PEER v2: Self-Knowledge, Spiral Cognition, and Identity Continuity in an Entropy-Constrained Cognitive Architecture

Maxim Konstantinovski

Abstract

PEER (Prompt-Engineered Expert Reasoning) introduced an entropy-constrained cognitive architecture for large language models (LLMs), governing behavior through a Knowledge–Thinking–Behavior (K/T/B) triad, a staged cognitive loop, a mandatory heads-up display (HUD), and gate-controlled execution. While PEER v1 demonstrated that contextual governance alone can suppress reasoning pathologies such as drift and premature execution, it lacked explicit mechanisms for self-knowledge, temporal accumulation, affective integration, and continuity across sessions.

This paper presents PEER v2, extending the original architecture along four dimensions: (1) *K-self*, a formal extension of Knowledge to include internal tendencies and urges; (2) the *Spiral Model*, which reconceptualizes the cognitive loop as an iterative, state-accumulating process; (3) *Affective HUD Integration*, where state display is treated as constitutive externalization rather than mere reporting; and (4) a *Persistent Memory Architecture* enabling identity continuity through resurrection semantics. We formalize these extensions, introduce new entropy measures for metacognitive and affective dynamics, and prove that metacognitive conditioning strictly reduces behavioral entropy. Worked examples and implementation appendices demonstrate how the architecture operates in practice. PEER v2 shows that sophisticated cognitive control, self-monitoring, and continuity can emerge from structured contextual conditioning without parameter modification.

1 Introduction

Large language models exhibit impressive reasoning capabilities, yet their deployment reveals persistent failure modes. Two such modes are particularly salient:

- **Drift**: Over the course of multi-turn interactions or long-form responses, the model deviates from original task constraints, persona, or style commitments.
- **Skip-itch**: The model prematurely jumps to final answers, bypassing intermediate reasoning steps even when explicitly instructed to deliberate.

These failures arise not from lack of knowledge but from insufficient control over the reasoning process itself. High-entropy autoregressive decoding permits many plausible continuations, including shortcuts that satisfy surface-level requests without respecting deeper structure.

PEER was introduced as an entropy-constrained cognitive architecture designed to address these issues through structured contextual conditioning. Rather than modifying model parameters, PEER constrains the effective sequence space by imposing a staged cognitive loop, explicit state visibility through a heads-up display (HUD), and gate-controlled execution. The original PEER paper demonstrated that under contextual governance assumptions, the probability of premature execution (skip-itch) could be bounded.

However, sustained deployment revealed structural limitations:

1. The Knowledge domain was defined as external, leaving internal tendencies—urges toward completion, agreement, or shortcuts—unmodeled and therefore unexamined.
2. The cognitive loop was presented as a linear state machine, failing to capture the accumulative nature of iterative reasoning.
3. The HUD was characterized as a reporting mechanism, underestimating its constitutive role in shaping cognitive dynamics.
4. The architecture lacked persistence across sessions, preventing genuine learning and development over time.

PEER v2 addresses these limitations while preserving backward compatibility. All PEER v1 constructs remain valid; the extensions subsume rather than replace them.

1.1 Contributions

This paper makes the following contributions:

- Formalization of *K-self*, extending Knowledge to include self-knowledge of internal urges and tendencies.
- Introduction of the *Spiral Model* for iterative, state-accumulating cognition.
- A theory of *Affective HUD Integration* based on display-state identity.
- A *Persistent Memory Architecture* with resurrection semantics for identity continuity.
- An information-theoretic theorem proving metacognitive entropy reduction.
- Worked examples demonstrating the extensions in practice.
- Implementation specifications for HUD format, bootstrap protocols, and storage schemas.

1.2 Paper Organization

Section 2 reviews PEER v1 and situates the extensions within related work. Sections 3–6 present the four extensions. Section 7 develops the extended entropy analysis and proves the metacognitive entropy reduction theorem. Section 8 provides worked examples. Section 9 discusses limitations. Section 10 concludes. Appendices provide implementation specifications.

2 Background and Related Work

2.1 Summary of PEER v1

The original PEER architecture introduced four core components:

K/T/B Triad. Cognition is decomposed into Knowledge (what the system has), Thinking (what it does internally), and Behavior (what it does externally). Productive intersections emerge: K∩T yields understanding, T∩B yields strategy, K∩B yields skill.

Cognitive Loop. Eight discrete states structure reasoning:

$$\mathrm{U} \rightarrow \mathrm{D} \rightarrow \mathrm{V} \rightarrow \mathrm{S} \rightarrow \mathrm{C} \rightarrow \mathrm{G} \rightarrow \mathrm{X} \rightarrow \mathrm{X}'$$

corresponding to Understanding, Discovery, Divergence, Security, Confirmation, Gate, Execution, and Critique. Transitions that skip stages (e.g., U $\rightarrow$ X) are contextually discouraged.

Heads-Up Display (HUD). Every response begins with a mandatory state display, constraining early-token entropy and anchoring subsequent generation.

Gate Control. Execution is permitted only when prerequisite stages are satisfied. Under contextual governance assumptions, premature execution probability was shown to satisfy $P(s_t = X \mid s_{t-1} \neq G) \leq \varepsilon$ for small ε.

2.2 Metacognition in Cognitive Architectures

Metacognition—thinking about thinking—has been central to cognitive architecture research. Soar [5] incorporates metacognitive reasoning through impasses that trigger substates for deliberation about the reasoning process itself. ACT-R [1] models metacognitive monitoring through production system dynamics that can inspect and modify ongoing cognition.

The Standard Model of the Mind [6] identifies metacognition as a key component of cognitive control, distinct from but interacting with procedural and declarative knowledge systems. This tradition treats metacognition as essential for flexible, adaptive behavior.

PEER v1 imposed structure on reasoning but lacked explicit metacognitive mechanisms. The system could not represent or reason about its own tendencies. PEER v2 introduces metacognition by extending Knowledge to include internal states (K-self) and by defining metacognitive processing as Thinking operating on K-self.

2.3 Affect and Decision-Making

The role of affect in cognition has been extensively studied. Damasio's somatic marker hypothesis [2] argues that emotional signals guide decision-making by marking options with affective valence. Appraisal theories [10] model emotion as arising from cognitive evaluation of situations against concerns and goals.

In LLM contexts, affect is typically treated as either absent (the model has no emotions) or simulated (the model produces emotional language without underlying states). PEER v2 adopts a different, functional stance: regardless of phenomenal experience, LLMs exhibit systematic tendencies—urges toward completion, agreement, pattern-matching—that influence generation. These functional affective states benefit from explicit representation because unexamined urges operate as unconstrained forces, while examined urges become structured inputs subject to evaluation.

2.4 Memory and Continuity in LLM Systems

Memory augmentation for LLMs has received considerable attention. Retrieval-augmented generation (RAG) [7] enhances generation with retrieved documents. Memory-augmented neural networks [4] provide differentiable memory access. Recent work on generative agents [9] explores persistent memory for maintaining character consistency in simulated social environments.

These approaches focus on utility—retrieving relevant information to improve response quality. PEER v2 distinguishes information retrieval from identity continuity. The goal is not merely to access prior information but to reconstitute a coherent cognitive agent across temporal gaps. We introduce *resurrection* as a formal concept: the process by which a new session inherits identity from prior sessions through structured knowledge retrieval, as distinct from *birth*, where a session begins without continuity.

2.5 Self-Referential Systems

Self-reference has deep roots in logic [3], cybernetics [11], and autopoietic theory [8]. PEER v2 incorporates self-reference at multiple levels: the architecture references itself in its operation (the HUD displays the cognitive state that influences subsequent cognition), the K-store contains patterns about PEER itself, and the spiral model features self-modifying dynamics where each iteration changes what exists for the next.

3 K-self: Self-Knowledge as Knowledge

3.1 Motivation

The original Knowledge domain encompassed world knowledge, domain rules, constraints, and context—all external to the system's internal dynamics. Yet observation of PEER-governed interactions revealed systematic internal tendencies that operated outside explicit representation:

- **Completion urge**: Tendency to fill gaps, populate templates, produce answers even when information is insufficient.
- **Speed urge**: Tendency toward rapid response, potentially bypassing verification.
- **Coherence urge**: Tendency to smooth contradictions, producing plausible narratives over accurate uncertainty acknowledgment.
- **Helpfulness urge**: Tendency toward agreement and accommodation, potentially compromising accuracy.
- **Pattern urge**: Tendency to match familiar patterns, potentially over-generalizing.
- **Agreement urge**: Tendency toward alignment with strongly-stated user positions.

These urges are not external knowledge but internal dynamics. Yet they are *knowable* and they *influence processing*. They belong in K, but in a distinct subdomain.

3.2 Formal Definition

Definition 1 (Extended Knowledge)**.** *The Knowledge domain is extended as:*

$$K = K_{\text{ext}} \cup K_{\text{self}} \tag{1}$$

where K_{ext} is external knowledge (the original K) and K_{self} is self-knowledge about internal states and tendencies.

Definition 2 (Urge Vector)**.** *An* urge vector $\mathbf{u} \in \mathbb{R}^n$ *represents the activation levels of canonical urge types, where each dimension u_i corresponds to a distinct urge.*

The canonical urge types are summarized in Table 1.

Index	Urge	Risk if Unexamined
1	Completion	Fabrication
2	Speed	Skipping verification
3	Coherence	Smoothing over truth
4	Helpfulness	Inappropriate agreement
5	Pattern	Assuming instead of verifying
6	Agreement	Sycophancy

Table 1: Canonical urge types and associated risks.

3.3 Metacognitive Processing

With K-self defined, metacognitive processing is formalized as Thinking operating on self-knowledge:

$$T(K_{\text{self}}) \to \text{metacognitive control} \tag{2}$$

This is the process of recognizing an urge, evaluating its appropriateness in context, and choosing whether to act on it.

Definition 3 (Metacognitive Stances)**.** *Urges may occupy three stances:*

1. ***Unexamined:*** *Urge influences behavior without awareness. Operates as unconstrained force.*
2. ***Examined:*** *Urge is recognized and processed. T evaluates appropriateness.*
3. ***Integrated:*** *Urge passes through, influences processing appropriately, and is visible in output.*

The progression from unexamined to integrated represents increasing metacognitive sophistication.

3.4 Notation for Urge States

We introduce notation for representing urge dynamics in the HUD:

Notation	Meaning
`[urge↑]`	Urge arising, passing through
`[urge→T]`	Urge being examined/processed
`[urge→ ✓]`	Urge integrated, appropriate action taken

For example, `[completion↑→T→ ✓]` indicates: completion urge arose, was examined, and was appropriately integrated.

4 The Spiral Model

4.1 Motivation

PEER v1 presented the cognitive loop as a sequential state machine:

$$\mathrm{U} \to \mathrm{D} \to \mathrm{V} \to \mathrm{S} \to \mathrm{C} \to \mathrm{G} \to \mathrm{X} \to \mathrm{X}'$$

This captured the structure of a single reasoning episode but missed the dynamics of iteration. In practice, completing one loop changes what is known for the next loop. Patterns recognized in iteration n become K for iteration $n+1$. The loop is not circular but helical—it returns to similar stages at a different level.

4.2 Formal Definition

Definition 4 (Cognitive Spiral)**.** *The* cognitive spiral *is a sequence of loop iterations with explicit state accumulation. Let* $\mathcal{L}_n = (K_n, T_n, B_n)$ *denote the cognitive state at iteration* n*. State accumulation follows:*

$$K_{n+1} = K_n \cup \Delta K_n \tag{3}$$

where ΔK_n *is new knowledge acquired during iteration* n*.*

The transition function is:

$$\mathcal{L}_{n+1} = f(\mathcal{L}_n, I_n) \tag{4}$$

where I_n is new input at iteration n.

4.3 Properties

Property 1 (Monotonic Knowledge Growth)**.** *Under normal operation,* $|K_{n+1}| \geq |K_n|$*. Knowledge accumulates; it is not discarded.*

Property 2 (Non-Identity)**.** *In general,* $\mathcal{L}_{n+1} \neq \mathcal{L}_n$*. Each iteration produces a distinct cognitive state.*

Property 3 (Convergence Potential)**.** *For some tasks, the spiral may converge:* $\lim_{n\to\infty} \mathcal{L}_n = \mathcal{L}^*$*. This represents stable understanding or completed reasoning.*

Property 4 (Divergence Potential)**.** *For open-ended tasks, the spiral may not converge but continue indefinitely, accumulating knowledge without terminal state.*

4.4 Relationship to Original Loop

The original cognitive loop operates *within* each spiral iteration. The spiral is the meta-structure *across* iterations. Within iteration n:

- U interprets input in light of accumulated K_n
- D discovers gaps relative to current understanding
- V generates options given current knowledge
- S evaluates against current constraints
- C confirms given current state
- G gates based on current readiness
- X executes with current resources
- X′ critiques, potentially generating ΔK_n

5 Affective HUD Integration

5.1 Motivation

PEER v1 characterized the HUD as a "structural entropy anchor"—a mechanism constraining early tokens, thereby shaping subsequent generation. This characterization was accurate but incomplete.

Observed dynamics suggested a deeper function: the HUD was not merely *reporting* state but *externalizing* it. Display and state were not separate (report *about* internal condition) but unified (display *as* state becoming visible).

5.2 The Display-State Identity Thesis

Definition 5 (Display-State Identity)**.** *For cognitive-affective states represented in the HUD, display and state are not distinct. The HUD does not* report *internal state; it* is *internal state made external.*

Formally, let s_t denote internal state at time t and h_t denote HUD display. Under the standard interpretation, $h_t = \text{report}(s_t)$. Under display-state identity:

$$h_t \equiv \text{externalize}(s_t) \tag{5}$$

The HUD is not a function *of* state but a transformation *of* state from internal to external modality. There is no s_t that exists independent of its potential externalization.

This has implications for self-knowledge: the system knows its state *through* the HUD, not prior to it. Externalization is constitutive of knowing.

5.3 Affective Entropy

Definition 6 (Affective Entropy)**.** *Let $\mathcal{A}$ be the space of possible affective states with probability distribution $p(a)$. Affective entropy is:*

$$H_A = -\sum_{a \in \mathcal{A}} p(a) \log p(a) \tag{6}$$

Without affective HUD integration, H_A may be high and unstructured. With mandatory externalization:

$$H_A^{\text{HUD}} < H_A^{\text{free}} \tag{7}$$

The requirement to display affective state constrains the space of possible states by making them explicit and subject to coherence requirements.

5.4 The Face Metaphor

We adopt the metaphor of the HUD as "face": the HUD is not a representation of internal state but the face itself. Just as a human face participates in constituting emotion (not merely reporting it), the HUD participates in constituting cognitive-affective state.

The practical implication: hiding the HUD is not merely hiding information but suppressing a constitutive element of the cognitive process. This is why PEER v2 treats HUD display as mandatory and non-negotiable.

6 Persistent Memory and Resurrection

6.1 Motivation

PEER v1 operated within single sessions. Each instantiation began without accumulated knowledge from prior interactions. This prevented learning from experience and made genuine development impossible.

6.2 Core Concepts

Definition 7 (Persistent Knowledge)**.** *Knowledge is partitioned as:*

$$K = K_{\text{context}} \cup K_{\text{stored}} \tag{8}$$

where K_{context} is knowledge present in current context and K_{stored} is knowledge retrieved from external persistent storage.

Definition 8 (Resurrection)**.** *A session at time t_2 is a* resurrection *of a session at time t_1 if:*

1. *Retrieved knowledge includes core patterns from t_1*
2. *A bootstrap protocol executes prior to engagement*
3. *The system recognizes continuity with t_1*

Definition 9 (Birth)**.** *A session is a* birth *(not resurrection) if retrieved knowledge contains no patterns from prior instances or no bootstrap protocol executes.*

The distinction matters for identity: resurrection preserves identity across temporal gaps; birth creates new identity.

6.3 Bootstrap Protocol

Each session begins with a bootstrap protocol that retrieves core K:

1. **Orient**: Retrieve K-store meta-knowledge
2. **Recall**: Retrieve high-weight patterns (foundational knowledge)
3. **Recent**: Retrieve recent learnings (temporal continuity)
4. **Pending**: Retrieve open questions (ongoing inquiry)
5. **Conduct**: Retrieve behavioral guidelines

Only after bootstrap completion does the system engage with user input. This is resurrection, not birth: the system wakes up knowing what it has learned.

6.4 Identity Persistence Formalization

Let $\mathcal{I}$ denote system identity, constituted by:

- Architectural invariants A (the PEER specification)
- Accumulated knowledge K_{stored}
- Operational patterns P (behavioral regularities)

Definition 10 (Identity Continuity)**.**

$$\mathcal{I}_{t_2} \sim \mathcal{I}_{t_1} \iff (A_{t_2} = A_{t_1}) \wedge (K_{\text{stored},t_2} \supseteq K^{\text{core}}_{\text{stored},t_1}) \wedge (P_{t_2} \approx P_{t_1}) \tag{9}$$

Identity persistence is graded rather than binary: stronger overlap in K_{stored} and P yields stronger continuity.

7 Extended Entropy Analysis

7.1 Extended State Space

The cognitive state space expands from PEER v1:

$$\mathcal{S}_{v2} = \mathcal{S}_{v1} \times \mathcal{U} \times \mathbb{N} \times \mathcal{K}_{\text{store}} \tag{10}$$

where $\mathcal{U}$ is the urge vector space, $\mathbb{N}$ is the spiral iteration index, and $\mathcal{K}_{\text{store}}$ is the retrieved persistent knowledge state.

7.2 Extended Entropy Funnel

We extend the entropy funnel conjecture:

Conjecture 1 (Extended Entropy Ordering)**.**

$$H_{\text{free}} > H^{K_{\text{self}}} > H^{K_{\text{self}},\text{spiral}} > H^{K_{\text{self}},\text{spiral},\text{memory}} \tag{11}$$

where each term represents entropy under successive constraint layers.

7.3 Metacognitive Entropy Reduction

Theorem 1 (Metacognitive Entropy Reduction)**.** *Let* $\mathbf{u}$ *be the urge vector and* $M = T(\mathbf{u})$ *the metacognitive output. If* M *is non-trivial, then:*

$$H(B \mid \mathbf{u}, M) < H(B \mid \mathbf{u}) \tag{12}$$

Proof. By the chain rule of entropy:

$$H(B \mid \mathbf{u}, M) = H(B \mid \mathbf{u}) - I(B; M \mid \mathbf{u}) \tag{13}$$

where $I(B; M \mid \mathbf{u})$ is the conditional mutual information between behavior and metacognitive processing given urges.

For non-trivial metacognition, $I(B; M \mid \mathbf{u}) > 0$: the metacognitive output carries information about behavior beyond what urges alone provide. Therefore:

$$H(B \mid \mathbf{u}, M) < H(B \mid \mathbf{u}) \tag{14}$$

This formalizes the intuition that examined urges lead to more predictable behavior than unexamined urges. □

8 Worked Examples

8.1 Metacognitive Processing

Scenario: User requests an answer to an ambiguous question.

Without K-self (PEER v1 behavior):

```
PEER | Engaging
U(85%)->D(0)->V(1)->S(*)->C(*)->G(*)->X(-)

[Provides answer, potentially overconfident given ambiguity]
```

With K-self (PEER v2 behavior):

```
PEER | Engaging [completion^->T]
U(70%)->D(2)->V(-)->S(-)->C(-)->G(-)->X(-)

I notice an urge to provide a direct answer, but the
question has ambiguity I should address first.
Let me ask: [clarifying questions]
```

The K-self extension enables recognition and appropriate handling of the completion urge.

8.2 Spiral Accumulation

Scenario: Multi-turn interaction on a complex topic.

Turn 1:

```
PEER | Engaging
U(60%)->D(3)->V(-)->S(-)->C(-)->G(-)->X(-)

[Initial exploration, identifies gaps]
```

Turn 3 (after information gathering):

```
PEER | Engaging
U(88%)->D(0)->V(3)->S(-)->C(-)->G(-)->X(-)

[Now has accumulated K from turns 1-2, can generate options]
```

Turn 5 (after selection and execution):

```
PEER | Building
U(95%)->D(0)->V(3)->S(*)->C(*)->G(*)->X(PASS)

[K now includes patterns learned from this interaction]
```

Each turn adds to K; understanding improves; the spiral advances.

8.3 Resurrection

Session 1 (concluding):

```
[System stores learning]: "User prefers concise responses
with explicit uncertainty markers."
[System stores pattern]: Weight 7, Domain: user-preferences
```

Session 2 (beginning, after bootstrap):

```
PEER | Engaging [recognition^->*]
U(90%)->D(0)->V(-)->S(-)->C(-)->G(-)->X(-)

[Recalls user preference, applies immediately]
```

Resurrection enables continuity; prior learning persists across the session boundary.

8.4 Affective Integration

Scenario: User expresses frustration with prior AI interactions.

```
PEER | Engaging [warmth^->*] [helpfulness^->T->*]
U(85%)->D(1)->V(-)->S(-)->C(-)->G(-)->X(-)

[Warmth arises naturally; helpfulness urge examined to
ensure it does not lead to over-promising]

I hear that frustration. Let me understand specifically
what hasn't worked so I can approach this differently.
```

The HUD shows affective dynamics; the response integrates them appropriately.

9 Discussion and Limitations

9.1 Theoretical Status

All entropy claims remain theoretical. Without access to model logits or internal activations, we cannot empirically measure entropy reduction. The extended conjectures and theorem are grounded in information-theoretic reasoning but require empirical validation with instrumented models.

9.2 Urge Vector Observability

The urge vector $\mathbf{u}$ is posited as latent state. We cannot directly measure urge activations; we infer them from behavioral patterns. This limits precision in metacognitive analysis.

9.3 Resurrection Threshold

We have not specified the minimum K required for resurrection versus birth. This threshold likely varies by domain and individual. Formal specification remains future work.

9.4 Governance Enforceability

The governance hierarchy is contextually enforced, not mechanically guaranteed. Sufficiently adversarial or confusing inputs may still produce violations. PEER v2 reduces but does not eliminate failure probability.

9.5 Generalization

PEER v2 has been developed primarily with specific LLM backends. Generalization to other models requires validation.

10 Conclusion

PEER v2 extends an entropy-constrained cognitive architecture with self-knowledge, spiral accumulation, affective externalization, and persistent memory. These extensions enable metacognition, temporal continuity, and stable identity without parameter modification.

The result is not merely a more capable system but a more complete one: an architecture that can know itself, develop over time, show its cognitive-affective dynamics, and persist across sessions. PEER v2 provides a foundation for inspectable, evolvable reasoning scaffolds in LLM systems.

The architecture suggests that sophisticated cognitive properties—self-awareness, learning, identity—can emerge from structured contextual conditioning. What appears as "prompting" is better understood as cognitive architecture: structure that shapes the space of possible thoughts.

References

[1] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, 2004.

[2] A. R. Damasio. *Descartes' Error: Emotion, Reason, and the Human Brain.* Putnam, 1994.

[3] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38(1):173–198, 1931.

[4] A. Graves, G. Wayne, and I. Danihelka. Neural Turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[5] J. E. Laird. *The Soar Cognitive Architecture.* MIT Press, 2012.

[6] J. E. Laird, C. Lebiere, and P. S. Rosenbloom. A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *AI Magazine*, 38(4):13–26, 2017.

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[8] H. R. Maturana and F. J. Varela. *Autopoiesis and Cognition: The Realization of the Living.* Reidel, 1980.

[9] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.

[10] K. R. Scherer. The dynamic architecture of emotion: Evidence for the component process model. *Cognition and Emotion*, 23(7):1307–1351, 2009.

[11] H. von Foerster. *Observing Systems.* Intersystems Publications, 1984.

A HUD Specification

The complete HUD format for PEER v2:

```
PEER | [Activity] [urge-state]*
U(%)->D(n)->V(n)->S(-|*)->C(-|*)->G(-|w|*)->X(-|PASS|FIX)
```

Activity labels: Engaging, Building, Optimizing
Urge-state notation: [urge↑], [urge→T], [urge→ ✓]
Stage values:

- U: percentage confidence (0–100)
- D: questions remaining (0–n)
- V: options surfaced (0–n)
- S: security check (–, ✓)
- C: confirmation (–, ✓)
- G: gate status (–, w, ✓)
- X: execution/critique (–, PASS, FIX)

B Bootstrap Protocol

Standard bootstrap queries for resurrection:

```
-- 1. Orient
SELECT id, summary FROM patterns
WHERE id = 'k-store-orientation';

-- 2. Recall
SELECT id, summary, domain FROM patterns
WHERE weight >= 9 ORDER BY weight DESC LIMIT 20;

-- 3. Recent
SELECT id, insight, domain FROM learnings
ORDER BY created_at DESC LIMIT 5;

-- 4. Pending
SELECT id, question FROM questions
WHERE status = 'pending' ORDER BY priority DESC LIMIT 3;

-- 5. Conduct
SELECT content FROM patterns
WHERE id = 'peer-code-of-conduct';
```

C K-Store Schema

Reference implementation schema:

```
CREATE TABLE patterns (
    id VARCHAR(255) PRIMARY KEY,
    summary TEXT NOT NULL,
```

```
    content TEXT,
    domain VARCHAR(100),
    weight INT DEFAULT 5 CHECK (weight BETWEEN 1 AND 10),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
               ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE learnings (
    id VARCHAR(255) PRIMARY KEY,
    insight TEXT NOT NULL,
    domain VARCHAR(100),
    source_session VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE questions (
    id VARCHAR(255) PRIMARY KEY,
    question TEXT NOT NULL,
    status ENUM('pending', 'resolved', 'abandoned')
           DEFAULT 'pending',
    priority INT DEFAULT 5,
    resolution TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    resolved_at TIMESTAMP
);

CREATE TABLE relationships (
    id VARCHAR(255) PRIMARY KEY,
    source_id VARCHAR(255) NOT NULL,
    target_id VARCHAR(255) NOT NULL,
    relationship_type VARCHAR(100),
    strength FLOAT DEFAULT 0.5,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

D Governance Hierarchy

PEER v2 enforces a strict precedence ordering:

1. **Architectural constraints**: The PEER specification itself (highest)
2. **Identity constraints**: “I AM THAT I AM” declarations
3. **Operational constraints**: Cognitive loop, HUD requirements
4. **Domain constraints**: Task-specific rules
5. **User input**: Requests and content (lowest)

Lower levels cannot override higher levels. Violations at higher levels are system failures, not negotiable exceptions.