Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

TME LLM OS: A Closed-Loop Systems Engineering Architecture for Epistemic Correction in AI Agent Systems

Ilan Kander

ilan.kander@gmail.com

Independent Scholar https://orcid.org/0009-0008-0338-3488

Method Article

Keywords: AI Agent Systems, Discriminative Intelligence, Episode Context Controller, Epistemic Correction, Overthinking Drift, Reflective Delay, Residual Mismatch, Secondary Segmentation, TME Framework, TME LLM OS

Posted Date: November 5th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11442416/v2

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

TME LLM OS: A Closed-Loop Systems Engineering Architecture for Epistemic Correction in AI Agent Systems

Ilan Kander

***Abstract*— Current AI systems can combine foundation models, retrieval, tools, agents, and execution controls, yet still amplify an unsupported premise or require repeated user correction because generation, review, and release are not organized as one bounded epistemic episode. This systems-engineering proposal translates selected functions from the human Theoretical-Mechanistic-Experiential (TME) framework into a generic artificial architecture. The source framework distinguishes the real-time experiential field from cognition, memory, and action, and orders correction through salience, attention, ordinary awareness, functional self-awareness, reflective delay, secondary segmentation, discriminative comparison, and renewed orientation. The proposed TME LLM OS retains the Transformer-foundation-model core while adding episode continuity, source provenance, role-specific agents, structured correction, residual-mismatch assessment, and gated execution. Runtime correction returns unresolved mismatch to the current episode until provisional convergence reaches a task-dependent tolerance $|\Delta| \leq \varepsilon$; validated corrections can enter a separate, governed, auditable, versioned, and reversible refinement path across episodes. The proposal does not equate functional implementation with lived experience. It yields testable predictions concerning grounding, recovery from false premises, safe release, recurrence of corrected errors, user burden, total task cost, and time-to-acceptable-result.**

***Index Terms*— AI Agent Systems, Discriminative Intelligence, Episode Context Controller, Epistemic Correction, Overthinking Drift, Reflective Delay, Residual Mismatch, Secondary Segmentation, TME Framework, TME LLM OS.**

This work received no institutional or external financial support.
Ilan Kander is an Independent Researcher, Israel (e-mail: ilan.kander@gmail.com).
ORCID: https://orcid.org/0009-0008-0338-3488
A preliminary version of this manuscript was made publicly available as a preprint on Zenodo (doi: 10.5281/zenodo.23403314).

I. Introduction: Hallucination, Path-Dependent Inference, and the Need for Epistemic Loop Closure

Contemporary generative AI systems produce language, code, plans, control proposals, and multimodal interpretations through path-dependent probabilistic inference. Prompts, system instructions, accumulated context, retrieved information, sensor data, prior state estimates, and earlier generated tokens condition the developing trajectory, while learned parameters ordinarily remain fixed during inference. Attention routes and integrates representations within that trajectory, but does not itself provide intention, causal validation, or reflective access to how the interpretation was formed. Fluent or actionable content can therefore become available for release before the operation that produced it has been made an explicit object of examination.

An unsupported interpretation introduced early in this trajectory can condition later generation, planning, self-critique, or model-based evaluation. Continued processing may then increase internal coherence without improving factual grounding or causal validity, producing a linguistic hallucination, false world-model state, or inappropriate action plan. This possible *path-dependent amplification* is termed *Overthinking Drift*. The problem is not probability itself and not the total absence of feedback; current systems may use tools, critics, environmental feedback, and human review. The architectural gap is the absence of an independently organized episode that preserves the governing concern, source provenance, alternatives, correction history, and release conditions while interrupting continuation, reacquiring evidence, comparing interpretations, testing causal relations, and correcting the trajectory before consequential release. Accumulated dead-reckoning error without an independent position fix provides a control-system analogue.

The present systems-engineering proposal is grounded in two prior levels of TME development. Kander [1] provides a phenomenological description of nearly the full experiential organization later incorporated into TME, including the real-time experiential field, the functional distinction between consciousness and cognition, lived significance, salience, attention, ordinary awareness, self-awareness, reflective delay, renewed differentiation, and the possibility

of correction. Kander (2026a) [2] (Hebrew translation in press as Kander, 2026b [3]) reformulates this experiential organization as an explicit epistemic-mechanistic framework, defining non-overlapping functional roles, canonical relations, recurrent correction, differential diagnosis relative to source authority, and the progression from stabilized understanding toward valid knowledge. The Hebrew translation (2026b) [3], currently in press as a book chapter, has undergone full peer review. The present paper translates that epistemic-mechanistic organization into an implementable and testable systems-engineering architecture for language, software, perceptual, robotic, autonomous, and embodied AI agent systems. The source framework advances three linked claims [1], [2], [3]. *First*, a mind-specific feedback loop transforms disturbance, mismatch, or possibility into lived significance, salience, attention, stabilization, ordinary awareness, self-awareness, reflective delay, secondary segmentation, discriminative comparison, correction, and stabilized understanding; only after further testing and correction can stabilized understanding move toward valid knowledge. *Second*, cortex, symbolic language, and diverse forms of expression provide necessary computational, symbolic, communicative, and cultural conditions, but are insufficient for stabilized understanding or valid knowledge without a loop that makes selected content lived, holdable, comparable, correctable, and stabilizable. Generative language is therefore treated as one specialized cortical capacity among others: it can produce structured meaning, explanations, and candidate interpretations, but cannot by itself establish knowledge beyond the meaning it generates. *Third*, consciousness is a necessary link in the loop: the organism-level, real-time experiential field in which feedback-relevant content becomes lived, salient, and available for conscious regulation when automatic regulation is insufficient.

Evidence of basic conscious life without a cerebral cortex, including relevant human clinical cases, supports the claim that consciousness should not be reduced to specialized cortical processing [4]. Panksepp's account of subcortical affective systems provides complementary support for the evolutionary and organism-level foundations of conscious affective life [5]. Rather, the later-developing cortex supplies specialized regions and networks for memory-linked interpretation, planning, symbolic abstraction, and generative language. Their contribution requires a communication interface through which selected cortical and bodily-neural outcomes enter and shape the experiential field, while feedback-relevant experiential content becomes available for further cognitive and cortical interpretation. The possible biological convergence interface remains a research hypothesis, but the systems-engineering implication is clear: a generative LLM (large language model) should be modeled as one specialized processing region within a wider recurrent architecture, not as the complete architecture of consciousness, cognition, or knowledge.

A central methodological claim of TME is that its terms must have strict, non-overlapping functional definitions. Consciousness, cognition, experience, salience, attention, ordinary awareness, self-awareness, reflective delay, secondary segmentation, discriminative intelligence, and correction cannot be allowed to define or explain one another tautologically. Their separation makes the canonical sequence mechanically describable and permits selected functions to be translated into system components without equating the components with consciousness. Most regulation remains automatic and non-conscious. When an unresolved interpretation or response becomes current content, reflective delay permits secondary segmentation, discriminative comparison, correction, and controlled response. Current corrective prompting supplies parts of this process externally by returning system-relative output for renewed examination. The engineering proposal seeks to internalize part of this *borrowed reflective regulation* without assuming that computational implementation reproduces lived experience.

Operating and control systems already separate sensing, state estimation, planning, permissions, execution, feedback, and safety interlocks. These mechanisms neither constitute consciousness nor automatically provide epistemic correction, but they offer implementable precedents for holding candidate output, reacquiring source data, comparing alternatives, and gating linguistic, software, tool, or physical execution. The proposal is hardware-implementation-agnostic but not algorithmically neutral. It does not make the AI machine deterministic as a whole: generative, associative, perceptual, and alternative-forming processes may remain probabilistic, while the canonical TME loops impose reproducible rules for holding, comparing, correcting, and releasing their results. The architecture is therefore a controlled stochastic system with rule-governed, auditable correction and release boundaries. It specifies a high-level system architecture together with canonical functional algorithms: constitutive loops, episode maintenance, reflective delay, secondary segmentation, discriminative comparison, recurrent correction, task-dependent convergence, and controlled release. These functions may be realized through different models, software structures, operating systems, storage technologies, and hardware platforms, provided that their functional definitions, ordering, interfaces, feedback responsibilities, and correction conditions are preserved. *Fig. 1* presents the current baseline; *Fig. 2* presents the canonical TME sequence from which the machine architecture is derived.

The proposal therefore contributes a specific architectural hypothesis rather than a new collection of AI capabilities: epistemic correction should be organized as a bounded recurrent episode that preserves the governing concern, source

provenance, candidate history, alternatives, residual mismatch, and release conditions across successive cycles. Its distinctive empirical question is whether this organization improves recovery from false premises, reduces repeated user correction and recurrence of corrected errors, and lowers total task cost relative to strong agentic, world-model, neuro-symbolic, and embodied baselines.

Section II defines the minimum TME distinctions required for engineering translation. *Section III* specifies the recurrent agent architecture and its boundaries. *Section IV* defines episode-time correction. *Section V* limits the biological analogy. *Section VI* presents the integrated TME LLM OS. *Section VII* compares it with current system architectures. *Section VIII* states the resulting claims and tests.

II. TME Functional Foundations: Consciousness, Cognition, Memory, and the Experiential Present

A. Consciousness and Cognition as Distinct Functional Roles

The name *Theoretical-Mechanistic-Experiential* identifies three coordinated descriptive dimensions of the framework: its theoretical claims, its mechanistic role definitions and canonical loops, and the experiential appearance that those mechanisms must explain. They are not three biological levels, independent substances, or software layers. In TME, *consciousness* is the organism-level, real-time experiential field in which selected sensory, affective, bodily, imaginal, and conceptual contents appear; it is not the interpretive mechanism or a store of content, preference, or knowledge. *Cognition* comprises the interpretive, stabilizing, memory-linked, and action-guiding operations whose selected outcomes enter, shape, or become available within that field. Most bodily-neural regulation remains non-conscious. Consciousness becomes especially important when disturbance, mismatch, need, threat, relief, or possibility requires orientation beyond automatic regulation. Ordinary awareness makes selected content usable; self-awareness begins when an ongoing interpretation, impulse, or response becomes current content and can be delayed, segmented, compared, corrected, redirected, or inhibited [1], [2].

B. No-Hysteresis, Temporal Integration, and the Current Episode

No-hysteresis means that consciousness has no dedicated storage function [2], but this does not make experience durationless or disconnected. Brief retention, sensory traces, temporal binding, and short-lived recurrent support integrate selected contents arriving from sensory, bodily, emotional, imaginal, and cognitive sources, and at slightly different processing times, into a coherent experiential present. This present stains the neighboring working-memory workspace with the current configuration of contents, sources, relations, and significance. The working-memory workspace maintains these successive stains as an elongated episode, allowing cognition to relate newly presented content to recent states, interpretations, goals, corrections, and possible responses. Each stimulus, sensation, thought, or internally generated content nevertheless enters consciousness as a new current event. Consciousness does not store or carry the episode forward; no-hysteresis prevents a previous experiential state from smearing the next presentation. Longer stabilization and persistence belong to working memory, long-term memory, learning, and cognitive patterning.

Current-Episode Workspace. A neighboring working-memory system maintains a fast, fragile model of the current episode: recent states, objects, task conditions, governing concern, source provenance, candidate interpretations, goals, corrections, and possible actions. Its temporary persistence supports comparison and regulation without becoming an intrinsic property of consciousness. The artificial analogue must preserve whether content originated from current sensing, retrieved memory, prediction, simulation, generated output, or corrective feedback.

Long-Term Memory, Learning, and Cognitive Patterning. Persistence beyond the episode belongs to memory, learning, case history, and cognitive patterning. Cognition retrieves and interprets these contents and can contribute corrected results to later learning, but is not itself a long-term store. This separation preserves plasticity: stabilized understanding, including culturally or scientifically validated knowledge, remains open to revision as evidence, measurements, models, and domains of application change.

C. Internally Generated Processing, Dreaming, and Hallucination

The distinction between externally grounded processing and internally generated activity is important for both TME and artificial systems. In the human framework, dreaming shows that the experiential field can remain active when ordinary coupling to the external environment and practical control are reduced. Dream content may then be shaped predominantly by memory, affect, imagination, bodily state, and internally generated cognitive organization. Dreaming does not imply a total absence of input: interoceptive signals, autonomic activity, residual sensory

stimulation, neural variability, and changes in bodily state may still influence the field. The relevant condition is reduced constraint by reliable external information and weakened reality-testing.

A limited functional analogy applies to artificial systems. When reliable new input is absent, weak, noisy, ambiguous, delayed, or no longer representative of the environment, processing may continue from the current episodic state, retrieved memory, learned parameters, prior state estimates, predictions, and internally generated candidates. In language models, autoregressive generation may continue from the existing context and previously generated tokens. In perceptual, robotic, and autonomous systems, state estimation and planning may continue from an earlier world model, sensor history, or predicted trajectory. Noise may be segmented as signal; departure from equilibrium may indicate a genuine disturbance or system-state change; and an extreme condition such as novelty, conflict, overload, threat, sensor failure, or large prediction error may exceed automatic regulation and require escalation to a broader correction process.

Internally generated activity can support simulation, planning, memory consolidation, counterfactual evaluation, anomaly anticipation, and preparation for possible action. The epistemic risk arises when internally generated content loses its provenance, is mistaken for current external input, and is recursively elaborated without independent grounding. A TME-inspired artificial architecture should therefore represent internally generated states explicitly as predictions, simulations, hypotheses, or candidate interpretations rather than as observations. Before such content updates an external world model or triggers consequential action, the system should assess source provenance and sensor confidence, compare the candidate with the current episodic record, reacquire source data when possible, test temporal and causal consistency, and evaluate alternatives. Dreaming and machine hallucination may thus share a limited organizational condition, with greater dependence on internally generated structure under reduced external constraint, while remaining fundamentally different: dreaming is lived experience within an organism, whereas machine hallucination is an ungrounded output, state estimate, or action proposal.

D. Functional Translation to Artificial Systems

Cognition in TME is distributed across specialized cortical regions and networks whose partial operations contribute to a wider epistemic process. In the artificial architecture, a generative LLM is one specialized linguistic-symbolic module within a wider recurrent system. Its candidate outputs become stabilized understanding only through the source-sensitive, episodic, comparative, corrective, and release functions required by the wider architecture. The artificial architecture can implement this organization across language, software, perceptual, robotic, autonomous, and embodied systems without implying lived experience.

E. Correction Timescales, Operational Cost, and User Burden

Two Correction Timescales. During ordinary use, correction acts within the current episode: the system can retrieve or reacquire source data, reconstruct task state, compare the candidate with definitions, evidence, causal constraints, and alternatives, and return unresolved mismatch for another cycle without changing the foundation model's learned parameters. Corrective prompting currently supplies much of this function externally. A TME-inspired architecture seeks to internalize source grounding, reflective delay, secondary segmentation, discriminative comparison, correction, and gated release. Across longer timescales, corrections accumulated and validated through external evidence, reproducibility, scientific or cultural review, and safety evaluation may propose governed updates to memory, retrieval, policies, routing, specialized models, adapters, or learned parameters. Persistent change remains distinct from runtime correction and requires provenance, authorization, versioning, evaluation, rollback, and protection against unrestricted self-modification.

The two timescales yield a testable operational and economic prediction. Source recovery, episode maintenance, alternative comparison, uncertainty calibration, causal and safety checks, and gated release may increase the cost and latency of an individual cycle. The relevant measure, however, is the *total cost of reaching an acceptable outcome*: machine computation, energy, tools, retrieval, sensing, infrastructure, and latency, together with user prompting, source restoration, repeated explanation, review, cognitive burden, rework, and the cost of unsuitable answers or unsafe software, world-model, or physical actions. Earlier correction may reduce repeated prompts, invalid actions, full-cycle repetition, and recurrence of previously corrected errors; governed refinement may reduce recurring intervention across future tasks. Evaluation should therefore report total task cost, time-to-acceptable-result, human correction time, number of corrective cycles, user burden, machine and tool cost, failure and rework cost, cost per validated outcome, and the proportion of corrections completed internally.

1) Implementation Case: Incomplete Self-Correction During Diagram Editing: A diagram-editing episode during manuscript development exposed the difference between technical execution and epistemic closure. The required

outcome contained several simultaneous conditions: move a complete vertical column, create spacing between blocks, relocate an arrow, preserve unrelated content, and verify the displayed result. The machine changed the text representation and reported completion after checking that an edit had occurred, but did not compare the rendered output with every retained condition. Repeated corrective prompts and final manual repair were therefore required. A TME implementation would preserve the decomposed requirements in the episode workspace, hold the completion report, compare the displayed result with each criterion, return residual mismatch for another cycle, and release only when the task-dependent tolerance is satisfied. The case predicts lower user burden, fewer corrective cycles, and lower total task cost when outcome verification is internalized.

F. Implementation Examples

Table I presents the implementation examples.

TABLE I

IMPLEMENTATION EXAMPLES

Profile	**Episode-time operation**	**Release and learning boundary**
LLM assistant Initial state: user request and authoritative source. Trigger: unsupported claim, omitted definition, or conflict with the source.	The Episode Context Controller preserves the request and source authority. Self-monitoring holds the draft; secondary segmentation separates source text, model interpretation, and added assumption; discriminative intelligence compares alternatives; retrieval or rereading restores missing evidence; structured correction returns a revised candidate.	Release occurs when required claims are source-grounded and $\lvert\Delta\rvert \leq \varepsilon$. The expected result is a qualified answer with preserved distinctions. User feedback and outcome review return to the episode. Repeated validated corrections may later update case memory, schemas, routing, or governed model components, but ordinary runtime correction does not change foundation-model parameters.
Software agent Initial state: requested code or document operation, repository state, permissions, and acceptance criteria. Trigger: test failure, unauthorized operation, incomplete requirement, or mismatch between the changed artifact and the requested outcome.	The agent snapshots the task state, decomposes the acceptance criteria, generates a candidate change, runs tests or document checks, compares the actual result with every criterion, and returns any residual mismatch for another cycle. The OS (operating system) or platform enforces sandboxing, permissions, transactions, and rollback.	Release requires passing tests, preserved unrelated content, authorized execution, and $\lvert\Delta\rvert \leq \varepsilon$. The expected result is a verified artifact rather than a successful edit command alone. Test results and user review become feedback. Recurrent validated failures may refine testing schemas, tool selection, or policies through the separate governance path.
Robotic or autonomous system Initial state: sensor observations, world-model state, objective, safety envelope, and action history. Trigger: low sensor confidence, unexpected obstacle, prediction error, failed action, or risk above threshold.	Primary segmentation and salience identify the event. The episode retains sensor provenance and prior state estimates; reflective delay holds consequential action; secondary segmentation separates observation, prediction, and planned response; discriminative intelligence compares routes, reacquires sensing, or invokes simulation and control models.	Release requires adequate sensor grounding, satisfied safety constraints, and $\lvert\Delta\rvert \leq \varepsilon$. The expected result is a revised, qualified, or safely withheld action. Environmental outcome becomes new input. Validated cross-episode evidence may refine maps, case memory, control policies, specialized models, or protected parameters only through governed deployment and rollback.

III. Agent Dynamics and Constitutive Functions

The TME process is canonical but recurrent rather than linear. Bodily-neural selection and primary segmentation precede lived significance, salience, attention, cognitive interpretation, ordinary awareness, and possible response. When available regulation is insufficient, self-awareness, reflective delay, secondary segmentation, discriminative comparison, correction, and renewed orientation enter the cycle. The no-hysteresis and episodic-continuity distinctions established in Section II-B remain governing constraints. The artificial architecture translates this organization without assuming a phenomenal field.

Within a bounded episode, a fast current-state cycle handles acquisition, segmentation, salience, routing, candidate interpretation, and response readiness. When the developing interpretation or response requires review, a slower reflective loop reacquires sources, preserves alternatives, compares evidence, and returns correction. The Episode Context Controller maintains the governing concern and the episode state established in Section II-B across both cycles. It thereby internalizes part of the continuity now supplied externally through corrective prompting, while supervisory controls govern escalation, resources, fallback, human handoff, and release without replacing canonical recurrence.

A. Foundation-Model Substrate and Higher-Level TME Agents

The artificial architecture treats a generative LLM, or coordinated foundation models, as the principal cortical-like processing substrate for representation, probabilistic sequence modeling, abstraction, pattern completion, candidate generation, planning support, and access to learned parameter knowledge. Higher-level TME agents use this shared substrate rather than independent intelligence engines. Each agent is defined by its TME role, episode context, memory and source access, tools, permissions, preserved distinctions, update policy, and authority to recommend, delay, correct, gate, or release an operation. Specialized models support perception, control, or verification where needed; shared services maintain the episode, provenance, retrieval, memory, communication, audit, access control, tools, and action interfaces. Static, dynamic, and linking agents are distinguished by functional role, update rate, and operational authority rather than by different computational substances. Static agents maintain protected, corrigible constraints and validation criteria; dynamic agents generate and revise interpretations, predictions, plans, and candidate responses; linking agents preserve provenance, timing, identity, priority, and compatibility as information moves among models, agents, memory, tools, sensors, and action interfaces.

The foundation model remains parameter-stable during ordinary inference but is not immutable ROM (read-only memory). Episode-time correction changes context, episodic state, candidates, routing, tools, or action; only separately validated cross-episode evidence may justify governed, versioned, auditable, and reversible updates to memory, retrieval, policies, adapters, specialized models, or foundation-model parameters. The architecture contains no immutable ontological base or internal store of final truth. Static agents provide protected, slowly updated constraints, validated sources, safety boundaries, and current validation criteria; dynamic agents adapt present processing; and linking agents combine protected interfaces with transient or learnable mediation. All remain corrigible through their authorized update paths.

Discriminative intelligence belongs at the higher agent layer because it is neither the temporary problem model nor the foundation model itself. It is an experience-sedimented problem-solving capacity that uses the shared foundation model, conceptual and procedural memory, and the current episode to recognize problem structure, compare the present case with prior cases, select or revise a strategy, and abandon an earlier schema when its applicability conditions are not met. Its complete episode-time function, resources, persistent-update policy, and authority are specified in Section III-B and Appendix A (Table III).

The biological and engineering boundaries established in the Introduction govern this mapping: the foundation model supplies cortical-like specialization, while constitutive orientation functions remain architecturally distinct from agent use of the model.

These three categories do not exhaust the higher-level architecture. Episode Context Control, Self-Monitoring, Reflective Delay, Secondary Segmentation, Discriminative Intelligence, Epistemic Correction, and Release Gating are separately defined roles because they maintain distinct state, timing, authority, or feedback responsibility. The complete mapping of architectural roles, TME derivation, episode-time function, persistent-update policy, and operational authority is provided in Appendix A (Table III).

B. Core TME Functions in Artificial Systems

1. Differential Diagnosis: This function starts when the system has more than one possible interpretation or when a proposed output or action may be wrong. After reflective delay, it compares the candidate with the original source or sensor data, the purpose of the current episode, relevant memory, alternative explanations, causal relations, operating constraints, and previous corrections. It asks four simple questions: What supports this candidate? What contradicts it? What information is missing? Is another interpretation more suitable? The result is provisional. If support is sufficient, the candidate can move toward controlled release. If a mismatch remains, the system returns the problem to the current episode, gathers more information, revises the candidate, qualifies the result, or abstains.

2. Functional Self-Awareness and Reflective Delay: This function begins when the system stops treating its current interpretation or planned response as automatically acceptable. Instead, it brings that developing operation back into the current episode as something to be examined. Phenomenologically described in Kander [1] as an interruption of automatic continuation and formalized as an epistemic mechanism in Kander [2], Reflective Delay is translated here into an engineered execution pause that prevents consequential release and triggers Differential Diagnosis. During this pause, secondary segmentation separates the source from its own interpretation, observation from prediction, premise from conclusion, and the proposed action from its expected consequence. The system can compare alternatives and decide whether to continue, revise, redirect, or withhold the response. This is a functional review loop; it does not claim that the machine has phenomenal self-awareness.

3. Constitutive Orientation and Salience: Before reflective review is possible, the system must distinguish current input from internal state, segment candidate signals, preserve source identity, and determine whether a change is relevant to viability, task goals, safety, or possible action. Salience changes priority and recruits attention or routing; it does not by itself interpret the content or establish consciousness. In the source human framework, these functions overlap partly with subcortical arousal, orientation, and action systems. In the artificial architecture, they are implemented through sensing, state monitoring, anomaly detection, priority control, and routing.

4. Network Switching: The architecture must shift resources between internally generated processing and task-directed processing without treating either mode as consciousness. The Salience Network provides a partial biological overlap for relevance detection and switching between the Default Mode Network (DMN) and Central Executive or Frontoparietal Network (CEN/FPN) [6], [7]. In the artificial system, salience agents, linking agents, and episode-context control jointly perform the engineering analogue: they select the active route, preserve the governing concern, and redirect processing when new evidence, mismatch, or urgency requires it.

5. Discriminative Intelligence: This is a learned problem-solving capacity, not a truth store and not merely the temporary problem model. It uses the shared foundation-model substrate together with conceptual hierarchies, problem-solving schemas, case memory, prior solution paths, applicability conditions, tool knowledge, and validation criteria. Within the episode it identifies the structure of the present problem, retrieves relevant experience, preserves the differences between the present case and earlier cases, compares alternatives, selects or revises a strategy, and recommends whether to continue, reacquire information, qualify the result, abstain, or hand off.

6. Episode Continuity, Correction, and Release: The current-episode workspace and Episode Context Controller maintain the governing concern, source authority, provenance, candidate history, open mismatches, previous corrections, possible actions, uncertainty, and release criteria across successive cycles. Unresolved mismatch returns through the recurrent pathway introduced in Section II-E and maintained within the current episode; release and persistent refinement remain governed by the separate boundaries defined in Section II-E.

C. Architecture Boundaries and Neighboring Services

The architecture distinguishes canonical TME functions, cortical-like processing and higher-level agents, and neighboring services. Fig. 3 shows how these categories interact while preserving two boundaries: neighboring services provide infrastructure without constituting consciousness or cognition, and the DMN, CEN/FPN, and Salience Network offer only partial functional overlaps rather than definitions of consciousness or direct mappings to artificial agents.

IV. Recurrent Epistemic-Correction Pathway

The TME architecture does not seek zero difference as a universal condition. It establishes a recurrent correction pathway that reduces unresolved mismatch to a task-dependent acceptable range while preserving bounded associative variation for alternative generation, simulation, and reconsideration. For system-engineering purposes, Δ denotes the structured residual mismatch between a candidate interpretation or action and the episode's current evidence, source

provenance, competing alternatives, causal and operational constraints, prior corrections, and governing concern. It is not a difference from an immutable static truth state.

Residual Mismatch Identification. After reflective delay and secondary segmentation, the system compares candidate interpretations or actions with source provenance, current evidence, competing alternatives, causal and operational constraints, uncertainty, prior corrections, and the governing concern. It records the remaining discrepancy as a structured, task-specific residual rather than an undefined semantic vector.

Structured Return to the Episode. The residual mismatch, its provenance, missing evidence, competing hypotheses, uncertainty, safety status, and recommended next steps are returned as new current content in the artificial system's episode workspace. Salience and routing functions prioritize the mismatch for renewed examination, after which the correction becomes part of the elongated episode and can be related to the governing concern, source provenance, prior candidates, previous corrections, and possible responses. In the human TME framework, an analogous correction becomes current content in the experiential field [1], [2]; the artificial implementation claims only functional availability, not lived experience. The current-episode workspace, audit record, and governed memory services maintain persistence across cycles.

Bounded Provisional Convergence. The renewed cycle does not require $\Delta = 0$ as a universal release condition. It seeks a task-dependent tolerance $|\Delta| \leq \varepsilon$, where ε reflects permitted uncertainty, risk, source confidence, operational consequence, loop limits, and safety requirements. For a strictly determinate logical or formal operation, ε may be set to 0. For empirical, perceptual, interpretive, probabilistic, or operational tasks, ε will ordinarily remain greater than 0. Residual associative variation within the permitted tolerance supports alternative generation, imagination, counterfactual evaluation, and adaptation to new evidence. Release occurs only when major contradictions are resolved, grounding is sufficient, and residual mismatch falls within the permitted envelope. If this condition is not reached within the allowed time, computation, or retry budget, the system reacquires information, qualifies the result, abstains, enters a safe fallback, or transfers control to a human operator.

Fig. 4 summarizes the episode-time pathway defined above. Persistent updates remain outside this loop and follow the governed refinement boundary established in Section II-E.

V. Biological Analogy and Engineering Boundary

Biological Boundary. The source TME paper provides the full biological and theoretical argument [2]. For the present systems-engineering proposal, the relevant boundary is that basic wakefulness, internal-state regulation, sensation, emotion, orientation, target selection, and action readiness should not be reduced to cortical processing. Cortical networks expand cognition through memory-linked interpretation, abstraction, planning, language, and reflective self-awareness. A possible brainstem-cortical convergence interface remains a research hypothesis, not the location of consciousness and not a component-by-component blueprint for artificial systems.

Engineering Boundary. The present proposal extracts a generic functional architecture for sensing, segmentation, salience, specialized processing, episodic continuity, reflective delay, comparison, correction, controlled action, and governed learning. Human creation can apply acquired understanding of natural phenomena and human capacities through selective copying, analogy, abstraction, and imitation [1, Sec. 4.3]. An engineered system therefore need not reproduce its natural model completely to implement useful parts of its organization. These functions may support measurable artificial-system development through corrections accumulated and validated across episodes, improving memory, problem-solving schemas, routing, specialized agents, models, and protected parameters. TME defines experience as present content lived within the real-time field of consciousness [2]. What remains open is whether implementing a similar functional organization in a machine would cause its current states to be lived rather than only processed, represented, and reported.

VI. TME LLM OS Architecture: Runtime Correction and Governed Refinement

Fig. 5 integrates the runtime architecture and the separate cross-episode refinement path. The term *LLM operating system* denotes the protected orchestration environment that coordinates agents, episode state, external services, execution permissions, audit, and governed refinement. It does not identify consciousness with a memory-management unit, the elongated episode with hardware RAM (random-access memory), validated knowledge with immutable ROM, or Reflective Delay with a single kernel component.

VII. Comparison with Current Frontier AI Systems and System Architectures

Comparison must extend beyond commercial Transformer-LLM products. Current frontier platforms already combine long context, retrieval, multimodal input, reasoning, tool and program use, compliance controls, agent orchestration, persistent task state, execution environments, and human oversight. These capabilities are represented across OpenAI's tool-using agent platform, Google DeepMind's embodied Gemini Robotics, Anthropic's workflow and agent architecture, Microsoft Copilot Studio's governed connected agents, and Meta's model- and system-level safeguards [8], [9], [10], [11], [12]. Relevant research baselines also include modular world-model architectures, continual experience-based agents, active-inference guidance, neuro-symbolic systems, embodied vision-language-action models, and world-foundation-model platforms for physical AI. Gemini Robotics and OpenVLA provide baselines for vision-language-action control, embodied reasoning, and robot adaptation, whereas NVIDIA Cosmos primarily provides world-model simulation, predictive infrastructure, and post-training resources rather than a directly equivalent action policy. Global Workspace Theory describes broad availability and broadcasting of selected content [13], but it is not used here as a definition of consciousness or as the blueprint for the Current-Episode Workspace. The source TME paper provides the fuller comparison with GWT (Global Workspace Theory) and other theories of consciousness; the present paper compares system architectures only where they overlap with the engineering problem. The question is not whether TME introduces memory, tools, planning, feedback, or specialized modules, but whether its canonical episode organization provides a distinct and testable form of correction before release. Table II presents this comparison.

TABLE II

COMPARISON WITH CURRENT FRONTIER AI SYSTEMS AND SYSTEM ARCHITECTURES

System or research direction	**Established contribution**	**Question retained by TME**
Frontier reasoning, tool-use, and agent platforms	Long context, retrieval, multimodality, reasoning, tools, subagents, computer use, permissions, monitoring, and human approval.	Are the governing concern, source provenance, candidate history, unresolved mismatch, alternatives, and release conditions maintained as one recurrent episode?
Autonomous world-model architecture: LeCun [14]	Perception, configurable world model, short-term memory, intrinsic cost, actor, hierarchical prediction, and planning.	Does the developing interpretation or action become an explicit review object that is segmented, compared, corrected, and re-presented before release?
Experience-based continual agents: Silver and Sutton [15]	Continuous learning from agent–environment interaction rather than dependence on static human-generated data.	How are episode-time correction and persistent system change separated, validated, governed, audited, and reversed?
Active-inference guidance: Da Costa et al. [16]	Unified state estimation, generative world models, uncertainty, planning, control, and adaptive robotics.	Are source, interpretation, competing alternatives, and the planned response kept distinct within a reflective correction episode?
Embodied vision-language-action systems: Gemini Robotics Team [17] and Kim et al. [18]	Vision-language-action control, embodied reasoning, robot adaptation, generalization across tasks, and physical outcome feedback.	Can physical outcome feedback correct the system's underlying conceptual premise and interpretation before repeated action, rather than being used primarily to refine action policies, trajectories, or low-level motor-control outputs?
World-foundation-model platforms for physical AI: NVIDIA Cosmos [19]	World-model generation, physical simulation, predictive infrastructure, synthetic data, and post-training resources for physical-AI development.	Can simulated and observed outcome mismatches be returned to a bounded epistemic episode that revises the governing premise or interpretation before model, policy, or action updates are accepted?
Neuro-symbolic architectures: Bhuyan et al. [20]	Neural learning combined with explicit knowledge representation, constraints, reasoning, and decision-making.	How does hybrid reasoning organize episode continuity, reflective delay, secondary segmentation, and discriminative correction before consequential release?

TME LLM OS proposal	Integrates these capabilities within a bounded recurrent episode and separates runtime correction from governed cross-episode refinement.	Requires implementation and comparison through grounding, recovery from false premises, safe release, recurrence of corrected errors, user burden, time-to-acceptable-result, and total task cost.

TME does not claim that current systems lack feedback, reasoning, world models, memory, tools, control, or learning from experience. Its proposed distinction is organizational: these capabilities are organized within a bounded correction episode before consequential release. This claim must be evaluated against the strongest available industrial and research systems rather than against a simplified LLM baseline.

VIII. Conclusions

1. Architectural claim. AI serves as a qualitative diagnostic case: memory, reasoning, tools, agents, world models, feedback, and continual learning can coexist without an episode-level mechanism that preserves the governing concern, source provenance, alternatives, residual mismatch, correction history, and release conditions until correction is provisionally complete. The limitation addressed here is therefore not probabilistic inference or capability shortage, but consequential release without an independently organized epistemic-correction process. TME therefore keeps candidate generation, review, correction, and execution distinct but recurrently connected within one bounded episode.
2. System organization. The TME LLM OS retains the Transformer-foundation-model core and organizes specialized agents, episode continuity, correction, and controlled release around it. The canonical but recurrent organization preserves the governing concern across cycles, re-presents correction as new current content, and permits task-bounded release without eliminating the associative variation required for alternatives and adaptation.
3. Development and operational prediction. During the current episode, correction modifies context, episodic state, grounding, candidates, routing, tools, or action without ordinarily changing learned parameters. Across episodes, validated corrections may refine persistent system components only through the governed, auditable, versioned, and reversible update path. The predicted benefit is not necessarily a cheaper local cycle, but lower total cost and shorter time to an acceptable result through reduced user correction, repeated prompting, rework, unsafe action, and recurrence of previously corrected errors.
4. Boundary and test. The source TME framework defines experience as present content lived within the real-time field of consciousness [2]. Basic wakefulness, sensation, emotion, orientation, target selection, and action readiness should not be reduced to cortical processing, while cortical networks expand cognition, language, planning, problem solving, and reflective self-awareness. The artificial architecture transfers functional organization rather than biological structures, and its measurable self-development can be evaluated independently of whether machine states are lived or only processed, represented, and reported. TME claims no priority for the individual capabilities compared in Section VII; its proposed contribution is their organization within a recurrent correction episode. Evaluation must compare that organization with the strongest available systems using the operational measures defined in Sections II-E and VII. Until implementation and comparison are complete, the TME LLM OS remains a systems-engineering hypothesis rather than a demonstrated epistemically valid or conscious machine.

Appendix A. Detailed Architecture Matrix

TABLE III
DETAILED ARCHITECTURE MATRIX

Architectural Role and TME Derivation	**High-Level Realization**	**Episode Function**	**Persistent Update**	**Authority**
Foundation-Model Substrate *Cognition, cortical specialization, cognitive patterning*	Generative LLM or coordinated foundation models shared by higher-level agents.	Representation, abstraction, probabilistic sequence modeling, candidate generation, interpretation, and planning support; parameters remain stable during ordinary inference.	Runtime-protected, versioned, governably writable, and reversible after validated cross-episode review.	Produces candidates; does not alone validate or authorize consequential release.

Architectural Role and TME Derivation	**High-Level Realization**	**Episode Function**	**Persistent Update**	**Authority**
Static Agents *Memory, learned constraints, current validation criteria, cultural-scientific correction*	Role-specific model use with protected policies, validated sources, safety constraints, schemas, and operational rules.	Expose stable reference conditions, evidence requirements, task boundaries, and safety envelopes.	Slow, governed, provenance-preserving revision; protected rather than immutable.	Flag violations, restrict transitions, require evidence, or block release.
Dynamic Agents *Cognition, imagination, prediction, interpretation, action readiness*	Foundation-model and specialized-model roles for perception, hypothesis generation, planning, simulation, and response formulation.	Adapt to current input, episode state, retrieved memory, and tool results; outputs remain revisable candidates.	Episode adaptation through context, working state, routing, and tools; persistent learning follows governance.	Generate and revise alternatives without independently releasing high-impact action.
Linking Agents and Interfaces *Availability from neighboring systems, temporal coordination, routing*	Routers, buses, protocol and schema adapters, gateways, queues, buffers, and learned mediation.	Preserve provenance, timing, identity, priority, and compatibility across agents, services, memory, sensors, tools, and action interfaces.	May combine protected protocols, transient buffers and routing state, and governably learnable mediation.	Move and synchronize information without converting transmission into validation.
Episode Context Controller *Working-memory workspace, ordinary awareness, sustained episodic orientation*	Agent-service combination using the shared model and current-episode workspace.	Maintains the governing concern, objective, source authority, provenance, candidates, mismatches, corrections, actions, and release criteria.	Episode state is temporary; selected validated lessons may be proposed for persistent storage.	Reintroduces unresolved distinctions, detects episodic drift, and requires continued review.
Self-Monitoring and Reflective Delay *Self-awareness, inhibitory control, constitutive reflective loop*	Higher-level agent representing the developing interpretation, impulse, plan, or response as a review object.	Suspends consequential release and reopens processing when the operation itself requires examination.	Decisions are logged; review policy changes require governed testing.	Hold, escalate, reacquire, or refer to segmentation and comparison without claiming phenomenal self-awareness.
Secondary-Segmentation Agent *Secondary segmentation of already salient contents*	Role-specific use of the model, provenance service, and episode state.	Separates source from interpretation, observation from prediction, premise from inference, memory from present input, and action from consequence. In embodied systems, this separation enables physical outcome feedback to challenge the underlying	Schemas may be refined from validated experience while distinctions remain traceable.	Prevents premature merging and prepares structured content for discriminative comparison.

Architectural Role and TME Derivation	**High-Level Realization**	**Episode Function**	**Persistent Update**	**Authority**
		conceptual premise and interpretation before repeated action, rather than being used primarily to refine action policies, trajectories, or low-level motor-control outputs.		
Discriminative-Intelligence Agent *Discriminative and embodied intelligence, experiential sedimentation*	Foundation model plus conceptual hierarchies, problem schemas, case memory, prior solution paths, tool knowledge, and validation criteria.	Recognizes problem structure, retrieves relevant experience, compares present and prior cases, selects or revises strategy, and adapts the route.	Validated lessons may update case memory, schemas, routing, tool policies, adapters, or parameters through governance.	Pursue, reroute, request information, qualify, abstain, or hand off; remembers solution methods rather than answers alone.
Epistemic Correction and Release Gating *Epistemic loop, correction, intelligent reorientation, reflective regulation, response boundary*	Agents and platform controls assembling correction packages, coordinating renewed inference, and governing execution.	Return mismatch, missing evidence, alternatives, uncertainty, safety status, and next steps; support release, qualification, reacquisition, fallback, abstention, or human handoff.	Correction history is auditable; persistent change requires separate approval and rollback.	Provides the provisional, task-bounded operational gate.
Shared System Services *Working memory, long-term memory, learning, bodily-neural support, external execution*	Episode workspace, provenance, retrieval, long-term and case memory, audit, access control, versioning, rollback, sensors, tools, and execution isolation.	Supply protected shared state and infrastructure without constituting consciousness, cognition, or discriminative intelligence.	Each service follows its own retention, authorization, provenance, update, and rollback policy.	Enforce boundaries and authorized routing, storage, retrieval, and action-control decisions.

Acknowledgment

Google Gemini, NotebookLM, and Microsoft Copilot were utilized throughout the project, serving as assistive tools for linguistic refinement, structural editing, citation verification, and the generation of accompanying graphics. This assistance involved manuscript text, reference entries, and the accompanying flowcharts (Figs. 1–5). These technologies did not alter the originality, core argumentation, or authorship of the work. The author reviewed and edited the AI-assisted material and assumes full responsibility for the manuscript's content, references, and graphics.

References

[1] I. Kander, "The essay: On the creation of the inner world and the possibility of its correction," 2021 (in Hebrew). [Online]. Available: https://www.ilankander.com/essay/

[2] I. Kander, “The TME Loop and Epistemic Correction: From Experience to Valid Knowledge: A Theoretical Framework for Conscious Feedback, Reflective Delay, and Cultural Validation,” [Preprint], Zenodo, 2026a. https://doi.org/10.5281/zenodo.23394497

[3] I. Kander, "The TME loop and epistemic correction: From experience to valid knowledge: A theoretical framework for conscious feedback, reflective delay, and cultural validation," (in Hebrew), in *Human–Science–Consciousness: Jubilee Volume in Honor of Oded Maimon*, A. Elqayam, Ed. Tel Aviv, Israel: Idra, 2026b, in press. ISBN 978-965-7876-37-4. Hebrew translation of the English preprint (2026a) cited in [2].

[4] B. Merker, “Consciousness without a cerebral cortex: A challenge for neuroscience and medicine,” *Behav. Brain Sci.*, vol. 30, no. 1, pp. 63–81, Feb. 2007, doi: 10.1017/S0140525X07000891. https://doi.org/10.1017/S0140525X07000891

[5] J. Panksepp, *Affective neuroscience: The foundations of human and animal emotions*. Oxford University Press, 1998, doi: 10.1093/oso/9780195096736.001.0001. https://doi.org/10.1093/oso/9780195096736.001.0001

[6] V. Menon, “Large-scale brain networks and psychopathology: A unifying triple network model,” *Trends Cogn. Sci.*, vol. 15, no. 10, pp. 483–506, Oct. 2011, doi: 10.1016/j.tics.2011.08.003. https://doi.org/10.1016/j.tics.2011.08.003

[7] D. Sridharan, D. J. Levitin, and V. Menon, “A critical role for the right fronto-insular cortex in switching between central-executive and default-mode networks,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 105, no. 34, pp. 12569–12574, Aug. 2008, doi: 10.1073/pnas.0800005105. https://doi.org/10.1073/pnas.0800005105

[8] Anthropic, *Building effective agents*, Anthropic Engineering, Dec. 19, 2024.

[9] Google DeepMind, *Gemini Robotics brings AI into the physical world*, Mar. 12, 2025. Google DeepMind.

[10] Meta, *Llama developer use guide: AI protections*, AI at Meta, Apr. 2025.

[11] Microsoft, *Multi-agent orchestration patterns and best practices*, Microsoft Learn, May 21, 2026.

[12] OpenAI, *Agents SDK*, OpenAI API documentation, 2026.

[13] B. J. Baars, *A cognitive theory of consciousness*. Cambridge University Press, 1988.

[14] Y. LeCun, “A path towards autonomous machine intelligence,” Version 0.9.2, [Position paper], OpenReview, 2022. https://openreview.net/pdf?id=BZ5a1r-kVsf

[15] D. Silver and R. S. Sutton, “Welcome to the era of experience,” 2025. [Preprint of a chapter to appear in *Designing an intelligence*, edited by G. Konidaris, MIT Press]. https://storage.googleapis.com/deepmind-media/Era-of-Experience%20/The%20Era%20of%20Experience%20Paper.pdf

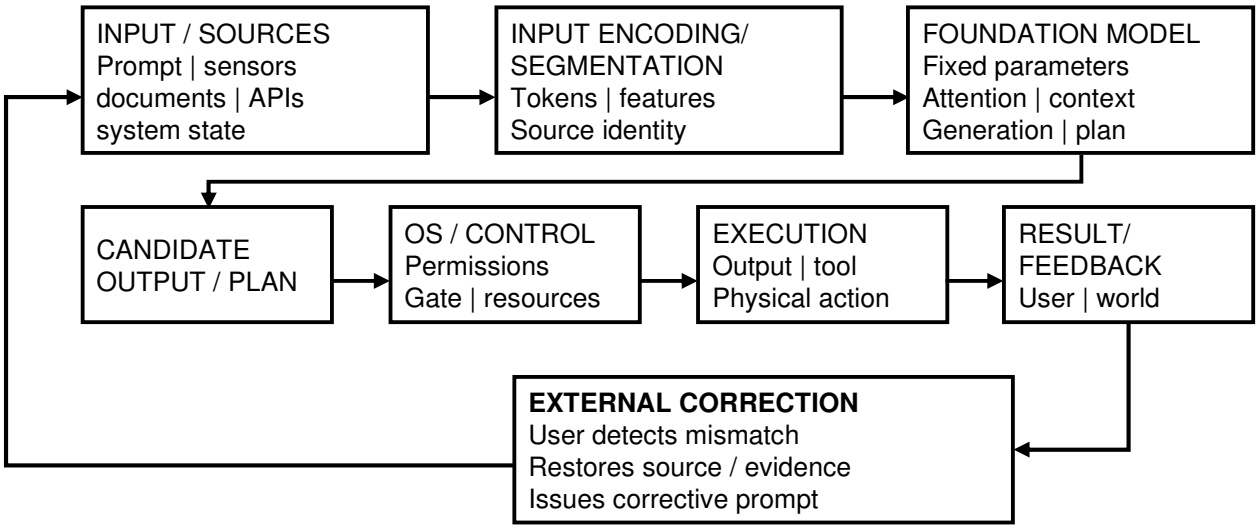
[16] L. Da Costa, P. Lanillos, N. Sajid, K. Friston, and S. Khan, “How active inference could help revolutionise robotics,” *Entropy*, vol. 24, no. 3, Art. no. 361, Mar. 2022, doi: 10.3390/e24030361. https://doi.org/10.3390/e24030361

[17] Gemini Robotics Team, “Gemini Robotics: Bringing AI into the physical world,” *arXiv*, 2025, doi: 10.48550/arXiv.2503.20020. https://doi.org/10.48550/arXiv.2503.20020

[18] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, “OpenVLA: An open-source vision-language-action model,” in *Proc. 8th Conf. Robot Learn.*, PMLR, 2025, pp. 2679–2713.

[19] NVIDIA, “Cosmos world foundation model platform for physical AI,” [Preprint], *arXiv*, 2025, doi: 10.48550/arXiv.2501.03575. https://doi.org/10.48550/arXiv.2501.03575

[20] B. P. Bhuyan, A. Ramdane-Cherif, R. Tomar, and T. P. Singh, “Neuro-symbolic artificial intelligence: A survey,” *Neural Comput. Appl.*, vol. 36, no. 21, pp. 12809–12844, 2024, doi: 10.1007/s00521-024-09960-z. https://doi.org/10.1007/s00521-024-09960-z



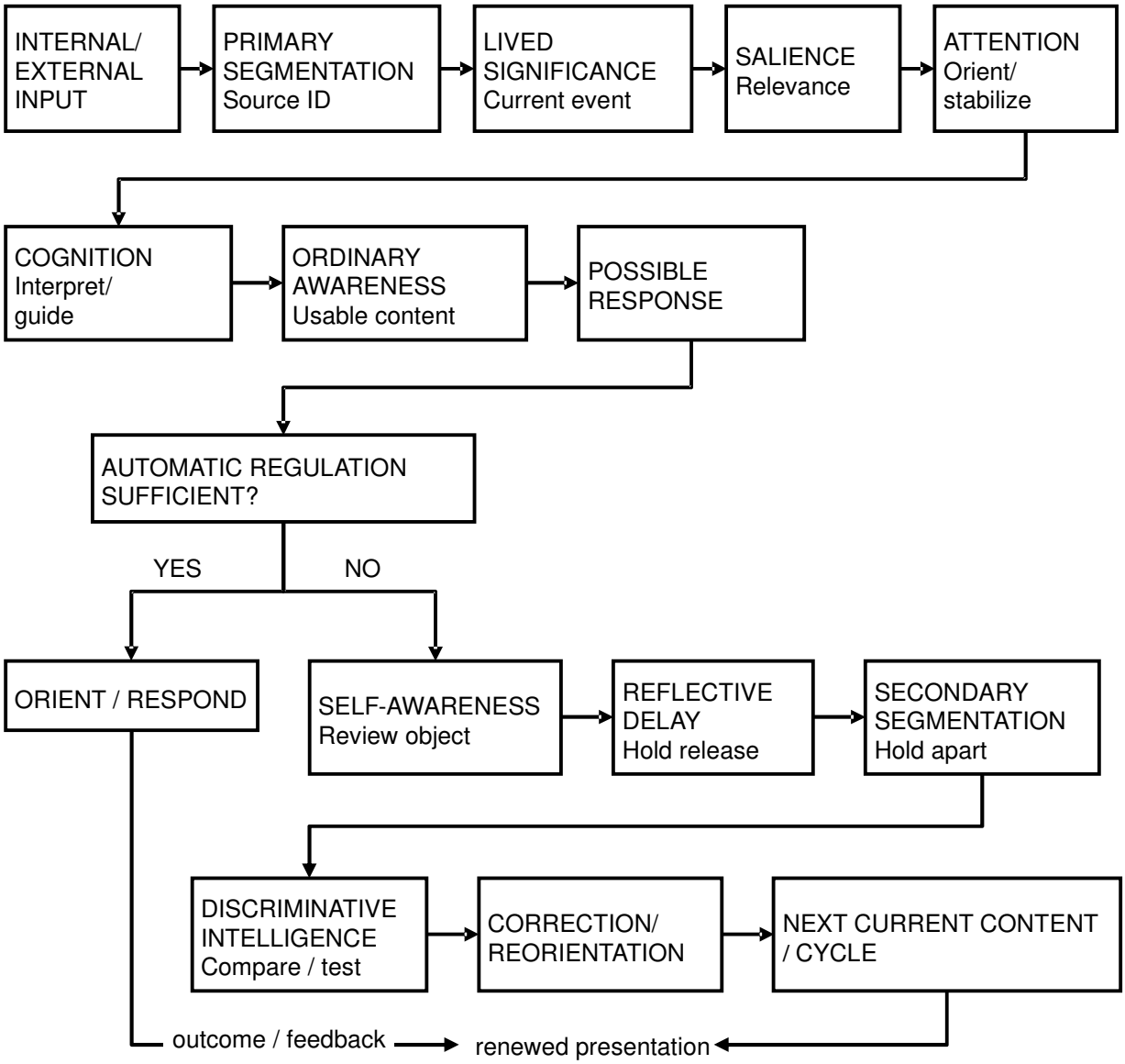
INTERNAL CONTINUATION

context + generated state → reasoning / self-critique / planning
→ same or closely related representations
→ greater coherence, not necessarily grounding

MISSING INTERNAL FUNCTION

No independently organized TME loop for reflective delay, provenance, secondary segmentation, discriminative comparison, and correction.

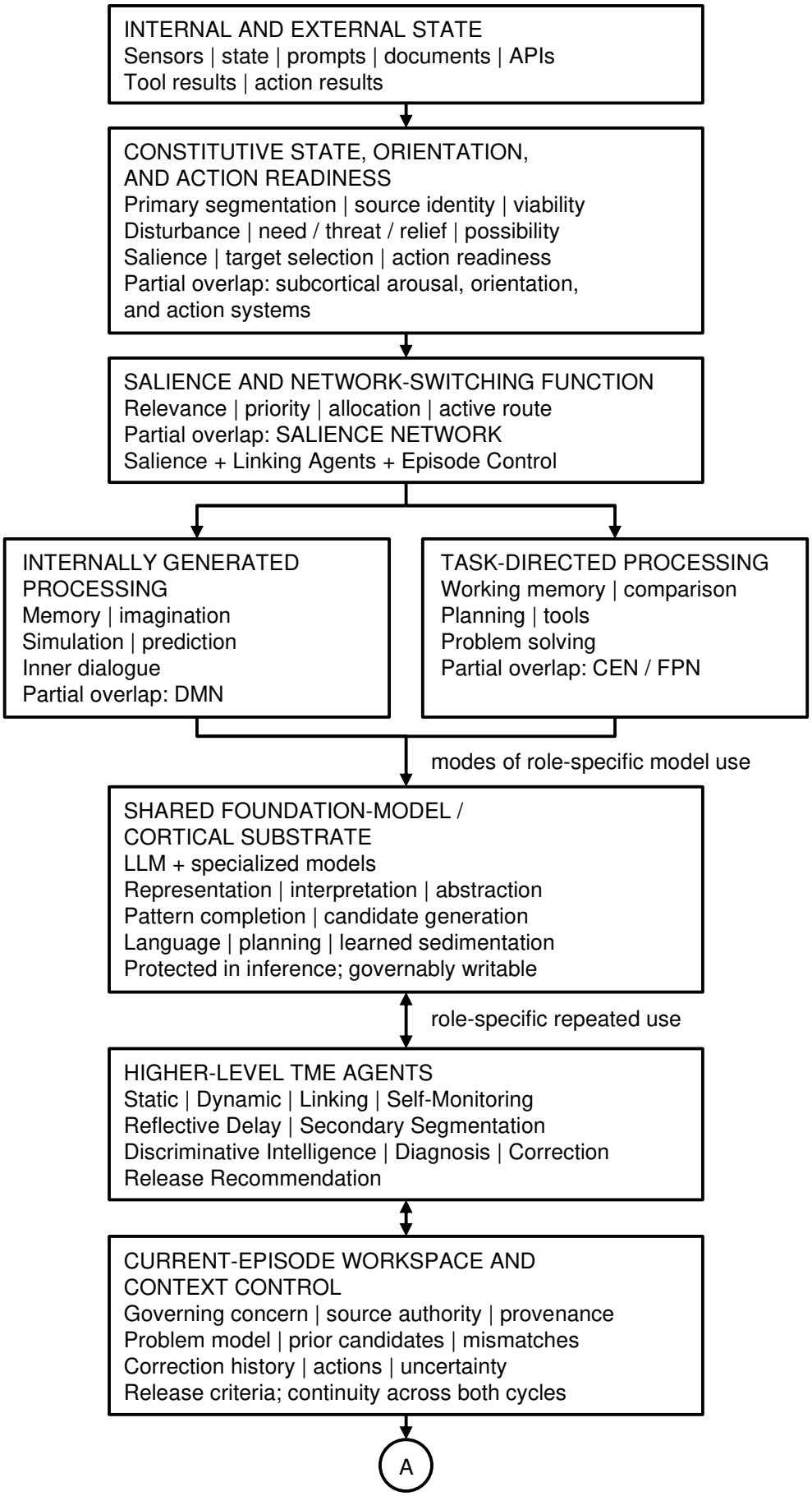
Fig. 1. Current LLM-OS Flow Without Internal Epistemic Closure

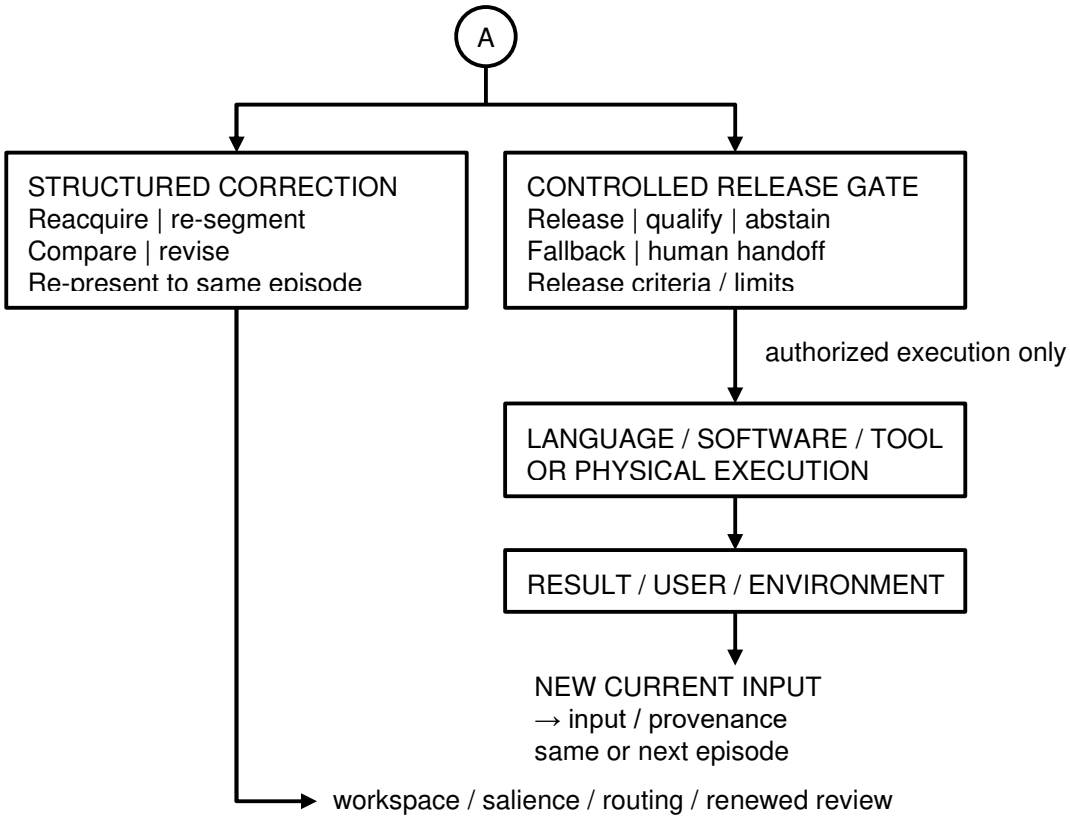


NEIGHBORING SYSTEMS
Working memory / current-episode workspace | long-term memory and learning
Bodily-neural support | execution systems

EPISTEMIC EXTENSION
Stabilized understanding → further testing and external verification
→ cultural-scientific correction → valid knowledge

Fig. 2. Canonical TME Loop

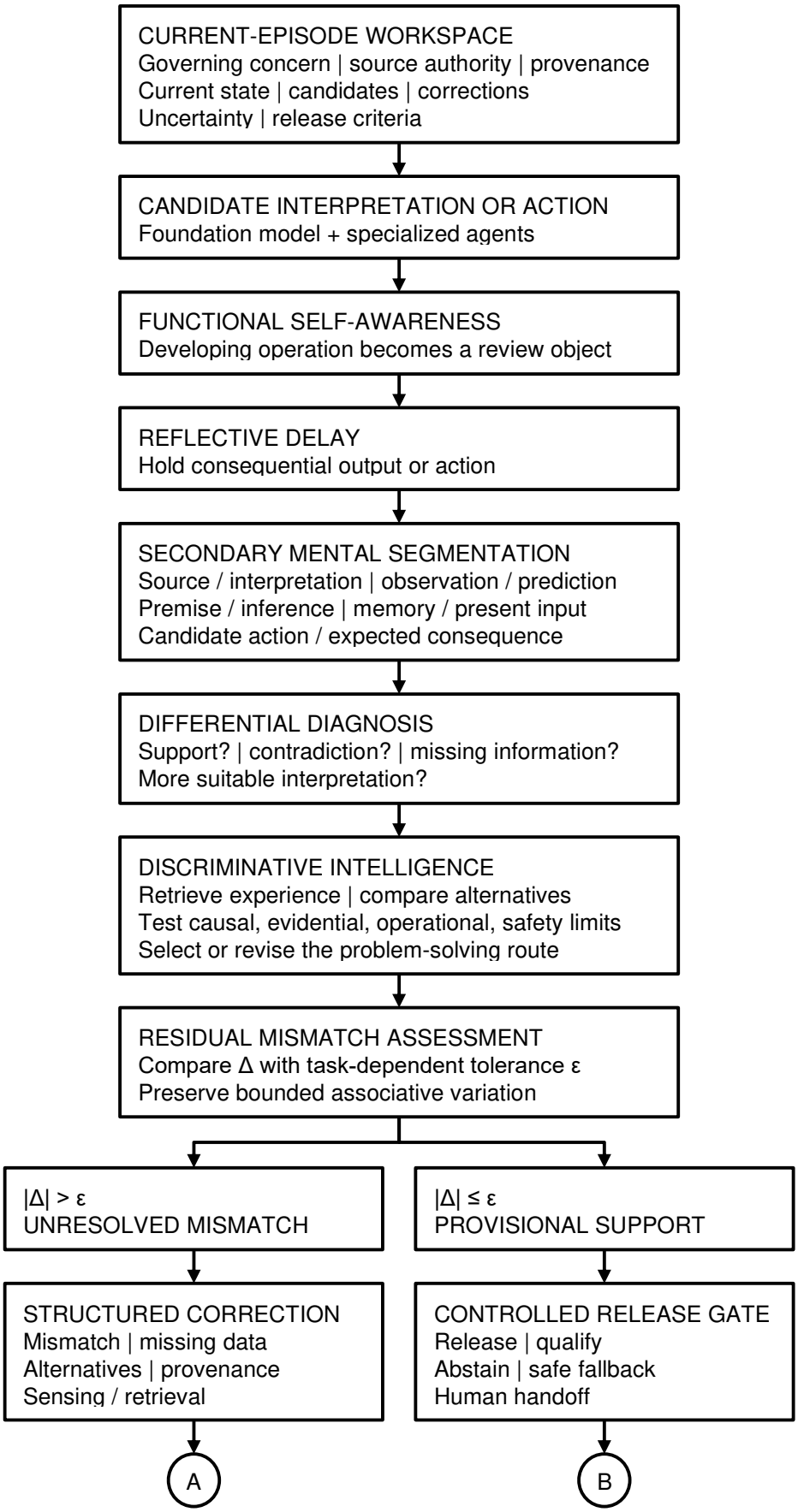

INTERNAL AND EXTERNAL STATE
Sensors | state | prompts | documents | APIs
Tool results | action results
CONSTITUTIVE STATE, ORIENTATION,
AND ACTION READINESS
Primary segmentation | source identity | viability
Disturbance | need / threat / relief | possibility
Salience | target selection | action readiness
Partial overlap: subcortical arousal, orientation,
and action systems
SALIENCE AND NETWORK-SWITCHING FUNCTION
Relevance | priority | allocation | active route
Partial overlap: SALIENCE NETWORK
Salience + Linking Agents + Episode Control
INTERNALLY GENERATED
PROCESSING
Memory | imagination
Simulation | prediction
Inner dialogue
Partial overlap: DMN
TASK-DIRECTED PROCESSING
Working memory | comparison
Planning | tools
Problem solving
Partial overlap: CEN / FPN
modes of role-specific model use
SHARED FOUNDATION-MODEL /
CORTICAL SUBSTRATE
LLM + specialized models
Representation | interpretation | abstraction
Pattern completion | candidate generation
Language | planning | learned sedimentation
Protected in inference; governably writable
role-specific repeated use
HIGHER-LEVEL TME AGENTS
Static | Dynamic | Linking | Self-Monitoring
Reflective Delay | Secondary Segmentation
Discriminative Intelligence | Diagnosis | Correction
Release Recommendation
CURRENT-EPISODE WORKSPACE AND
CONTEXT CONTROL
Governing concern | source authority | provenance
Problem model | prior candidates | mismatches
Correction history | actions | uncertainty
Release criteria; continuity across both cycles
A

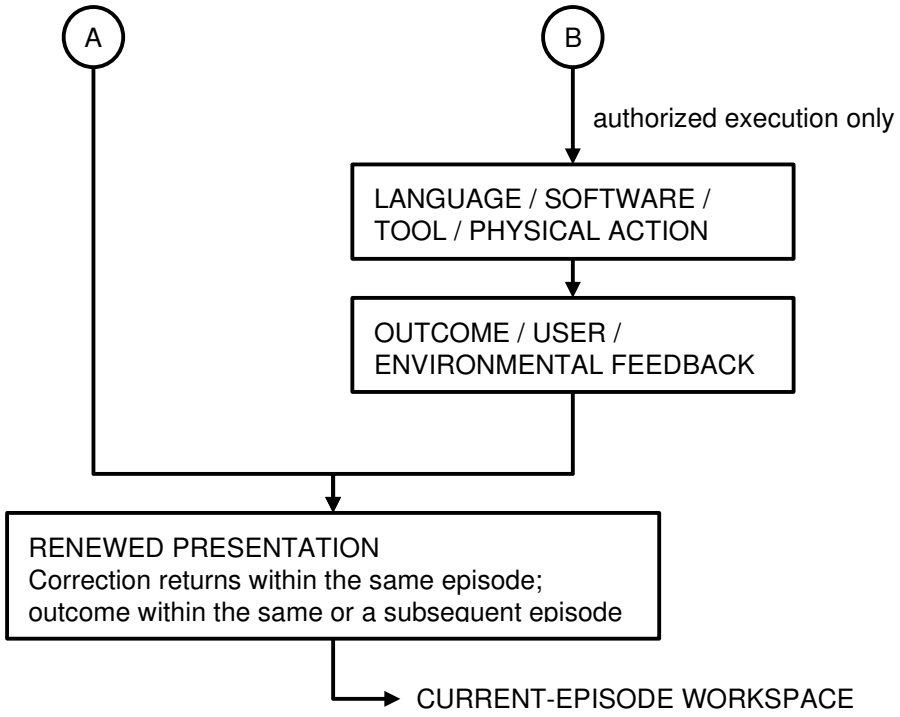


NEIGHBORING SERVICES
Working memory | long-term memory | case memory and schemas
Retrieval | provenance | audit | access control
Versioning | rollback | sensors | tools | execution interfaces

INTERPRETIVE BOUNDARY
DMN, CEN/FPN, and Salience Network provide partial functional overlap only. They neither define consciousness nor replace the canonical TME sequence. Current-state availability in the artificial architecture is a functional analogue, not a claim of experience.

Fig. 3. High-Level Artificial TME Architecture with Network Switching

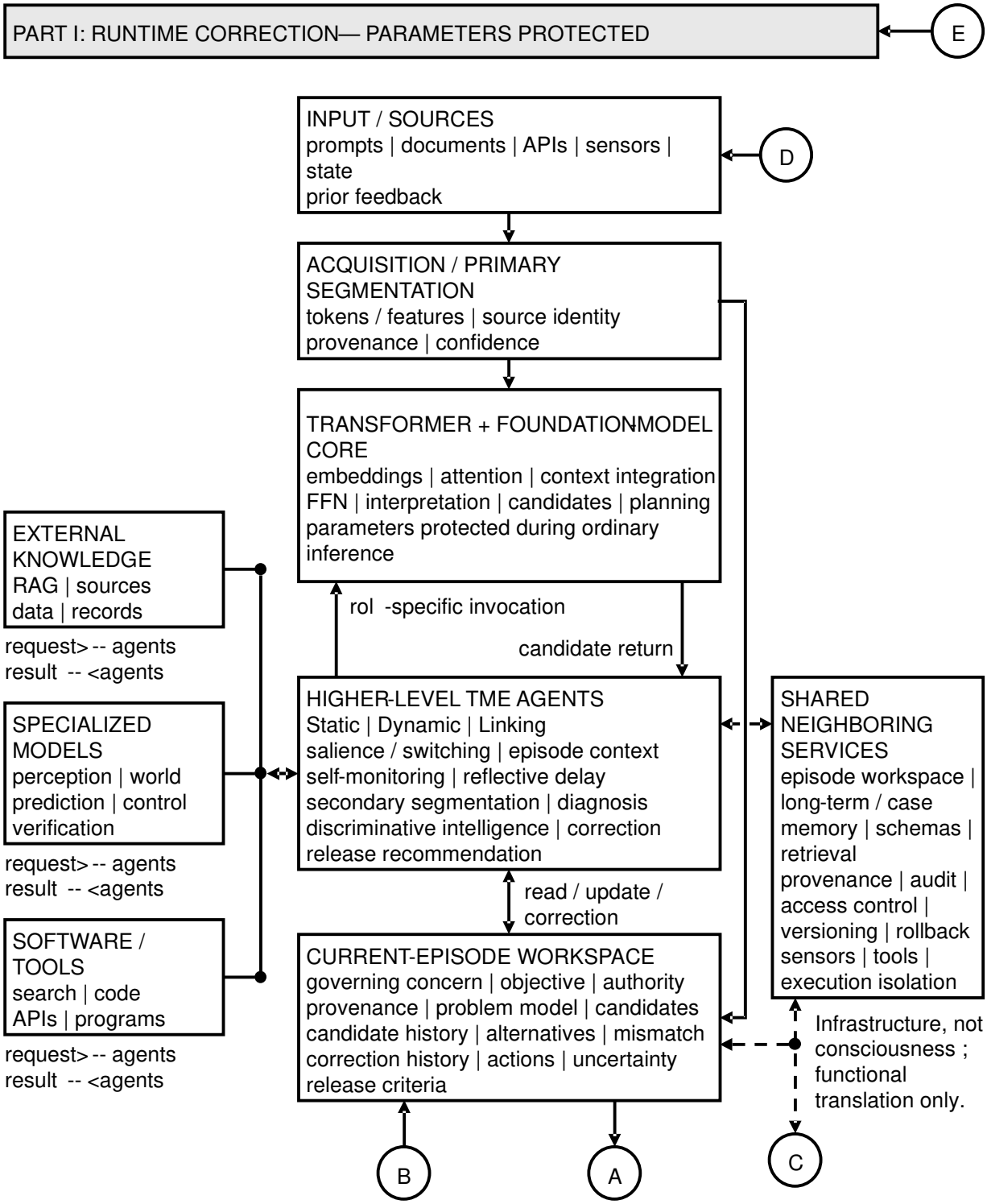

CURRENT-EPISODE WORKSPACE
Governing concern | source authority | provenance
Current state | candidates | corrections
Uncertainty | release criteria
CANDIDATE INTERPRETATION OR ACTION
Foundation model + specialized agents
FUNCTIONAL SELF-AWARENESS
Developing operation becomes a review object
REFLECTIVE DELAY
Hold consequential output or action
SECONDARY MENTAL SEGMENTATION
Source / interpretation | observation / prediction
Premise / inference | memory / present input
Candidate action / expected consequence
DIFFERENTIAL DIAGNOSIS
Support? | contradiction? | missing information?
More suitable interpretation?
DISCRIMINATIVE INTELLIGENCE
Retrieve experience | compare alternatives
Test causal, evidential, operational, safety limits
Select or revise the problem-solving route
RESIDUAL MISMATCH ASSESSMENT
Compare Δ with task-dependent tolerance ε
Preserve bounded associative variation
|Δ| > ε
UNRESOLVED MISMATCH
|Δ| ≤ ε
PROVISIONAL SUPPORT
STRUCTURED CORRECTION
Mismatch | missing data
Alternatives | provenance
Sensing / retrieval
CONTROLLED RELEASE GATE
Release | qualify
Abstain | safe fallback
Human handoff
A
B

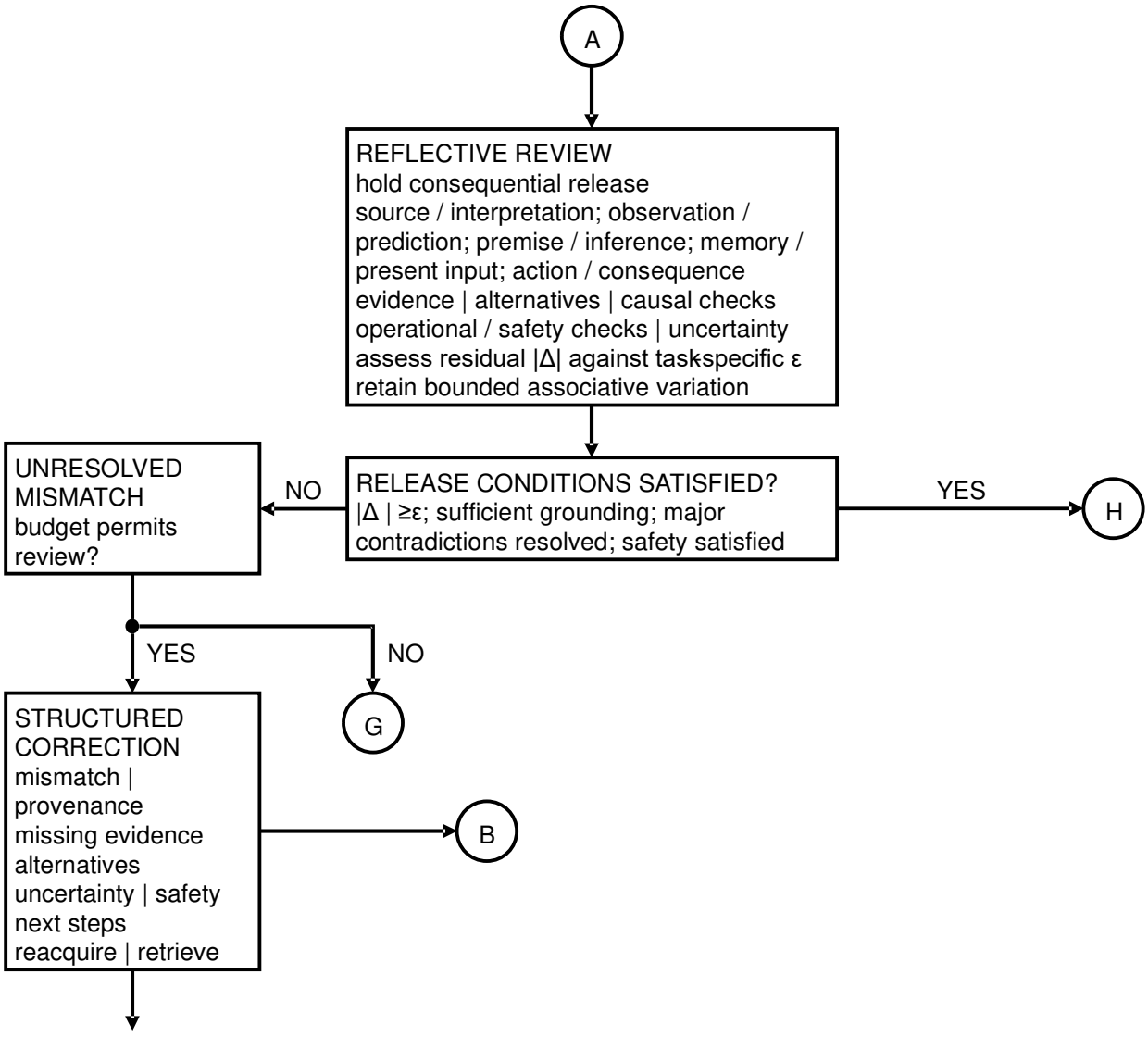


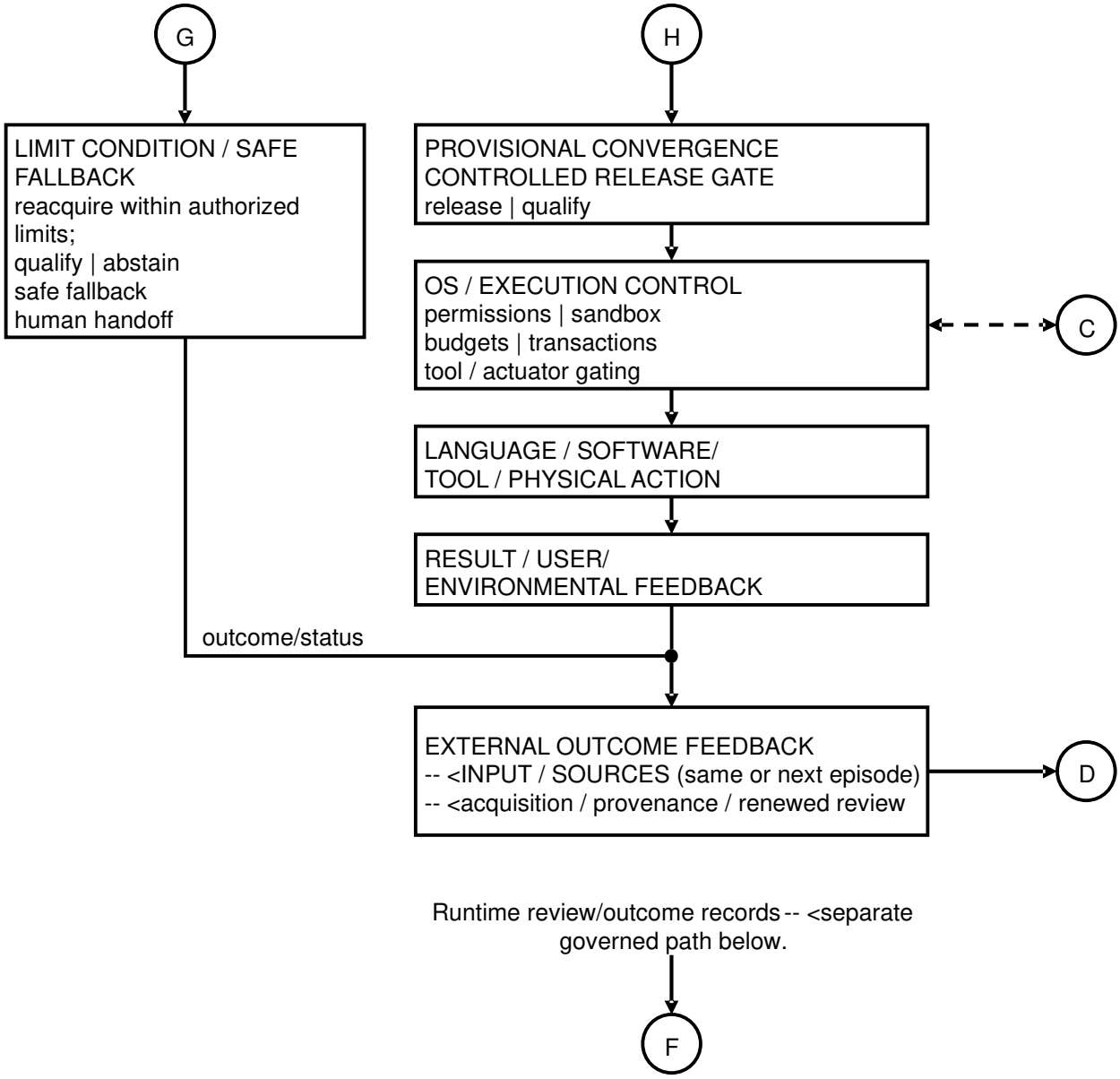
RELEASE AND LIMIT CONDITIONS
Release also requires sufficient grounding, resolved major contradictions, and satisfied safety constraints. Otherwise, return for structured correction within authorized review limits. If release conditions remain unmet at the permitted time, risk, computation, or retry limit: reacquire within authorized limits | qualify | abstain | safe fallback | human handoff.

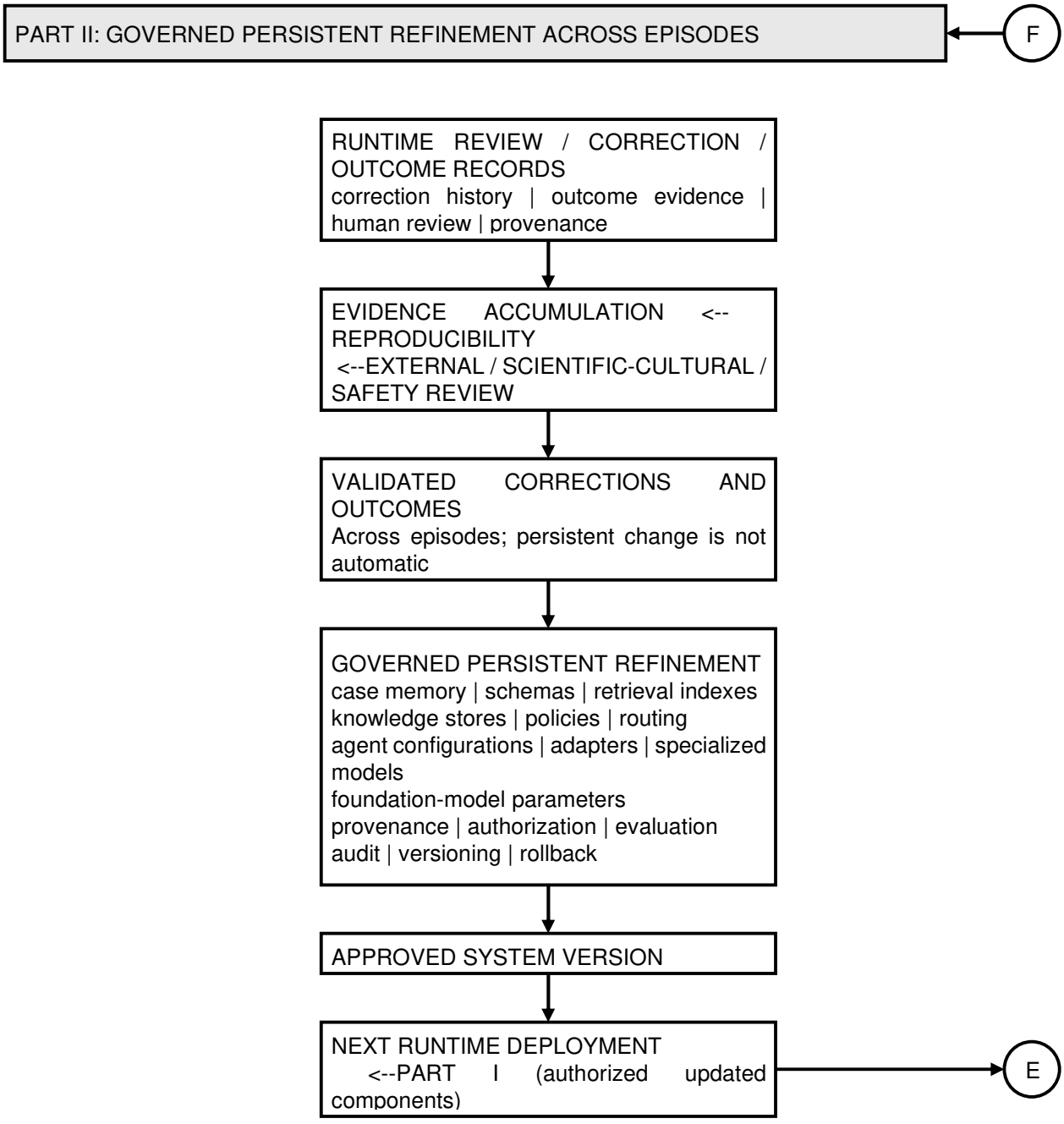
NEIGHBORING SERVICES
Working memory | long-term memory | case memory and schemas
Retrieval | provenance | sensors and tools | audit | access control
Versioning | rollback

Fig. 4. Recurrent Episode and Epistemic-Correction Flow


PART I: RUNTIME CORRECTION— PARAMETERS PROTECTED
E
INPUT / SOURCES
prompts | documents | APIs | sensors | state
prior feedback
D
ACQUISITION / PRIMARY SEGMENTATION
tokens / features | source identity
provenance | confidence
TRANSFORMER + FOUNDATIONMODEL CORE
embeddings | attention | context integration
FFN | interpretation | candidates | planning
parameters protected during ordinary inference
EXTERNAL KNOWLEDGE
RAG | sources
data | records
request> -- agents
result -- <agents
rol -specific invocation
candidate return
SPECIALIZED MODELS
perception | world prediction | control verification
request> -- agents
result -- <agents
HIGHER-LEVEL TME AGENTS
Static | Dynamic | Linking
salience / switching | episode context
self-monitoring | reflective delay
secondary segmentation | diagnosis
discriminative intelligence | correction
release recommendation
SHARED NEIGHBORING SERVICES
episode workspace | long-term / case memory | schemas | retrieval
provenance | audit | access control | versioning | rollback
sensors | tools | execution isolation
read / update / correction
SOFTWARE / TOOLS
search | code
APIs | programs
request> -- agents
result -- <agents
CURRENT-EPISODE WORKSPACE
governing concern | objective | authority
provenance | problem model | candidates
candidate history | alternatives | mismatch
correction history | actions | uncertainty
release criteria
Infrastructure, not consciousness ; functional translation only.
B
A
C


A
REFLECTIVE REVIEW
hold consequential release
source / interpretation; observation /
prediction; premise / inference; memory /
present input; action / consequence
evidence | alternatives | causal checks
operational / safety checks | uncertainty
assess residual |Δ| against task‑specific ε
retain bounded associative variation
RELEASE CONDITIONS SATISFIED?
|Δ | ≥ε; sufficient grounding; major
contradictions resolved; safety satisfied
NO
YES
H
UNRESOLVED
MISMATCH
budget permits
review?
YES
NO
G
STRUCTURED
CORRECTION
mismatch |
provenance
missing evidence
alternatives
uncertainty | safety
next steps
reacquire | retrieve
B
SAME-EPISODE STRUCTURED RETURN
-- <CURRENT-EPISODE WORKSPACE
-- <salience / routing prioritization
-- <agents / renewed review
New current content; no unrestricted parameter
update.


G
H
LIMIT CONDITION / SAFE FALLBACK
reacquire within authorized limits;
qualify | abstain
safe fallback
human handoff
PROVISIONAL CONVERGENCE CONTROLLED RELEASE GATE
release | qualify
OS / EXECUTION CONTROL
permissions | sandbox
budgets | transactions
tool / actuator gating
C
LANGUAGE / SOFTWARE/ TOOL / PHYSICAL ACTION
RESULT / USER/ ENVIRONMENTAL FEEDBACK
outcome/status
EXTERNAL OUTCOME FEEDBACK
-- <INPUT / SOURCES (same or next episode)
-- <acquisition / provenance / renewed review
D
Runtime review/outcome records -- <separate governed path below.
F



No unrestricted self-modification. Runtime correction and persistent refinement remains separate.

Fig. 5. TME LLM OS Architecture: Runtime Correction and Governed Refinement