is deliberate. Both systems run wholly inside the trusted operational network, on a self-contained H200 GPU node attached to the same network. The models are served locally, the databases are the unit's own, and the institutional library is a local corpus of doctrine and past assessments. Nothing leaves the enclave. The design also uses more than one model rather than one. A smaller, faster model handles the lighter steps, such as understanding the question and summarising a result, while a larger model is reserved for the heavier analytical reasoning. This keeps response times workable on organic compute, and it is a practical expression of the principle that runs through the whole architecture: that the largest model should be called only where it is genuinely needed. A design that instead depended on an external service or an internet search would be unusable in exactly the conditions a field formation cares about most.

The Model Should be the Smallest Part of the System

The common assumption in current AI design puts the model at the centre, with data, tools and controls arranged around it. The experience recorded here points the other way. Truth came from authoritative data. Analysis came from deterministic computation. Grounding came from retrieval against doctrine. Security came from hierarchy. Durable learning came from validated human judgement. The model's legitimate role, understanding the question, reasoning over evidence it did not generate, and composing the result, is real but narrow. Each system is better understood not as an AI system that draws on data, but as an intelligence system that uses AI for the parts of the task to which AI is actually suited. Put as a single claim: artificial intelligence should not establish truth; it should reason over truth established by deterministic computation and authoritative data.

It is on that basis that the shared paradigm is named *hierarchy-aware axiomatic agentic intelligence*. The word order is the argument in miniature. Hierarchy comes first because access governance computed from the order of battle is not an implementation detail but the outermost layer of the design. Axiomatic comes next because authoritative data and deterministic analytics, not a model, establish factual truth. Agentic comes last because specialist agents perform only bounded reasoning over that settled material, under doctrinal grounding and final human oversight. Defined fully: an architecture in which a security boundary computed from the command hierarchy precedes all retrieval, authoritative data and deterministic analytics establish factual truth, and specialist AI agents then reason within that truth under institutional grounding and human authority.

Operational Status and Early Use

The architecture is not a paper proposal. Both systems have been implemented and exercised against operational data, drawn from a decision support database of more than two hundred tables spanning the live intelligence sources, their master lookups, the order of battle and the access tables. The assessment engine deploys up to eight specialist agents in parallel and computes its evidence base through around sixteen deterministic enrichment categories before any agent reasons, while the geospatial agent answers direct questions, holds a conversation, and plots activity on the map over the same governed data.

In field use, a full assessment is produced in about sixty to one hundred and eighty seconds, with the spread driven almost entirely by how much data Layer 2 must process for the question asked: a focused query over a short window returns quickly, while a year-long, all-source assessment sits at the upper end. These figures are achieved on enterprise-grade GPU infrastructure comprising a single node equipped with eight NVIDIA H200 accelerators. The deployment uses a tiered model strategy, with a larger reasoning model of approximately 120 billion parameters distributed across four GPUs for specialist-agent analysis and assessment synthesis, while smaller models responsible for routing, retrieval, summarisation and embeddings are hosted on the remaining GPUs. The design currently supports approximately twenty to thirty concurrent users. Concurrency is constrained less by retrieval and more by the assessment engine itself, since a single comprehensive assessment may invoke multiple specialist agents in parallel, each consuming substantial inference resources. The figures reported here are therefore not intended as benchmarks, but as evidence that the architecture can operate at a tempo suitable for command decision support while handling realistic operational workloads. The architecture is model agnostic and may be

deployed on smaller or larger compute footprints with corresponding trade-offs in concurrency, response time, and analytical depth

Limitations and Future Directions

Both systems confine the language model, but neither removes it. Understanding the question, agent-level reasoning, and narrative synthesis are still probabilistic, so the design is best described as a hybrid: a deterministic, hierarchy-governed foundation with a bounded agentic layer above it. The honest claim is not that probabilistic reasoning has been eliminated, but that it has been confined to interpretation and barred from ever being the source of fact. The quality of the enrichment engine also sets a ceiling on the quality of analysis, since the agents can only reason as well as the metrics computed for them, which puts a premium on the rigour of Layer 2.

Several directions follow naturally. The enrichment engine could be extended with validated predictive analytics, kept separate from the agents so that forecasts stay auditable. The geospatial agent path to larger schemas, through semantic retrieval of relevant tables, is a concrete next step as the number of sources grows. The institutional knowledge agent invites deeper work on how strongly a doctrinal conflict should weigh against an operational conclusion, and how best to present that tension to a commander. The tiered model could widen to further intelligence disciplines and to cross-formation federation, under the same hierarchy governance. And the memory layer raises real questions about how validated human feedback should be represented, weighted and audited over time without ever crossing into autonomous retraining. Each of these can be pursued without disturbing the central ordering of the architecture, which is its main contribution.

Field evaluation will judge the two systems by different yardsticks, because they do different work. For the geospatial agent, the measures are accuracy of the generated query against a curated test suite, the share of questions answered without a failed attempt, the precision and recall of the handoff decision, and time to a rendered layer. For the assessment engine, the measures are analyst-rated quality and usefulness of the product, the completeness of its audit trail, the rate at which the institutional knowledge agent correctly surfaces real doctrinal conflicts without crying wolf, and the speed at which the memory layer brings a product into line with formation preferences. Cutting across both is the test that matters most in the field, which is operator trust and sustained adoption, and a standing requirement that every access control check pass without exception, since a single hierarchy breach would undo the rest. These criteria, rather than model benchmark scores, are what should decide whether the design earns a place in the command post.

Conclusion

The route from a natural language to SQL assistant to two complementary, hierarchy-governed, axiomatic agentic systems was not planned. It was forced, one principle at a time, by the failure of the conventional approach against real questions, and by the realisation that one design cannot serve both a quick map request and a year-long assessment. The lessons travel beyond any single system. Context windows do not scale to operational evidence, so the model must not be the holder of the evidence. Schema-coupled designs do not absorb new sources, so the model must be decoupled from the schema. Retrieval is not analysis, so a system meant to assess must be built to assess rather than to fetch. Security written as instruction is not security, so authorisation must be computed before the model is ever involved, which is what lets us say plainly that the model cannot mishandle data it was

prevented from ever seeing. And an assessment built from general patterns can fail on the ground, so a human who knows that ground must be able to correct it and make the correction stick.

Taken together, these lessons move the model out of the centre of the design and put authoritative data, deterministic computation, doctrinal grounding, hierarchy and human judgement in its place. They reduce to a single design doctrine, the one this paper has returned to throughout: artificial intelligence should not establish truth; it should reason over truth established by deterministic computation and authoritative data. The future of trustworthy military AI may belong less to larger models than to architectures that know precisely what a model should, and should not, be allowed to do.

For more than a century, military advantage has come from extending reach: first through platforms, then through networks, and later through sensors and data. The next advantage may come from something less visible but more decisive: the ability to convert vast quantities of information into trusted decisions faster than an adversary. If that proves true, the decisive question for military AI will not be how powerful a model is, but whether commanders trust the system enough to act on its assessments. Trust is not a model capability. It is an architectural property. The systems described here are one attempt to build that trust into the design itself.

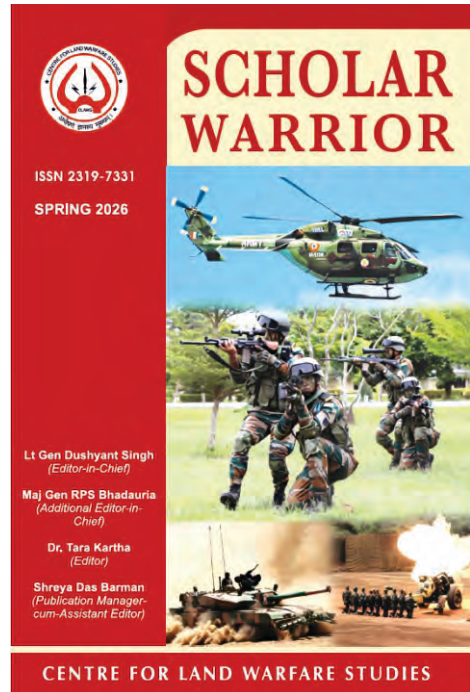
“The future battlefield is unlikely to reward the force with the largest model. It is more likely to reward the force that can transform data into trusted decisions fastest. In that contest, trust will be as important as intelligence, and architecture will matter as much as algorithms”.

References

1. Alberts, D. S., Garstka, J. J., & Stein, F. P. (1999). *Network Centric Warfare: Developing and Leveraging Information Superiority* (2nd ed.). CCRP Publication Series.
https://www.dodccrp.org/files/Alberts_NCW.pdf.
2. Cebrowski, A. K., & Garstka, J. J. (1998). *Network-Centric Warfare: Its Origin and Future*. *Proceedings*, 124(1), U.S. Naval Institute.
<https://www.usni.org/magazines/proceedings/1998/january/network-centric-warfare-its-origin-and-future>.
3. Hu, V. C., Ferraiolo, D., Kuhn, D. R., Schnitzer, A., Sandlin, K., Miller, R., & Scarfone, K. (2014). *Guide to Attribute Based Access Control (ABAC) Definition and Considerations* (NIST Special Publication 800-162). National Institute of Standards and Technology. DOI: <https://doi.org/10.6028/NIST.SP.800-162>
PDF: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-162.pdf>.
4. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. *Advances in Neural Information Processing Systems (NeurIPS 2020)*.
<https://arxiv.org/abs/2005.11401>.
5. Amershi, S., Weld, D. S., Vorvoreanu, M., Fournery, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S. T., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R., & Horvitz, E. (2019). *Guidelines for Human-AI Interaction*. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*.
DOI: <https://doi.org/10.1145/3290605.3300233>.
6. Gunning, D., & Aha, D. W. (2019). *DARPA's Explainable Artificial Intelligence (XAI) Program*. *AI Magazine*, 40(2), 44–58.
DOI: <https://doi.org/10.1609/aimag.v40i2.2850>.
7. U.S. Army Training and Doctrine Command. (2018). *TRADOC Pamphlet 525-3-1: The U.S. Army in Multi-Domain Operations 2028*.
<https://adminpubs.tradoc.army.mil/pamphlets/TP525-3-1.pdf>.
8. Wang, L., Ma, Y., Zhang, Q., Tian, S., Zhang, Y., Wang, J., et al. (2024). *A Survey on Large Language Model-Based Autonomous Agents*. *Frontiers of Computer Science*, 18(6).
DOI: <https://doi.org/10.1007/s11704-024-40231-1>.

9. OpenAI. (2025). *A Practical Guide to Building Agents*. <https://cdn.openai.com/business-guides-and-resources/a-practical-guide-to-building-agents.pdf>.
10. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": *Explaining the Predictions of Any Classifier*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. <https://doi.org/10.1145/2939672.2939778>.
11. Anthropic. (2024). *Building Effective Agents*. <https://www.anthropic.com/engineering/building-effective-agents>.
12. Microsoft Research. (2024). *Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks*. <https://arxiv.org/abs/2411.04468>.

SUBSCRIBE NOW



SUBSCRIPTION RATES

IN INDIA

☐ Rs.500/- per copy

☐ Rs.1000/- Annual Subscription (2 issues)

SAARC COUNTRIES

☐ US \$ 15 per copy

OTHER COUNTRIES

☐ US \$ 20 per copy

TO SUBSCRIBE SEND YOUR REQUEST TO



Centre for Land Warfare Studies (CLAWS)
RPSO Complex, Parade Road, Delhi Cantt, New Delhi - 110010
Tel: +91-11-25691308
• Fax: +91-11-25692347 • Army: 33098
E-mail: landwarfare@gmail.com
www.claws.co.in

Modern military headquarters generate enormous volumes of operational data, yet the challenge increasingly lies not in collecting information but in converting it into trusted understanding at the speed of command. The Model Should Be the Smallest Part of the System: Hierarchy-Aware Axiomatic-Agent Intelligence for Military Decision Support examines this challenge through the practical evolution of an AI-enabled military intelligence assistant. The paper argues that Artificial Intelligence should not establish truth, but reason over truth established through authoritative data and deterministic computation. It proposes a hierarchy-aware architecture in which security is enforced before data retrieval, specialist agents perform bounded analytical tasks, institutional knowledge grounds reasoning, and human authority governs learning and correction. The paper presents two complementary systems for direct operational queries and deeper intelligence assessment, offering a framework for secure, explainable, and human-governed AI-enabled military decision support.

...



Major T Eswara Rao was commissioned into the Corps of Electronics and Mechanical Engineers (EME) and is currently holding the appointment of GSO-1 (System Analyst) at the Directorate General of Information Systems (DGIS). A technology enthusiast, he has extensive experience in Artificial Intelligence, Large Language Models (LLMs), Automatic Speech Recognition (ASR), Power BI, software engineering, and AI-enabled decision support systems. He is also a military drone pilot with over 300 flying hours. Additionally, he is an award-winning filmmaker, with his travel documentary "20 Days on Wheels" receiving the Best Documentary Award at the National Tourism Short Film Festival 2023. He is also the author of a book titled Something Different.

The Centre for Land Warfare Studies (CLAWS), New Delhi, is an independent Think Tank dealing with contemporary issues of national security and conceptual aspects of land warfare, including conventional & sub-conventional conflicts and terrorism. CLAWS conducts research that is futuristic in outlook and policy oriented in approach.

CLAWS Vision: To be a premier think tank, to shape strategic thought, foster innovation, and offer actionable insights in the fields of land warfare and conflict resolution.

CLAWS Mission: Our contributors aim to significantly enhance national security, defence policy formulation, professional military education, and promote the attainment of enduring peace.

Website: www.claws.co.in

Contact us: landwarfare@gmail.com

