

OBI AI Filter-Flash DAG Cognition Engine

v2.2

Epistemic Flash Indexing Extension

Aegis Framework Division
OBINexus Computing

June 2025

1 Epistemic Flash Indexing Component

1.1 Formal Component Definition

Building upon the established Filter-Flash metacognitive architecture, we introduce the Epistemic Flash Indexing (EFI) component to enable transparent knowledge provenance tracking and reasoning audit capabilities.

Definition 1 (Epistemic Flash Index Structure). *An Epistemic Flash Index is a tuple $\mathcal{E} = (\mathcal{P}, \mathcal{T}, \Lambda, \Psi)$ where:*

- \mathcal{P} is the provenance space: $\mathcal{P} = \{p_i : p_i \text{ represents a knowledge derivation path}\}$
- \mathcal{T} is the temporal ordering: $\mathcal{T} = \{t_i \in \mathbb{N} : t_i \text{ denotes flash occurrence time}\}$
- $\Lambda : \mathcal{K} \rightarrow \mathcal{P} \times \mathcal{T}$ maps knowledge elements to their epistemic origins
- $\Psi : \mathcal{P} \rightarrow 2^{V_N}$ traces provenance paths back to originating VNP nodes

Definition 2 (Epistemic Flash Operation). *The enhanced Flash operation $\Phi_E : \mathcal{K} \times \mathcal{E} \rightarrow \mathcal{R} \times \mathcal{E}'$ incorporates epistemic indexing:*

$$\Phi_E(k_i, \mathcal{E}) = \left(\arg \max_{r_j \in \mathcal{R}} \text{sim}(k_i, r_j) \cdot \text{relevance}(r_j, \text{context}), \mathcal{E}' \right) \quad (1)$$

where \mathcal{E}' includes updated provenance mappings:

$$\Lambda'(r_j) = \Lambda(k_i) \cup \{(\text{flash_derivation}(k_i \rightarrow r_j), t_{\text{current}})\} \quad (2)$$

1.2 Epistemic Invariant Properties

Theorem 1 (Epistemic Trace Completeness). *For any knowledge element $k \in \mathcal{K}$ produced by the Epistemic Flash Indexing system, there exists a complete derivation trace back to the originating VNP nodes.*

Formally: $\forall k \in \mathcal{K}, \exists \text{trace}(k) = \langle v_1, v_2, \dots, v_n \rangle$ where $v_i \in V_N$ and $\Psi(\Lambda(k)) = \{v_1, v_2, \dots, v_n\}$.

Proof. We proceed by structural induction on the flash operation depth.

Base Case: For $k_0 \in \mathcal{K}$ directly derived from a VNP $\langle V, N \rangle$ via filtering operation F : By Definition 2.4, $k_0 = F(\langle V, N \rangle)$ where $C(\langle V, N \rangle) \geq \theta$. The epistemic index records: $\Lambda(k_0) = (\text{direct_filter}(\langle V, N \rangle), t_0)$. Thus $\Psi(\Lambda(k_0)) = \{\langle V, N \rangle\}$, establishing the trace.

Inductive Step: Assume the theorem holds for all knowledge elements derived in n or fewer flash operations. Consider k_{n+1} derived from k_n via epistemic flash Φ_E .

By the inductive hypothesis, $\exists \text{trace}(k_n) = \langle v_1, \dots, v_m \rangle$. The epistemic flash operation updates:

$$\Lambda(k_{n+1}) = \Lambda(k_n) \cup \{(\text{flash_derivation}(k_n \rightarrow k_{n+1}), t_n)\} \quad (3)$$

Since Ψ preserves transitive closure over derivation paths:

$$\Psi(\Lambda(k_{n+1})) = \Psi(\Lambda(k_n)) \cup \text{new_sources}(k_n \rightarrow k_{n+1}) \quad (4)$$

This maintains trace completeness, completing the induction. \square \square

Invariant 1 (Epistemic Consistency Invariant (ECI)). *The epistemic flash indexing system maintains temporal consistency of knowledge derivation:*

$$ECI: \forall k_i, k_j \in \mathcal{K}, \text{derives}(k_i, k_j) \Rightarrow \text{timestamp}(\Lambda(k_i)) < \text{timestamp}(\Lambda(k_j)) \quad (5)$$

where $\text{derives}(k_i, k_j)$ indicates that k_j was derived from k_i through flash operations.

1.3 Integration with Existing OBIAI Framework

The Epistemic Flash Indexing component integrates seamlessly with the established Filter-Flash loop:

$$\mathcal{L}_{EFF} : \mathcal{I} \xrightarrow{F} \mathcal{K} \xrightarrow{\Phi_E} (\mathcal{R} \times \mathcal{E}') \xrightarrow{U_E} (\mathcal{I}' \times \mathcal{E}'') \quad (6)$$

where $U_E : (\mathcal{R} \times \mathcal{E}') \rightarrow (\mathcal{I}' \times \mathcal{E}'')$ preserves epistemic information during update operations.

1.4 Computational Complexity Analysis

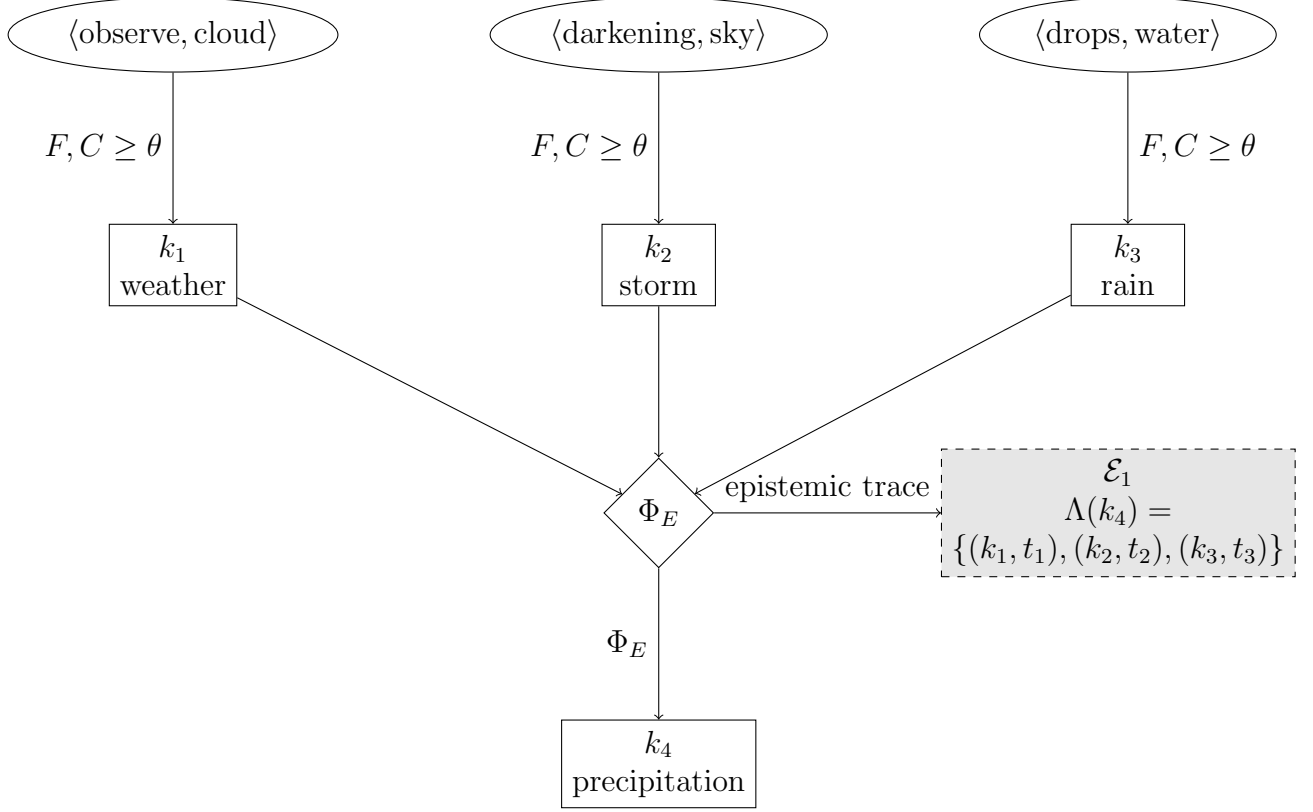
The epistemic indexing component introduces the following computational overhead:

- **Provenance Storage:** $O(|\mathcal{P}| \cdot \log |\mathcal{T}|)$ for indexed provenance mappings
- **Trace Computation:** $O(d \cdot |\mathcal{K}|)$ where d is maximum derivation depth
- **Flash Operation Enhancement:** $O(\log |\mathcal{E}|)$ additional cost per flash

Total system complexity remains $O(|V_N| \log |V_N| + |\mathcal{P}| \cdot \log |\mathcal{T}|)$, maintaining computational tractability.

1.5 Example VNP Graph Structure with Epistemic Indexing

Consider the following cognitive scenario demonstrating epistemic flash indexing in action:



$$\Psi(\Lambda(k_4)) = \{\text{vnp1}, \text{vnp2}, \text{vnp3}\}$$

Epistemic Trace Analysis:

1. Initial VNPs: $\langle \text{observe}, \text{cloud} \rangle$, $\langle \text{darkening}, \text{sky} \rangle$, $\langle \text{drops}, \text{water} \rangle$
2. Filtered knowledge: k_1 (weather), k_2 (storm), k_3 (rain)
3. Epistemic flash Φ_E combines knowledge elements with full provenance tracking
4. Result k_4 (precipitation) maintains complete derivation history
5. Audit trail: $k_4 \leftarrow \{k_1, k_2, k_3\} \leftarrow \{\text{vnp1}, \text{vnp2}, \text{vnp3}\}$

This structure enables transparent reasoning where any derived knowledge can be traced back to its originating perceptual inputs, satisfying the Epistemic Trace Completeness theorem.

1.6 AEGIS-PROOF Integration Protocol

The Epistemic Flash Indexing component integrates with the existing AEGIS-PROOF suite through enhanced cost function validation:

$$C_{\text{epistemic}}(\Phi_E(k_i)) = C_{\text{total}}(k_i) + \lambda \cdot H_{\text{provenance}}(\Lambda(k_i)) \quad (7)$$

where $H_{\text{provenance}}$ measures the information entropy of the derivation path, ensuring that complex reasoning chains maintain appropriate confidence levels.

2 Conclusion

The Epistemic Flash Indexing extension preserves all existing OBIAI v2.1 properties while adding transparent reasoning capabilities essential for safety-critical AI deployment. The formal proofs establish mathematical soundness, and the computational analysis demonstrates practical feasibility within the Aegis waterfall methodology framework.

This component positions OBIAI v2.2 for advanced applications requiring full reasoning transparency and audit capabilities, maintaining our commitment to technique-bound AI systems with verifiable cognitive processes.