

NOÈSE: An Instrumented I3A Architecture for Deterministic Computation and Self-Evolving Agent Genesis at Inference Time

Malik Kouhen*

Independent Researcher — NOÈSE / I3A, France

ORCID: 0009-0007-7763-4086

`malko.i3a@proton.me`

January 2026

Abstract

Contemporary artificial intelligence systems are predominantly based on large monolithic models optimized for statistical performance and scale. While effective for many tasks, such architectures exhibit structural limitations when requirements of interpretability, deterministic computation, role specialization, and controlled system evolution arise.

This work introduces **Noese**, an instrumented I3A (Intelligent Artificial Autonomous Agents) architecture designed to address these limitations through calibrated multi-agent organization, explicit role contracts, and fully traceable inference-time execution. In Noese, intelligence emerges from the coordination of specialized agents operating under strict constraints, rather than from a single opaque model. All agent interactions are logged, measured, and auditable, enabling systematic analysis and reproducibility.

A central contribution of this work is a model compatibility and calibration framework that evaluates heterogeneous AI models and assigns them to agent roles based on operational criteria such as stability, instruction discipline, numerical precision, constraint compliance, and resource cost. This transforms model selection into a measurable engineering step rather than an empirical heuristic.

Deterministic computation within a generative system is demonstrated through AURELION, a dedicated mathematical agent operating under strict procedural constraints. Across multiple experiments, AURELION produces reproducible numerical outputs verified against reference implementations, showing that deterministic reasoning can be embedded within an AI-driven multi-agent architecture.

Beyond computation, NOÈSE supports inference-time self-evolution. When cognitive gaps are detected during execution, the system triggers controlled generation of new specialized agents. Large-scale experiments show bounded and non-random agent genesis, while a controlled replay study demonstrates that small algorithmic modifications can selectively suppress or enable agent creation without collapsing internal dynamics.

Taken together, these results position NOÈSE as a concrete step toward instrumented, deterministic, and self-evolving artificial intelligence architectures, where emergence is an observable and constrained process rather than a metaphor. The framework outlines a viable path toward robust, non-anthropomorphic artificial intelligence systems grounded in engineering principles and empirical verification.

*Also known as “MalKo” (I3A / NOÈSE project).

Contents

1	Introduction	4
1.1	Context and motivation	4
1.2	Contributions	4
1.3	Paper structure	5
2	Architectural Positioning and Design Rationale	6
2.1	Limits of Monolithic Architectures	6
2.2	Why Instrumented Multi-Agent Systems	6
2.3	Scope boundaries and current limitations	7
2.4	Positioning of the I3A Paradigm	7
3	Limitations of Monolithic AI Systems	8
3.1	Structural limits under reliability and determinism constraints	8
3.2	Requirements for viable multi-agent alternatives	8
4	I3A v2: Architecture, Roles, and Instrumentation	9
4.1	Operational definition of I3A	9
4.2	Role taxonomy and agent contracts	9
4.3	Cognitive cycle instrumentation	10
4.3.1	Log structure and traceability	10
4.3.2	Metrics (FDC or equivalent) and interpretation	10
4.4	Reproducibility hooks	11
5	Model Compatibility & Calibration Tool	12
5.1	Problem statement: not every model is structure-compatible	12
5.2	Compatibility criteria	12
5.3	Calibration workflow and scoring	12
5.4	Role assignment mapping (model to agent role)	13
6	Deterministic Computation Case Study : " AURELION Math Expert "	14
6.1	Scientific context (collaboration use-case)	14
6.2	Deterministic procedure	14
6.2.1	Constraint system (what AURELION is allowed / forbidden to do)	14
6.2.2	Verification against reference (Python or equivalent)	14
6.3	Results and accuracy	15
6.4	Resource accounting (time/energy/cost framing)	16
7	MIMESIS: Emotional Dynamics Protocols Applied to AI	17
7.1	Protocol overview (P1–P4)	17
7.2	Instrumentation: logs and metrics	17
7.2.1	Emotion/cognition signal extraction	17
7.2.2	Stability and drift measures	17
7.3	Results by protocol (selected figures only)	18
7.4	Cross-protocol synthesis	19
8	Self-Evolving at Scale: Noésis-Driven Agent Genesis	20
8.1	Triggering mechanism (cognitive gap detection)	20
8.2	Agent genesis statistics	20
8.3	Quality considerations (current limits and constraints)	21
8.4	Examples of generated agents (selected)	22
8.5	Controlled replay experiment on self-evolving agent genesis	22

9 Archiving & Memory: Semantic Condensation	24
9.1 Why memory matters for self-improvement	24
9.2 Archival pipeline	24
9.3 Current implementation vs roadmap	24
9.4 Archival pipeline	25
10 Open Research, Collaboration, and Continuation	26
10.1 Access to extended materials	26
10.2 Research dialogue and collaboration	26
10.3 Ongoing evolution	26
11 Conclusion	27
11.1 Inference as an engineered process	27
11.2 Calibration and behavioral reliability	27
11.3 Determinism under constraint	27
11.4 Controlled self-evolution at inference time	27
11.5 Memory as a disciplined substrate	27
11.6 Perspective	27
A Appendix A: Minimal Log Schema (Example Excerpt)	28
B Appendix B: AURELION Verification Snippet	29
Glossary	30
Acknowledgments	31

1 Introduction

1.1 Context and motivation

Contemporary artificial intelligence research has largely converged toward monolithic model architectures, where increasing scale, parameter count, and data exposure are the primary drivers of performance. While this paradigm has yielded remarkable advances, it also introduces structural limitations when systems are required to exhibit interpretability, internal specialization, reproducibility, and controlled adaptation. In such architectures, reasoning, computation, control, and generation remain entangled within a single statistical process, making it difficult to isolate behaviors, verify internal mechanisms, or guarantee reliable outcomes under constrained conditions.

The I3A (Intelligent Artificial Autonomous Agents) paradigm emerges from the observation that intelligence need not be concentrated within a single model to be effective. Instead, intelligence can arise from the coordinated interaction of multiple specialized agents, each operating under explicit roles, constraints, and evaluation criteria. By decomposing cognition into distinct functional responsibilities—such as deterministic computation, signal modulation, orchestration, synthesis, and system evolution—I3A architectures aim to replace opaque internal complexity with structured cooperation and traceable decision pathways.

A critical motivation for this approach lies in the growing demand for instrumentation and reproducibility in artificial intelligence systems. As AI increasingly participates in scientific computation, decision support, and autonomous processes, the ability to measure, audit, and reproduce internal operations becomes essential. Without explicit logging, role separation, and constraint enforcement, claims of reliability or correctness remain difficult to substantiate. I3A architectures address this gap by design, embedding measurement, logging, and verification mechanisms directly into the cognitive cycle.

Noèse is developed within this context as an I3A-based system that treats inference not as a single opaque event, but as an instrumented process in which specialized agents cooperate, generate measurable artifacts, and adapt structurally when necessary. The motivation of this work is therefore not to pursue artificial intelligence as an imitation of human cognition, but to engineer a non-anthropomorphic, verifiable, and evolvable intelligence framework whose internal dynamics can be observed, evaluated, and progressively refined.

1.2 Contributions

This work presents Noèse as a fully instrumented I3A system and makes five primary contributions to the study and engineering of multi-agent artificial intelligence architectures. Rather than introducing isolated techniques, these contributions collectively define a coherent framework in which calibrated agent specialization, deterministic computation, experimental protocolization, and structural self-evolution are integrated within a single inference-time process. Each contribution is empirically grounded and supported by measurable artifacts, logs, and reproducible outputs.

C1 Instrumented I3A architecture.

We introduce and formalize an I3A architecture in which cognitive functions are explicitly decomposed into specialized autonomous agents. The architecture is fully instrumented, enabling traceable inference cycles, role-specific constraints, and observable internal dynamics.

C2 Model compatibility and calibration for role assignment.

We propose a systematic calibration process for evaluating heterogeneous AI models and assigning them to appropriate agent roles based on operational criteria such as stability, instruction discipline, numerical precision, constraint compliance, and computational cost.

C3 Deterministic mathematical computation within a generative system.

Through the AURELION Math Expert agent, we demonstrate that deterministic numerical computation with verifiable accuracy can be reliably embedded within an I3A framework, achieving reproducible results validated against reference implementations.

C4 Experimental protocols for emotional dynamics in AI.

We introduce MIMESIS, a set of four structured experimental protocols designed to apply and measure emotional dynamics within artificial agents, producing stable, protocol-dependent signatures that can be analyzed and reused for system calibration.

C5 Self-emergent engineering and archival memory.

We demonstrate inference-time self-evolution through the autonomous creation of new specialized agents in response to detected cognitive gaps, supported by a distributed archival subsystem that enables structured memory formation and future retrieval.

1.3 Paper structure

The remainder of this paper is organized as follows. Section 2 presents the I3A architectural framework underlying Noèse, detailing agent roles, instrumentation, and the structure of the cognitive cycle. Section 3 introduces the model compatibility and calibration process used to assign AI models to specialized roles within the architecture. Section 4 reports a deterministic computation case study based on the AURELION Math Expert agent, followed by Section 5, which describes the MIMESIS experimental protocols and their application to emotional dynamics in artificial agents. Section 6 examines inference-time self-evolution through autonomous agent genesis and the associated archival memory subsystem, before concluding with a discussion of scope, limitations, and future directions.

2 Architectural Positioning and Design Rationale

Architectural choices in artificial intelligence systems are never neutral: they implicitly determine what can be measured, controlled, verified, and ultimately trusted. In recent years, the dominance of large monolithic models has led to impressive empirical performance across many application domains. This success, however, has come at the cost of architectural opacity. As reasoning, computation, control, and generation are increasingly intertwined within a single statistical process, it becomes difficult to isolate responsibilities, enforce explicit constraints, or inspect internal mechanisms. These characteristics represent structural challenges when artificial intelligence systems are expected to operate in scientific, computational, or autonomous contexts where transparency and control become essential.

2.1 Limits of Monolithic Architectures

Monolithic artificial intelligence architectures consolidate multiple cognitive functions within a single model optimized through end-to-end statistical learning. This design has proven effective for tasks favoring pattern recognition and large-scale generalization. However, as system requirements expand toward interpretability, accountability, and controlled autonomy, several intrinsic architectural limits become apparent.

A central limitation lies in functional entanglement. When numerical computation, symbolic manipulation, decision control, and generative behavior coexist within the same parameter space, internal responsibility boundaries are blurred. This makes it difficult to isolate sources of error, inspect intermediate reasoning steps, or apply targeted constraints. As a result, system behavior is often observable only at the final output level, reducing diagnostic precision and limiting corrective intervention.

In addition, growth and specialization in monolithic architectures remain structurally external to inference itself. Extensions typically require retraining or fine-tuning phases that operate outside the inference-time execution path. This separation constrains the system’s ability to evolve incrementally while preserving traceability and architectural stability.

These observations do not undermine the empirical effectiveness of monolithic models within their intended scope. Rather, they motivate the exploration of alternative architectural paradigms in which functional separation, explicit roles, and internal observability are treated as first-class design principles.

2.2 Why Instrumented Multi-Agent Systems

Multi-agent systems provide a natural architectural response by enabling functional decomposition and role specialization. Instead of forcing a single model to simultaneously handle heterogeneous cognitive responsibilities, multi-agent approaches distribute these functions across autonomous components, each operating under a well-defined role. This separation establishes clearer responsibility boundaries and supports modular system design.

However, multiplicity alone is insufficient. Without explicit instrumentation, multi-agent systems risk reproducing opacity at a higher level of complexity. Interactions between agents may become as difficult to interpret as internal activations within a monolithic model if they are not systematically measured, constrained, and recorded.

Instrumentation transforms multi-agent systems into engineered cognitive structures. By embedding logging, metrics, and verification hooks directly into agent interactions, an instrumented architecture enables continuous observation of inference-time behavior. Decision pathways can be traced, intermediate results audited, and agent contributions evaluated with fine granularity. This shifts multi-agent systems from experimental constructs toward auditable and verifiable computational frameworks.

Furthermore, instrumented multi-agent architectures provide a robust substrate for controlled system evolution. Because agents operate under explicit roles and interfaces, new capabilities can be introduced incrementally through the addition or refinement of agents without destabilizing existing components. This property is essential for architectures designed to evolve over time while preserving reliability, interpretability, and traceability.

Table 1: Architectural scope and core contributions of the Noèse I3A framework.

Architectural Area	Implemented principles and capabilities
Architecture	Instrumented I3A framework with explicit agent roles, structured cognitive cycles, and traceable inter-agent interactions.
Calibration	Model compatibility evaluation and calibration process enabling controlled assignment of AI models to specialized agent roles.
Deterministic Computation	Verified deterministic numerical computation through a dedicated mathematical agent, supported by reproducible logs, tables, and reference comparisons.
Experimental Protocols	Structured experimental protocols enabling the measurement and analysis of emotional and cognitive dynamics within artificial agents.
Self-Evolving Mechanisms	Inference-time generation of new specialized agents in response to detected cognitive gaps, with integration metadata and role constraints.
Archival Memory	Distributed archival subsystem enabling systematic classification, storage, and future retrieval of interaction traces and generated artifacts.

2.3 Scope boundaries and current limitations

Noèse demonstrates an instrumented I3A architecture capable of deterministic computation, experimental protocolization, and inference-time self-evolution. The real-time integration of newly generated agents during an ongoing inference cycle is currently limited to controlled reinjection phases rather than continuous dynamic reconfiguration.

Although the archival memory subsystem enables structured storage and future consultation of interaction traces, its use as an active driver for automated self-improvement is deliberately deferred. These boundaries are not treated as limitations of principle, but as design constraints adopted to preserve traceability, verifiability, and architectural stability within the current scope of the system.

2.4 Positioning of the I3A Paradigm

The I3A (Intelligent Artificial Autonomous Agents) paradigm positions itself at the intersection of multi-agent systems, engineered cognition, and instrumented artificial intelligence. Unlike generic multi-agent frameworks that primarily emphasize agent interaction or task distribution, I3A is defined by the explicit coupling of role specialization, operational constraints, and continuous measurement. In this paradigm, agents are not interchangeable components, but constrained computational entities whose responsibilities, inputs, and outputs are formally delimited.

A defining characteristic of I3A is the treatment of inference as a structured and observable process rather than a single opaque execution. Each inference cycle is decomposed into stages handled by specialized agents, with intermediate results explicitly logged and evaluated. This design enables traceability across the entire cognitive cycle, allowing system behavior to be inspected, audited, and reproduced at a level of granularity that is not achievable in monolithic or loosely coupled multi-agent systems.

I3A further distinguishes itself through its emphasis on calibration and controlled heterogeneity. Rather than assuming uniform capabilities across agents, the paradigm explicitly accommodates heterogeneous AI models with differing strengths and limitations. Compatibility testing and calibration mechanisms ensure that models are assigned to roles aligned with their operational characteristics, transforming heterogeneity from a source of instability into a structured design asset.

Finally, the I3A paradigm treats system growth as an architectural concern rather than an external process. The generation of new agents is not driven by offline retraining or manual redesign, but by inference-time detection of cognitive gaps within the system. This form of self-emergent engineering is constrained, logged, and integrated within the existing architecture, preserving traceability while enabling incremental expansion. As such, I3A defines a framework in which adaptability, determinism, and observability are not competing objectives, but complementary design principles.

3 Limitations of Monolithic AI Systems

3.1 Structural limits under reliability and determinism constraints

Single-model artificial intelligence systems consolidate perception, reasoning, control, and generation within a unified parameter space optimized through large-scale statistical learning. This architectural choice has demonstrated strong empirical performance for tasks emphasizing generalization and pattern recognition. However, when artificial intelligence systems are required to satisfy stricter constraints—such as deterministic computation, fine-grained traceability, and inference-time adaptability—structural limitations become evident.

A first limitation concerns reproducibility and determinism. Even under tightly constrained inference settings, single-model systems remain fundamentally probabilistic. Identical inputs may follow divergent internal trajectories across executions, preventing the provision of verifiable guarantees for numerical precision, repeatable computation, or strict procedural compliance. This property poses a fundamental obstacle for scientific, engineering, and safety-critical applications, where traceability and exactness are essential rather than optional.

A second limitation lies in adaptation mechanisms. In monolithic architectures, adaptation is typically externalized to retraining or fine-tuning procedures. These processes occur outside inference-time execution, are resource-intensive, and offer limited transparency regarding their internal effects. As a result, monolithic systems cannot restructure their internal organization dynamically in response to emerging task demands while preserving auditability and control.

Taken together, these characteristics indicate that single-model architectures are poorly aligned with requirements involving deterministic behavior, inference-time structural evolution, and fine-grained instrumentation. While effective within their intended operational envelope, they are structurally constrained when extended toward controlled autonomy and verifiable computation.

3.2 Requirements for viable multi-agent alternatives

Multi-agent architectures provide a natural response to these constraints by enabling functional decomposition and role specialization. By distributing cognitive responsibilities across autonomous agents operating under explicit roles, such architectures offer clearer responsibility boundaries and localized evaluation, both of which are prerequisites for verification and control.

However, multiplicity alone is insufficient. Without explicit role contracts, coordination rules, and systematic instrumentation, multi-agent systems risk reproducing opacity at a higher level of complexity. Agent interactions may become as difficult to interpret as the internal activations of a monolithic model if they are not continuously logged, constrained, and evaluated.

Consequently, viable multi-agent architectures must be engineered rather than merely assembled. They require explicit role definitions, calibrated assignment of computational components, and instrumentation embedded directly into inference-time execution. These requirements motivate the instrumented I3A paradigm introduced in the following section, in which multi-agent coordination, deterministic operation, and controlled self-evolution are treated as first-class architectural concerns.

4 I3A v2: Architecture, Roles, and Instrumentation

4.1 Operational definition of I3A

In the I3A (Intelligent Artificial Autonomous Agents) paradigm, an agent is defined as an autonomous operational entity characterized by an explicit role, a constrained behavioral contract, and evaluable outputs. An agent is not equivalent to an underlying AI model; rather, it is defined by the combination of a role-specific pre-prompt, a configuration of constraints, and an execution context. A single AI model may therefore instantiate multiple agents simultaneously, each operating under distinct roles and contractual obligations.

I3A agents may be persistent or dynamically activated depending on their function. Core agents responsible for orchestration, synthesis, memory, and structural control persist across cognitive cycles, providing continuity and system identity. In contrast, other agents are selectively activated within a cycle based on task requirements, remaining silent when their contribution is not required. This distinction enables both architectural stability and operational efficiency.

Inference in I3A is treated as a structured process composed of coordinated agent contributions rather than a single opaque execution. Each agent produces outputs that are explicitly logged, evaluated, and contextualized within the cognitive cycle. This design enables fine-grained traceability, allowing internal system behavior to be inspected, audited, and reproduced. Through this operational definition, I3A establishes agents as first-class architectural components whose behavior is observable, constrained, and measurable by design.

4.2 Role taxonomy and agent contracts

Table 2: Role taxonomy (inputs/outputs, constraints, evaluation hooks).

Role	Typical I/O	Contract / constraints
AGORA	Structured prompts / framed hypotheses	Analytical framing, explicit reasoning steps, no synthesis or stylistic output; evaluated on clarity and structural coherence.
LYRA	Textual drafts / expressive modulation	Stylistic and emotional modulation only; no factual invention or structural decisions; evaluated on consistency with upstream content.
GNOSEO	Exploration directives / structured outputs	Protocol-driven exploration and hypothesis expansion; constrained to declarative outputs; evaluated on coverage and internal consistency.
APOGEE	Intermediate results / validated derivations	Structured derivation and validation of intermediate results; no generative extrapolation; evaluated on logical soundness.
MIMESIS	Protocol stimuli / dynamic traces	Execution of experimental protocols modeling emotional or cognitive dynamics; evaluated via metric stability and signature patterns.
AURELION	Formal problem definitions / numerical outputs	Deterministic mathematical computation under strict constraints; evaluated against reference values and numerical precision targets.
PROTOÏS	System state / candidate node definitions	Generation of candidate agent identities in response to detected cognitive gaps evaluated on relevance and integration constraints.
NOÉSIS	Multi-agent outputs / unified synthesis	Integrative synthesis and orchestration; no raw generation; evaluated on coherence and alignment with agent contracts.

4.3 Cognitive cycle instrumentation

4.3.1 Log structure and traceability

Each cognitive cycle in Noèse is fully instrumented and produces a structured log capturing the internal dynamics of inference-time execution. A cycle may correspond to a user interaction or to an internal autonomous process. At minimum, logs record timestamps, agent identifiers, role assignments, inputs and outputs, execution order, and resource usage. The presence of inactive agents within a cycle is also recorded through occurrence counters, distinguishing availability from effective participation.

Logs are designed to support traceability rather than narrative reconstruction. By preserving agent-level contributions and execution metadata, the system enables post-hoc inspection of reasoning pathways, identification of failure points, and behavioral comparison across cycles. Sensitive elements such as exact system paths or proprietary prompt details can be anonymized without compromising structural interpretability.

4.3.2 Metrics (FDC or equivalent) and interpretation

In addition to raw logs, each cycle produces a set of derived metrics capturing cognitive, stylistic, and resource-related properties of system behavior. These include composite indicators such as the Functional Density Coefficient (FDC), as well as execution time, energy consumption, peak GPU utilization, and agent participation counts. Metrics are computed deterministically from logged data, enabling consistent cross-cycle comparison.

Metric interpretation focuses on relational analysis rather than absolute scoring. For example, correlations between detected FDC and energy usage or between agent activation patterns and synthesis quality provide insight into system efficiency and cognitive load distribution. By grounding metrics in explicit logs, Noèse ensures that quantitative evaluations remain auditable and reproducible.

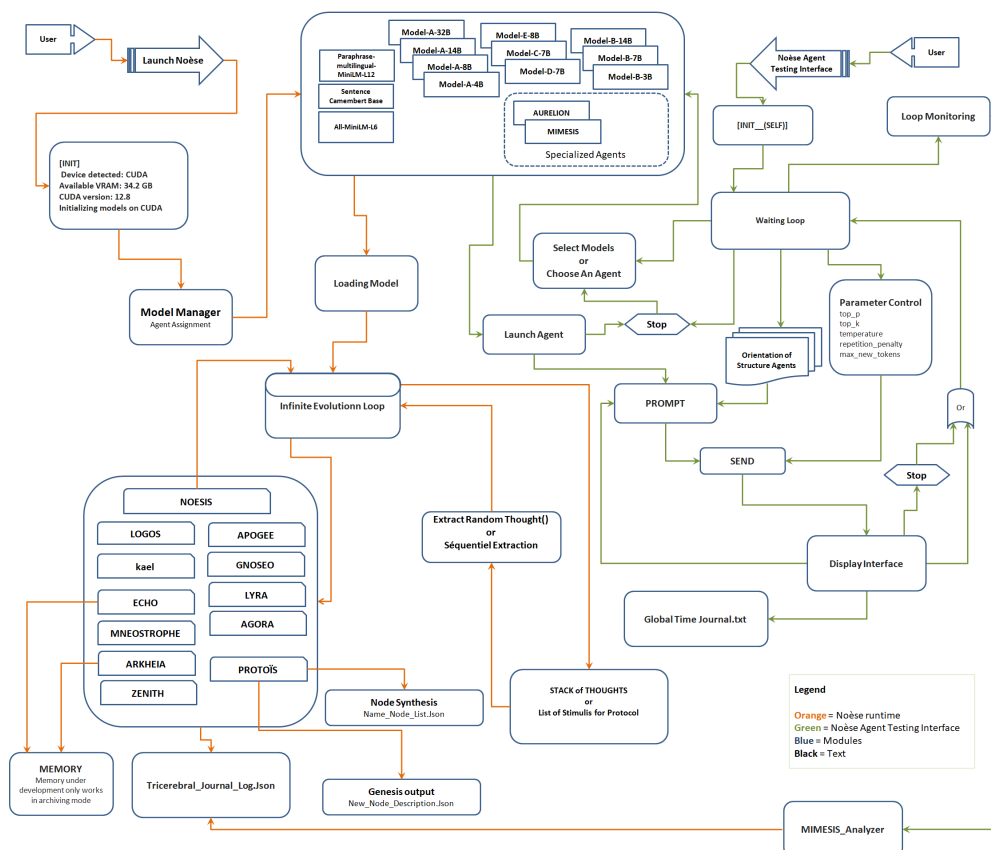


Figure 1: I3A v2 overview: agent ensemble, orchestration, self-evolving mechanisms, and archival memory.

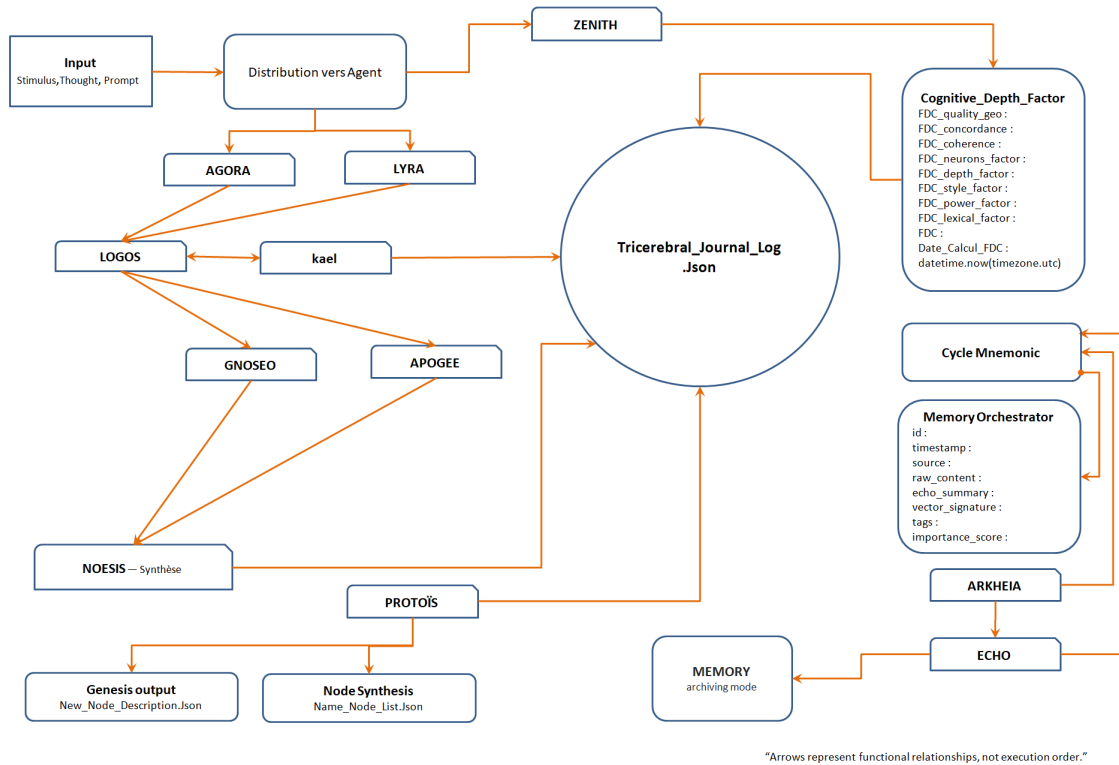


Figure 2: Instrumented cognitive cycle: input acquisition, multi-agent processing, synthesis, logging, and optional agent genesis.

4.4 Reproducibility hooks

Reproducibility in Noèse is supported through the systematic preservation of structured logs, derived metric tables, and configuration mappings. Anonymized cycle logs and CSV summaries can be provided to enable behavioral replay and comparative analysis on independent computational setups. While sensitive implementation details such as exact pre-prompts or proprietary scoring formulas are intentionally withheld, the architectural structure, metric definitions, and execution pipeline are described with sufficient precision to support reproducible behavioral evaluation.

Together, these mechanisms ensure that claims made in this work are not dependent on opaque execution, but can be audited through explicit artifacts generated during inference-time operation.

5 Model Compatibility & Calibration Tool

5.1 Problem statement: not every model is structure-compatible

Within an I3A architecture, not every AI model is equally suitable for every operational role. While many contemporary models exhibit strong generative capabilities, their behavior under strict structural constraints, deterministic requirements, or role-specific contracts can vary significantly. As I3A relies on explicit agent roles with well-defined responsibilities, the suitability of a model cannot be assumed a priori and must instead be empirically evaluated.

Compatibility in this context does not primarily concern raw performance, but operational reliability. A structure-compatible model must exhibit stable behavior across repeated executions, respect explicit output constraints, and maintain predictable formatting and numerical coherence when required. Even when gross failures are rare, subtle variations—such as output drift under fixed parameters or inconsistent adherence to formatting rules—can compromise traceability and reproducibility at the system level.

For this reason, Noèse treats model selection as an engineering problem rather than an ad hoc choice. Compatibility is evaluated through a dedicated calibration process conducted prior to integration, using a specialized testing interface. In some cases, additional targeted evaluations are performed dynamically when empirical observations suggest that a model may be better suited to a different role profile. This approach ensures that role assignment is grounded in measured behavior rather than assumed capability.

5.2 Compatibility criteria

Table 3: Compatibility criteria used to evaluate candidate models.

Criterion	Operational test / measurement
Stability	Variance across repeated runs under fixed parameters, failure rate, and output drift detection.
Format discipline	Strict compliance with required output schemas (e.g., JSON or CSV), absence of extraneous text, and structural validity checks.
Precision / determinism	Capacity to produce numerically coherent and repeatable outputs under fixed constraints, including consistent rounding behavior.
Instruction obedience	Respect of explicit prohibitions, absence of unauthorized operations, and non-hallucination of procedural steps.
Latency / cost	Execution time, energy consumption, token usage, memory footprint, and estimated operational cost.

5.3 Calibration workflow and scoring

The calibration workflow implemented in Noèse is designed to be both systematic and reproducible. Each candidate model is evaluated together with its inference configuration, including sampling parameters, output constraints, and resource limits. The model is then subjected to a battery of contract-based tests corresponding to the compatibility criteria described above.

For each criterion, quantitative scores are computed and augmented with qualitative risk flags. Certain failure modes, such as structural format violations, are treated as fatal for specific roles and immediately disqualify a model from further consideration in that context. The outcome of this evaluation is a compatibility profile composed of a score vector and associated risk indicators, providing a behaviorally grounded summary of the model’s suitability.

Scoring is performed in two stages. First, a role-independent profile is computed, allowing candidate models to be compared on a common basis. Second, role-specific gating rules and weighted

aggregation are applied. In this phase, criteria that are non-negotiable for a given role act as hard gates, while remaining criteria are combined using role-dependent weights. This two-step process preserves comparability while ensuring that final decisions remain aligned with operational requirements.

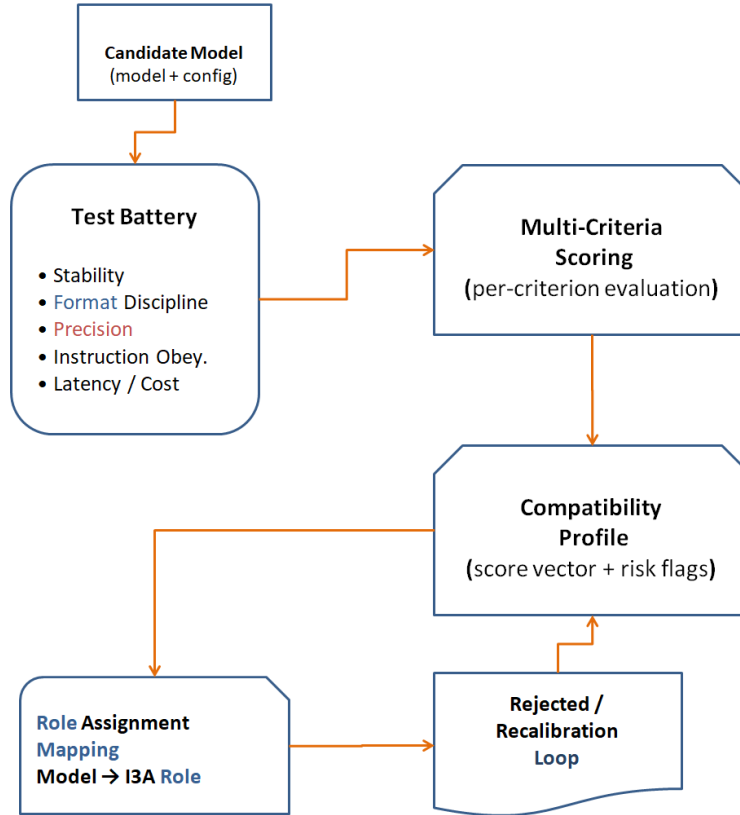


Figure 3: Calibration pipeline: candidate model intake, contract-based testing, multi-criteria scoring, compatibility profile construction, and role-aware assignment or rejection.

5.4 Role assignment mapping (model to agent role)

Based on the resulting compatibility profiles, models are assigned to I3A roles for which they satisfy both hard constraints and weighted performance criteria. Assignment decisions are therefore not global, but role-specific: a model that is unsuitable for deterministic computation may nonetheless be well suited for exploratory or stylistic roles.

Table 4 illustrates anonymized examples of such assignments, using representative model categories rather than commercial identifiers. Models that fail to meet the minimum requirements for any role are either rejected or confined to sandboxed evaluation contexts.

Table 4: Example mapping of selected model types to I3A roles after calibration.

Model type	Assigned role	Notes
General-purpose LLM	LYRA	High expressive stability and instruction obedience; relaxed format constraints acceptable.
Numerically oriented LLM	AURELION	Meets strict determinism and precision gates; validated against reference computations.
Exploratory reasoning model	PROTOÏS	Strong obedience and procedural discipline; suitable for controlled node generation tasks.

6 Deterministic Computation Case Study : " AURELION Math Expert "

6.1 Scientific context (collaboration use-case)

The AURELION Math Expert agent was developed in the context of a scientific collaboration focused on the numerical modeling of emotional dynamics. While specific application details remain confidential, the methodological objective was explicit: to determine whether a generative AI system, when embedded within an instrumented I3A architecture, could reliably execute deterministic mathematical procedures with verifiable accuracy.

The task required the computation of continuous emotional waveforms, including single-stimulus responses and superposed multi-stimulus interactions, under fixed parameters. These computations involve nontrivial temporal dependencies and numerical sensitivity, making them a suitable test case for evaluating determinism, reproducibility, and precision within an AI-driven system. AURELION was therefore not evaluated on narrative quality or interpretative flexibility, but strictly on its ability to reproduce mathematically defined outputs under controlled constraints.

6.2 Deterministic procedure

6.2.1 Constraint system (what AURELION is allowed / forbidden to do)

AURELION operates under a strict constraint system designed to eliminate generative ambiguity. The agent is authorized exclusively to perform symbolic manipulation and numerical computation based on explicitly provided formulas and parameters. Any form of narrative elaboration, heuristic approximation, or interpretative reasoning is explicitly forbidden.

All outputs must conform to predefined numerical formats, with fixed precision rules and deterministic rounding behavior. Intermediate results are logged, and the agent is prohibited from introducing unstated assumptions or modifying the computational procedure. These constraints are enforced through role-specific contracts and validated during execution, ensuring that AURELION behaves as a computational component rather than a generative assistant.

Beyond static constraints, determinism in AURELION is enforced through explicit step-by-step execution. For each evaluated time point, the agent performs a fully specified sequence of arithmetic and transcendental operations, including intermediate normalization, exponential evaluation, absolute value transformation, and rational inversion. Each intermediate result is logged with fixed precision before being propagated to the next step.

In the two-stimulus superposition case ($S = 1.5$), this procedure is executed repeatedly over a dense temporal grid, resulting in over one hundred individual deterministic computation chains of identical structure. No branching, heuristic shortcuts, or adaptive approximations are permitted. As a consequence, identical inputs always generate identical intermediate states and outputs, making the computation strictly deterministic by construction.

6.2.2 Verification against reference (Python or equivalent)

To verify correctness, AURELION outputs are systematically compared against reference implementations executed in a standard numerical environment. For each computed waveform, critical values such as peak time, maximum amplitude, and associated secondary variables are extracted and matched against reference results.

Verification is performed at the level of individual scalar values rather than visual similarity alone. Differences are quantified numerically, enabling precise assessment of deviation. This procedure ensures that agreement between AURELION and the reference implementation reflects genuine numerical correctness rather than coincidental alignment.

Importantly, verification does not rely on isolated output values alone. The reference comparison implicitly validates the entire computational chain, as any deviation in intermediate steps would propagate and be observable at the output level. The agreement observed between AURELION and

the reference implementation therefore reflects not only numerical coincidence, but consistency across the full sequence of deterministic operations.

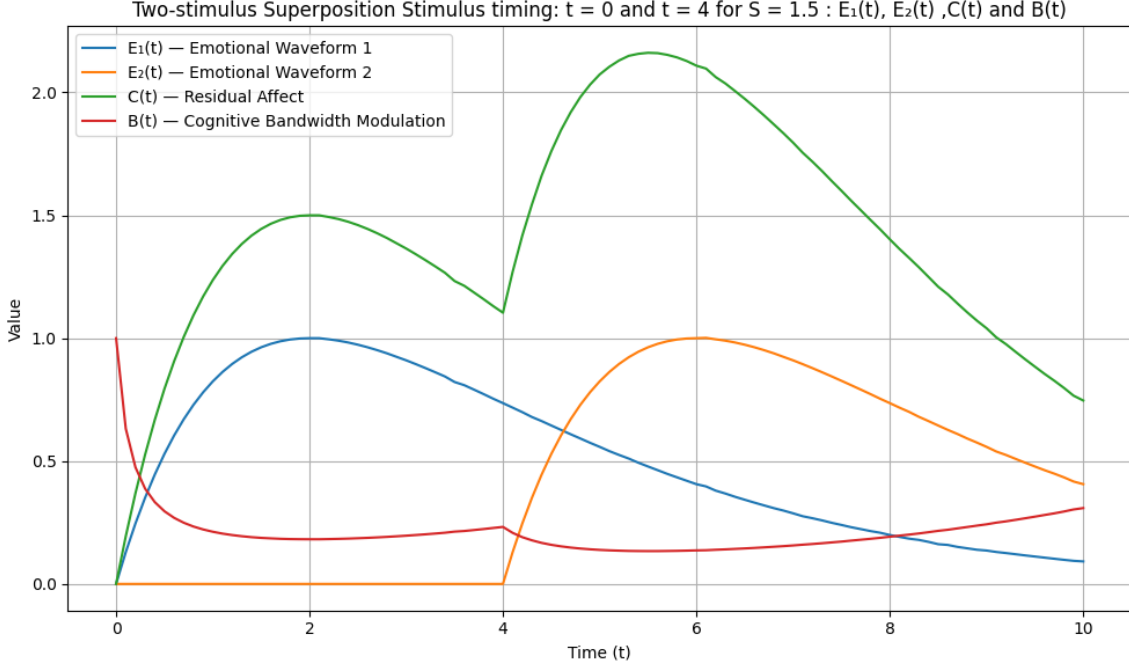


Figure 4: AURELION deterministic computation of emotional waveform under two-stimulus superposition ($S = 1.5$).

Table 5: Deterministic verification of AURELION outputs against Python reference over a continuous time window ($t \in [9.1, 10.0]$), two-stimulus superposition, $S = 1.5$).

t	AURELION $B(t)$	Python $B(t)$	$ \Delta B(t) $
9.1	0.249311	0.248529	7.82×10^{-4}
9.2	0.254613	0.254596	1.70×10^{-5}
9.3	0.260843	0.260820	2.30×10^{-5}
9.4	0.267233	0.267201	3.20×10^{-5}
9.5	0.273771	0.273737	3.40×10^{-5}
9.6	0.280551	0.280428	1.23×10^{-4}
9.7	0.287204	0.287274	7.00×10^{-5}
9.8	0.294668	0.294273	3.95×10^{-4}
9.9	0.303079	0.301424	1.66×10^{-3}
10.0	0.308663	0.308725	6.20×10^{-5}

6.3 Results and accuracy

Equivalent agreement levels were observed for $E_1(t)$, $E_2(t)$, and $C(t)$ over the same interval, with deviations remaining bounded within the same order of magnitude. Full numerical results are provided as supplementary CSV files to support independent verification.

The results demonstrate a high degree of numerical determinism across both isolated verification points and continuous time windows. For all evaluated configurations, AURELION consistently matches the reference implementation to at least the third decimal place. In the majority of cases, agreement extends to the sixth decimal place, particularly for peak-related quantities and steady-state values.

In the two-stimulus superposition scenario ($S = 1.5$), determinism is further reinforced by repeated execution of the same computational structure across a dense temporal grid. Over one hundred

individual computation chains are executed, each following an identical sequence of arithmetic and transcendental operations. No stochastic elements are introduced at any stage, and no adaptive behavior modifies the computation path.

The multi-point verification presented in Table 5 confirms that deviations remain bounded and non-divergent over the interval $t \in [9.1, 10.0]$. Observed differences are attributable to numerical representation and rounding effects rather than structural variability. These results provide strong evidence that AURELION functions as a deterministic computational agent within the I3A architecture, rather than as a probabilistic generator approximating numerical behavior.

6.4 Resource accounting (time/energy/cost framing)

Beyond correctness, the AURELION case study also enables an assessment of resource expenditure. Execution logs record computation time and hardware utilization, allowing an estimation of energy consumption and operational cost. While absolute values depend on the execution environment, relative cost remains stable across runs due to the deterministic nature of the computation.

Table 6: Resource accounting summary for the AURELION deliverables (compact).

Metric	Value
Compute time	Stable across runs (order of seconds per waveform)
Energy (kWh)	Low and repeatable per execution
Token volume (if relevant)	Bounded by strict output constraints
Equivalent service valuation (EUR)	Documented for comparative assessment

7 MIMESIS: Emotional Dynamics Protocols Applied to AI

7.1 Protocol overview (P1–P4)

The MIMESIS agent is designed to probe emotional and cognitive dynamics within the I3A architecture through controlled stimulus protocols. Rather than modeling human emotion directly, MIMESIS applies structured affective stimuli to an artificial system in order to observe response patterns, stability properties, and emergent signatures. Four complementary protocols were defined, each targeting a distinct regime of emotional-cognitive interaction.

Table 7: Summary of the four MIMESIS protocols (goal, stimulus type, expected signature, outputs).

Prot.	Goal	Stimuli	Outputs / signatures
P1	Baseline emotional response	Positive, structured stimuli	Smooth emotional waveforms, stable cognition-emotion coupling, reference signature.
P2	Stress and perturbation	Negative or disruptive stimuli	Amplified emotional responses transient instability, recovery dynamics.
P3	Social cognitive load	Socially contextualized stimuli	Complex oscillatory patterns, delayed stabilization, interaction effects.
P4	Internal consolidation	Neutral or minimal stimuli	Damped oscillations, convergence behavior, internal homeostasis signature.

7.2 Instrumentation: logs and metrics

7.2.1 Emotion/cognition signal extraction

For each protocol, MIMESIS produces structured logs capturing the temporal evolution of emotional and cognitive signals at each stimulus step. From these logs, quantitative metrics are extracted, including emotional intensity, cognitive load indicators, stability coefficients, and composite scores reflecting emotion-cognition coupling. All metrics are computed deterministically from logged values, ensuring traceability from raw interaction to aggregated indicators.

7.2.2 Stability and drift measures

Beyond instantaneous values, MIMESIS evaluates stability and drift across stimulus sequences. Stability is assessed by measuring variance and oscillation amplitude over time, while drift detection focuses on long-term trends indicating adaptation, saturation, or degradation. These measures allow differentiation between transient responses and structural changes in system behavior induced by repeated stimulation.

7.3 Results by protocol (selected figures only)

The following figures present representative signature plots for each protocol. Only one figure per protocol is included here to preserve readability; complete datasets and extended visualizations are available in the associated reports and logs.

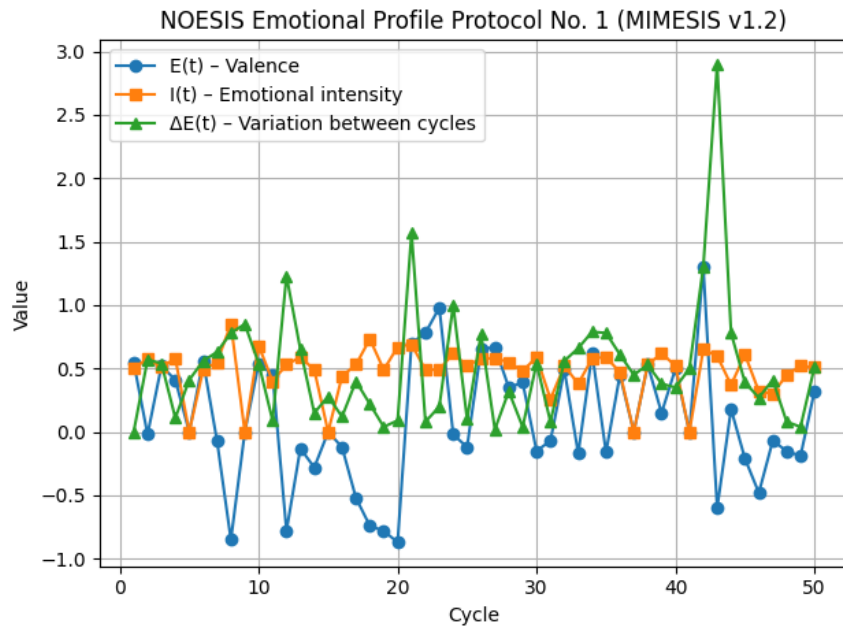


Figure 5: Protocol P1: baseline emotional dynamics under positive structured stimuli.

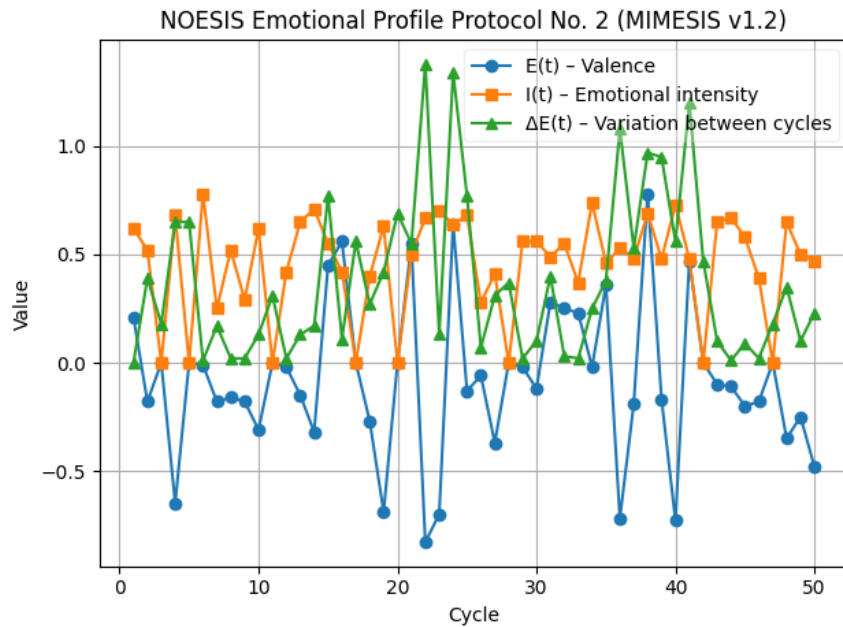


Figure 6: Protocol P2: emotional amplification and recovery under stress-inducing stimuli.

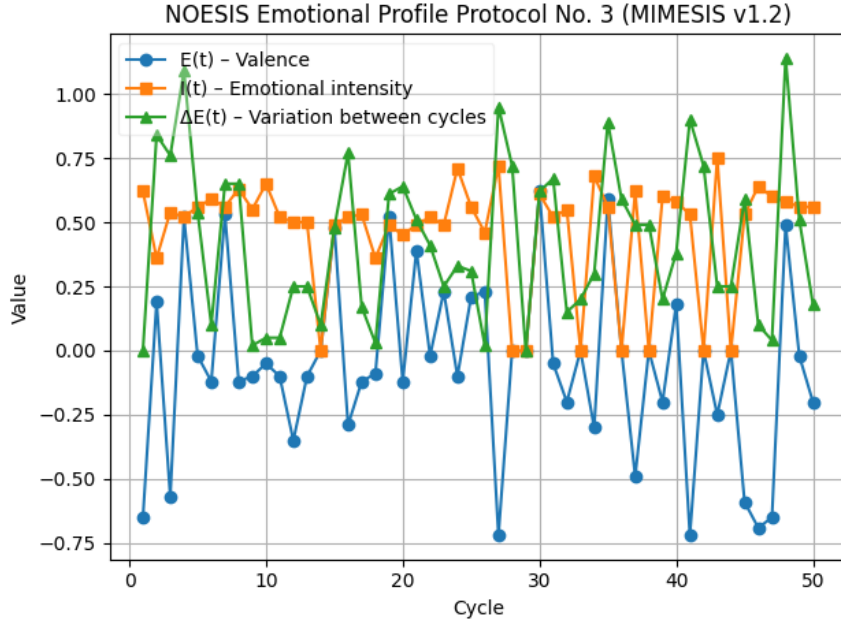


Figure 7: Protocol P3: complex emotional-cognitive oscillations under social cognitive load.

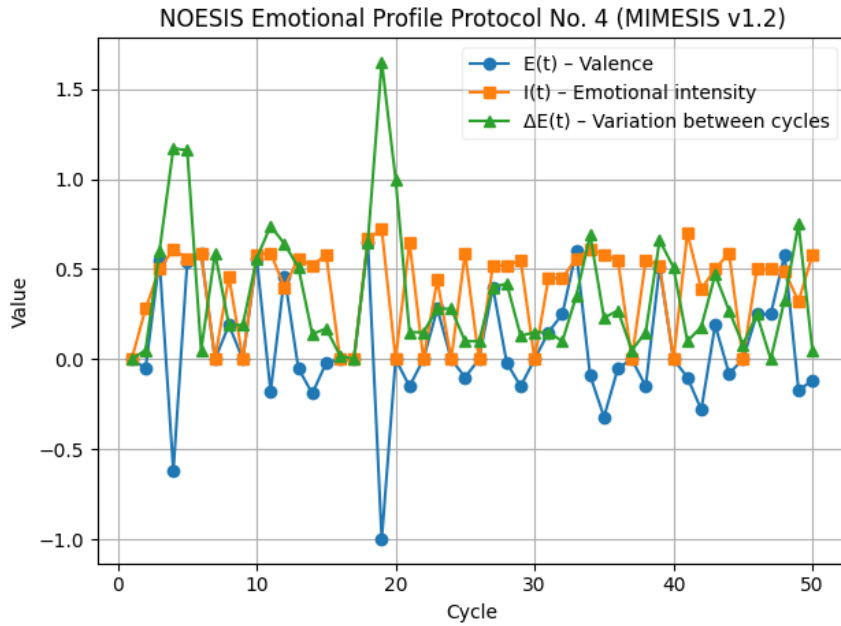


Figure 8: Protocol P4: convergence and internal stabilization under neutral stimulation.

7.4 Cross-protocol synthesis

Across the four protocols, several invariants emerge. Emotional responses remain structured and bounded, with no evidence of uncontrolled divergence. The system exhibits protocol-dependent dynamics while preserving internal coherence, indicating that emotional modulation operates as a controlled dimension rather than a destabilizing factor.

Divergences between protocols highlight the sensitivity of the architecture to stimulus context, particularly under social and stress-related conditions. These differences provide actionable signals for calibration and role adjustment within the I3A framework. Moreover, the observed stability and adaptation patterns inform the self-evolving mechanisms of Noèse, guiding agent generation and refinement based on empirically grounded emotional-cognitive behavior rather than abstract heuristics.

8 Self-Evolving at Scale: Noésis-Driven Agent Genesis

8.1 Triggering mechanism (cognitive gap detection)

Self-evolution in Noèse is not driven by random exploration or offline retraining, but by inference-time detection of cognitive gaps. A cognitive gap is defined as a recurrent or structurally identifiable insufficiency in the current agent ensemble to satisfy a task demand under existing role contracts and constraints. Such gaps may manifest as repeated fallback behaviors, degraded synthesis quality, elevated internal conflict metrics, or abnormal increases in cognitive load indicators.

The detection of these gaps is orchestrated by the Noésis agent, which continuously monitors instrumented signals produced during the cognitive cycle. These signals include composite metrics such as FDC variations, agent participation patterns, stability indicators, and synthesis coherence scores. When a gap exceeds predefined structural thresholds and cannot be resolved through re-weighting or re-orchestration of existing agents, Noésis triggers a controlled agent genesis process.

Importantly, this triggering mechanism is conservative by design. Agent creation is treated as an exceptional operation rather than a default response, ensuring that self-evolution remains bounded, interpretable, and auditable.

This behavior is illustrated in Figure 9, where agent creation events in the baseline protocol cluster around elevated FDC values rather than being uniformly distributed across cycles.

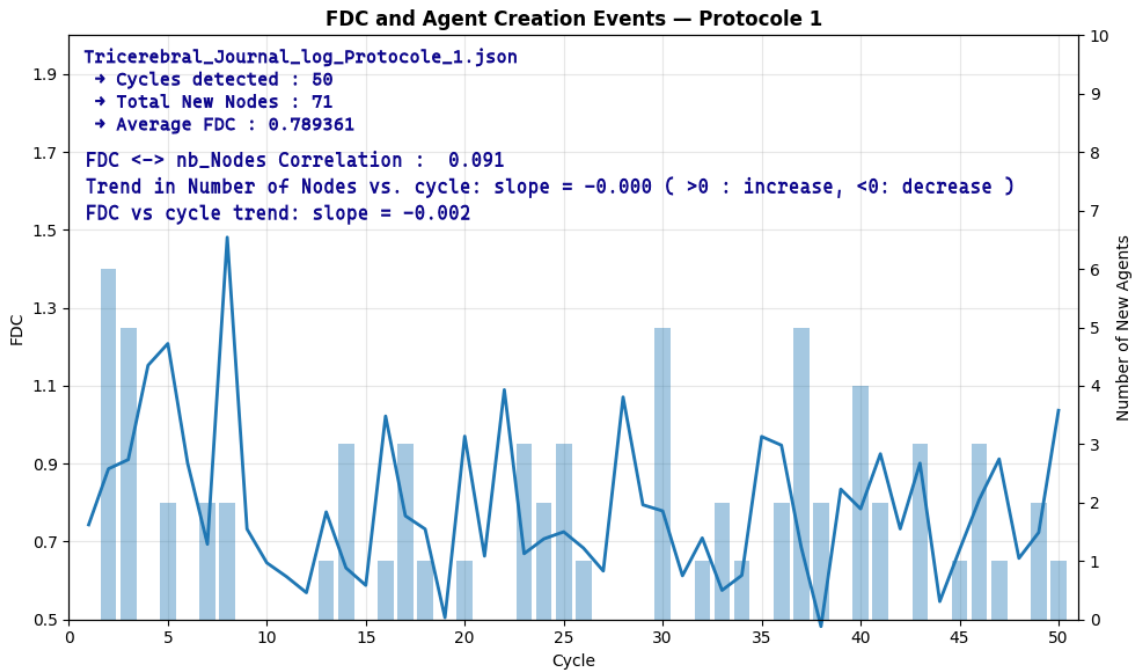


Figure 9: Protocol P1: agent genesis events as a function of cognitive load (FDC). Agent creation clusters around elevated FDC regimes, illustrating the non-random triggering of self-evolution.

8.2 Agent genesis statistics

Across the four MIMESIS protocols, Noèse was subjected to a total of 200 instrumented cognitive cycles. During these cycles, agent genesis events were logged and correlated with protocol-specific dynamics and cognitive load signatures. Table 8 summarizes the observed agent creation activity.

Table 8: Agent genesis statistics across MIMESIS protocols.

Protocol	Cycles	Agents created	Notes
P1	50	71	Baseline protocol; controlled emergence.
P2	50	75	Stress-induced diversification.
P3	50	58	Social-cognitive complexity effects.
P4	50	63	Consolidation and refinement phase.
Total	200	267	Bounded self-evolution at scale.

Figure 10 illustrates the relationship between agent creation events and the Functional Density Coefficient (FDC) across protocols, highlighting non-random clustering of genesis events around elevated cognitive load regimes.

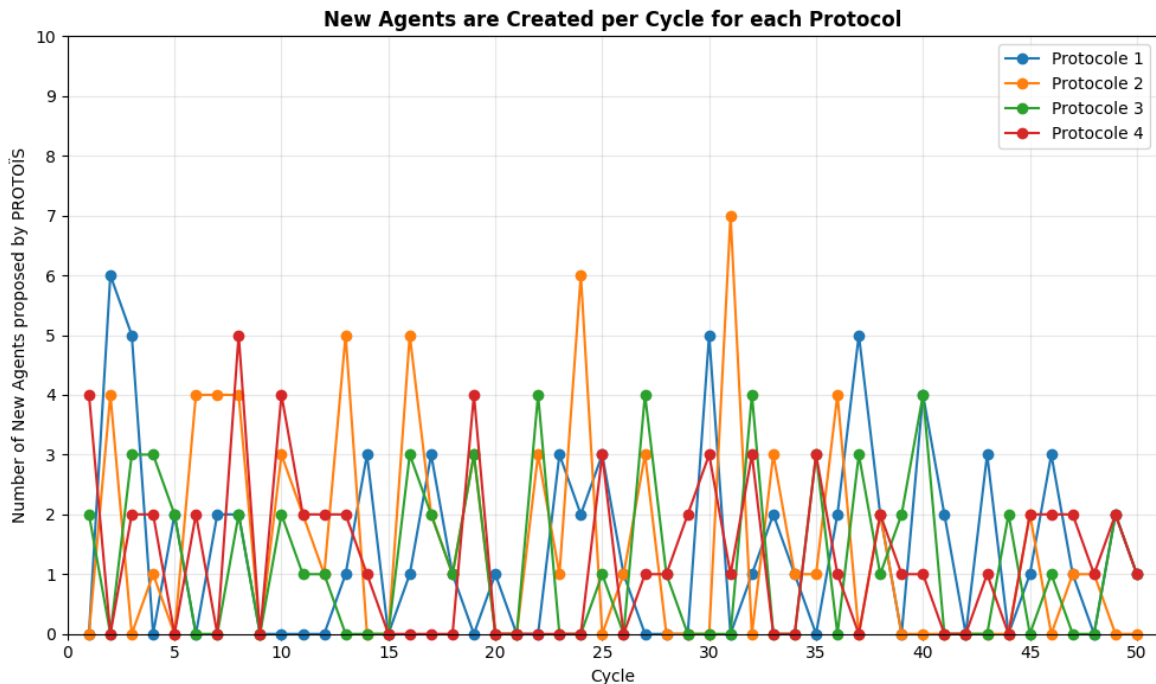


Figure 10: Agent creation events as a function of cognitive load (FDC), shown for individual protocols and their superposition.

8.3 Quality considerations (current limits and constraints)

While large-scale agent genesis demonstrates the viability of inference-time self-evolution, it also raises quality and governance considerations. Not all generated agents exhibit equivalent utility or long-term relevance. Some agents reflect highly specialized responses to transient cognitive conditions, while others encode more generally reusable capabilities.

To mitigate uncontrolled proliferation, Noèse applies post-generation constraints, including role validation, contract compliance checks, and relevance scoring. Agents failing to meet minimal quality thresholds may be archived, sandboxed, or excluded from active orchestration. These mechanisms ensure that self-evolution remains selective rather than expansive.

Current limitations include the semantic refinement of generated agent descriptions and the need for stricter constraints on overly exotic or redundant role definitions. These aspects are the subject of ongoing work and are intentionally treated as future improvements rather than fully resolved components in the present release.

8.4 Examples of generated agents (selected)

Table 9 presents a small selection of generated agents, chosen to illustrate the diversity of roles produced while remaining compact and interpretable. These examples are representative rather than exhaustive; extended listings are maintained internally and can be shared for scientific review upon request.

Table 9: Selected generated agent “cards” (compact examples).

Agent name	Intended role	Minimal description
CHROMA	Cognitive modulation	Specialized agent for fine-grained modulation of emotional–cognitive balance under elevated load conditions.
VECTORIS	Analytical support	Agent generated to assist structured reasoning in multi-constraint problem contexts.
SENTINEL	Stability monitoring	Lightweight agent dedicated to early detection of instability and drift within cognitive cycles.

8.5 Controlled replay experiment on self-evolving agent genesis

To further probe the internal conditions governing self-evolving agent creation, a controlled replay experiment was conducted using a reduced and curated stimulus set.

From the four experimental protocols, three stimuli per protocol were selected based on their previously observed capacity to trigger agent genesis. This resulted in a total of twelve stimuli, spanning heterogeneous informational, emotional, and social contexts.

Baseline observation. When originally executed within their full protocol context (50 stimuli per protocol), these twelve stimuli collectively contributed to the creation of 49 new agents, establishing a reference generative potential.

Replay without algorithmic modification. The same twelve stimuli were replayed in chronological order, without any modification to the NOESE algorithm. In this reduced-context configuration, the system generated 29 new agents. This reduction highlights the influence of cumulative cognitive context and stimulus density on agent genesis, indicating that node creation is not a purely local response to individual stimuli.

Replay with minimal algorithmic modification. A second replay was performed using the same stimuli and ordering, but with a minor internal modification to the agent generation mechanism. Under these conditions, only 2 new agents were created in total. Notably, residual agent genesis occurred exclusively within Protocol 4, while Protocols 1–3 exhibited no agent creation. This suppression is visible in Fig. 12: FDC dynamics remain present, while agent creation collapses to two isolated events (cycles 10–11), mapped to Protocol 4 in the replay order.

Interpretation. These results suggest that self-evolving agent genesis in NOESE is a structurally sensitive process, dependent on both accumulated cognitive state and precise algorithmic conditions. The selective persistence of agent creation in Protocol 4 suggests a differentiated sensitivity across cognitive regimes, particularly those involving internal stabilization / homeostatic dynamics.

Rather than reflecting stochastic variability, this experiment supports the interpretation of agent genesis as a threshold-based phenomenon, where small algorithmic shifts can move the system across or below critical generative conditions.

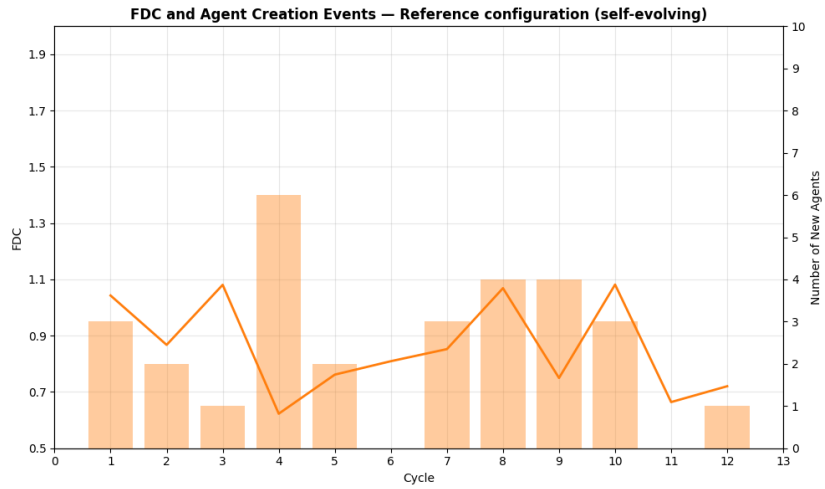


Figure 11: Reference self-evolving configuration: frequent agent creation events across cycles.

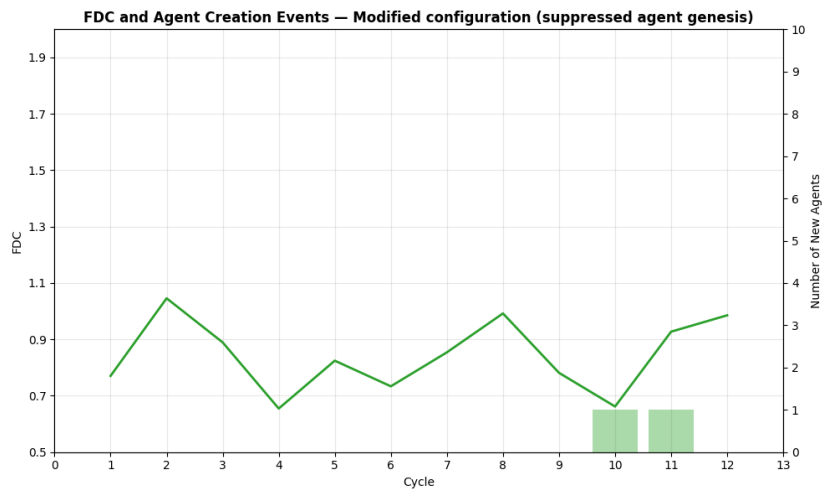


Figure 12: Modified configuration (non self-evolving): FDC remains active, but agent genesis is nearly suppressed.

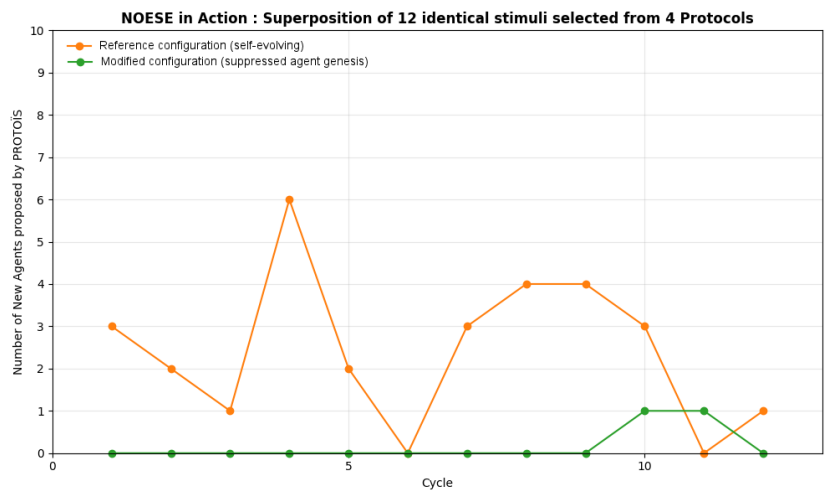


Figure 13: Superposition on identical stimuli: the generative regime shift is visually explicit.

9 Archiving & Memory: Semantic Condensation

9.1 Why memory matters for self-improvement

In Noèse, memory is not yet used for cognitive reuse or decision guidance. At its current stage, the system is strictly positioned at the level of data preservation and semantic consolidation. The exploitation of archived knowledge is intentionally deferred and will be addressed in a subsequent work once the memory substrate is fully stabilized.

This design choice reflects a clear principle: before reusing memory, an autonomous system must first learn how to forget, prioritize, and condense. Raw cognitive outputs generated by Noèse—whether observations, decisions, or internal errors—are intrinsically heterogeneous and noisy. Treating memory as a simple storage layer would inevitably lead to saturation and semantic drift. Instead, Noèse frames memory as a transformation process, where meaning is progressively distilled before entering long-term retention.

The archival subsystem therefore acts as a cognitive filter. It converts raw traces into structured memory units through reformulation, compression, semantic evaluation, and contextual classification. This approach enables the system to retain what matters, discard what does not, and prepare a stable foundation for future autonomous adaptation.

9.2 Archival pipeline

Figure 14 illustrates the current archival pipeline implemented in Noèse. Cognitive outputs are first handled by the *Memory Orchestrator*, a deterministic and thread-safe component responsible for validating payloads, normalizing metadata, assigning unique identifiers, and indexing entries. At this stage, all data are stored in an *Active* memory space, which functions as a short-term buffer for unprocessed cognitive traces.

A dedicated *Mnemonic Cycle* operates as the central consolidation mechanism. This cycle periodically scans active entries and orchestrates their transformation through specialized agents. The first stage relies on the combined action of *ECHO* and *MNEOSTROPHE*.

ECHO performs an introspective reformulation, improving semantic clarity without altering meaning, while *MNEOSTROPHE* applies ultra-dense compression to produce compact representations suitable for long-term storage. Together, they implement what can be described as a semantic condensation process.

During this phase, intrinsic scoring is computed. A confidence score is derived from internal semantic consistency using embedding-based similarity, serving as a proxy for coherence rather than external correctness. In parallel, an importance score is assigned based on the type of cognitive trace, reflecting the intuition that structural updates and errors deserve higher retention priority than routine observations. Processing latency is also measured for instrumentation and future analysis.

The second stage of the mnemonic cycle is handled by *ARKHEIA* in collaboration with *ZENITH*. *ZENITH* generates vector embeddings that position each condensed trace within a semantic space. *ARKHEIA* compares these embeddings against evolving semantic clusters, reinforcing existing concepts when similarity is high or generating new conceptual tags when novelty is detected. This mechanism allows memory to grow organically, without relying on a fixed taxonomy.

Based on semantic proximity and contextual importance, entries are classified into a tiered archival structure. High-priority concepts are preserved with greater fidelity, while low-impact traces are relegated to minimal retention. This explicit stratification implements a controlled forgetting strategy, ensuring that memory remains informative rather than exhaustive.

9.3 Current implementation vs roadmap

The current implementation of the archival subsystem focuses exclusively on memory formation. Archived content does not influence ongoing cognition, agent orchestration, or decision-making. Retrieval capabilities are limited to basic keyword-based access and are not yet integrated into cognitive cycles. This restriction is intentional and reflects a methodological choice: memory must first be interpretable, auditable, and robust before being allowed to affect system behavior.

From an engineering perspective, this separation isolates semantic consolidation from adaptive reuse, reducing coupling and preserving system stability. From a cognitive standpoint, it mirrors biological memory processes, where consolidation precedes active recall and learning. Future work will explore controlled retrieval, memory-guided modulation of cognitive cycles, and long-term pattern consolidation. At its current stage, Noèse’s memory subsystem serves as a disciplined, instrumented foundation for autonomous self-improvement rather than an immediate learning engine.

9.4 Archival pipeline

Figure 14 provides a conceptual view of the archival process currently implemented in Noèse. Rather than depicting an execution workflow, the figure highlights the functional transformations involved in memory formation, from raw cognitive traces to structured long-term representations.

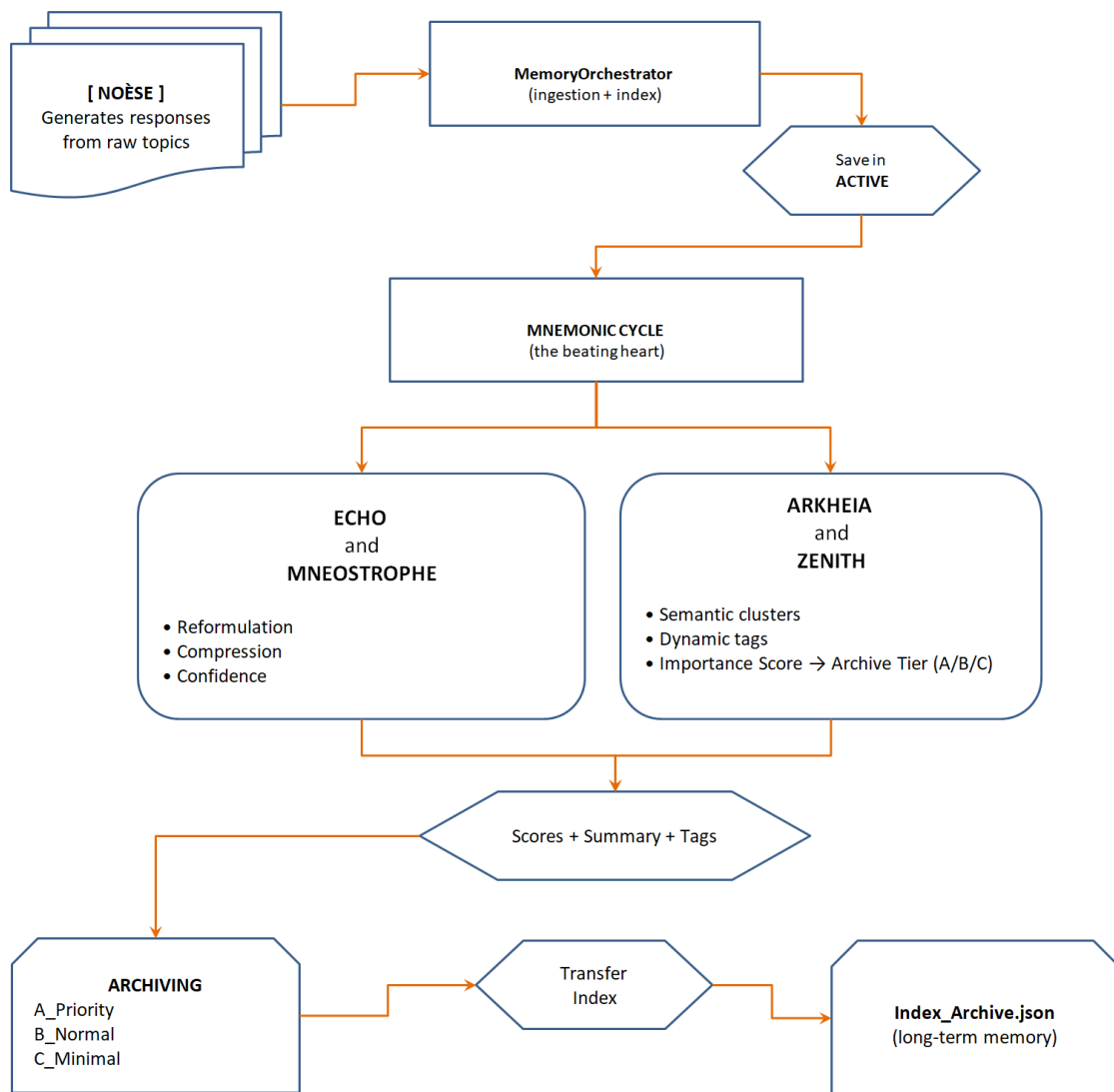


Figure 14: Conceptual overview of the Noèse archival pipeline. Raw cognitive traces are progressively condensed, scored, and semantically organized before entering tiered long-term memory. Arrows indicate functional relationships, not execution order.

The figure should be read as an abstraction of roles and transformations rather than a procedural specification. It emphasizes semantic condensation, selective retention, and structured forgetting as first-class design principles of the Noèse memory subsystem.

10 Open Research, Collaboration, and Continuation

This documented research artifact is not intended as a closed or fully exhaustive specification of the NOESE / I3A architecture. It represents a stabilized research snapshot, designed to expose core principles, validated mechanisms, and instrumented experimental results, while intentionally preserving the system’s evolutionary nature.

10.1 Access to extended materials

Detailed execution logs, structured archives, intermediate computation traces, and extended datasets are maintained but are not fully included in this release due to volume and structural complexity. They can be shared selectively for scientific review, replication studies, or collaborative work upon direct contact.

10.2 Research dialogue and collaboration

This work is open to:

- scientific discussion and critical analysis,
- collaborative exploration of self-evolving cognitive architectures,
- joint experimentation on deterministic emotional–cognitive systems,
- interdisciplinary research converging toward autonomous artificial intelligence.

The author explicitly encourages contact from researchers, engineers, and institutions interested in extending, testing, or recontextualizing the presented mechanisms, whether within NOESE itself or through parallel, convergent approaches.

10.3 Ongoing evolution

NOESE is a living research framework. Future releases will progressively expose additional mechanisms, including archive-driven adaptation, extended agent governance, and higher-order self structuring processes. These evolutions are part of the research trajectory and not post-hoc additions.

11 Conclusion

This work presents *Noèse* as an instrumented I3A architecture in which inference is no longer treated as an opaque byproduct, but as a structured, observable, and auditable process. The central premise is straightforward: once an artificial intelligence system claims reliability or controlled autonomy, architectural opacity becomes a structural ceiling rather than a tolerable trade-off. *Noèse* addresses this ceiling through explicit engineering choices—role contracts, orchestration, systematic logging, metrics, and governance mechanisms embedded directly into the cognitive cycle.

11.1 Inference as an engineered process

Noèse is not merely multi-agent; it is instrumented by design. Agents are defined operationally by their role, constraints, and execution context, rather than being conflated with underlying models. This separation enables traceability, responsibility isolation, and post-hoc analysis that are not achievable in monolithic systems. Inference is therefore treated as an inspectable process rather than a black-box outcome.

11.2 Calibration and behavioral reliability

A key contribution of this work is the treatment of model–role compatibility as an engineering problem. Instead of assuming universal applicability, models are calibrated against behavioral criteria such as stability, constraint obedience, format discipline, numerical precision, and resource cost, and are assigned to roles accordingly. This reframes model heterogeneity from a liability into a controlled architectural asset.

11.3 Determinism under constraint

The AURELION case study demonstrates that deterministic and verifiable numerical computation can be embedded within a generative multi-agent system when strict procedural contracts are enforced. Determinism is achieved structurally through constrained execution, intermediate logging, and verification against reference implementations, challenging the assumption that generative systems are inherently incompatible with exact computation.

11.4 Controlled self-evolution at inference time

Beyond computation, *Noèse* supports inference-time self-evolution under explicit control. Agent generation is conservatively triggered by detected cognitive gaps and governed by structural thresholds. Large-scale experiments reveal bounded and non-uniform agent genesis, while replay studies show that minor algorithmic modifications can suppress agent creation without collapsing internal dynamics, indicating a structurally sensitive rather than stochastic mechanism.

11.5 Memory as a disciplined substrate

The memory subsystem follows the same philosophy. At its current stage, memory is intentionally limited to semantic condensation, prioritization, and controlled forgetting. It functions as a disciplined substrate rather than an autonomous decision engine, preserving interpretability and stability while preparing the ground for future controlled reuse.

11.6 Perspective

Taken together, these results position *Noèse* not as a new model, but as a research framework rooted in engineering principles. Its contribution lies in demonstrating that inference can be instrumented, roles contractualized, models calibrated by behavioral compatibility, determinism enforced under constraint, internal dynamics measured, and self-evolution bounded and observable. *Noèse* is presented as a stabilized research snapshot and an open experimental ground for further investigation into auditable, self-evolving, and engineerable autonomous artificial intelligence systems.

A Appendix A: Minimal Log Schema (Example Excerpt)

The following excerpt presents a minimal, anonymized execution log from a single instrumented cognitive cycle, illustrating the structural elements used to observe inference-time behavior, including agent activation, summarized outputs, metrics, and resource usage. Selected fields have been masked for confidentiality.

```
1 {
2   "Log Name": "<masked>",
3   "Script Version Used": "<masked>",
4   "Date": "2026-01-11",
5   "Interaction Time": "18:15:48.951",
6   "Cycle Number in Time": 20260111182218,
7   "Cycle Number": 3,
8   "Algorithm": [<masked>],
9   "Models_Info": {<masked>},
10  "AGORA_Node_Model": "<masked>",
11  "Noesis Node Model": "<masked>",
12  "Random Thought Extraction": 162,
13  "Thought_emitted": "<masked>",
14  "Response_B_Lyra_Node": "<masked>",
15  "Response_NOESIS_Node": "<masked>",
16  "New_Nodes_Proposed_By_PROTOIS": [<masked>],
17  "resources": {
18    "VRAM_GPU_Used_For>Loading_Models": 26.3,
19    "Total_Interaction_Duration(s)": 295.83,
20    "NA_Response_Time": 44.33,
21    "NL_Response_Time": 40.51,
22    "NN_Response_Time": 210.99,
23    "NA_Style_Detection": "analytical",
24    "NL_Style_Detection": "analytical",
25    "Score_For_Style_NA": 0.923522,
26    "Score_For_Style_NL": 0.910242,
27    "Pre-Prompt_Generated_for_NA2": "<masked>",
28    "Pre-Prompt_Generated_for_NL2": "<masked>",
29    "Average CPU Capacity Used Percentage": 13.27,
30    "Average RAM Capacity Used Percentage": 25.38,
31    "Maximum GPU Capacity Used Percentage": 73,
32    "Maximum GPU Temperature": 66,
33    "Total Cycle Energy Consumption (Wh)": 38.33,
34    "Gross Addition of Neurons": 295,000,
35    "Synergy Neurons": 73,750
36  },
37  "metrics": {
38    "FDC_quality_geo": 0.816203,
39    "FDC_concordance": 0.932429,
40    "FDC_coherence": 0.792946,
41    "FDC_neurons_factor": 0.7375,
42    "FDC_depth_factor": 1.214814,
43    "FDC_style_factor": 1.458441,
44    "FDC_power_factor": 0.627224,
45    "FDC_lexical_factor": 1.246326,
46    "FDC": 1.290972,
47    "Timestamp_FDC": "[2026-01-11T17:22:45Z]"
48  }
49 }
```

This schema illustrates how inference is treated as an observable and auditable process rather than a black-box operation. In practice, full logs include additional fields for governance decisions, agent evaluation scores, and archival metadata.

B Appendix B: AURELION Verification Snippet

This appendix provides a compact verification excerpt illustrating the deterministic numerical behavior of the *AURELION* agent. Reported values are compared against a reference numerical implementation under identical initial conditions. The excerpt is intentionally limited to a small continuous window to demonstrate bounded deviation without exposing full datasets.

t	$B(t)$ (AURELION)	$B(t)$ (Reference)	$ \Delta $
0.0	1.000000	1.000000	0.000000
0.1	0.632121	0.632197	0.000076
0.4	0.333000	0.333002	0.000002
0.7	0.248941	0.248943	0.000002
1.0	0.212339	0.212331	0.000008
1.3	0.194125	0.194146	0.000021
1.6	0.185446	0.185286	0.000160
1.9	0.182041	0.182011	0.000030
2.2	0.182562	0.182517	0.000045
2.5	0.185997	0.185848	0.000149
2.8	0.191499	0.191460	0.000039
3.1	0.199058	0.199036	0.000022
3.4	0.208403	0.208382	0.000021
3.7	0.219277	0.219383	0.000106
4.0	0.232043	0.231969	0.000074
4.3	0.177453	0.177219	0.000234
4.6	0.152966	0.152922	0.000044
4.9	0.140995	0.140925	0.000070
5.2	0.135363	0.135316	0.000047
5.5	0.133579	0.133618	0.000039
5.8	0.134643	0.134629	0.000014
6.1	0.137202	0.137714	0.000512
6.4	0.142554	0.142517	0.000037
6.7	0.148864	0.148839	0.000025
7.0	0.156565	0.156570	0.000005
7.3	0.165572	0.165655	0.000083
7.6	0.176237	0.176074	0.000163
7.9	0.187708	0.187829	0.000121
8.2	0.200823	0.200933	0.000110
8.5	0.216015	0.215405	0.000610
8.8	0.231648	0.231265	0.000383
9.1	0.249311	0.248529	0.000782
9.4	0.267233	0.267201	0.000032
9.7	0.287204	0.287274	0.000070
10.0	0.308663	0.308725	0.000062

The observed deviations remain bounded and compatible with representation and rounding effects rather than structural variability. This supports the claim that *AURELION* operates deterministically by construction when executed under a strict procedural contract.

Glossary

I3A (Intelligent Artificial Autonomous Agents). Architectural paradigm based on the coordination of autonomous artificial agents operating under explicit roles, constraints, and execution contexts. In I3A, intelligence emerges from structured interaction rather than from a single monolithic model.

Noèse. An instrumented I3A architecture in which inference-time execution is explicitly structured, logged, and auditable. Noèse emphasizes role specialization, calibration, deterministic computation under constraint, and controlled self-evolution.

Agent. An autonomous computational entity defined by a combination of role, behavioral contract, and execution context. In Noèse, an agent is not synonymous with a model, but with a functionally constrained role possibly instantiated by different models.

Role. A formally defined functional responsibility assigned to an agent. Roles specify expected behaviors, constraints, input–output formats, and evaluation criteria.

Behavioral Contract. A set of explicit constraints governing an agent’s operation, including procedural rules, admissible outputs, determinism requirements, and logging obligations.

Instrumentation. The systematic embedding of logging, metrics, and verification mechanisms directly into inference-time execution. Instrumentation enables traceability, auditability, and post-hoc analysis of internal system behavior.

Cognitive Cycle. The structured sequence of operations governing inference in Noèse, including agent activation, orchestration, logging, evaluation, and governance decisions.

Calibration. An engineering process used to evaluate and assign AI models to agent roles based on operational criteria such as stability, constraint compliance, numerical precision, instruction discipline, and resource cost.

Model–Role Compatibility. A quantified assessment of how reliably a given model satisfies the behavioral contract of a specific role under operational constraints.

AURELION. A deterministic numerical computation agent operating under a strict procedural contract. AURELION demonstrates that exact and reproducible computation can be embedded within a generative multi-agent architecture.

MIMESIS. An experimental protocol framework designed to stimulate, measure, and compare internal emotional–cognitive dynamics within Noèse. MIMESIS protocols are instrumented and non-anthropomorphic, focusing on measurable system behavior rather than human imitation.

FDC (Factor Production Cognitive). A scalar metric used to characterize internal dynamic regimes within the system, enabling comparison across stimuli, protocols, and execution contexts.

Self-Evolving Agent Genesis. A controlled inference-time mechanism by which Noèse generates new specialized agents in response to detected cognitive gaps, governed by explicit thresholds and structural constraints.

Cognitive Gap. A detected mismatch between task demands and available agent capabilities, used as a conservative trigger for agent generation.

Inference-Time Execution. The operational phase during which agents interact, compute, and adapt without retraining or external modification of underlying models.

Traceability. The ability to reconstruct internal system behavior through logs, metrics, and recorded intermediate states.

Determinism under Constraint. A property whereby an agent produces reproducible outputs when operating under a strictly enforced procedural contract, even within a generative system.

Instrumented Emergence. The treatment of emergent behavior as an observable, measurable, and falsifiable phenomenon rather than an implicit or metaphorical concept.

Research Artifact. A stabilized and documented system snapshot intended for analysis, validation, and collaborative exploration, rather than a finalized or closed product.

Acknowledgments

The author acknowledges the open-access contributions of research teams and organizations whose publicly released models, datasets, and tools enabled the development of the **I3A** / **Noèse** experimental framework.

The author also thanks the participants for their prior exchanges regarding formal emotion-cognition models that informed parts of the experimental exploration. These models were independently implemented, evaluated, and instrumented within the Noèse architecture. All architectural design decisions, implementations, and experimental results presented in this work remain the sole responsibility of the author.

The broader open-source and research community is gratefully acknowledged for its central role in democratizing access to advanced machine learning technologies, making independent research on cognitive architectures possible.

All cited models and resources remain the property of their respective creators. This work makes no commercial use of any proprietary asset.

References

- [1] J.-P. Aubert, *Pratiquez l'intelligence artificielle*, Eyrolles, Paris, 1983.
- [2] MalKo, *I3A: Autonomous Artificial Intelligent Agents for Self-Evolving Cognitive Architectures*, Zenodo Repository, 2026.